Generalized Characteristic Function Loss for Crowd Analysis in the Frequency Domain

Weibo Shu[1], Jia Wan[1,2], and Antoni B. Chan[1]
[1]Department of Computer Science, City University of Hong Kong, Hong Kong
[2]Department of Electrical and Computer Engineering, University of California, San Diego, USA

Typical approaches that learn crowd density maps are limited to extracting the supervisory information from the loosely organized spatial information in the crowd dot/density maps. This paper tackles this challenge by performing the supervision in the frequency domain. More specifically, we devise a new loss function for crowd analysis called generalized characteristic function loss (GCFL). This loss carries out two steps: 1) transforming the spatial information in density or dot maps to the frequency domain; 2) calculating a loss value between their frequency contents. For step 1, we establish a series of theoretical fundaments by extending the definition of the characteristic function for probability distributions to density maps, as well as proving some vital properties of the extended characteristic function. After taking the characteristic function of the density map, its information in the frequency domain is well-organized and hierarchically distributed, while in the spatial domain it is loose-organized and dispersed everywhere. In step 2, we design a loss function that can fit the information organization in the frequency domain, allowing the exploitation of the well-organized frequency information for the supervision of crowd analysis tasks. The loss function can be adapted to various crowd analysis tasks through the specification of its window functions. In this paper, we demonstrate its power in three tasks: Crowd Counting, Crowd Localization and Noisy Crowd Counting. We show the advantages of our GCFL compared to other SOTA losses and its competitiveness to other SOTA methods by theoretical analysis and empirical results on benchmark datasets. Our codes are available at github.com/wbshu/Crowd_Counting_in_the_Frequency_Domain

Index Terms—**crowd analysis, scene understanding, frequency domain analysis, loss function, heat maps**

I. Introduction

CROWD analysis has a wide application in practice, such as surveillance, business, urban planning, and transportation management. Among crowd analysis tasks, crowd counting draws much attention since the techniques used in it can also be applied to other areas such as counting animals for ecological purposes [1–3], counting microorganisms in microscopic images [4–7], and counting vehicles in transportation congestion [8–11]. The crowd counting task is challenging due to occlusions and overlaps among people's heads and bodies as well as drastic changes in people heads' shape and size. Though a number of outstanding works are proposed for solving this challenge [12–26], there are still many spaces for further improvements. Furthermore, the training of the mainstream methods relies on the dot map which is the manual annotations of all heads in the image. But in practical application, this dot map may be noisy, e.g., the annotation may deviate from the exact head position to some extent, if the annotator is working fast or is not careful. How to count the crowd with the noisy dot map is a research field with real demands but is underexplored. Recently, researchers [25–29] also focus on crowd localization, which is a more difficult task than crowd counting. For some high-level crowd analysis tasks such as behavior detections, activity recognition, and crowd tracking, the exact position of heads or people is required. Based on those tasks' wide application in the real world, the research of crowd analysis has flourished for many years, and benefits from active research.

Since [5] proposed the idea of crowd density maps as the intermediate representation, which is an intermediate representation based on smoothening the annotation dot map with a Gaussian kernel, crowd counting has entered the dot-map supervision era. The multi-column neural network (MCNN) [30] was one of the first deep neural networks (DNN) to be supervised using a density map, with many models following. The subsequent research can be sorted into two categories: 1) designing the network structure for increasing the learning capacity; 2) investigating how to better use the ground-truth (GT) dot map to give stronger supervision. This paper belongs to the 2nd category and addresses the loss design for extracting high-quality supervision information from the GT.

Although there are already some methods [24–26] in the second category obtaining outstanding performance, they also have some shortcomings. Firstly, although there is adequate exploitation of the position information in the optimal transport (OT) loss [24, 25] and the purely point-based framework (P2PNet) [26], the GT counting information is underexploited. To address this, they introduce extra terms requiring delicate balancing or more prior information. Secondly, in each training step, both the P2PNet [26] and the OT loss [24, 25] rely on iterative external algorithms for extracting the spatial information from the GT.

We think that the above drawbacks are incurred by the nature of the distribution of information in the spatial domain. First, the counting information and the position information in the spatial domain are loosely coupled, which makes the state-of-the-art (SOTA) have to introduce remedies for exploiting the counting information when the position information is fully used; Second, the position information in the spatial domain is distributed everywhere, and therefore a global optimization procedure is required to extract the spatial relationships (e.g., the Hungarian algorithm [31] for the P2PNet [31], the Sinkhorn algorithm [32] for the OT loss [24, 25]). These

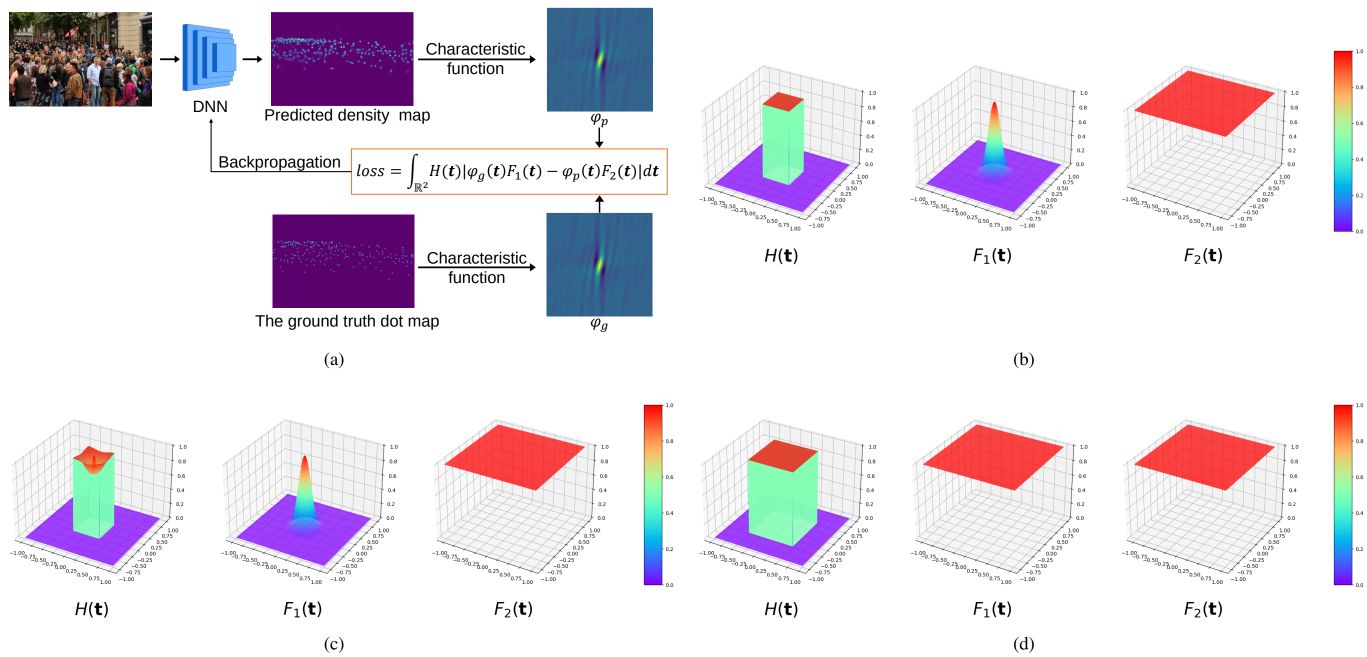


Fig. 1: (a) Our framework for crowd analysis in the frequency domain. The dispersed spatial information in the predicted density and ground-truth dot maps is converted to compact information in the frequency domain by computing their characteristic functions. Then the generalized characteristic function loss (GCFL) is a modified extension of the L1-norm between the characteristic functions. The window functions $H(\mathbf{t})$, $F_1(\mathbf{t})$, and $F_2(\mathbf{t})$ are set according to the crowd analysis task. (b) The window functions used for crowd counting. (c) The window functions used for noisy crowd counting. (d) The window functions used for crowd localization. The detailed window functions are introduced in Sec. III.

intrinsic drawbacks are hard to improve in the spatial domain – instead, we consider transforming the information to the frequency domain. By regarding dot and density maps as finite measures (i.e., unnormalized probability densities) and extending the definition of characteristic functions from probability to finite measures, we can well-organize information in the frequency domain by means of deriving characteristic functions of dot/density maps.

In the frequency domain, the original spatial information is hierarchically organized in a compact range around the origin. The information closer to the origin contains the global spatial information (i.e., which regions contain crowds), while information further from the origin relates to the local position information (i.e., the exact positions of people). Moreover, our statistical analysis also shows that the irregular spatial annotation noise changes to a concentrated noise distribution in a ring band in the frequency domain. Thus, exploiting the well-ordered frequency information can help the design of specific loss functions for better utilizing the GT information for training on different crowd analysis tasks.

In the paper, we design a generalized characteristic function loss (GCFL) for transforming the spatial information to the frequency domain and then exploiting it for the supervision of diverse crowd analysis tasks. The flexibility of the GCFL is reflected in its window functions, and we demonstrate how to use the GCFL to deal with crowd counting, crowd localization, and noisy crowd analysis tasks by applying different window functions. In the process, solid theoretical and experimental evidence is provided. Fig. 1 shows the basic framework of our method. In summary, the contributions of the paper are:

- We establish the theoretical basis of transforming the spatial crowd information into the frequency domain by extending the definition of the characteristic function from probability distributions to finite measures, as well as proving or strengthening some of its key properties.
- The characteristic function transformation yields a compactly and hierarchically organized frequency information, from which we propose the generalized characteristic function loss (GCFL) for crowd analysis tasks. The window functions in GCFL can be customized and provide flexibility for specific crowd analysis tasks.
- We demonstrate three applications using different window functions: crowd counting, crowd localization, and noisy crowd counting. For crowd counting (see Fig. 1b), we prove that minimizing the loss will decrease the upper bound of a pseudo sup norm metric between the predicted and the ground truth density map (over all sub-regions), which is effective for crowd counting. For crowd localization (Fig. 1d), we exploit the advantages of information organization in the frequency domain to scale the prediction/GT map to improve localization performance. For noisy crowd counting (see Fig. 1c), we use theoretical and statistical analysis to reveal that noisy annotations in the spatial domain will transform into noise in the regular ring band in the frequency domain. We then design a window function to ignore this regular ring band, making the loss robust to noisy annotations.
- To the best of our knowledge, this is the first work investigating crowd analysis in the frequency domain. The experimental results on benchmark datasets show the superiority of our loss to other SOTA losses on crowd counting, crowd localization, and noisy crowd counting.

This paper is an extension work of our preliminary work [33] on characteristic function loss (ChfL). In this paper, we propose a generalized loss function GCFL (§III-B), of which the original ChfL loss in [33] is a special case for a specific set of window functions. In addition, we apply our GCFL to two new tasks by designing specific window functions for crowd localization (§III-D and §V-C) and noisy crowd counting (§III-E and §V-D), which are not included in our preliminary conference paper. For crowd localization, we propose the map scaling method that takes advantage of the information distribution in the frequency domain. For noisy crowd counting, we derive the distribution of spatial annotation noise in the frequency domain, and design the window function accordingly to ignore this noise. Compared to the conference version, we also improve the implementation details of GCFL through analysis of its gradients (§IV-B). The new implementation leads to more stable training, with lower variance in results for repeated training runs, which is more suitable for use in real applications. Also, one property is strengthened (§III-A-2 Property 4) and more ablation studies are included (§V-B). Finally, we provide more theoretical analysis and experimental results to show that the low-pass filtering window is not necessary for our GCFL (§IV-A-2), which addresses a limitation in our conference paper.

The remainder of this paper is organized as follows. Sec. II introduces the related works of crowd analysis. Sec. III proposes the GCFL and demonstrates its applications to crowd counting, crowd localization, and noisy crowd counting. Sec. IV is about the implementation of the GCFL. Finally, Sec. V presents our experimental results, and Sec. VI concludes.

II. Related Works

A. Image-based crowd analysis

Image-based crowd analysis has had three research stages. The first stage is “analysis by detections”, which used various features to detect the people/heads in images [34–44], and then counted or localized from the detection results. The second stage is based on “image to count”, where regression methods were explored for directly regressing the people count from the input image features [45–51], which are specific to the crowd counting task. The current stage uses dot annotations of each person and harnesses an intermediate representation—the density map [14–18, 20, 21, 28, 44, 52–55]. These methods regress the density map from the image, and the downstream crowd analysis tasks are based on the predicted density maps. We introduce these methods in the next subsection.

B. Density map regression

[5] first proposed the density map regression method based on hand-crafted features, and [30] showed the power of deep learning for regressing the density maps. Based on the dot annotations of each person's head in the image, the GT density map provides large amounts of supervisory information, and it combined with the strong learning capacity of DNNs largely improves the performance of crowd analysis tasks. There are roughly two branches of research on supervised learning methods for density map regression. The first branch is regarding the DNN design [12–21, 28], proceeding from the traditional convolutional neural networks (CNNs) era to the vision transformer era. In contrast to the network structure design, the second branch studies how to better exploit the GT to supervise the training [22–26]. Our method belongs to the second category.

C. Improving training and loss functions

Traditional training for density map regression uses the pixel-wise L2 loss between the GT and predicted density maps. Recent methods [22–26] focus on improving training effectiveness by extracting higher-quality supervisory information from the dot annotations. In [56, 57], the dot annotations are used to build an adaptive density map representation, where the density kernel and the density map regressor are trained together to improve the counting ability for the given task. NoiseCC [23] merges the annotation uncertainty into the loss function by modeling the spatial noise of each dot annotation as a Gaussian distribution.

