2D Gaussians Spatial Transport for Point-supervised Density Regression

Miao Shang, Xiaopeng Hong*

The Faculty of Computing, Harbin Institute of Technology
miaos0522@gmail.com, hongxiaopeng@ieee.org

Abstract

This paper introduces Gaussian Spatial Transport (GST), a novel framework that leverages Gaussian splatting to facilitate transport from the probability measure in the image coordinate space to the annotation map. We propose a Gaussian splatting-based method to estimate pixel-annotation correspondence, which is then used to compute a transport plan derived from Bayesian probability. To integrate the resulting transport plan into standard network optimization in typical computer vision tasks, we derive a loss function that measures discrepancy after transport. Extensive experiments on representative computer vision tasks, including crowd counting and landmark detection, validate the effectiveness of our approach. Compared to conventional optimal transport schemes, GST eliminates iterative transport plan computation during training, significantly improving efficiency.

Code — https://github.com/infinite0522/GST

Introduction

Optimal transport (OT) has gained great attention in computer vision due to its powerful ability to model and solve various problems involving distances and distributions (Peyré, Cuturi et al. 2019; Villani 2021; Wang et al. 2020a; Lin and Chan 2023; Zhang et al. 2024a; Ge et al. 2021). By treating the prediction and the ground truth as probability distributions, OT determines their optimal match with the transport cost measured by, for example, the Wasserstein distance (Kantorovich 1960).

However, despite their effectiveness, the application of OT in deep learning is plagued by high computational costs. This issue stems from their inherent bi-level optimization nature: at each training iteration, an inner loop must solve a costly OT problem (e.g., via the Sinkhorn algorithm (Cuturi 2013)) to compute the loss, before an outer loop can update the network's parameters. This tight coupling makes the entire optimization process extremely time-consuming.

To address the computational bottleneck, the primary idea of this paper is to decouple the transport plan generation from the network optimization. The goal is to pre-compute a fixed transport pattern based on the input image and its annotations, and then use it throughout the training. However, this seemingly simple idea poses significant challenges.

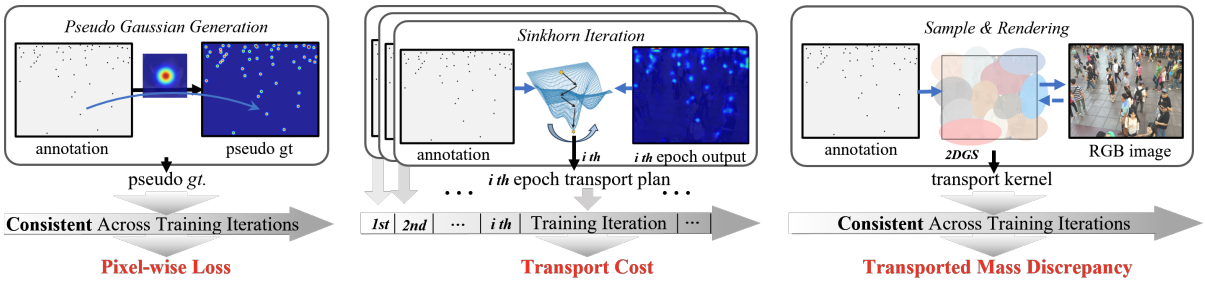
First, standard *Optimal Transport* lacks a mechanism to derive a transport plan from the input image prior to network optimization. OT treats the estimated density maps as probability measures in the image coordinate space and solves for the optimal plan to the annotation space. As a result, it requires the network to first generate the density map, followed by the computation of the transport plan and cost. Second, in practical applications such as crowd counting, the images often contain thousands of targets. To minimize annotation costs, only object locations are typically provided without detailed contours (Cao et al. 2018; Liu, Salzmann, and Fua 2019; Lin et al. 2025). However, object contours are crucial for determining the correspondence between pixels and annotation points, which directly impacts the transport plan. Without contour information, accurately establishing pixel-to-object correspondence becomes challenging.

In response to these challenges, we propose Gaussian Spatial Transport (GST), a novel framework that efficiently transports probability measures from image coordinates to annotation space. At its core, GST establishes an interpretable probabilistic correspondence between pixels and annotations by leveraging 2D Gaussian Splatting. This correspondence is then distilled into a fixed transport kernel, a matrix that encodes the static mapping from each pixel to all target annotations, which can be pre-computed before training. During optimization, the model's predicted density map is pushed forward to the space of ground truth annotations via a single matrix multiplication with this pre-computed kernel. Whereas OT-based methods require iteratively solving a costly optimization problem to determine the transport plan and compute the loss, our loss is simply the discrepancy between the transported density and the ground truth. Extensive experiments demonstrate that GST achieves a strong balance of high accuracy and computational efficiency.

The main contributions of this paper are threefold:

- We propose a novel spatial transport framework that decouples plan generation from network optimization. It relies on a pre-computed transport kernel, which transforms the costly iterative optimization of OT into a single, efficient matrix multiplication during training.
- We bridge Gaussian splatting and spatial transport by de-

*Corresponding author.



(a) Direct Regression (b) Optimal Transport (c) Gaussian Spatial Transport

Figure 1: Comparison of loss computing of Direct Regression, OT, and the proposed GST.

riving a Bayesian framework to compute the transport kernel using results obtained from 2D Gaussian splatting.

- We successfully apply the proposed GST framework to representative computer vision tasks, including crowd counting and landmark detection.

Related work

Spatial Transport

Conventional point-supervised density regression methods (Sindagi and Patel 2017; Li, Zhang, and Chen 2018; Sun et al. 2019) rely on handcrafted Gaussian kernels to generate pseudo-ground-truth density maps for direct pixel-to-pixel regression, creating a critical dependency on the quality of these synthetic approximations, as in Fig. 1(a). Although subsequent improvements (Zhang et al. 2016; Idrees et al. 2018; Wan, Wang, and Chan 2020) introduce adaptive kernels to enhance quality, they still suffer from inherent localization ambiguity in high-density scenarios.