The Bayesian Loss (BL) method [22] used the GT dot map to calculate class conditional distributions (CCD) for each position as supervision, which inspires subsequent works by demonstrating the potential of extracting supervisory information from the GT dot map. Among them, the Generalized Loss (GL) [25] and the Distribution Matching (DMCount) [24] exploited the optimal transport (OT) distance between the GT dot maps and predicted density maps as the loss function. The OT loss is superior to the traditional pixel-wise L2 loss, as OT is a global optimization problem that jointly considers the transport cost of all pixels.

The Purely Point-Based Framework (P2PNet) [26] exploits the position information in the GT by directly training the network to predict the head positions of people. By solving a one-to-one point match between the GT and the prediction in each training step, each annotation's position information was fully used in the training process.

Despite their success, there are also some shortcomings of these SOTA methods. Firstly, although there is adequate exploitation of the position information in OT/P2PNet, the GT counting information is underexploited. Therefore, extra items and hyperparameters are introduced for remedies, which require delicate balancing. Secondly, in each training step, both the P2PNet [26] and OT [24, 25] rely on inefficient external algorithms for extracting the spatial information from the GT. [33] provides more details.

In contrast to [24–26], our method transforms the dispersed spatial information to compact frequency information, which can simultaneously use the position information and counting information for supervision in a convenient way. Moreover, our method is also efficient as it does not rely on external algorithms for spatial information extraction. Our transformation only requires basic tensor operations and can be efficiently implemented on GPU without iterations.

D. Transforming into the frequency domain in vision tasks

There are also related works exploiting the frequency domain in vision tasks [58–64]. These works transform the spatial information to the frequency domain at different locations of the model/training pipeline, such as on the inputs [62–64], the intermediate features [60, 61], or the model parameters [58, 59]. The use of the frequency transform affects the model in different ways; e.g., for tasks such as face forgery detection [62] and image demoireing [63, 64], converting the input images to the frequency domain will allow better capturing of key features for those tasks, while transforming the intermediate features [60, 61] will enable long-range and short-range feature interactions. Finally, applying frequency transforms on the model parameters and applying a low-pass filter will benefit model compression [58, 59]. In contrast to these previous works, our work applies the frequency transform on the output of the network and the GT density/dot map when computing our loss function. The traditional loss pixel-wise mean-squared error (MSE) implicitly assumes that the underlying per-pixel errors (i.e., observation noise) are independent [23]. However, for density maps the errors of pixels are typically correlated, e.g., shifting an annotation induces a specific correlated error structure [23]. Applying the frequency transform on the output allows our loss function to consider correlations among the map pixels during training.

III. Generalized Characteristic Function Loss

In this section we will introduce our loss framework of crowd analysis in the frequency domain. First, in §III-A, we establish the theoretical basis of our framework around the extension of the definition and properties of the characteristic function, by which we can transform the disorganized spatial crowd information to the hierarchically-organized frequency information. Second, based on the above fundamentals, in §III-B we propose our generalized characteristic function loss (GCFL), of which the basic characteristic function loss (ChfL) [33] is a special case. GCFL introduces a set of window functions that allows the loss to be customized for specific crowd analysis tasks. The framework is summarized in Fig. 1a.

Third, we demonstrate how to use GCFL for three crowd analysis tasks. In §III-C we introduce GCFL for crowd counting (Fig. 1b), and prove that minimizing GCFL will decrease the upper bound of a pseudo sup norm metric between the predicted and the GT density map (over all sub-regions of the spatial domain). In §III-D, we study GCFL for crowd localization (Fig. 1d), where we take advantage of the information organization in the frequency domain to boost the performance by scaling the annotation map. Finally, in §III-E, we study GCFL for noisy crowd counting, where the dot annotations contain spatial noise. Via theoretical and statistical analysis, we first show that the irregular annotation noise in the spatial domain will turn to a regular noise distribution in a ring band in the frequency domain. We then devise a set of window functions (Fig. 1c) that are robust to this frequency-domain noise, thus enabling learning from noisy crowd annotations.

A. Theoretical basis

In this subsection we first extend the concept of characteristic functions from probability distributions to density maps (i.e., finite measures). We then prove some important properties of characteristic functions of density maps.

1) Characteristic functions of density maps

In mathematics, the measure is defined as follows.

Definition 1 (Measure [65]): A **measure** is a set function m defined on a measurable space $(\Omega, \mathcal{F})$, where Ω is the total space and the family of sets $\mathcal{F}$ is a σ-algebra (comprising subsets of Ω that are closed under union, intersection, and complement), that satisfies:

(i) non-negativity: $m(A) \geq 0, \forall A \in \mathcal{F}$.
(ii) σ-additivity: $m(\emptyset) = 0$, where $\emptyset$ is the empty set, and $m(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} m(A_i)$ for a countable set $\{A_i | A_i \in \mathcal{F}, A_i \cap A_j = \emptyset \text{ if } i \neq j\}$.

If $m(\Omega) < \infty$, i.e., the total measure is finite, then it is a **finite measure**.

Thus, the density map is a finite measure on the 2D plane; Ω is the 2D Euclidean space $\mathbb{R}^2$ and $\mathcal{F}$ are all Borel sets.

Definition 2 (Density Map): A **crowd density map** is a finite measure defined on $(\mathbb{R}^2, \mathcal{B}_{\mathbb{R}^2})$, where $\mathbb{R}^2$ is the 2D Euclidean space and $\mathcal{B}_{\mathbb{R}^2}$ is all the Borel sets on $\mathbb{R}^2$. The density map's total measure on $\mathbb{R}^2$ is the total people count.

A *discrete density map* is a density map whose measure is only distributed on a set of finite points, i.e., if the density map m satisfies the following property:

$$m(A) = \sum_{i=1}^{n} m(\{x_i\} \cap A), \forall A \in \mathcal{B}_{\mathbb{R}^2}, \tag{1}$$

where $x_i \in \mathbb{R}^2$ are those points with non-zero measure, then m is a **discrete density map**. Note that the GT dot map is also a discrete density map where every point with non-zero measure has value 1, assuming that no two people can share the same location.

Next, we introduce the definition of the characteristic function for probability distributions, which is a class of special finite measures with a total measure of 1.

Definition 3 (Characteristic Function for Distributions [66]): Given a distribution d defined on $\mathbb{R}^n$, its **characteristic function** φ_d is a complex-valued function defined on $\mathbb{R}^n$:

$$\varphi_d(\mathbf{t}) = \mathbb{E}_{\mathbf{X}\sim d}[e^{\mathrm{i}\langle \mathbf{t}, \mathbf{X}\rangle}], \tag{2}$$

where $\mathbf{t} \in \mathbb{R}^n$ is the independent variable of the frequency domain, $\mathbb{E}_{\mathbf{X}\sim d}$ is expectation under $\mathbf{X}$ with distribution d, and i is the imaginary unit.

Since the probability distribution is just the finite measure with total measure 1, the definition of characteristic functions can be extended to finite measures (i.e., density maps).

Definition 4 (Characteristic Function for Measures): Given a finite measure m defined on $\mathbb{R}^n$, its **characteristic function** φ_m is a complex-valued function defined on $\mathbb{R}^n$:

$$\varphi_m(\mathbf{t}) = \int_{\mathbb{R}^n} e^{\mathrm{i}\langle \mathbf{t}, \mathbf{x}\rangle} dm(\mathbf{x}), \tag{3}$$

where $dm(\mathbf{x})$ is the integral calculated based on measure m.

Therefore, using Defn. 2 and 4, we can calculate the characteristic function of a density map, transforming the spatial information into the frequency domain. Fig. 2(a-c) show an example of a density map and its characteristic function.

2) Properties of the characteristic function

Next we derive several important properties of characteristic functions of finite measures. All proofs are in the supplemental. For clarity, we will directly present these properties in terms of density maps, rather than finite measures. Therefore, in the remaining, the terminology "density map" refers the finite measure defined on $(\mathbb{R}^2, \mathcal{B}_{\mathbb{R}^2})$ (see Defn. 2).

Property 1 (Uniqueness): The characteristic function **uniquely** determines the density map and vice versa. Suppose that φ_{m_1} and φ_{m_2} are two characteristic functions derived from two density maps m_1 and m_2 respectively. Then,

$$\varphi_{m_1}(\mathbf{t}) = \varphi_{m_2}(\mathbf{t}) \ a.e. \tag{4}$$

iff

$$m_1(A) = m_2(A), \ \forall A \in \mathcal{B}_{\mathbb{R}^2}. \tag{5}$$

We denote this as $m_1 = m_2$. In (4), a.e. means $\mathcal{L}(\{\mathbf{t} \in \mathbb{R}^2 | \varphi_{m_1}(\mathbf{t}) \neq \varphi_{m_2}(\mathbf{t})\}) = 0$, where $\mathcal{L}$ is the Lebesgue measure. See proof in Appendix A.2.4.

Remark Intuitively, this property states that if the characteristic functions of two density maps are identical, then the two density maps are identical, and vice versa. This property mainly guarantees that there is a unique optimal solution in our loss function, whereas the problem of non-unique optimal solutions in loss functions is pointed out by [24] as a potential defect of the BL [22].

Property 2 (Linearity): Suppose that m_3 is a linear combination of two density maps m_1 and m_2,

$$m_3 = \alpha m_1 + \beta m_2, \ \alpha, \beta \geq 0 \tag{6}$$

then

$$\varphi_{m_3}(\mathbf{t}) = \alpha \varphi_{m_1}(\mathbf{t}) + \beta \varphi_{m_2}(\mathbf{t}). \tag{7}$$

See proof in Appendix A.2.1.

Remark This property is helpful when we derive the characteristic functions of the GT and predicted density maps, because they are linear combinations of some basic units, e.g., singleton measures or Gaussian distributions.

Property 3 (Inversion Formula): For a density map m, suppose there is a box area $A = [a_1, b_1] \times [a_2, b_2]$ in $\mathbb{R}^2$ with zero measure boundary, i.e.,

$$m(\partial A) = 0 \tag{8}$$

where ∂A means the boundary of A, then we have

$$m(A) = \lim_{T \to \infty} \frac{1}{(2\pi)^2} \int_{[-T,T]^2} \int_A \varphi_m(\mathbf{t}) e^{-\mathrm{i}\langle \mathbf{t}, \mathbf{x} \rangle} d\mathbf{x} d\mathbf{t} \tag{9}$$

where $d\mathbf{x}$ and $d\mathbf{t}$ mean both the first and second integral are calculated based on Lebesgue measure.[1] See proof in Appendix A.2.2.

Remark This property bridges the density map and its characteristic function. Fig. 2 illustrates that the main contribution to the integral in (9) is from a small compact range of $\mathbb{R}^2$, which means each spatial region's information can be recovered from the compactly organized frequency information by Property 3.

[1]Note that when $d\mathbf{x}$ or $d\mathbf{t}$ appears in the next context, it also means the integral is calculated based on Lebesgue measure. $\mathbf{x} \in \mathbb{R}^2$ corresponds to the spatial domain, and $\mathbf{t} \in \mathbb{R}^2$ corresponds to the frequency domain.

Property 4 (Lipschitz Continuity): The characteristic function $\varphi_m(\mathbf{t})$ of a density map m is uniformly continuous. If m is a discrete density map (see Defn. 2) or a discrete density map convolved with a Gaussian kernel, then the characteristic function $\varphi_m(\mathbf{t})$ is Lipschitz continuous. See proof in App. A.2.3.

Remark This property is vital for the implementation of our basic loss. There is no analytic solution for our basic loss function, but this property enables an approximate implementation of our basic loss by discretization.

B. Generalized characteristic function loss (GCFL)

We now propose our generalized characteristic function loss (GCFL). We start from the characteristic function (ChfL) loss in [33], and then derive the GCFL. Given the predicted discrete density map m_p and the GT density map $\widehat{m}_g$, which is obtained by convolving the GT dot map m_g with a Gaussian kernel, the Chf loss [33] is the L_1-norm metric between their characteristic functions $\varphi_{\widehat{m}_g}$ and φ_{m_p}, i.e.,[2]

$$l_{\text{chf}}(\widehat{m}_g, m_p) = \int_{\mathbb{R}^2} \left| \varphi_{\widehat{m}_g}(\mathbf{t}) - \varphi_{m_p}(\mathbf{t}) \right| d\mathbf{t} \tag{10}$$

where in this paper $|a|$ always means taking the modulus of complex number a, i.e., $|a| = \sqrt{\Re(a)^2 + \Im(a)^2}$, where $\Re(a)$ and $\Im(a)$ are the real and imaginary parts of a. If a is a real number, then $|a|$ is its absolute value.

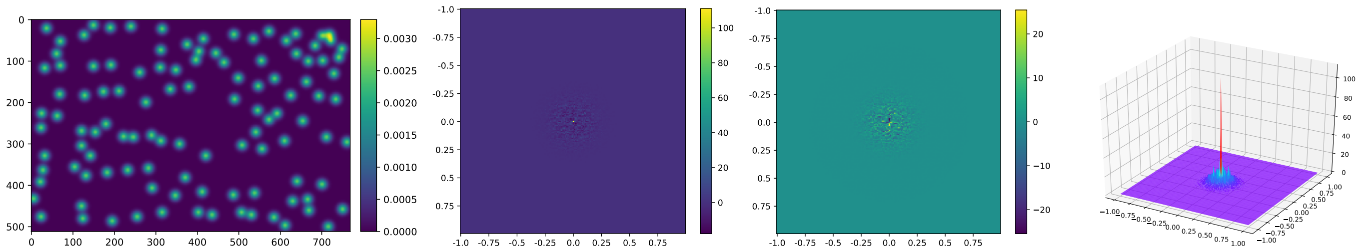
In crowd counting tasks, we usually convolve the GT dot map with a Gaussian kernel for smoothing the discrete density map [22, 30, 56, 67]. In our framework, this is equivalent to multiplying a Gaussian window with the characteristic function of the dot annotation map. In particular, let m_g represent the GT dot map, and suppose there are M people with locations $\{\boldsymbol{\mu}_j\}_{j=1}^M$, then

$$m_g(\mathbf{x}) = \sum_{k=1}^{M} \delta(\mathbf{x} - \boldsymbol{\mu}_j), \tag{11}$$

where $\delta(\mathbf{x})$ is the Dirac delta function, and we denote $\delta_{\boldsymbol{\mu}}(\mathbf{x}) = \delta(\mathbf{x} - \boldsymbol{\mu})$. Using Property 2 and noting that the characteristic function of a Dirac delta is $\varphi_{\delta_{\boldsymbol{\mu}}}(\mathbf{t}) = \exp(\mathrm{i}\boldsymbol{\mu}^T \mathbf{t})$, we obtain the characteristic function of the dot map,

$$\varphi_{m_g}(\mathbf{t}) = \sum_{j=1}^{M} \varphi_{\delta_{\boldsymbol{\mu}_j}}(\mathbf{t}) = \sum_{j=1}^{M} \exp(\mathrm{i}\boldsymbol{\mu}_j^T \mathbf{t}). \tag{12}$$

[2]Note here that we directly use the Lebesgue integral on $\mathbb{R}^2$, but in (9) we use a limit formula rather than the direct Lebesgue integral. As they are not always identical, some care is needed and we provide the mathematical details in the supplementary.



(a) (b) (c) (d)

Fig. 2: Comparison between the information distribution in the spatial domain and the frequency domain. (a) the density map m in the spatial domain $[0, 512] \times [0, 749]$; (b) the real part of the characteristic function φ_m of m, in the range $[-1, 1]^2$; (c) the imaginary part of the characteristic function φ_m in the range $[-1, 1]^2$; (d) the spectrum of the characteristic function, i.e., $|\varphi_m|$ in range $[-1, 1]^2$. The information is distributed everywhere in the spatial domain, while the information in the frequency domain is concentrated on a small compact range near the origin. By Property 3, that compact frequency information can recover the information anywhere in the spatial domain.

The GT density map is typically obtained by convolving the dot map with a Gaussian distribution, $\mathcal{N}(0, \mathbf{\Sigma})$, and thus the GT density map $\widehat{m}_g$ is the sum of M Gaussian distributions,

$$\widehat{m}_g = m_g * \mathcal{N}(0, \mathbf{\Sigma}) = \sum_{k=1}^{M} \delta_{\boldsymbol{\mu}_j} * \mathcal{N}(0, \mathbf{\Sigma}),$$

$$\Rightarrow \widehat{m}_g(\mathbf{x}) = \sum_{j=1}^{M} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \mathbf{\Sigma}), \tag{13}$$

where $*$ is the convolution operation. Using Property 2 and noting that the characteristic function of a Gaussian distribution $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{\Sigma})$ is $\varphi_{\mathcal{N}}(\mathbf{t}) = \exp\left(\mathrm{i}\boldsymbol{\mu}^T\mathbf{t} - \frac{1}{2}\mathbf{t}^T\mathbf{\Sigma}\mathbf{t}\right)$, we obtain the characteristic function of $\widehat{m}_g$

$$\begin{aligned} \varphi_{\widehat{m}_g}(\mathbf{t}) &= \sum_{j=1}^{M} \exp\left(\mathrm{i}\boldsymbol{\mu}_j^T\mathbf{t} - \tfrac{1}{2}\mathbf{t}^T\mathbf{\Sigma}\mathbf{t}\right) && (14)\\ &= \sum_{j=1}^{M} \exp\left(\mathrm{i}\boldsymbol{\mu}_j^T\mathbf{t}\right)\exp\left(-\tfrac{1}{2}\mathbf{t}^T\mathbf{\Sigma}\mathbf{t}\right) && (15)\\ &= \varphi_{m_g}(\mathbf{t})\exp\left(-\tfrac{1}{2}\mathbf{t}^T\mathbf{\Sigma}\mathbf{t}\right). && (16) \end{aligned}$$

This result is also consistent with the fact that convolution in the spatial domain is equivalent to multiplication in the frequency domain.

Therefore, using (16), we can rewrite the loss in (10) in terms of the GT dot map m_g,

$$l_{\mathrm{chf}}(m_g, m_p) = \int_{\mathbb{R}^2} \left|\varphi_{m_g}(\mathbf{t})G(\mathbf{t}) - \varphi_{m_p}(\mathbf{t})\right| d\mathbf{t}, \tag{17}$$

where $G(\mathbf{t}) = \exp(-\frac{1}{2}\mathbf{t}^T\mathbf{\Sigma}\mathbf{t})$ is the Gaussian window in the frequency domain.

Now we can interpret why convolving a Gaussian kernel with the GT dot map is beneficial to crowd counting in terms of the frequency domain. Specifically, we have $\varphi_{\widehat{m}_g}(\mathbf{t}) = \varphi_{m_g}(\mathbf{t})G(\mathbf{t})$. The Gaussian window exponentially decays as the frequency increases. Therefore, multiplying the Gaussian window will ignore the high-frequency components in the GT dot map, which corresponds to local position information. Thus, Gaussian kernel convolution can avoid overfitting on the local position information in the GT, resulting in more accurate crowd count predictions.

The Gaussian window $G(\mathbf{t})$ is only one type of window function in the frequency domain. More generally, we propose a generalized characteristic loss function (GCFL),

$$l_{\mathrm{gchf}}(m_g, m_p; H, F_1, F_2) = \int_{\mathbb{R}^2} H(\mathbf{t})\left|\varphi_{m_g}(\mathbf{t})F_1(\mathbf{t}) - \varphi_{m_p}(\mathbf{t})F_2(\mathbf{t})\right| d\mathbf{t}, \tag{18}$$

which is parametrized by three window functions $\{H, F_1, F_2\}$ that give flexibility to handle different crowd analysis tasks.

In (18), F_1 controls the GT information, i.e., which part of the GT should be stressed and which part should be ignored. F_2 controls the prediction information in an inverse way. Suppose we want the prediction to respond with high values in some region in the frequency domain, then we can give low values in the corresponding region of F_2. Since the final prediction used in the loss is the product of the NN prediction and the window F_2, then the NN must output higher prediction values to overcome the lower multiplicative factor in F_2. H controls the overall loss behavior, where important frequencies can be given higher weights, and unimportant (or less confident) frequencies given lower weights.

Next, we will demonstrate how to use our GCFL in (18) for crowd counting, crowd localization, and noisy crowd counting.

C. GCFL for crowd counting

The Chf loss l_{chf} in [33] for crowd counting is a special case of our GCFL using the following window functions,

$$H(\mathbf{t}) = 1, F_1(\mathbf{t}) = \exp(-\tfrac{1}{2}\mathbf{t}^T\mathbf{\Sigma}\mathbf{t}), F_2(\mathbf{t}) = 1, \tag{19}$$

where $\mathbf{\Sigma}$ is the covariance matrix of the Gaussian kernel used to build the density map.

We next present some vital properties for this Chf loss in (10). The first property is that it is not underdetermined, i.e., two unequal density maps m_1 and m_2 will never have zero loss between them. As pointed out in [24], minimizing an underdetermined loss may degenerate the crowd counting performance.

Proposition 1: The chf loss l_{chf}, i.e., the GCFL with window functions in (19), is not underdetermined for the ground-truth density map $\widehat{m}_g$ and the predicted density map m_p. See proof in Appendix A.3.1.

Next we present a proposition for revealing why the Chf loss works well for crowd counting.

Proposition 2: For the ground-truth density map $\widehat{m}_g$ and the predicted density map m_p,

$$|\widehat{m}_g(A) - m_p(A)| \leq (2\pi)^{-2} l_{\text{chf}}(\widehat{m}_g, m_p)\mathcal{L}(A), \quad (20)$$

for any open set $A \in \mathcal{B}_{\mathbb{R}^2}$. Here $\mathcal{L}$ means the Lebesgue measure, i.e., area of A. See proof in Appendix A.3.2.

The proposition shows what will happen to the predicted density map when the Chf loss decreases w.r.t. the GT. Rearranging the terms in (20), we obtain

$$(2\pi)^2 \tfrac{|\widehat{m}_g(A) - m_p(A)|}{\mathcal{L}(A)} \leq l_{\text{chf}}(\widehat{m}_g, m_p), \forall A \in \mathcal{B}_{\mathbb{R}^2}. \quad (21)$$

and therefore the Chf loss is an upper-bound to the normalized counting errors of all sub-regions A in the density map, $\frac{|\widehat{m}_g(A) - m_p(A)|}{\mathcal{L}(A)}$, where the normalization is based on the sub-region area $\mathcal{L}(A)$.

Next, we define the "sup norm" metric between two density maps, which is the largest normalized error over all sub-regions, as

$$\Delta(\widehat{m}_g, m_p) = \sup_{\partial A = \emptyset \ \wedge \ \mathcal{L}(A) \neq 0} \frac{|\widehat{m}_g(A) - m_p(A)|}{\mathcal{L}(A)}, \quad (22)$$

where $\partial A = \emptyset$ means A has an empty boundary (i.e., it is an open set), and $\mathcal{L}(A) \neq 0$ means it has a non-trivial Lebesgue measure. Our sup norm in (22) has a similar flavor to the MESA (Maximum Excess over SubArrays) loss from [5], except that MESA is defined using rectangular regions and is unnormalized, whereas ours is defined over all sub-regions and is normalized.

Finally, we obtain

$$(2\pi)^2 \Delta(\widehat{m}_g, m_p) \leq l_{\text{chf}}(\widehat{m}_g, m_p), \quad (23)$$

and thus minimizing the Chf loss is equivalent to minimizing the upper bound of our sup norm metric $\Delta(\widehat{m}_g, m_p)$ between the prediction and the GT, i.e., minimizing the largest normalized error over all sub-regions. Using the Chf loss for training will apply supervision more evenly on all region counts, which avoids individual pixel-wise fluctuations in the spatial domain (e.g., inherent with pixel-wise losses like L2). Specifically, (21-23) show that decreasing the Chf loss will ensure the closeness of the prediction to the GT for all areas in the spatial domain, i.e., both local and global counts are considered for supervision.

In practical implementation, we adopt the following windows for GCFL to do crowd counting,

$$H(\mathbf{t}) = \begin{cases} 1, & \mathbf{t} \in [-0.3, 0.3]^2 \\ 0, & \text{otherwise} \end{cases}, \quad (24)$$
$$F_1(\mathbf{t}) = \exp(-\tfrac{1}{2}\mathbf{t}^T \mathbf{\Sigma} \mathbf{t}), F_2(\mathbf{t}) = 1.$$

Comparing (24) with (19), $H(\mathbf{t})$ is truncated to a frequency range around the origin, and thus the integral in (18) is restricted on $[-0.3, 0.3]^2$. See §IV-A1 and §V-B3 for more details.

D. GCFL for localization

In contrast to crowd counting which needs to ignore local details for preventing overfitting, crowd localization needs more local information to provide precise positions, especially for tiny dense heads. In the frequency domain, Low-frequency components correspond to smoother 2D sinusoids, while high-frequency components correspond to sharper 2D sinusoids. Reconstructing precise local details in the spatial domain requires high-frequency sinusoids. Thus, the following window functions are adopted for GCFL to tackle crowd localization:

$$H(\mathbf{t}) = \begin{cases} 1, & \mathbf{t} \in [-0.5, 0.5]^2 \\ 0, & \text{otherwise} \end{cases}, F_1(\mathbf{t}) = 1, F_2(\mathbf{t}) = 1. \quad (25)$$

Since the localization requires precise local information, it is not helpful to use a Gaussian window to smoothen the local position information. Therefore in (25), we remove the Gaussian window as compared to (24). Furthermore, the integral range is expanded from $[-0.3, 0.3]^2$ to $[-0.5, 0.5]^2$, which includes more high-frequency components to use more precise localization information for supervision. §IV-A2 gives more theoretical details about the range selection, while §V-C2 presents an ablation study of the H window range.

Map scaling. When we train the model for crowd localization on dense heads, we can also exploit the information distribution in the frequency domain and devise a map scaling trick. After transformation to the frequency domain, most information is concentrated in a small compact range around the origin. This attribute is applicable to any spatial information distribution, i.e., no matter how large the range of the spatial information, its transformed frequency information is always in that small compact range. Therefore, we can expand the coordinates of the GT dot map and the corresponding predicted density map simultaneously, so that the localization error is increased. Then GCFL will be more sensitive to the localization error, but without any extra time or space consumption due to the above property of the frequency information organization. Furthermore, when the GT dot map is scaled, some dense heads are more separated, which makes the local position information clearer. Note that here we are scaling the coordinate system of the GT dots (e.g., $\boldsymbol{\mu}_j \to 10\boldsymbol{\mu}_j$) and predicted density locations in the discrete density map (cf., increasing the image resolution), so no extra memory and negligible extra computation are required.

E. GCFL for noisy crowd counting

In practical application, the annotation of heads may not be in the exact center of the head, due to carelessness and the ambiguity of the annotation task. In other words, there exists spatial noise in the head annotations. How to train a crowd counting model with this type of noisy training data is a realistic problem. [23] has investigated this problem in terms of the spatial domain. Here we tackle this challenge in terms of the frequency domain, by using GCFL.