Recent works (Wang et al. 2020a; Qu et al. 2022) mitigate these limitations by reframing the task as distribution matching via transport theory. Based on Optimal Transport, they treat the predicted density map and the ground truth as two measures, and define the training loss through the minimal cost to transport one to the other. For instance, DM-Count (Wang et al. 2020a) first introduced this using balanced OT, while subsequent works explored unbalanced OT to relax the mass conservation constraint (Ma et al. 2021; Lin et al. 2021; Wan, Liu, and Chan 2021). However, it introduces a significant computational bottleneck due to the inherent bi-level optimization nature: at each training step, an inner-loop optimization (e.g., using the Sinkhorn algorithm) must be solved to find the optimal transport plan, before the outer-loop can update the network's parameters. This iterative re-computation of transport plans makes the entire process time-consuming. Through BL (Ma et al. 2019) avoids this cost by using a fixed, handcrafted transport plan, the reliance on a task-specific prior limits its adaptability.

Our work addresses the limitations of both. We propose a novel transport scheme, which decouples the transport plan from the network optimization. Instead of using a handcrafted prior, we pre-compute a transport kernel by analyzing the input image's content via Gaussian Splatting. During training, the transport is reduced to a single, efficient matrix multiplication, enabling high performance with significant computational gains. Fig. 1 illustrates the key differences between our method and other major paradigms like direct regression and OT.

Gaussian Splatting

Gaussian splatting (GS) is widely used for efficient appearance modeling and rendering. It approximates complex textures and colors of 3D scenes with overlapping Gaussian balls. Benefiting from the explicit differentiable rendering and GPU-parallelized, tiles-based rasterization pipeline, GS has become a new paradigm for fast differentiable rendering, making it a popular choice for real-time 3D scene reconstruction and editing (Kerbl et al. 2023; Huang et al. 2024).

Recently, GS's application has expanded to 2D tasks like image reconstruction (Zhu et al. 2025; Dong et al. 2025; Ye et al. 2024), compression (Zhang et al. 2024b; Wang, Shi, and Ooi 2025), and super-resolution (Hu et al. 2024; Chen et al. 2025). In the 2D image domain, Gaussian units can serve as novel representation units, replacing traditional pixels. For instance, GaussianImage (Zhang et al. 2024b) innovatively replaces pixels with Gaussian units for image representation, further enhancing efficiency by simplifying the parameterization of Gaussian representations and the computationally intensive α-blending process in 2D scenes.

Existing methods primarily utilize 2D GS for image reconstruction and restoration. Instead, our work pioneers a novel application of Gaussian Splatting via its connection to spatial transport. We rely on its implicit geometric encoding ability to establish the pixel-to-target correspondence and formulate a transport plan that maps the measures from coordinate spaces to annotation spaces, which is clearly beyond the scope of previous GS studies.

Method

In this section, we elaborate on our Gaussian Spatial Transport method. We begin with the problem definition and preliminaries. As the design of the loss function is paramount to address the computational bottleneck of OT-based approaches, we provide a general analysis of transport-based losses, which is distinct to the conventional OT formulation with our proposed Bayesian Transport framework. On this basis, the subsequent sections will then detail the full GST pipeline: the generation of its transport kernel via Gaussian Splatting and the training procedure.

Problem Definition. Let $\mathcal{X}=\{x_i\}_{i=1}^{I}$ be pixel coordinates of an RGB image $\boldsymbol{\zeta}_{\text{img}}$ and $\mathcal{Y}=\{y_n\}_{n=1}^{N}$ be the discrete ground truth point locations. The density map $\boldsymbol{\zeta}_d \in \mathbb{R}_+^I$ assigns non-negative density values on each pixel in $\mathcal{X}$. The annotation map $\boldsymbol{\zeta}_g \in \mathbb{R}_+^N$ is defined on $\mathcal{Y}$ with $\zeta_g(y_n)=1$ for all n, representing the once presence of each object. A regression network f with trainable parameter θ is adopted to approximate the mapping from input to ideal density map $\boldsymbol{\zeta}_d$. The loss function for network training is denoted by $L(\cdot)$.

Preliminaries on Transport. To establish a quantitative relationship between the density map and annotation points, we first formalize the concept of transport.

By normalizing the two maps into probability functions $\boldsymbol{P}_X = \frac{\boldsymbol{\zeta}_d}{||\boldsymbol{\zeta}_d||_1}$ and $\boldsymbol{P}_Y = \frac{\boldsymbol{\zeta}_g}{||\boldsymbol{\zeta}_g||_1}$, we define the their probability measures $\boldsymbol{\mu} = \sum_{i=1}^{I} P_X(x_i)\delta_{x_i}$ on $\mathcal{X}$ and $\boldsymbol{\nu} = \sum_{n=1}^{N} P_Y(y_n)\delta_{y_n}$ on $\mathcal{Y}$, where δ_{x_i}, δ_{y_n} are Dirac measures.

A *spatial transport* between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ is a joint probability measure $\boldsymbol{P} = \sum_{i,n} P(x_i, y_n)\delta_{(x_i,y_n)}$ on $\mathcal{X}\times\mathcal{Y}$ that satisfies the marginal constraints. The admissible set is:

$$\mathcal{U}(\boldsymbol{P}_X, \boldsymbol{P}_Y) \triangleq \{\boldsymbol{P} \in \mathbb{R}_+^{I\times N} : \boldsymbol{P}\mathbf{1}_N = \boldsymbol{P}_X,\, \boldsymbol{P}'\mathbf{1}_I = \boldsymbol{P}_Y\}. \quad (1)$$

General Form of Transport Plan-based Loss

Conceptually, any transport-based optimization loss function combines two key aspects: *transport discrepancy*, which measures the difference between the resulting distribution of transported mass and the desired target distribution, and *transport cost*, referring to the expense required to move mass from source to target. Given the density map $\tilde{\boldsymbol{\zeta}}_d^t = f(\boldsymbol{\zeta}_{\text{img}}; \theta_t)$ estimated by f at training iteration t and transport plan $\boldsymbol{P}$, a generic loss function is formulated as:

$$L(\tilde{\boldsymbol{\zeta}}_d^t, \boldsymbol{\zeta}_g) = \lambda_1 D\big(\tau(\tilde{\boldsymbol{\zeta}}_d^t), \boldsymbol{\zeta}_g\big) + \lambda_2 W(\boldsymbol{P}), \quad (2)$$

where $\tau : \mathcal{X} \to \mathcal{Y}$ signifies the push-forward movement of the source mass transported from $\mathcal{X}$ to $\mathcal{Y}$. The loss is a weighted sum of two terms: the transport discrepancy, measured by $D(\cdot)$, and the transport cost, given by $W(\cdot)$, which are balanced by parameters $\lambda_1, \lambda_2 \geq 0$.