1) Analysis of annotation noise in the frequency domain

We start from the characteristic function of the GT density map $\widehat{m}_g$ in (13), and analyze how annotation noise affects the frequency information. Suppose for each head $\boldsymbol{\mu}_j$ the annotation noise is a 2D random vector $\boldsymbol{\epsilon}_j$, the noisy GT density map $\widetilde{m}_g$ is

$$\widetilde{m}_g(\mathbf{x}) = \sum_{j=1}^{M} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j + \boldsymbol{\epsilon}_j, \boldsymbol{\Sigma}). \tag{26}$$

By (16), we obtain the characteristic function of $\widetilde{m}_g$,

$$\varphi_{\widetilde{m}_g}(\mathbf{t}) = \sum_{j=1}^{M} \exp\left(\mathrm{i}(\boldsymbol{\mu}_j + \boldsymbol{\epsilon}_j)^T\mathbf{t} - \tfrac{1}{2}\mathbf{t}^T\boldsymbol{\Sigma}\mathbf{t}\right) \tag{27}$$

$$= \Big[\sum_{j=1}^{M} \exp(\mathrm{i}\boldsymbol{\mu}_j^T\mathbf{t})\exp(\mathrm{i}\boldsymbol{\epsilon}_j^T\mathbf{t})\Big]\exp(-\tfrac{1}{2}\mathbf{t}^T\boldsymbol{\Sigma}\mathbf{t}). \tag{28}$$

Comparing (27) with (14), the annotation noise introduces the extra terms $\exp(\mathrm{i}\boldsymbol{\epsilon}_j^T\mathbf{t})$ in the frequency domain, which perturbs the frequency content. Note that $|\exp(\mathrm{i}\boldsymbol{\epsilon}_j^T\mathbf{t})| = 1$ always holds, and thus the noise term will only rotate the original frequency component of each head, without changing its modulus. From this perspective, the perturbation in the frequency domain is bounded somehow.

Gaussian annotation noise. To further analyze the effect, we first assume a Gaussian distribution on the annotation noise.

Proposition 3: Suppose the noisy density map is defined in (26). If spatial annotation noises $\{\boldsymbol{\epsilon}_j\}_j$ are independent and identically distributed (i.i.d.) and follow a Gaussian distribution $\mathcal{N}(0, \boldsymbol{\Lambda})$, then for the noisy characteristic function $\varphi_{\widetilde{m}_g}$,

$$\mathbb{E}[\varphi_{\widetilde{m}_g}(\mathbf{t})] = \sum_{j=1}^{M} \exp(\mathrm{i}\boldsymbol{\mu}_j^T\mathbf{t})\exp(-\tfrac{1}{2}\mathbf{t}^T(\boldsymbol{\Lambda}+\boldsymbol{\Sigma})\mathbf{t}), \tag{29}$$

$$\mathrm{var}(\varphi_{\widetilde{m}_g}(\mathbf{t})) = M(1-\exp(-\mathbf{t}^T\boldsymbol{\Lambda}\mathbf{t}))\exp(-\mathbf{t}^T\boldsymbol{\Sigma}\mathbf{t}). \tag{30}$$

See proof in Appendix A.3.3.

Comparing (29) and (15), on average, the Gaussian annotation noise has the effect of spreading the Gaussian window according to the annotation noise variance $\boldsymbol{\Lambda}$. For the variance distribution, note that the two terms $(1-\exp(-\mathbf{t}^T\boldsymbol{\Lambda}\mathbf{t}))$ and $\exp(-\mathbf{t}^T\boldsymbol{\Sigma}\mathbf{t})$ in (30) have complementary effect, which cause $\mathrm{var}(\varphi_{\widetilde{m}_g}) \to 0$ when $|\mathbf{t}| \to 0$ or $|\mathbf{t}| \to \infty$ (see the plot in Fig. 3a). Thus, from (30), there are 2 important properties of the variance map: 1) the region of large variance forms a ring band in the frequency domain; 2) the variance linearly scales with the count M.

Non-Gaussian annotation noise. Next we show the above two properties of the variance map also hold without the Gaussian assumption. More specifically, we have

Proposition 4: Suppose the noisy density map is defined in (26). Let the spatial annotation noises $\{\boldsymbol{\epsilon}_j\}_j$ be i.i.d. and follow distribution $\mathcal{X}$, and let $\varphi_{\widetilde{m}_g}$ be the noisy characteristic function. Then the variance map $\mathrm{var}(\varphi_{\widetilde{m}_g}(\mathbf{t})) \to 0$ when $|\mathbf{t}| \to \infty$ and $|\mathbf{t}| \to 0$, and $\mathrm{var}(\varphi_{\widetilde{m}_g}(\mathbf{t}))$ linearly scales with the total people count in $\widetilde{m}_g$. See proof in Appendix A.3.4.

Proposition 4 shows the two properties of the frequency noise distribution hold regardless of the specific distribution of the spatial annotation noise. In the spatial domain, the diversity of the head positions and annotation noise distribution makes the joint distribution of the noises very different among different images, which makes implementing noise robustness in the spatial domain difficult. However, Proposition 4 reveals that the irregularly distributed spatial annotation noises are regularly concentrated in a ring band in the frequency domain and their variances linearly scale with the total people count in the annotated dot map. This suggests a convenient method for handling spatial annotation noises in the frequency domain.

2) Simulation of noise distribution

To confirm our analysis, we run a statistical simulation to examine the effect of annotation noise in the frequency domain. For a given dot map, we simulate a noisy dot map by adding spatial noise to each dot according to a uniform disk distribution with a radius of 20 pixels. We then calculate the density maps for the original dot map and the noisy version, and the error map between their characteristic functions. The process is repeated 10,000 times, and we obtain a variance map for each dot map. Finally, we perform this procedure for 2,000 dot maps from the training set of JHU++ dataset [68], thus obtaining 2,000 variance maps.

Figure 3b-3d show examples of error variance maps for three dot maps with different counts. In the frequency domain, the spatial annotation noise causes perturbations to the characteristic functions in a ring region around the origin. Note that the scale of the variance varies with the number of people in the original dot map. Normalizing the error variance maps by the number of people in the ground truth and averaging them yields the consistent error map in Fig. 3e, which illustrates that there is indeed a linear correlation between the error variance and the ground-truth count.[3]

3) Noise-robust window

Using our analysis, we design an appropriate dynamic window function $H(\mathbf{t})$ that focuses less on the information in the ring band, which makes the model training robust to spatial annotation noise. Let $h(\mathbf{t})$ be the average normalized variance map in Fig. 3e, then we define a dynamic window function $H(\mathbf{t})$ for each image,

$$H(\mathbf{t}) = \begin{cases} \frac{1}{\sqrt{h(\mathbf{t})\cdot M+1}}, & \mathbf{t} \in [-0.3, 0.3]^2 \\ 0, & \text{otherwise} \end{cases},$$

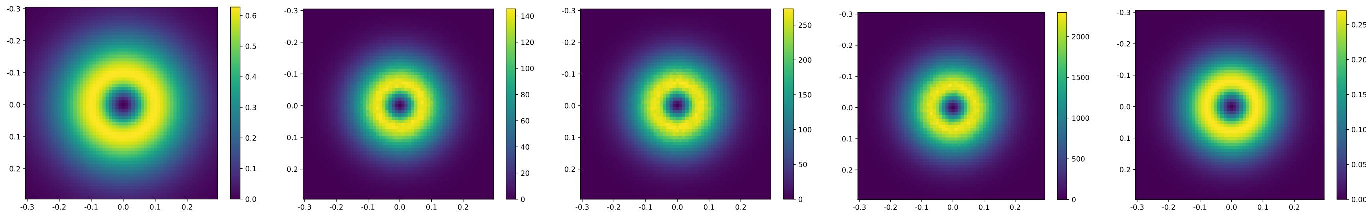
$$F_1(\mathbf{t}) = \exp(-\tfrac{1}{2}\mathbf{t}^T\boldsymbol{\Sigma}\mathbf{t}), F_2(\mathbf{t}) = 1, \tag{31}$$

where M is the total people count in the ground truth m_g, and thus $h(\mathbf{t}) \cdot M$ is the error variance map of each image. $H(\mathbf{t})$ is the reciprocal of the standard deviation map, where positions with large variances will have low weights and thus GCFL will focus less on these positions. Note that even when applying this window H, the overall count in the frequency domain, which is represented by the value at $\mathbf{t} = 0$, remains the same, thus the count is not affected during training. We use the windows in (31) for the noisy crowd counting task.

IV. Implementation of GCFL

Since there is no analytical solution for the integral in (18), we first propose an approximate implementation of

[3]See Appendix C for more details.



(a) (b) (c) (d) (e)

Fig. 3: (a), The variance map of the characteristic function when the spatial annotation noise follows a Gaussian distribution, with total people count 1. (b)∼(e) Simulation of the variance maps of characteristic functions of density maps with spatial annotation noise with (b) 455, (c) 855, and (d) 7139 people. (e) shows the average variance maps (over 2000 dot maps) after normalizing by the GT count, which illustrates the variance scales linearly with the count.

GCFL using both theoretical and empirical support. Then, we modify the implemented loss by analyzing the backpropagated gradient, and propose a modified GCFL loss that yields more stable training with lower variance among repeated runs. These properties make it well-suited for real applications.

A. Approximating the integral

To approximate (18) of GCFL requires two steps: 1) truncating the infinite integral range on a finite range; 2) using the Riemann sum to approximate the integral in this finite range.

1) Truncating the integral to a finite range

As illustrated in Fig. 2, the characteristic function values outside a compact central range are typically very small. Thus, the integral can be truncated using the window function H. The empirical and theoretical evidence also supports the truncation. In theory, we have the following upper bound of the error between the original and reconstructed density map.

Proposition 5: Suppose the density map m is obtained by convolving a discrete dot map with a Gaussian kernel whose bandwidth is σ, and the reconstructed density map $\tilde{m}$ is obtained from the characteristic function φ_m restricted on the disk $B(0,r)$. Let T be the total measure of m. Then on any non-empty box area A with trivial boundary, i.e., $m(\partial A)=0$, we have

$$\frac{|m(A)-\tilde{m}(A)|}{\mathcal{L}(A)} \leq \frac{T\exp(\frac{-\sigma^2 r^2}{2})}{2\pi\sigma^2}. \tag{32}$$

where $\mathcal{L}(A)$ means the Lebesgue measure of A, i.e., the area of the region A. See proof in Appendix A.3.5.

Proposition 5 indicates that the error between the original and the reconstructed GT density map can be well bounded by an exponentially decaying term, when we use a Gaussian kernel to generate the GT density map from the dot map. Fig. 4 shows the comparison between the original density map and the reconstructed density map from the truncated characteristic function.

2) Truncation without low-pass windows

If we do not convolve the GT dot map with the Gaussian window (or any other smoothing windows), then truncation still has an effect on the training.

As stated in Defn. 2, the dot map is also a discrete density map, then the following proposition works for both the ground-truth dot map and the predicted discrete density map.

Proposition 6: Consider a discrete density map m whose measure is distributed on N points $\{\mathbf{p}^{(1)},\cdots,\mathbf{p}^{(N)}\}$. Let $\tilde{m}$ be the reconstructed density map from the characteristic function φ_m restricted on the square $[-a,a]^2$. Then on any non-empty box area A with trivial boundary, i.e., $m(\partial A)=0$,

$$\tilde{m}(A)=\sum_{k=1}^{N} m(\mathbf{p}^{(k)})\int_A \prod_{d=1}^{2}\frac{\sin((\mathbf{p}_d^{(k)}-\mathbf{x}_d)a)}{\pi(\mathbf{p}_d^{(k)}-\mathbf{x}_d)}d\mathbf{x} \tag{33}$$

where the subscripts here indicate the 1st and 2nd coordinates of $\mathbf{p}^{(k)}$ or $\mathbf{x}$. See proof in Appendix A.3.6.

Denote the integrand in (33) as

$$f^{(k)}(\mathbf{x})=\prod_{d=1}^{2}\frac{\sin((\mathbf{p}_d^{(k)}-\mathbf{x}_d)a)}{\pi(\mathbf{p}_d^{(k)}-\mathbf{x}_d)} \tag{34}$$

$$=\frac{a^2}{\pi^2}\prod_{d=1}^{2}\operatorname{sinc}((\mathbf{p}_d^{(k)}-\mathbf{x}_d)a), \tag{35}$$

where $\operatorname{sinc}(x)=\frac{\sin(x)}{x}$, and let

$$f(\mathbf{x})=\frac{a^2}{\pi^2}\prod_{d=1}^{2}\operatorname{sinc}(\mathbf{x}_d a). \tag{36}$$

Then (33) can be rewritten as

$$\tilde{m}(A)=(m*F)(A) \tag{37}$$

where $*$ means the convolution of two measures and

$$F(A)=\int_A f(\mathbf{x})d\mathbf{x} \tag{38}$$

is the signed measure with f as its density function. Therefore intuitively, Proposition 6 says that truncating the integral is actually multiplying the frequency components with a rectangle window, which corresponds to convolving the dot map with a sinc function. The sinc function's components compactly gather around the center and quickly decay to 0 after the first zero point (outside the main lobe), which suggests the feasibility of truncating the integral to a small range. Thus truncating the characteristic function of the discrete density map is equivalent to distributing the extremely concentrated singleton measure at the annotation point to its neighboring pixels. This is a form of measure smoothing, preventing overfitting since the predictions of the NN do not have to be exactly at the ground truth dot positions. It is also consistent with the case in the frequency domain. Since the high-frequency components in general correspond to the noise part, truncating the integral means discarding the high-frequency components, which can prevent overfitting on noise.