Existing Optimal Transport-based methods (Wang et al. 2020a; Qu et al. 2022) define their loss by finding the minimal-cost transport from model prediction to ground truth in each training iteration. We, instead, derive a pre-computable transport scheme, termed *Bayesian Transport*, realizing our loss under this fixed-pattern transport.

Optimal Transport-based Loss OT-based methods first solve for the minimal cost transport plan $\boldsymbol{P}_*^t$ between the normalized predicted density $\tilde{\boldsymbol{P}}_X^t = \frac{\tilde{\boldsymbol{\zeta}}_d^t}{||\tilde{\boldsymbol{\zeta}}_d^t||_1}$ and the ground truth $\boldsymbol{P}_Y$, and define the cost term in Eq. 2 under this $\boldsymbol{P}_*^t$:

$$W(\boldsymbol{P}_*^t) = \min_{\boldsymbol{P}^t} \langle \mathbf{C}, \boldsymbol{P}^t \rangle, \;\; s.t.\; \boldsymbol{P}^t \in \mathcal{U}(\tilde{\boldsymbol{P}}_X^t, \boldsymbol{P}_Y), \quad (3)$$

where $\mathbf{C} \in \mathbb{R}_+^{I\times N}$ is the cost matrix.

For the transport discrepancy term, it's important to note two aspects. since $\boldsymbol{P}_*^t$ is solved for normalized probability distributions, $D\big(\tau(\tilde{\boldsymbol{P}}_X^t), \boldsymbol{P}_Y\big) = 0$ due to marginal constraints. However, a mass discrepancy still remains when transporting unnormalized distributions (proportionally by $\boldsymbol{P}_*^t$), defined as $D\big(\tau(\tilde{\boldsymbol{\zeta}}_d^t), \boldsymbol{\zeta}_g\big) = \big|\|\tilde{\boldsymbol{\zeta}}_d^t\|_1 - \|\boldsymbol{\zeta}_g\|_1\big|$ in DM-Count (Wang et al. 2020a), which can be proven equivalent to the L_1 norm of the difference between the pushed-forward unnormalized distribution and the ground truth $\boldsymbol{\zeta}_g$[1].

This approach leads to a **bi-level optimization**: the inner loop solves the OT problem in Eq. 3 through iterative computation within each training step, while the outer loop updates the task-specific model parameters θ across iterations. This tight coupling requires recomputing $\boldsymbol{P}_*^t$ at every step of training, leading to significant computational overhead.

Bayesian Transport-based Loss To address OT's computational overhead, we propose a Bayesian Transport-based Loss. This method derives a pre-computable transport plan from probabilistic principles, decoupling its calculation from the regression model's iterative optimization. Its foundation lies in the following theorem.

Theorem 1. *For probability distributions $\boldsymbol{P}_X$ on $\mathcal{X}$ and $\boldsymbol{P}_Y$ on $\mathcal{Y}$, there exists a transport plan $\hat{\boldsymbol{P}} \in \mathcal{U}(\boldsymbol{P}_X, \boldsymbol{P}_Y)$ that can be expressed as $\hat{\boldsymbol{P}} = diag(\boldsymbol{P}_X)\cdot\boldsymbol{\mathcal{K}}$. Here, $\boldsymbol{\mathcal{K}}$, termed transport kernel, has elements defined as* [2]*:*

$$\boldsymbol{\mathcal{K}}_{i,n} = \frac{P(x_i|y_n)P_Y(y_n)}{\sum_{n=1}^{N} P(x_i|y_n)P_Y(y_n)}. \quad (4)$$

From problem definition, $P_Y(y_n)=\zeta_g(y_n)/||\boldsymbol{\zeta}_g||_1=1/N$ for all n. Substituting this uniform probability into Eq. 1, the kernel can be further simplified as:

$$\boldsymbol{\mathcal{K}}_{i,n} = \frac{P(x_i|y_n)/N}{\sum_{n=1}^{N} P(x_i|y_n)/N} = \frac{P(x_i|y_n)}{\sum_{n=1}^{N} P(x_i|y_n)}, \quad (5)$$

As suggested in Eq. 5, the kernel becomes fixed and pre-computable when the conditional distribution $P(x_i|y_n)$ is known or approximated (e.g., via a Gaussian assumption).

To define the loss function, we first discuss the transport cost term $W(\hat{\boldsymbol{P}})$. From Theorem 1, $\hat{\boldsymbol{P}}$ is defined by a predeterminable $\boldsymbol{\mathcal{K}}$ and $\boldsymbol{P}_X$ reflecting input's ideal density. Since $\hat{\boldsymbol{P}}$ remains fixed during training, its cost is constant and does not contribute to model parameter optimization. Consequently, we effectively set the coefficient $\lambda_2 = 0$.