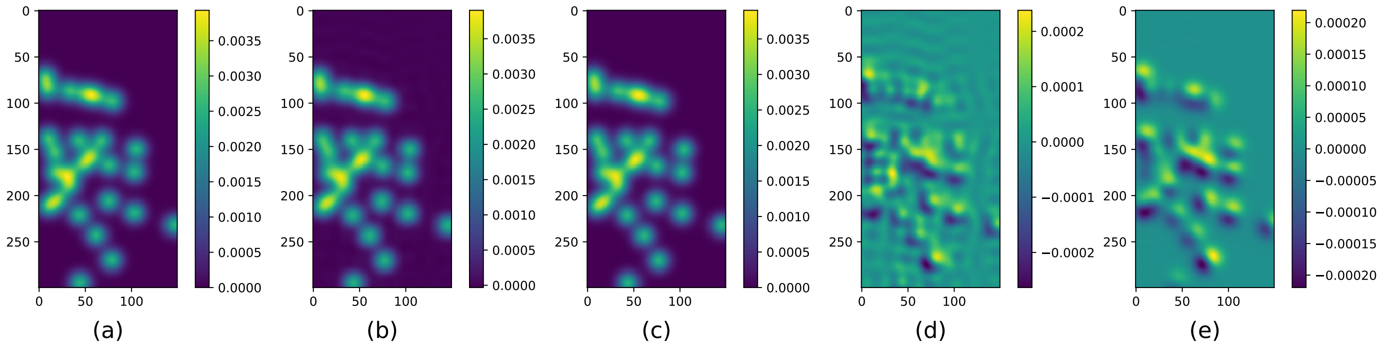


Fig. 4: Comparison between the original density map and the reconstructed density map from the characteristic function confined on a small range. (a) the original density map (with a Gaussian kernel of bandwidth 8) ; (b) the reconstructed density map from its characteristic function truncated on $[-0.3, 0.3]^2$, and on (c) $[-0.5, 0.5]^2$; (d) the difference between (a) and (b); (e) the difference between (a) and (c). The reconstructed density map and the original density map are nearly the same. Note the range of difference values in (d) and (e) is much smaller than the range of the density values. This indicates that the characteristic function confined in a small range carries nearly all the information in the spatial domain. Hence, it is appropriate to restrict the integral to a small range when we calculate the GCFL.

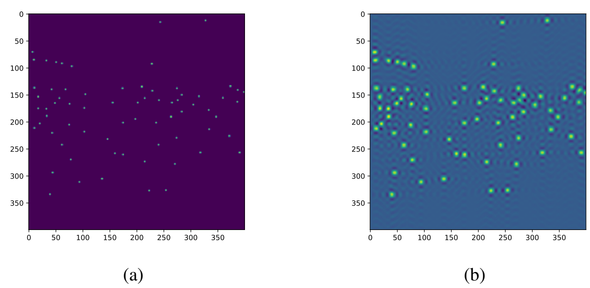


Fig. 5: (a) the original dot map. (b) the recovered density map from the characteristic function truncated on $[-0.5, 0.5]^2$.

Fig. 5 shows an example. In the spatial domain, the spatial unit is an image pixel. Thus a single pixel with measure v in the discrete density map is similar to a disk centered at the pixel location whose radius is several pixels and whose total measure is also near to v. Specifically, we set $a = 0.3$ in the crowd counting task and $a = 0.5$ in the crowd localization task, which corresponds to a sinc function with a main-lobe width of 21 pixels and 13 pixels respectively according to (36). In crowd localization, the main lobe is narrower so the density map is more separated and sharper, which is useful for localization in dense crowds.

3) Approximating the integral with Riemann sum

Although the GCFL integral is confined to a small range, it still needs to be approximated with the Riemann sum. Prop. 4 shows the Lipschitz continuity of the characteristic function, which gives a firm theoretical guarantee for the Riemann sum approximation. Furthermore, some empirical results will be shown in §V-B3.

The implementation of GCFL using the Riemann sum approximation is illustrated in Fig. 6. For a given image, let there be M people in the ground truth with locations $\{\boldsymbol{\mu}_j\}_{j=1}^M$. The characteristic function of the dot map is (12). Let $P(\mathbf{x})$ be the values in 2D matrix corresponding to the predicted density map at spatial locations $\mathbf{x}$. The prediction density map m_p is also a stack of singleton measures, and by Property 2 we have

$$\varphi_{m_p}(\mathbf{t}) = \sum_{\mathbf{x}} P(\mathbf{x})\varphi_{\delta_{\mathbf{x}}} = \sum_{\mathbf{x}} \exp(\mathrm{i}\mathbf{x}^T\mathbf{t})P(\mathbf{x}). \tag{39}$$

In the Riemann sum approximation, the integral range on $[-a, a]^2$ is divided evenly into small square grids. Let $\mathcal{R}$ be the set of center points on the grids, where the edge size of the square grid is c. Then, the implementation of GCFL is

$$\hat{l}_{\text{gchf}}(m_g, m_p) = c^2 \sum_{\mathbf{t}\in\mathcal{R}} H(\mathbf{t}) \left|\varphi_{m_g}(\mathbf{t})F_1(\mathbf{t}) - \varphi_{m_p}(\mathbf{t})F_2(\mathbf{t})\right| \tag{40}$$

$$= c^2 \sum_{\mathbf{t}\in\mathcal{R}} H(\mathbf{t}) \Big| F_1(\mathbf{t}) \sum_{j=1}^{M} \exp(\mathrm{i}\boldsymbol{\mu}_j^T\mathbf{t}) - F_2(\mathbf{t}) \sum_{\mathbf{x}} \exp(\mathrm{i}\mathbf{x}^T\mathbf{t})P(\mathbf{x}) \Big|. \tag{41}$$

The approximation introduces two hyperparameters in our method: 1) the granularity of the grid in the Riemann sum (c); 2) the integral range (a). One of the important consequences of Property 4 is to decouple these two hyperparameters. Property 4 demonstrates a uniform continuity of the characteristic function, which means the intensity of the continuity is similar everywhere in the domain. Therefore, if the granularity of the Riemann sum approximation works fine around the origin, then it also works on any integral range. Hence, the granularity of the Riemann sum approximation is independent of the integral range. As a result, the hyperparameter search is converted from a two-dimensional grid search to two one-dimensional line searches, which is more efficient. In addition, the independence of the granularity to the integral range guarantees that we can use the same granularity setting for both crowd counting and crowd localization. No extra ablation study on granularity is needed once the ablation study for crowd counting is executed.

B. Improving training stability

The above implementation may have an unstable training process in crowd counting tasks, which causes large variances in results across different trials, which may limit its usefulness in real applications. Therefore, here we propose two variants to l_{gchf} for improving the training stability.

For clarity, we first assume $H(\mathbf{t}) = F_1(\mathbf{t}) = F_2(\mathbf{t}) = 1$ in (40) for analysis, and we will add the three window functions back at the end. Now (40) becomes

$$\hat{l}_{\text{gchf}}(m_g, m_p) = c^2 \sum_{\mathbf{t}\in\mathcal{R}} \left|\varphi_{m_g}(\mathbf{t}) - \varphi_{m_p}(\mathbf{t})\right|. \tag{42}$$

Let $\mathbf{t} = [t_1, t_2]^T$. We first write the set $\mathcal{R}$ as

$$\mathcal{R} = \{(t_1, t_2) | t_1 \in \mathcal{R}_1, t_2 \in \mathcal{R}_2\}, \tag{43}$$

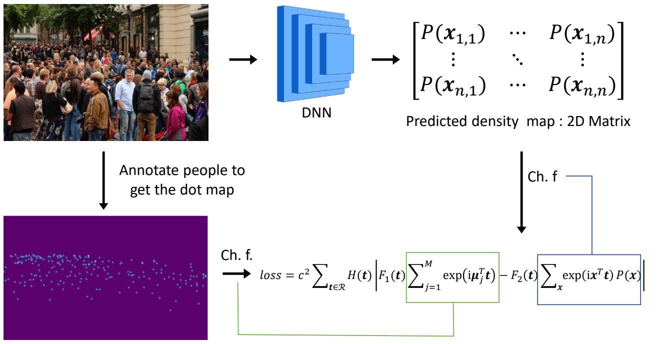


Fig. 6: The implementation of the GCFL. The DNN output is a density map represented as a 2D matrix, where each value $P(\mathbf{x})$ in the matrix corresponds to the singleton measure at a spatial position $\mathbf{x}$. Each dot in the GT has position $\boldsymbol{\mu}_j$. $\mathcal{R}$ is the set of points used for the Riemann sum approximation.

where $\mathcal{R}_1$ and $\mathcal{R}_2$ are the coordinate sets for the t_1 and t_2 axes in the frequency plane. Then we can expand (42) as

$$\hat{l}_{\text{gchf}}(m_g, m_p) = c^2 \sum_{t_1 \in \mathcal{R}_1} \sum_{t_2 \in \mathcal{R}_2} \sqrt{\Re(\Delta(t_1, t_2))^2 + \Im(\Delta(t_1, t_2))^2}, \tag{44}$$

where $\Re$ and $\Im$ mean taking the real and imaginary parts of a complex number, and $\Delta(t_1, t_2)$ is defined as

$$\Delta(\mathbf{t}) = \Delta(t_1, t_2) = \varphi_{m_g}(t_1, t_2) - \varphi_{m_p}(t_1, t_2). \tag{45}$$

We next define our two variants of GCFL,

$$\bar{l}_{\text{gchf}}(m_g, m_p) = c^2 \sum_{t_1 \in \mathcal{R}_1} \sqrt{\sum_{t_2 \in \mathcal{R}_2} \Re(\Delta(t_1, t_2))^2 + \Im(\Delta(t_1, t_2))^2}, \tag{46}$$

and[4]

$$\tilde{l}_{\text{gchf}}(m_g, m_p) = c \sqrt{\sum_{t_1 \in \mathcal{R}_1} \sum_{t_2 \in \mathcal{R}_2} \Re(\Delta(t_1, t_2))^2 + \Im(\Delta(t_1, t_2))^2}. \tag{47}$$

Note that decreasing the loss in (46) and (47) will also cause the decrease in (40), and vice versa. By analyzing their derivatives with respect to an output prediction value, we can examine the behaviors of training with these losses. Since the constants c and c^2 are absorbed into the learning rate, we first normalize the loss functions to remove these constants. For an output prediction value at position $\mathbf{x}$, the derivatives of the losses with respect to $P(\mathbf{x})$ are (see App. B for derivations):

$$\frac{1}{c^2} \frac{\partial \hat{l}_{\text{gchf}}(m_g, m_p)}{\partial P(\mathbf{x})} = \sum_{\mathbf{t} \in \mathcal{R}} \frac{\langle \mathbf{d}(\mathbf{t}), -\mathbf{f}_{\mathbf{x}}(\mathbf{t}) \rangle}{\|\mathbf{d}(\mathbf{t})\|_2}, \tag{48}$$

$$\frac{1}{c^2} \frac{\partial \bar{l}_{\text{gchf}}(m_g, m_p)}{\partial P(\mathbf{x})} = \sum_{\mathbf{t} \in \mathcal{R}} \frac{\langle \mathbf{d}(\mathbf{t}), -\mathbf{f}_{\mathbf{x}}(\mathbf{t}) \rangle}{\mathcal{Q}(\mathbf{t})}, \tag{49}$$

$$\frac{1}{c} \frac{\partial \tilde{l}_{\text{gchf}}(m_g, m_p)}{\partial P(\mathbf{x})} = \sum_{\mathbf{t} \in \mathcal{R}} \frac{\langle \mathbf{d}(\mathbf{t}), -\mathbf{f}_{\mathbf{x}}(\mathbf{t}) \rangle}{\frac{1}{c}\tilde{l}_{\text{gchf}}(m_g, m_p)}, \tag{50}$$

[4]For $\bar{l}_{\text{gchf}}$, we also considered first summing over $t_1 \in \mathcal{R}_1$ inside the square root, and then summing over $t_2 \in \mathcal{R}_2$ outside. A preliminary study showed little difference in performance, with MAE of 58.78±1.06 and 58.71±1.05 for the two versions on ShanghaiTech A.

Fig. 7: Comparison of the training process using GCFL $\hat{l}_{\text{gchf}}$ and its variant $\bar{l}_{\text{gchf}}$. The plot shows the test MAE on ShanghaiTech A test set during training (1000 epochs). Although the two losses can both explore some local optimal with the low test MAE, the more aggressive strategy in $\hat{l}_{\text{gchf}}$ makes its training more unstable.

Fig. 8: Comparison of test MAE results of ten runs using losses $\hat{l}_{\text{gchf}}$ and $\tilde{l}_{\text{gchf}}$ for training. The whiskers plot shows the mean, standard deviation, minimum and maximum test MAE over the trials. Training with $\hat{l}_{\text{gchf}}$ results in a larger variance in test MAE, as compared to training with $\tilde{l}_{\text{gchf}}$.

where

$$\mathbf{d}(\mathbf{t}) = \begin{bmatrix} \Re(\Delta(\mathbf{t})) \\ \Im(\Delta(\mathbf{t})) \end{bmatrix}, \quad \mathbf{f}_{\mathbf{x}}(\mathbf{t}) = \begin{bmatrix} \Re(\exp(\mathrm{i}\mathbf{x}^T\mathbf{t})) \\ \Im(\exp(\mathrm{i}\mathbf{x}^T\mathbf{t})) \end{bmatrix}. \tag{51}$$

In their derivatives, the numerator term $\langle \mathbf{d}(\mathbf{t}), -\mathbf{f}_{\mathbf{x}}(\mathbf{t}) \rangle$ is common, while the three denominators are different,

$$\|\mathbf{d}(\mathbf{t})\|_2 = \sqrt{\Re(\Delta(t_1, t_2))^2 + \Im(\Delta(t_1, t_2))^2}, \tag{52}$$

$$\mathcal{Q}(\mathbf{t}) = \sqrt{\sum_{t \in \mathcal{R}_2} \Re(\Delta(t_1, t))^2 + \Im(\Delta(t_1, t))^2}, \tag{53}$$

$$\frac{1}{c}\tilde{l}_{\text{gchf}}(m_g, m_p) = \sqrt{\sum_{t_1 \in \mathcal{R}_1} \sum_{t_2 \in \mathcal{R}_2} \Re(\Delta(t_1, t_2))^2 + \Im(\Delta(t_1, t_2))^2}. \tag{54}$$

From (52), (53), and (54), we have the following relationship among the denominators

$$\|\mathbf{d}(\mathbf{t})\|_2 < \mathcal{Q}(\mathbf{t}) < \tfrac{1}{c}\tilde{l}_{\text{gchf}}(m_g, m_p). \tag{55}$$

Therefore, the three losses behave differently during optimization. The $\hat{l}_{\text{gchf}}$ adopts the most aggressive optimization strategy at each training step, as its derivative possesses the smallest denominator. In contrast, $\tilde{l}_{\text{gchf}}$ is the most conservative optimization strategy, since its derivative has the largest denominator. Finally, $\bar{l}_{\text{gchf}}$ is between $\hat{l}_{\text{gchf}}$ and $\tilde{l}_{\text{gchf}}$.