We focus solely on the transport discrepancy term $D\big(\tau(\tilde{\boldsymbol{\zeta}}_d^t), \boldsymbol{\zeta}_g\big)$. Theorem 1 ensures $\big\|\boldsymbol{\mathcal{K}}'\boldsymbol{P}_X - \boldsymbol{P}_Y\big\|_1 = 0$. For the ideal density distribution, where $\|\boldsymbol{\zeta}_d\|_1 = \|\boldsymbol{\zeta}_g\|_1$, by scaling with their mass sums, $\big\|\boldsymbol{\mathcal{K}}'\boldsymbol{\zeta}_d - \boldsymbol{\zeta}_g\big\|_1 = 0$ holds. Consequently, for the estimated density distribution $\tilde{\boldsymbol{\zeta}}_d^t$, when the mass is proportionally pushed forward by $\boldsymbol{\mathcal{K}}$, the term $\Big\|\boldsymbol{\mathcal{K}}'\tilde{\boldsymbol{\zeta}}_d^t - \boldsymbol{\zeta}_g\Big\|_1 \geqslant 0$ reflects the aligned mass discrepancy $D\big(\tau(\tilde{\boldsymbol{\zeta}}_d^t), \boldsymbol{\zeta}_g\big)$. Therefore, the Bayesian transport-based loss is defined as:

$$L_{BT} = \Big\|\boldsymbol{\mathcal{K}}'\tilde{\boldsymbol{\zeta}}_d^t - \boldsymbol{\zeta}_g\Big\|_1. \quad (6)$$

[1] For more discussions about the definition of $D(\cdot)$ in other methods, please refer to the *Suppl. A*.

[2] The kernel is derived following Bayes' theorem and the law of total probability. Proof can be referred to *Suppl. B*.

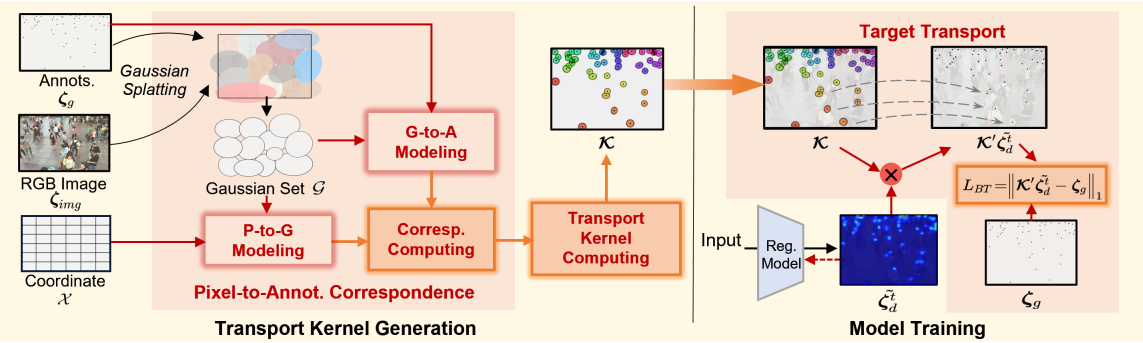


Figure 2: The GST Pipeline comprises two main components: transport kernel generation and model training. First, the transport kernel $\mathcal{K}$ is generated before training by reconstructing the RGB image via 2D Gaussian splatting and establishing pixel-to-annotation correspondences (Eq. 7) to then form $\mathcal{K}$ (Eq. 5). Second, during training, $\mathcal{K}$ transports the estimated density map to annotations, allowing for the computation of the transported mass discrepancy loss (Eq. 6).

As established, the efficacy of Bayesian Transport hinges on the pre-computation of the transport kernel $\mathcal{K}$, which in turn depends on estimating the conditional probability $P(x_i|y_n)$. We introduce Gaussian Spatial Transport as our concrete realization of this framework. GST leverages 2D Gaussian Splatting to effectively model the spatial relationships within the image and derive a high-quality kernel. The following sections will detail the complete GST pipeline, from kernel generation to model training, as shown in Fig. 2.

Transport Kernel Generation via 2DGS

Pixel to annotation Correspondence To construct $\mathcal{K}$, the core challenge lies in accurately approximating $P(x_i|y_n)$. We achieve this by using 2D Gaussian Splatting to explicitly model the underlying spatial correspondence between pixels and annotations, as its differentiable probabilistic representation is ideal for capturing the geometric features that govern this correspondence. The entire process is formally expressed using the law of total probability:

$$\begin{aligned} P(x_i|y_n) &= \sum\nolimits_{m=1}^{M} P(x_i|y_n, G_m)P(G_m|y_n) \\ &= \sum\nolimits_{m=1}^{M} P(x_i|G_m)P(G_m|y_n), \end{aligned} \tag{7}$$

where the Gaussian ellipse $G_m \in \mathcal{G}$ is obtained via Gaussian Splatting and encodes the spatial distribution of the input image. The final equality in Eq. 7 holds due to the conditional independence of x and y given G_m. This effectively decomposes the original pixel-to-annotation correspondence problem into two manageable steps: computing the pixel-to-Gaussian correspondence $P(x_i|G_m)$ and the Gaussian-to-annotation correspondence $P(G_m|y_n)$. Next, we elaborate on how to compute these two components.

To establish **pixel-to-Gaussian** correspondences, we use Gaussian splatting (Kerbl et al. 2023) to model the image as a composition of anisotropic Gaussian ellipses with fields of color and transparency. The image is approximated as:

$$\tilde{\boldsymbol{\zeta}}_{\text{img}}(x_i) = \sum\nolimits_{m=1}^{M} \alpha_m \boldsymbol{c}_m \exp\left(-\frac{1}{2} d_M^2(x_i, G_m)\right), \tag{8}$$
$$\text{where } d_M(x_i, G_m) = (x_i - \boldsymbol{\mu}_m)^\top \boldsymbol{\Sigma}_m^{-1}(x_i - \boldsymbol{\mu}_m),$$

where each Gaussian component combines visual properties for rendering (opacity α_m and color $\boldsymbol{c}_m$) with geometric parameters (mean $\boldsymbol{\mu}_m$ and covariance $\boldsymbol{\Sigma}_m$) that determine its spatial distribution. $\boldsymbol{\Sigma}_m = \boldsymbol{R}_m \boldsymbol{S}_m (\boldsymbol{R}_m \boldsymbol{S}_m)'$ is parameterized by the scaling matrix $\boldsymbol{S}_m$ and rotation matrix $\boldsymbol{R}_m$, ensuring positive semi-definiteness during gradient-based optimization, as in (Kerbl et al. 2023). The pixel-to-Gaussian correspondence is derived *exclusively* from the spatial probability distribution:

$$P(x_i|G_m) = \mathcal{N}(x_i; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m), \quad G_m \in \mathcal{G} \tag{9}$$

Deformity Elimination. To ensure Gaussians match annotated objects, we introduce a shape control regularization to maintain reasonable shapes:

$$L_{shape} = \max_m(\max(s_{ml}/s_{ms} - \delta, 0)), \tag{10}$$

where s_{ml},s_{ms} denote the scales along the major and minor axes of the Gaussian ellipse, respectively, and $\delta = 1.5$ controls the maximum permissible aspect ratio. This loss term prevents unconstrained Gaussians from forming excessively elongated deformations that could erroneously focus on irrelevant narrow regions, thereby maintaining accurate spatial correspondence between pixels and annotations. The total loss for Gaussian splatting process becomes $L_{gs} = L_{rec} + \beta L_{shape}$[3], where L_{rec} is the reconstruction loss (Zhang et al. 2024b) implemented by mean square distance between $\tilde{\boldsymbol{\zeta}}_{img}$ and $\boldsymbol{\zeta}_{\text{img}}$.

With Gaussians generated, we model the **Gaussian-to-Annotation** correspondence by a similarity metric, i.e., $P(G_m|y_n) = \frac{\exp(\text{sim}(G_m, y_n))}{\sum_m \exp(\text{sim}(G_m, y_n))}$. While it can be computed by evaluating the Gaussian probability density function at the annotation point, the pairwise computation of annotation-Gaussian similarities becomes computationally exhaustive and memory-intensive when dealing with large numbers of Gaussians and annotations. Moreover, in dense scenarios (e.g., in the task of crowd-counting), the image region corresponding to an annotation point is typically small and approximately elliptical.

[3] L_{gs} is specifically used to optimize the parameters of the Gaussians, rather than those of the task-tailored model f.

Based on above analysis, we simplify the correspondence modeling by establishing a one-to-one correspondence between annotations and Gaussians through a pre-assignment strategy within the Gaussian splatting process. Specifically, we partition a foreground Gaussian set $\mathcal{G}_{fg} \subseteq \mathcal{G}$, where each Gaussian $G_m \in \mathcal{G}_{fg}$ is uniquely pre-assigned to a specific annotation through an assignment mapping $\mathrm{asgn}(m)$. These Gaussians are centred at the position of the assigned annotation throughout the parameter optimization of GS. To maintain strict one-to-one correspondence, we ensure $|\mathcal{G}| \geq N$ and $|\mathcal{G}_{fg}| = N$, with the remaining Gaussians randomly initialized to model unannotated image regions. This configuration creates *binary correspondences* where each foreground Gaussian exclusively corresponds to its pre-assigned annotation with probability 1 while having zero correspondence with all other annotations.

Pixel to Background Correspondence Given the sparsity of foreground targets and the significant proportion of background pixels in an image, OT's approach of assigning all pixels to targets is suboptimal for background pixels. To address this, we augment the annotation set $\mathcal{Y}$ with a virtual background object and develop an auxiliary mechanism for background correspondence. Specifically, pixels are assigned to the background if their distance to the nearest foreground Gaussian exceeds a predefined threshold. Furthermore, for consistency, we extend the previously defined $\mathcal{G}, \mathcal{Y}$ by introducing a background pseudo-Gaussian G_0 and a virtual background object y_0, i.e., $\mathcal{G} \leftarrow \mathcal{G} \cup \{G_0\}$, $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{y_0\}$, with $\boldsymbol{\zeta}_g(y_0) = 0$ to prevent background mass allocation. The correspondence of pixel x_i given background region G_0 is defined as:

$$P(x_i|G_0) \triangleq \frac{1}{2\pi|\boldsymbol{\Sigma}_*|^{1/2}} \exp\left(-\frac{d^2 - d_M^2(x_i, G_*)}{2}\right), \quad \text{where } G_* = \arg\min_{G \in \mathcal{G}_{fg}} d_M(x_i, G). \tag{11}$$

G_* represents the nearest foreground Gaussian to x_i, determined by the Mahalanobis distance $d_M(\boldsymbol{x}_i, G)$. $\boldsymbol{\Sigma}_*$ denotes its covariance. d serves as a cut-off threshold to control the spatial extent of background region assignment.

The correspondence between G_m and target y_n for $m, n \geq 0$ is determined by the pre-assigned mapping:

$$P(G_m|y_n) = \mathbb{I}(\mathrm{asgn}(m), y_n) \tag{12}$$

where $\mathbb{I}$ is an indicator function returning 1 when inputs are equal. Integrating background terms from Eqs. 11 and 12 into Eq. 7 easily yields the complete pixel-to-background correspondence.

Model Training and Optimization

With the transport kernel $\mathcal{K}$ generated as described, the model training proceeds as the final phase of our pipeline, as in Fig.2. The optimization is driven by the loss previously defined in Eq. 6, , which leverages the pre-computed $\mathcal{K}$. In contrast to iterative OT-based methods, our loss is calculated via a single, efficient matrix multiplication. This design makes the training process highly efficient and directly optimizes the model to produce density maps that align with the rich spatial correspondences encoded in the kernel.

Experiment Results and Discussion

We evaluate the proposed Gaussian Spatial Transport (GST) on crowd counting and landmark localization, representative tasks for point-supervised density regression. For crowd counting, we use three benchmarks: UCF-QNRF (Idrees et al. 2018), JHU-Crowd++ (Sindagi, Yasarla, and Patel 2020) (Wang et al. 2020b), and NWPU, reporting MAE and MSE. For landmark localization, we use the MPII Human Pose benchmark (Simonyan and Zisserman 2014), reporting PCKh@0.5. 2DGS is implemented using gsplat (Ye et al. 2024), with the cut-off distance d set to 3. Further implementation details are in *Suppl. C*.