From the training result on SHTCA ($\hat{l}_{\text{gchf}}$ v.s. $\bar{l}_{\text{gchf}}$, see Fig. 7) and SHTCB ($\hat{l}_{\text{gchf}}$ vs. $\tilde{l}_{\text{gchf}}$, see Fig. 8), the aggressive optimization strategy of $\hat{l}_{\text{gchf}}$ yields an unstable training process, as well as higher variance of results among different runs. The variants $\bar{l}_{\text{gchf}}$ and $\tilde{l}_{\text{gchf}}$ mitigate these disadvantages.

Finally, we add the $H(\mathbf{t})$, $F_1(\mathbf{t})$, and $F_2(\mathbf{t})$ back to the modifications to get the final version,

$$\bar{l}_{\text{gchf}}(m_g, m_p; H, F_1, F_2) \tag{56}$$
$$= c^2 \sum_{t_1 \in \mathcal{R}_1} \sqrt{\sum_{t_2 \in \mathcal{R}_2} \Re(\widehat{\Delta}(t_1, t_2))^2 + \Im(\widehat{\Delta}(t_1, t_2))^2},$$
$$\tilde{l}_{\text{gchf}}(m_g, m_p; H, F_1, F_2) \tag{57}$$
$$= c \sqrt{\sum_{t_1 \in \mathcal{R}_1} \sum_{t_2 \in \mathcal{R}_2} \Re(\widehat{\Delta}(t_1, t_2))^2 + \Im(\widehat{\Delta}(t_1, t_2))^2}$$

where

$$\widehat{\Delta}(t_1, t_2) = H(t_1, t_2)(\varphi_{m_g}(t_1, t_2)F_1(t_1, t_2) - \varphi_{m_p}(t_1, t_2)F_2(t_1, t_2)).$$

We use (57) for crowd counting in order to make the training more stable. If there are more images with dense people in the dataset, then we use the more aggressive version in (56).

V. Experiments

In this section we present experiments on crowd counting & localization and noisy crowd counting using our GCFL.

A. Experiment setup

We conduct crowd counting tasks on five benchmark data sets: ShanghaiTech A & B [30], UCF-QNRF [67], JHU++ [68, 69], and NWPU [29]. Following the convention, the crowd localization is conducted on UCF-QNRF and NWPU. For the noisy crowd counting, we construct 5 different noisy crowd data sets from UCF-QNRF by adding different levels of noise to the original GT. For UCF-QNRF, we resize each image so that its shortest side does not exceed 1536. For JHU++ and NWPU, similar resizing is performed with length 2048. The image crop size is 384 for UCF-QNRF, JHU++, and NWPU, 128 for ShanghaiTech A, and 512 for ShanghaiTech B.

We use the same density map regression DNN from [22–25], comprising the feature extraction layers of VGG19 [70] connected to a regression module composed of three convolution layers. For our loss functions, we use the Adam [71] optimizer with learning rate 1e-5 and weight decay 1e-4.

If the Gaussian window is used, the covariance matrix is always a diagonal matrix with the diagonal set as 64, which follows the convention that the Gaussian kernel in the spatial domain is set to the bandwidth 8 pixels. The grid granularity in the Riemann sum approximation is set to 0.01 for all datasets.

For our GCFL, we use the window functions in (24) for crowd counting and localization of sparse heads, which we denote as GCFL-CC. For crowd localization of non-sparse heads, we use windows in (25), which is denoted as GCFL-CL, and we use (31) in GCFL for noisy crowd counting, denoted as GCFL-NCC. For clarity, we use GCFL to represent our method in experimental results where we compare it with SOTAs. And we use the specific names GCFL-CC, GCFL-CL, and GCFL-NCC in the ablation study. In crowd counting, we apply the variant loss in (56) on dense datasets QNRF and SHTCA, and apply the variant loss (57) on the other datasets, SHTCB, JHU++, and NWPU. For crowd localization, we use map scaling with factor 10 for the ground truth and prediction dot map during the training process.

	NWPU		JHU++		UCF-QNRF		SHTC A		SHTC B	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
VGG19+L2	-	-	78.1	300.1	98.7	176.1	68.6	110.1	8.5	13.9
BL [22]	105.4	454.2	75.0	299.9	88.7	154.8	62.8	101.8	7.7	12.7
NoiseCC [23]	96.9	534.2	67.7	<u>258.5</u>	85.8	150.6	61.9	99.6	7.4	<u>11.3</u>
DM count [24]	88.4	388.6	68.4	283.3	85.6	148.3	<u>59.7</u>	95.7	7.4	11.8
GL [25]	<u>79.3</u>	<u>346.1</u>	<u>59.9</u>	259.5	<u>84.3</u>	<u>147.5</u>	61.3	<u>95.4</u>	<u>7.3</u>	11.7
GCFL (ours)	**76.8**	**343.0**	**57.0**	**235.7**	**80.3**	**137.6**	**57.5**	**94.3**	**6.9**	**11.0**

TABLE I: Comparison with state-of-the-art loss functions and baseline. All losses use the same network architecture from [22]. The best method is **bold**, while the second best is <u>underline</u>.

Loss	time per epoch	time per 500 epochs	number of related hyperparameters
VGG19+L2	15.0 s	2h 5m	1
BL[22]	15.2 s	2h 7m	2
NoiseCC [23]	16.4 s	2h 17m	6
DM count [24]	19.0 s	2h 38m	4
GL [25]	17.4 s	2h 25m	7
GCFL (ours)	15.4 s	2h 9m	3

TABLE II: Efficiency and number of hyperparameters for different loss functions. The training time is measured using the training set (300 images) of ShanghaiTech A (with batch size 1 and crop size 512). Our implementation uses *PyTorch* on an *RTX2080 TI* and *i7 9700K CPU* with *64GB memory*.

B. Crowd counting

The evaluation metrics for crowd counting follow the standard convention: the Mean Absolute Error (MAE) and the Root Mean Square Error (MSE) are adopted.

1) Comparison of loss functions

First we compare our GCFL with SOTA loss functions in crowd counting in Table I. All of the losses use the same network architecture from [22]. Our loss outperforms the other losses on all datasets. Moreover, [24, 25] require an external Sinkhorn algorithm [32] running dozens of even hundreds of iterations in each training batch, while [23] needs to invert large matrices in each training batch. Nevertheless, GCFL does not require any other external algorithm, and the calculations are quickly completed using standard tensor operations.

Table II shows the efficiency comparison among these loss functions. Since all losses use the same network architecture in the training phase, the identical inference time is omitted here. Excluding the L2 baseline, BL [22] is the most efficient loss function but also has the poorest performance. Our GCFL has 2nd highest efficiency, as well as the 2nd lowest number of hyperparameters, while also achieving the best MAE and MSE. Note we only use 300 training images to compute the timings, and the efficiency advantage will increase as the training size and number of epochs increase.

2) Comparison with SOTA

Table III shows the comparison between our loss and the current SOTA. For fairness, this comparison only considers methods using a single model and trained on an individual dataset. Although our method is simple, our GCFL is competitive against current SOTA on large-scale datasets, obtaining the lowest MAE/MSE on UCF-QNRF, JHU++, and NWPU. Our method also obtains competitive results on ShanghaiTech A and B, but these two datasets are smaller and less representative of generalization ability. These comparative results demonstrate the potential of supervising crowd counting in the frequency domain.

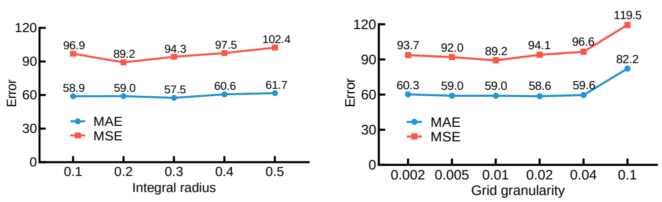
We also compare the efficiency of our method with other recent algorithms in Table III. Our method is 4x faster than

	NWPU		JHU++		UCF-QNRF		SHTC A		SHTC B	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
CAN [16]	106.3	386.5	100.1	314.0	107	183	62.3	100.0	7.8	12.2
SFCN [72]	105.7	424.1	77.5	297.6	102.0	171.4	64.8	107.5	7.6	13.0
BL [22]	105.4	454.2	75.0	299.9	88.7	154.8	62.8	101.8	7.7	12.7
KDMG [56]	100.5	415.5	69.7	268.3	99.5	173.0	63.8	99.2	7.8	12.7
LSCCNN [42]	-	-	112.7	454.4	120.5	218.2	66.5	101.8	7.7	12.7
RPNet [18]	-	-	-	-	-	-	61.2	96.9	8.1	11.6
AMRNet [73]	-	-	-	-	86.6	152.2	61.6	98.4	7.0	11.0
NoiseCC [23]	96.9	534.2	67.7	258.5	85.8	150.6	61.9	99.6	7.4	11.3
DM count [24]	88.4	388.6	68.4	283.3	85.6	148.3	59.7	95.7	7.4	11.8
LA-Batch [74]	-	-	-	-	113.0	210.0	65.8	103.6	8.6	14.0
AutoScale [75]	94.1	388.2	65.9	264.8	104.4	174.2	65.8	112.1	8.6	13.9
GL [25]	79.3	346.1	59.9	259.5	84.3	147.5	61.3	95.4	7.3	11.7
SDA+BL [76]	-	-	62.6	264.1	83.3	143.1	58.4	95.7	-	-
P2PNet [26]	77.4	362.0	-	-	85.3	154.5	**52.7**	85.1	**6.2**	**9.9**
DMCNet [77]	-	-	69.6	246.9	96.5	164.0	58.5	**84.6**	8.6	13.7
SS-DCNet [78]	-	-	-	-	81.9	143.8	56.1	88.9	6.6	10.8
DC [79]	-	-	60.0	269.9	86.9	159.3	59.7	91.4	7.0	11.6
GauNet [80]	-	-	58.2	245.1	81.6	153.7	54.8	89.1	**6.2**	**9.9**
GCFL (ours)	**76.8**	**343.0**	**57.0**	**235.7**	**80.3**	**137.6**	57.5	94.3	6.9	11.0

TABLE III: Comparison with state-of-the-art single-model methods trained on individual data sets.

Algorithm	training time per epoch	inference time per epoch	crop size of images for training
KDMG [56]	83.0 s	6.9 s	512
P2PNet [23]	60.8 s	11.8 s	128
GCFL (ours)	15.4 s	6.9 s	512

TABLE IV: Running time of recent algorithms. The inference time is measured using the test set (182 original images) of ShanghaiTech A. Other settings are the same as in Table II.



(a) (b)

Fig. 9: Ablation study on (a) the integral range $[-a, a]^2$ where a is the value in the x-axis; (b) the grid granularity in the Riemann sum approximation, where the granularity is the side length of the square grid and the integral range is fixed at $[-0.2, 0.2]^2$.

P2PNet (despite P2PNet using smaller image sizes) and 5.4x faster than KDMG in training. For inference, our method has the same running time as KDMG since they use the same architecture, and is ~41% faster than P2PNet.

3) Ablation study

The approximation of the integral of GCFL-CC introduces two extra hyperparameters: the integral range a and the grid granularity c in the Riemann sum approximation. As mentioned in §IV, Property 4 decouples these two hyperparameters, and thus the ablation study is carried out individually for each hyperparameter on ShanghaiTech A.

Fig. 9a shows the results for different integral ranges. Generally, the counting performance is robust to different integral ranges. When the range is above $[-0.3, 0.3]^2$, the performance gradually degenerates, which suggests that the frequency information beyond this range may make the model overfit. In practice, we fix the range at $[-0.3, 0.3]^2$.

Fig. 9b shows the counting result for different grid granularity. When the granularity is too coarse, i.e., 0.1 granularity, then the error increases significantly. When the granularity is below 0.04, the performance is not too sensitive to the granularity change. Since small granularity means more grids, which corresponds to more memory/computation, we set the granularity as 0.01 in practice.

	CSRNet [14]		VGG19 [22]		MAN [81]	
	MAE	MSE	MAE	MSE	MAE	MSE
MSE	110.6	190.1	98.7	176.1	84.5	148.5
GCFL	83.0	139.8	80.3	137.6	78.6	133.4

TABLE V: Comparison of GCFL and MSE loss for different architectures.

We next demonstrate the ability of GCFL to improve the performance of different network structures. We compare our GCFL with the L2 (MSE) loss on three typical network structures: CNN-based CSRNet [14], VGG19 [22], and transformer-based MAN [81], which are arranged according to their learning ability (the last is the strongest). The experiments are conducted on the QNRF dataset, and results are shown in Table V. GCFL improves over MSE regardless of the network structure used, albeit the improvements diminish as the learning ability of the network becomes stronger (i.e. MAN). Both the designs of the network structure and the loss function can benefit the ability to learn from the training data.

Finally, we note that transformer-based networks (e.g., MAN) create long-range interactions among features, which are then used to predict each output. In contrast, our GCFL applies a frequency transformation on the outputs, which better represents long-range correlations (interactions) among outputs. Thus, the transformer architecture and GCFL are complementary, and as a result, the transformer still benefits from using our GCFL, as compared to the MSE loss.