Evaluation Results

Crowd Counting. Table 1 focuses on the datasets UCF-QNRF (Idrees et al. 2018) and JHU-Crowd++ (Sindagi, Yasarla, and Patel 2020). We notably excel the L_2 Baseline (direct regression) and handcrafted fixed transport plan (BL). On JHU-Crowd++, we achieve the lowest MAE 53.9 and MSE 225.4, marginally outperforming APGCC. Compared to methods based on optimal transport (DMCount, UOT, GL), our approach achieved significant performance improvements On UCF-QNRF, reducing MSE by 17.2 against balanced OT (DMCount), and 11.2 and 16.4 against unbalanced OT (UOT and GL, respectively). Fig. 3 visualizes the estimated density map of BL, OT (DMCount), and GST. Besides accurate counting estimation, GST also obtains *sharp* density distributions as the OT-based method.

Table 2 lists the results on NWPU (Wang et al. 2020b), which is currently the largest, most diverse, and most challenging crowd counting dataset. We consistently achieves strong performance, highlighting the efficiency of our transport strategy, particularly advantageous for tackling high-density scenarios and complex environmental conditions.

Dataset	JHU++		UCF-QNRF	
Method	MAE	MSE	MAE	MSE
L_2 Baseline	81.7	304.5	107.2	164.6
BL (Ma et al. 2019)	75.0	299.9	88.7	154.8
DMCount (Wang et al. 2020a)	61.6	256.1	85.6	148.3
UOT (Ma et al. 2021)	60.5	252.7	83.3	142.3
GL (Wan, Liu, and Chan 2021)	59.5	259.5	84.3	147.5
P2PNet (Song et al. 2021)	–	–	85.3	154.5
(Liang, Xu, and Bai 2022)	59.5	240.6	85.8	141.3
PET (Liu et al. 2023)	58.5	238	**79.5**	144.3
APGCC (Chen et al. 2024)	54.3	225.9	80.1	136.6
Ours	**53.9**	**225.4**	80.7	**131.1**

Table 1: Counting Performance on JHU++ and UCF-QNRF.

Landmark Localization. We evaluate our transport strategy against two approaches: Gaussian-blurred pseudo-density map regression (HRNet) and optimal transport cost minimization (HDM-HPE). As in Table 3, our method achieves the best mean PCKh@0.5 across all joints, outperforming the baseline HRNets by 0.6% on HRNet-W32/W48 architectures, with over 1.0% gains on challenging joints

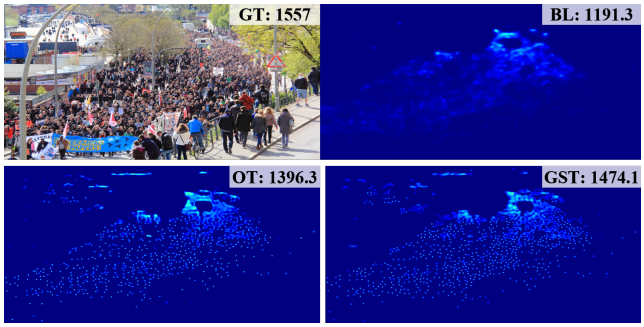


Figure 3: Visualization of crowd counting.

Method	MAE	MSE
L_2 Baseline	126.2	528.2
BL (Ma et al. 2019)	105.4	454.2
DMCount (Wang et al. 2020a)	88.4	388.6
UOT (Ma et al. 2021)	87.8	387.5
GL (Wan, Liu, and Chan 2021)	79.3	346.1
MAN (Lin et al. 2022)	76.5	323.0
ChfL (Shu et al. 2022)	76.8	343
PET (Liu et al. 2023)	74.4	328.5
Ours	**74.4**	**306.2**

Table 2: Performance of crowd counting on NWPU.

like shoulders, hips, knees, and ankles[4]. Comparable to OT-based HDM-HPE, our approach directly uses raw point annotations, avoiding computationally intensive subpixel smoothing (Qu et al. 2022). Fig. 4 shows that GST generates sharper density peaks, clarifying the location.

	HRNet-W32			HRNet-W48		
Metric	HRNet	HDM-HPE	Ours	HRNet	HDM-HPE	Ours
Mean	90.4	90.9	**91.0**	90.5	90.9	**91.1**
Head	97.1	97.3	96.2	96.9	97.1	96.4
Shldr.	95.9	96.2	97.1	96.0	96.3	97.0
Elb.	90.7	91.2	90.3	90.9	91.2	90.9
Wri.	86.1	86.8	86.9	86.2	87.0	86.5
Hip	89.4	90.1	90.7	89.6	90.2	91.1
Kne.	86.9	87.4	88.4	87.1	87.5	88.0
Ank.	83.2	84.1	85.1	83.5	84.2	84.7

Table 3: Performance of human pose estimation on MPII.

Contribution of Components and Parameters

Ablation study. Table 4 reports the ablation study results organized on a VGG (Simonyan and Zisserman 2014) backbone. The heuristic transport plan, which computes the transport kernel $\mathcal{K}$ via a Gaussian $P(x_i|y_n)$ with an empirical $\sigma = 8$, significantly outperforms the L_2 baseline, highlighting the advantage of Bayesian transport modeling. Furthermore, the 2DGS transport plan, derived from our proposed method leveraging Gaussian Splatting, demonstrates clear superiority over the heuristic plan. Finally, the inclusion of pixel-to-background correspondence further enhances performance, confirming its effectiveness in suppressing irrelevant background regions.

[4]The observed performance discrepancy in *head* stems from annotation bias: MPII's head joint annotations are at the periphery, not the center, degrading performance.

(a) ankle (b) head

Figure 4: Visualization of landmark location.

Meanwhile, we conducted experiments on VGG and Transformer architectures (as in (Lin et al. 2024)) to further compare the performance of the OT method and our GST. The results in Table 5 show that GST surpasses OT on both VGG and Transformer, highlighting its consistent robustness regardless of the backbone architecture.

transport plan	P2A Cor.	P2B Cor.	MAE	MSE
L_2 baseline			81.70	304.50
heuristic plan			65.45	268.50
GST-	✓		60.17	247.61
GST	✓	✓	**58.30**	**239.58**

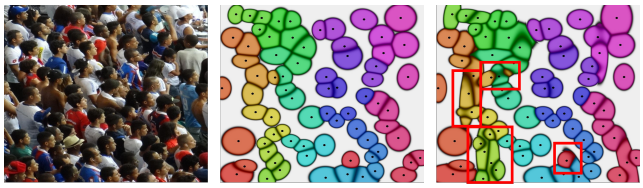
Table 4: Ablation Studies on JHU++ with VGG backbone. *P2A Cor.* and *P2B Cor.* represents the Pixel-to-Annotation and Pixel-to-background Correspondence, respectively.

Arch.	Method	UCF-QNRF		JHU++		NWPU	
		MAE	MSE	MAE	MSE	MAE	MSE
VGG	OT	87.2	150.8	61.6	256.1	88.4	388.6
	GST	**82.5**	**139.2**	**58.3**	**239.6**	**80.3**	**325.7**
Trans.	OT	87.8	153.0	58.5	240.0	80.9	323.5
	GST	**80.7**	**131.1**	**53.9**	**225.4**	**74.4**	**306.2**

Table 5: Counting Results on different model architectures. *Trans* is the abbreviation of Transformer.

Effect of Deformity Elimination. We investigate the necessity of deformity elimination in the splatting process in Fig. 5. Without it, over-elongated Gaussians erroneously cover irrelevant background regions, causing significant overlapping transport regions (Fig. 5(c), red boxes). This leads to penetrating-object transport and ambiguous correspondences, resulting in unreliable training supervision.

Effect of d in Pixel-to-Background Correspondence. We conduct experiments to evaluate the effect of cut-off distance d on VGG. As in Fig 6, the optimal performance is achieved at $d = 3$, which aligns with the 3σ principle in 2DGS rendering, ensuring effective foreground-background separation. Moreover, our method is robust to parameter selection near this optimum, consistently outperforming OT within the $2.4 \sim 3.6$ range.



(a) image (b) w/ DE (c) w/o DE

Figure 5: Visualization of transport plan w/ and w/o deformity elimination (DE) during Gaussian splatting. The map shows annotation-to-pixel correspondence: black points are annotations, unique colors represent individual targets with brightness indicating transport strength, and blank regions denote background transport.

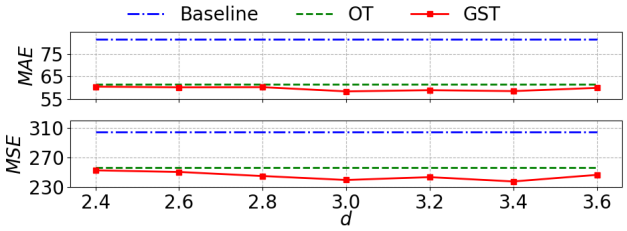


Figure 6: Results w.r.t. d on JHU++ with VGG backbone.

Discussion

The role of 2D Gaussian Splatting. Gaussian Splatting plays a central role in building a spatial transport plan from density maps to annotations as in Eqs. 5 and 7. Fig. 5 exemplifies a plan. It can be observed that GST establishes a discrete spatial mapping from image coordinate space to annotation space. To further quantify the effectiveness of GST, we compare it against a transport plan using *ideal* ground truth scale information, alongside heuristic-based scales and the OT plan, as shown in Table 6. Transport plans based on ground truth scales and 2DGS-estimated scales outperform both the OT plan and the heuristic-based plan.

These evaluations demonstrate that GST effectively estimates head scales using only point annotations. Moreover, its transport plan outperforms both the OT-based and the static heuristic-based approach. Finally, the results validate our assumption that a reliable transport plan can be derived from the input image and annotated points prior to task-specific network optimization.

Differences between OT and GST. GST is fundamentally different from traditional OT, representing *a distinct paradigm rather than an approximation*. The foundational difference lies in the source of information used for planning: while OT-based approaches must wait for and operate on the network's density estimation, our method directly constructs a correspondence from the input RGB image itself. This distinction in methodology leads to two key advantages. First, the RGB input space provides richer semantic information than just a one-dimensional density map. This allows GST to generate more interpretable transport plans that are grounded in the actual image content. Second, and more critically, GST's approach decouples the costly planning from the iterative optimization process. This is achieved through a one-time pre-computation of the transport kernel. To quantify this efficiency gain, consider an image with p annotated points and q pixels. The pre-computation is dominated by computing the probability density function between coordinates and kernels, costing $\mathcal{O}(pq)$. Crucially, this computation occurs before training, reducing the theoretical complexity *during training* to $\mathcal{O}(1)$. This complete separation of planning from training avoids the repeated calculations inherent to OT and creates a win-win solution, free from the trade-offs typical of two-stage optimization. In stark contrast, OT requires a much higher complexity of $\mathcal{O}(kpq)$ for iterative Sinkhorn optimization within each training step, typically k=100 for DM-Count (Wang et al. 2020a) and k=1,000 for HDM-HPE (Qu et al. 2022). This theoretical difference translates directly to practice, as GST reduces the runtime of OT almost by half on an NVIDIA 4090, as shown in the last column of Table 6.

As GST involves a pre-computation step, its cost is minimal. This one-time process typically takes only 3-7 seconds per high-resolution image, and remains under 10 seconds even for extreme cases with resolutions up to 4K and over 4,500 annotated points. This runtime could be further reduced with optimized Gaussian Splatting implementations. Consequently, the overall pipeline is significantly faster than end-to-end OT-based training. More discussion on differences between OT and GST can be referred to *Suppl. D*.

Method	MAE	MSE	Training Time
plan with *g.t.* scale	**56.60**	**238.24**	15h29min
Heuristic plan	65.45	268.50	14h35min
OT	61.55	256.10	28h36min
GST	<u>58.30</u>	<u>239.58</u>	15h32min

Table 6: Accuracy and runtime comparison of different transport plans on JHU++ with VGG backbone.