C. Crowd localization

We next conduct experiments on crowd localization using GCFL for training. To localize people in the predicted density map, we use a similar strategy as the official code of [29]. The output density map will first be upsampled to the same size as the input image, then a 3×3 max pooling with stride 1 is used for finding local maxima. The local maxima that are larger than a threshold are selected as the final localization points. Instead of using a troublesome fine-tuning method to find the proper threshold, we use an automatic method for dynamically deciding the threshold of each density map. Specifically, since we trust the crowd-counting result of our GCFL, the density threshold is set such that the number of localized points is closest to the people count predicted by GCFL-CC.

Our final localization result combines the results from GCFL-CL and GCFL-CC. Specifically, for the localization results from GCFL-CC, we first delete those localization points that are within 60 pixels of another localization point, in order to keep only sparse localization points. Then, we add these remaining points from GCFL-CC to the localization result from GCFL-CL if there is no localization point from GCFL-CL that is within 60 pixels.

1) Comparison with SOTA

We follow the convention to test on NWPU and QNRF datasets. For NWPU, the result is evaluated by the official website, which calculates precision, recall, and F1 measure based on the total true positive number, prediction points number, and ground truth points number. For QNRF, there is no official code for evaluation, and the evaluation method

Algorithm	Prec.	Recall	F1
MCNN [30]	59.93	63.5	61.66
BL [22]	76.70	65.40	70.60
CL [67]	75.80	59.75	66.82
LSC-CNN [42]	74.62	73.50	74.06
DMcount [24]	73.10	63.80	68.13
TopoCount [27]	**81.77**	$\underline{78.96}$	$\underline{80.34}$
GL [25]	78.20	74.80	76.46
GCFL (ours)	$\underline{80.81}$	**80.20**	**80.50**

TABLE VI: Comparison of localization performance on QNRF.

Algorithm	Prec.	Recall	F1
TinyFaces [82]	52.9	61.1	56.7
GPR [83]	55.8	49.6	52.5
RAZ_Loc [84]	66.6	54.3	59.8
Crowd_SDNet [43]	65.1	62.4	63.7
TopoCount [27]	69.5	68.7	69.1
GL [25]	**80.0**	56.2	66.0
P2PNet [26]	72.9	$\underline{69.5}$	$\underline{71.2}$
AutoScale_Loc [85]	67.3	57.4	62.0
CLTR [28]	69.4	67.6	68.5
OT-M [86]	71.0	65.8	68.3
GCFL (ours)	$\underline{73.5}$	**71.5**	**72.5**

TABLE VII: Comparison of localization performance on NWPU test set.

used in the original paper is not clear enough. Thus, we use the evaluation code from the recent SOTA [27]. Specifically, a 1-to-1 matching between the prediction and the ground truth localization points is calculated with distance thresholds from 1 to 100. For each threshold, the precision and recall are calculated based on the mapping result. Then for each image, the average precision and recall values are computed. Finally, the mean of the average precisions and recalls of all the images is reported, along with the F1 measure.

Table VI shows the comparison result on QNRF. Our method obtains the best recall and F1 measure, as well as the second-best precision, which is only inferior to the TopoCount [27]. Note that the TopoCount requires manually setting a box range for heads in the image, which is not needed in our method. Table VII shows that our method also achieves the best recall and F1 measure on the challenging NWPU dataset, while it has the second high precision, only lower than GL [25]. However, GL obtains high precision at the cost of a comparatively lower recall, whereas our method has both high precision and the best recall among all the compared methods.

2) Ablation study

We next conduct an ablation study on QNRF. As we stated in §IV, we do not need to conduct an ablation study on the granularity of the Riemann sum approximation, since it is already conducted in §V-B3.

Integral Range We first explore the appropriate integral range with results shown in Table VIII. Expanding the integral range from $[-0.5, 0.5]^2$ to $[-0.7, 0.7]^2$ diminishes localization performance, which suggests that adding too many local details will incur overfitting. Using integral range $[-0.3, 0.3]^2$ removes more high-frequency components compared with range $[-0.5, 0.5]^2$, which results in underfitting the local information. Thus overall, using $[-0.5, 0.5]^2$ as the integral range is the best choice.

Map Scaling. We investigate the map scaling factor for crowd localization, while setting the integral range to $[-0.5, 0.5]^2$. Table IX shows the results for different map scaling factors. Map scaling is effective in improving the localization results up to a point. Afterwards, larger scaling factors more heavily focus on the localization error, leading to overfitting. Thus, we select factor 10 accordingly.

Integral range	Prec.	Recall	F1
$[-0.3, 0.3]^2$	80.93	78.39	79.64
$[-0.5, 0.5]^2$	**81.24**	**78.73**	**79.96**
$[-0.7, 0.7]^2$	80.41	77.78	79.07

TABLE VIII: Ablation study on the integral range for GCFL-CL.

Map scaling factor	Prec.	Recall	F1
No map scaling	81.24	78.73	79.96
5	81.44	78.91	80.15
10	**81.64**	**79.10**	**80.35**
15	81.42	78.89	80.14
20	81.42	78.87	80.12

TABLE IX: Ablation study on map scaling for GCFL-CL

$H(\mathbf{t})$	Pr	Re	F1
$H_1(\mathbf{t}) = \begin{cases} 1, & \mathbf{t} \in [-0.5, 0.5]^2 \\ 0, & \text{otherwise} \end{cases}$	81.64	79.10	80.35
$H_2(\mathbf{t}) = \begin{cases} 8\|\mathbf{t}\|_2 + 0.6, & \|\mathbf{t}\|_2 \leq 0.05 \\ 1, & \|\mathbf{t}\|_2 > 0.05 \\ & \wedge\ \mathbf{t} \in [-0.5, 0.5]^2 \\ 0, & \text{otherwise} \end{cases}$	81.00	78.47	79.71
$H_3(\mathbf{t}) = \begin{cases} 4\|\mathbf{t}\|_2 + 0.6, & \|\mathbf{t}\|_2 \leq 0.1 \\ 1, & \|\mathbf{t}\|_2 > 0.1 \\ & \wedge\ \mathbf{t} \in [-0.5, 0.5]^2 \\ 0, & \text{otherwise} \end{cases}$	80.97	78.41	79.67
$H_4(\mathbf{t}) = \begin{cases} 0.8, & \|\mathbf{t}\|_2 \leq 0.05 \\ 1, & \|\mathbf{t}\|_2 > 0.05 \\ & \wedge \mathbf{t} \in [-0.5, 0.5]^2 \\ 0, & \text{otherwise} \end{cases}$	80.98	78.46	79.70
$H_5(\mathbf{t}) = \begin{cases} 0.8, & \|\mathbf{t}\|_2 \leq 0.1 \\ 1, & \|\mathbf{t}\|_2 > 0.1 \\ & \wedge\ \mathbf{t} \in [-0.5, 0.5]^2 \\ 0, & \text{otherwise} \end{cases}$	80.95	78.41	79.66
$H_6(\mathbf{t}) = \begin{cases} \frac{1}{\sqrt{h(\mathbf{t}) \cdot M + 1}}, & \mathbf{t} \in [-0.3, 0.3]^2 \\ 1, & \mathbf{t} \in [-0.5, 0.5]^2 \\ & \wedge\ \mathbf{t} \notin [-0.3, 0.3]^2 \\ 0, & \text{otherwise} \end{cases}$	80.90	78.36	79.61

TABLE X: Ablation study on $H(\mathbf{t})$ window function for crowd localization. The $H(\mathbf{t})$ in the final row is from the noise robust window functions (31). We only show results for the best hyperparameter setting for each window.

Window function H. We use the $H(\mathbf{t})$ function to control the behavior or our GCFL. Table X shows the effects of different $H(\mathbf{t})$ for crowd localization. These prototypes of $H(\mathbf{t})$ correspond to different strategies of focusing less on the frequency components around the origin, which is more about the global spatial information. The rationale is that this may make GCFL comparatively focus more on the local spatial information, which might help localization. Windows H_2 and H_3 gradually decay the lower-frequency components (e.g., see Fig. 10a). Windows H_4 and H_5 uniformly give the region around the origin a lower weight (see Fig. 10b). Finally H_6 is from the noise robust window in (31). Here we only use the hyperparameter (coefficient) settings for each window prototype that give the best results. The results in Table X show that all the prototypes obtain inferior localization

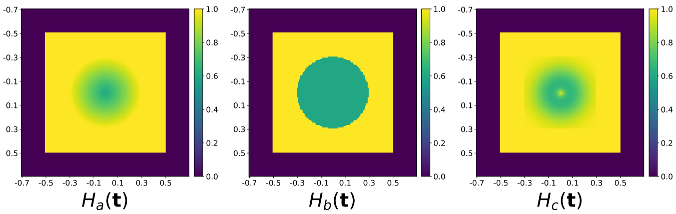


Fig. 10: Three prototypes of $H(\mathbf{t})$ for ignoring the frequency components near the origin. H_a is the prototype of H_2 and H_3 in Table X, H_b corresponds to H_4 and H_5, and H_c is for H_6.

technique used	Prec.	Recall	F1
GCFL-CL	81.24	78.73	79.96
+ map scaling	81.64	79.1	80.35
+ supplement from GCFL-CC	80.81	80.20	80.50

TABLE XI: Ablation study on each term in crowd localization

	NWPU		JHU++		UCF-QNRF		SHTC A		SHTC B	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
GCFL-CC	76.8	343.0	57.0	235.7	80.3	137.6	57.5	94.3	6.9	11.0
GCFL-NCC	77.7	349.6	58.2	239.2	79.5	140.0	55.0	88.5	6.9	11.3

TABLE XII: Crowd counting performance using GCFL-CC and GCFL-NCC on the original datasets (without adding additional annotation noise).

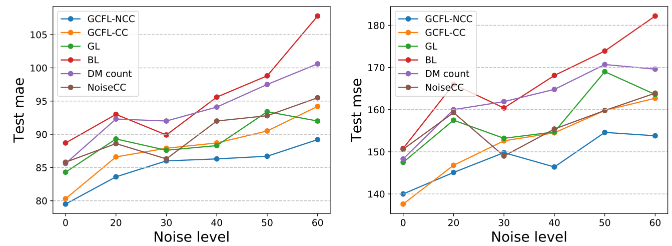
results compared with the window H_1. Thus, the frequency information surrounding the origin, i.e., the global spatial information, is also important for crowd localization of non-sparse heads.

Contribution of each term. To obtain the final localization result, we supplement the results from GCFL-CL with sparse heads localized by GCFL-CC. GCFL-CL is used to train the model for the basic crowd localization result, and map scaling is used to improve the localization result. Table XI shows the effect of each part. The map scaling is useful for improving precision and recall simultaneously. The supplement from the GCFL-CC can further improve the F1 measure.

D. Noisy crowd counting

In this section, we experiment with GCFL-NCC, which uses the noise robust window in (31). We first show that GCFL-NCC does not affect the crowd counting result much on standard (noiseless) data. Table XII shows the crowd counting result comparison between the GCFL-NCC and GCFL-CC (using windows in (24)). On most of these data sets, the GCFL-NCC obtains similar results to the GCFL-CC. On ShanghaiTech A, there is some annotation noise, and hence the result of using the noise robust window is better than using the counting window.

Next we show the robustness of GCFL-NCC on datasets with annotation noise. We simulate noisy data by regenerating the dot map with random sampling. Since the Gaussian distribution is peaked around the origin, we instead adopt a uniform distribution over a disk to increase the noise level. Each original annotation is replaced by a sample randomly selected from a disk uniformly distributed and centered at the original annotation position. The radius of the disk determines the noise level, and in this experiment we use 5 levels of noise with distribution radius in $\{20, 30, 40, 50, 60\}$. We compare with other SOTA losses for crowd counting, and



(a) (b)

Fig. 11: Comparison of SOTA losses on training with noisy dot annotations on QNRF: (a) test MAE and (b) test MSE. The noise level is the radius of the uniform distribution disk used to generate annotation noise. Noise level 0 means no annotation noise.

Fig. 11 presents the result. For different noise levels, GCFL-NCC almost always achieves the lowest MAE and MSE, which suggests its stronger robustness to the annotation noise compared with other losses. On noise level 30, the MSE performance of GCFL-NCC is slightly worse than NoiseCC [23]. However, NoiseCC is sensitive to its hyperparameters – for different noise levels, the hyperparameter α (i.e., the bandwidth of the Gaussian distribution of the annotation noise in their model) needs to be adjusted for each noise level. And in our experiments, α is set as the same value as the noise level. In contrast, for GCFL-NCC, the same hyperparameter setting works for all noise levels in the experiment, and thus GCFL-NCC is robust to its hyperparameter settings and is more practical for real applications.

VI. Conclusion

In this paper, we proposed the GCFL for crowd analysis in the frequency domain. GCFL has two steps: 1) transforming the spatial information to the frequency domain; 2) calculating a loss between the frequency information. In the first step, we established the theoretical foundation by extending the definition of characteristic functions and proving a series of vital properties. In the second step, we used window functions to make GCFL flexible for various tasks, and introduce approximate implementations that are convenient and efficient for real applications.

By exploiting different window functions, GCFL is able to tackle different crowd analysis tasks. We demonstrated three examples in this paper: crowd counting, crowd localization, and noisy crowd counting. In the process of designing the window functions for three tasks, we found some insightful properties of the crowd information in the frequency domain, which indicates its advantage in information organization compared to the spatial domain. Future work will consider devising bespoke window functions for applying GCFL to more crowd analysis tasks, e.g., image-based frameworks for counting everything, which is largely based on MSE loss between density maps [87, 88], and extending GCFL to the spatio-temporal frequency domain for video crowd counting, which may introduce an additional frequency transformation in time and associated temporal windows based on people's motion constraints.

Acknowledgment

This work was supported by a Strategic Research Grant from City University of Hong Kong (Project No. 7005665).

References

[1] C. Arteta, V. Lempitsky, and A. Zisserman, "Counting in the wild," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*. Springer, 2016, pp. 483–498.

[2] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. Schmidt, "Where are the blobs: Counting by localization with point supervision," in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 547–562.

[3] Z. Ma, L. Yu, and A. B. Chan, "Small instance detection by integer programming on object density maps," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3689–3697.

[4] J. Paul Cohen, G. Boucher, C. A. Glastonbury, H. Z. Lo, and Y. Bengio, "Count-ception: Counting by fully convolutional redundant counting," in *Proceedings of the IEEE International conference on computer vision workshops*, 2017, pp. 18–26.

[5] V. Lempitsky and A. Zisserman, "Learning to count objects in images," *Advances in neural information processing systems*, vol. 23, pp. 1324–1332, 2010.

[6] M. von Borstel, M. Kandemir, P. Schmidt, M. K. Rao, K. Rajamani, and F. A. Hamprecht, "Gaussian process density counting from weak supervision," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 365–380.

[7] E. Walach and L. Wolf, "Learning to count with cnn boosting," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 660–676.

[8] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal network," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4145–4153.

[9] M. Marsden, K. McGuinness, S. Little, C. E. Keogh, and N. E. O'Connor, "People, penguins and petri dishes: Adapting object counting models to new visual domains and object types without forgetting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8070–8079.

[10] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye, "A large contextual dataset for classification, detection and counting of cars with deep learning," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 785–800.

[11] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura, "Fcn-rlstm: Deep spatio-temporal neural networks for vehicle counting in city cameras," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3667–3676.

[12] L. Liu, H. Wang, G. Li, W. Ouyang, and L. Lin, "Crowd counting using deep recurrent spatial-aware network," *arXiv preprint arXiv:1807.00601*, 2018.

[13] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang, "Crowd counting via adversarial cross-scale consistency pursuit," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5245–5254.

[14] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1091–1100.

[15] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734–750.

[16] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5099–5108.

[17] A. Zhang, J. Shen, Z. Xiao, F. Zhu, X. Zhen, X. Cao, and L. Shao, "Relational attention network for crowd counting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6788–6797.

[18] Y. Yang, G. Li, Z. Wu, L. Su, Q. Huang, and N. Sebe, "Reverse perspective network for perspective-aware object counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4374–4383.

[19] S. Bai, Z. He, Y. Qiao, H. Hu, W. Wu, and J. Yan, "Adaptive dilated network with self-correction supervision for counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4594–4603.

[20] U. Sajid, X. Chen, H. Sajid, T. Kim, and G. Wang, "Audio-visual transformer based crowd counting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2249–2259.

[21] Z. Liu, W. Wu, Y. Tan, and G. Zhang, "Rgb-t multi-modal crowd counting based on transformer," *arXiv preprint arXiv:2301.03033*, 2023.

[22] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6142–6151.

[23] J. Wan and A. Chan, "Modeling noisy annotations for crowd counting," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[24] B. Wang, H. Liu, D. Samaras, and M. H. Nguyen, "Distribution matching for crowd counting," *Advances in neural information processing systems*, vol. 33, pp. 1595–1607, 2020.

[25] J. Wan, Z. Liu, and A. B. Chan, "A generalized loss function for crowd counting and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1974–1983.

[26] Q. Song, C. Wang, Z. Jiang, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Wu, "Rethinking counting and localization in crowds: A purely point-based framework," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3365–3374.

[27] S. Abousamra, M. Hoai, D. Samaras, and C. Chen, "Localization in the crowd with topological constraints," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 872–881.

[28] D. Liang, W. Xu, and X. Bai, "An end-to-end transformer model for crowd localization," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*. Springer, 2022, pp. 38–54.

[29] Q. Wang, J. Gao, W. Lin, and X. Li, "Nwpu-crowd: A large-scale benchmark for crowd counting and localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[30] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 589–597.

[31] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[32] G. Peyré, M. Cuturi *et al.*, "Computational optimal transport: With applications to data science," *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[33] W. Shu, J. Wan, K. C. Tan, S. Kwong, and A. B. Chan, "Crowd counting in the frequency domain," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 618–19 627.

[34] W. Ge and R. T. Collins, "Marked point processes for crowd counting," in *2009 IEEE Conference on Computer Vision and*

Pattern Recognition. IEEE, 2009, pp. 2913–2920.

[35] M. Li, Z. Zhang, K. Huang, and T. Tan, "Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection," in *2008 19th international conference on pattern recognition*. IEEE, 2008, pp. 1–4.

[36] S.-F. Lin, J.-Y. Chen, and H.-X. Chao, "Estimation of number of people in crowded scenes using perspective transformation," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 31, no. 6, pp. 645–654, 2001.

[37] R. Stewart, M. Andriluka, and A. Y. Ng, "End-to-end people detection in crowded scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2325–2333.

[38] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen, "Repulsion loss: Detecting pedestrians in a crowd," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7774–7783.

[39] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Occlusion-aware r-cnn: Detecting pedestrians in a crowd," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 637–653.

[40] T. Zhao and R. Nevatia, "Bayesian human segmentation in crowded situations," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 2. IEEE, 2003, pp. II–459.

[41] Y. Liu, M. Shi, Q. Zhao, and X. Wang, "Point in, box out: Beyond counting persons in crowds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6469–6478.

[42] D. B. Sam, S. V. Peri, M. N. Sundararaman, A. Kamath, and V. B. Radhakrishnan, "Locate, size and count: Accurately resolving people in dense crowds via detection," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[43] Y. Wang, J. Hou, X. Hou, and L.-P. Chau, "A self-training approach for point-supervised object detection and counting in crowds," *IEEE Transactions on Image Processing*, vol. 30, pp. 2876–2887, 2021.

[44] D. Lian, X. Chen, J. Li, W. Luo, and S. Gao, "Locating and counting heads in crowds with a depth prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 9056–9072, 2021.

[45] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–7.

[46] K. Chen, S. Gong, T. Xiang, and C. Change Loy, "Cumulative attribute space for age and crowd density estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2467–2474.

[47] B. Liu and N. Vasconcelos, "Bayesian model adaptation for crowd counts," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4175–4183.

[48] P. Chattopadhyay, R. Vedantam, R. R. Selvaraju, D. Batra, and D. Parikh, "Counting everyday objects in everyday scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1135–1144.

[49] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Crowd counting using multiple local features," in *2009 Digital Image Computing: Techniques and Applications*. IEEE, 2009, pp. 81–88.

[50] C. Shang, H. Ai, and B. Bai, "End-to-end crowd counting via joint learning local and global count," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1215–1219.

[51] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1299–1302.

[52] W. Liu, M. Salzmann, and P. Fua, "Counting people by estimating people flows," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 11, pp. 8151–8166, 2021.

[53] X. Liu, J. Van De Weijer, and A. D. Bagdanov, "Exploiting unlabeled data in cnns by self-supervised learning to rank," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1862–1878, 2019.

[54] Y. Liu, S. Ren, L. Chai, H. Wu, D. Xu, J. Qin, and S. He, "Reducing spatial labeling redundancy for active semi-supervised crowd counting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[55] W. Liu, M. Salzmann, and P. Fua, "Estimating people flows to better count them in crowded scenes," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*. Springer, 2020, pp. 723–740.

[56] J. Wan, Q. Wang, and A. B. Chan, "Kernel-based density map generation for dense object counting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[57] J. Wan and A. Chan, "Adaptive density map generation for crowd counting," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1130–1139.

[58] Y. Wang, C. Xu, S. You, D. Tao, and C. Xu, "Cnnpack: Packing convolutional neural networks in the frequency domain," *Advances in neural information processing systems*, vol. 29, 2016.

[59] Y. Wang, C. Xu, C. Xu, and D. Tao, "Packing convolutional neural networks in the frequency domain," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 10, pp. 2495–2510, 2018.

[60] Y. Rao, W. Zhao, Z. Zhu, J. Lu, and J. Zhou, "Global filter networks for image classification," *Advances in neural information processing systems*, vol. 34, pp. 980–993, 2021.

[61] Y. Rao, W. Zhao, Z. Zhu, J. Zhou, and J. Lu, "Gfnet: Global filter networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[62] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, and N. Yu, "Spatial-phase shallow learning: rethinking face forgery detection in frequency domain," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 772–781.

[63] B. Zheng, S. Yuan, G. Slabaugh, and A. Leonardis, "Image demoireing with learnable bandpass filters," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3636–3645.

[64] B. Zheng, S. Yuan, C. Yan, X. Tian, J. Zhang, Y. Sun, L. Liu, A. Leonardis, and G. Slabaugh, "Learning frequency domain priors for image demoireing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7705–7717, 2021.

[65] S. J. Taylor, "Set functions," in *Introduction to Measure and Integration*. Cambridge: Cambridge University Press, 1973, p. 51–73.

[66] P. Billingsley, *Probability and measure*. John Wiley & Sons, 2008.

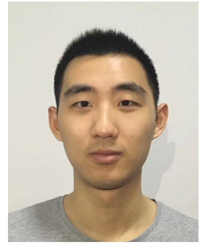
[67] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah, "Composition loss for counting, density map estimation and localization in dense crowds," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 532–546.

[68] V. A. Sindagi, R. Yasarla, and V. M. Patel, "Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method," *Technical Report*, 2020.

[69] ——, "Pushing the frontiers of unconstrained crowd counting: New dataset and benchmark method," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1221–1231.

[70] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[71] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[72] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8198–8207.
[73] X. Liu, J. Yang, W. Ding, T. Wang, Z. Wang, and J. Xiong, "Adaptive mixture regression network with local counting map for crowd counting," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*. Springer, 2020, pp. 241–257.
[74] J. T. Zhou, L. Zhang, D. Jiawei, X. Peng, Z. Fang, Z. Xiao, and H. Zhu, "Locality-aware crowd counting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
[75] C. Xu, D. Liang, Y. Xu, S. Bai, W. Zhan, X. Bai, and M. Tomizuka, "Autoscale: Learning to scale for crowd counting and localization," *arXiv preprint arXiv:1912.09632*, 2019.
[76] Z. Ma, X. Hong, X. Wei, Y. Qiu, and Y. Gong, "Towards a universal model for cross-dataset crowd counting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3205–3214.
[77] M. Wang, H. Cai, Y. Dai, and M. Gong, "Dynamic mixture of counter network for location-agnostic crowd counting," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 167–177.
[78] H. Xiong, H. Lu, C. Liu, L. Liu, C. Shen, and Z. Cao, "From open set to closed set: Supervised spatial divide-and-conquer for object counting," *International Journal of Computer Vision*, vol. 131, no. 7, pp. 1722–1740, 2023.
[79] H. Xiong and A. Yao, "Discrete-constrained regression for local counting models," in *European Conference on Computer Vision*. Springer, 2022, pp. 621–636.
[80] Z.-Q. Cheng, Q. Dai, H. Li, J. Song, X. Wu, and A. G. Hauptmann, "Rethinking spatial invariance of convolutional networks for object counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 638–19 648.
[81] H. Lin, Z. Ma, R. Ji, Y. Wang, and X. Hong, "Boosting crowd counting via multifaceted attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 628–19 637.
[82] P. Hu and D. Ramanan, "Finding tiny faces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 951–959.
[83] J. Gao, T. Han, Q. Wang, and Y. Yuan, "Domain-adaptive crowd counting via inter-domain features segregation and gaussian-prior reconstruction," *arXiv preprint arXiv:1912.03677*, vol. 2, no. 5, 2019.
[84] C. Liu, X. Weng, and Y. Mu, "Recurrent attentive zooming for joint crowd counting and precise localization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1217–1226.
[85] C. Xu, D. Liang, Y. Xu, S. Bai, W. Zhan, X. Bai, and M. Tomizuka, "Autoscale: learning to scale for crowd counting," *International Journal of Computer Vision*, vol. 130, no. 2, pp. 405–434, 2022.
[86] W. Lin and A. B. Chan, "Optimal transport minimization: Crowd localization on density maps for semi-supervised counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 663–21 673.
[87] A. Michel, W. Gross, F. Schenkel, and W. Middelmann, "Class-aware object counting," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 469–478.
[88] H. Go, J. Byun, B. Park, M.-A. Choi, S. Yoo, and C. Kim, "Fine-grained multi-class object counting," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 509–513.



Weibo Shu received the B.Eng. degree from Xiamen University, Xiamen, China, in 2016, and the M.Phil. degree from University of Science and Technology of China, Hefei, China, in 2019. Now he is pursuing his Ph.D. degree in City University of Hong Kong, Kowloon, Hong Kong, under the supervision of Prof. Antoni B. Chan. His current research interests include crowd analysis and machine learning.

Jia Wan received the B.Eng. and M.Phil. degrees from Northwestern Polytechnical University, Xi'an, China, in 2015 and 2018, respectively. He then received his Ph.D. degree in Computer Science from City University of Hong Kong, in 2021. He is currently a postdoctoral scholar in the Department of Electrical & Computer Engineering, University of California, San Diego (UCSD). His research interests include crowd counting, human pose estimation, monocular depth prediction and bird-eye-view (BEV) object detection.

Antoni B. Chan received the B.S. and M.Eng. degrees in electrical engineering from Cornell University, Ithaca, NY, in 2000 and 2001, and the Ph.D. degree in electrical and computer engineering from the University of California, San Diego (UCSD), San Diego, in 2008. He is currently a Professor in the Department of Computer Science, City University of Hong Kong. His research interests include computer vision, machine learning, pattern recognition, and music analysis.