Conclusion and Limitations

This paper proposes Gaussian Spatial Transport, which leverages Gaussian splatting to compute the transport plan. We apply it to representative tasks, including crowd counting and landmark detection. Experimental results demonstrate its clear advantages over traditional optimal transport schemes in both training efficiency and accuracy. Moreover, GST successfully extends the application of Gaussian splatting in computer vision. Validating it in a broader range of applications presents a promising research direction.

However, our method still has certain limitations. For instance, the hard binding of the Gaussian kernels to the annotations during splatting optimization may be suboptimal. More flexible soft-assignment mechanisms can be further explored in future work. Additionally, the 2D Gaussian splatting based on a single image lacks explicit depth modeling, which is crucial for image understanding. Considering recent progress in 3D Gaussian splatting for sparse-view reconstruction, future extensions could incorporate 3D Gaussians with explicit depth information for the transport in the multi-view scene to enhance spatial reasoning capabilities.

Acknowledgements

This work was funded in part by the National Natural Science Foundation of China (62376070, 62076195) and in part by the Fundamental Research Funds for the Central Universities (AUGA5710011522). Thanks to all anonymous reviewers and the area chair for their valuable comments. Thanks to Mr. Yupeng Wei for his discussion on the optimal transport theory.

References

Cao, X.; Wang, Z.; Zhao, Y.; and Su, F. 2018. Scale aggregation network for accurate and efficient crowd counting. In *Proceedings of the European conference on computer vision (ECCV)*, 734–750.

Chen, D.; Chen, L.; Zhang, Z.; and Zhang, L. 2025. Generalized and Efficient 2D Gaussian Splatting for Arbitrary-scale Super-Resolution. *arXiv preprint arXiv:2501.06838*.

Chen, I.-H.; Chen, W.-T.; Liu, Y.-W.; Yang, M.-H.; and Kuo, S.-Y. 2024. Improving point-based crowd counting and localization based on auxiliary point guidance. In *European Conference on Computer Vision*, 428–444. Springer.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

Dong, J.; Wang, C.; Zheng, W.; Chen, L.; Lu, J.; and Tang, Y. 2025. GaussianToken: An Effective Image Tokenizer with 2D Gaussian Splatting. *arXiv preprint arXiv:2501.15619*.

Ge, Z.; Liu, S.; Li, Z.; Yoshie, O.; and Sun, J. 2021. Ota: Optimal transport assignment for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 303–312.

Hu, J.; Xia, B.; Chen, B.; Yang, W.; and Zhang, L. 2024. GaussianSR: High Fidelity 2D Gaussian Splatting for Arbitrary-Scale Image Super-Resolution. *arXiv preprint arXiv:2407.18046*.

Huang, T.; Zhang, H.; Zeng, Y.; Zhang, Z.; Li, H.; Zuo, W.; and Lau, R. W. 2024. Dreamphysics: Learning physical properties of dynamic 3d gaussians with video diffusion priors. *arXiv preprint arXiv:2406.01476*.

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

Kantorovich, L. V. 1960. Mathematical methods of organizing and planning production. *Management science*, 6(4): 366–422.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4).

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

Liang, D.; Xu, W.; and Bai, X. 2022. An end-to-end transformer model for crowd localization. In *ECCV*.

Lin, H.; Hong, X.; Ma, Z.; Wei, X.; Qiu, Y.; Wang, Y.; and Gong, Y. 2021. Direct measure matching for crowd counting. *arXiv preprint arXiv:2107.01558*.

Lin, H.; Ma, Z.; Hong, X.; Shangguan, Q.; and Meng, D. 2024. Gramformer: Learning crowd counting via graph-modulated transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 3395–3403.

Lin, H.; Ma, Z.; Ji, R.; Wang, Y.; and Hong, X. 2022. Boosting crowd counting via multifaceted attention. In *CVPR*.

Lin, H.; Ma, Z.; Ji, R.; Wang, Y.; Su, Z.; Hong, X.; and Meng, D. 2025. Semi-supervised counting via pixel-by-pixel density distribution modelling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Lin, W.; and Chan, A. B. 2023. 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*, 21663–21673.

Liu, C.; Lu, H.; Cao, Z.; and Liu, T. 2023. Point-query quadtree for crowd counting, localization, and more. In *ICCV*, 1676–1685.

Liu, W.; Salzmann, M.; and Fua, P. 2019. Context-aware crowd counting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5099–5108.

Ma, Z.; Wei, X.; Hong, X.; and Gong, Y. 2019. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6142–6151.

Ma, Z.; Wei, X.; Hong, X.; Lin, H.; Qiu, Y.; and Gong, Y. 2021. Learning to count via unbalanced optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2319–2327.

Peyré, G.; Cuturi, M.; et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6): 355–607.

Qu, H.; Xu, L.; Cai, Y.; Foo, L. G.; and Liu, J. 2022. Heatmap distribution matching for human pose estimation. *Advances in Neural Information Processing Systems*, 35: 24327–24339.

Shu, W.; Wan, J.; Tan, K. C.; Kwong, S.; and Chan, A. B. 2022. Crowd counting in the frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19618–19627.

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

Sindagi, V. A.; and Patel, V. M. 2017. Generating high-quality crowd density maps using contextual pyramid cnns. In *Proceedings of the IEEE international conference on computer vision*, 1861–1870.

Sindagi, V. A.; Yasarla, R.; and Patel, V. M. 2020. Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. *IEEE transactions on pattern analysis and machine intelligence*, 44(5): 2594–2609.

Song, Q.; Wang, C.; Jiang, Z.; Wang, Y.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; and Wu, Y. 2021. Rethinking Counting and Localization in Crowds: A Purely Point-Based Framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Sun, K.; Xiao, B.; Liu, D.; and Wang, J. 2019. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5693–5703.

Villani, C. 2021. *Topics in optimal transportation*, volume 58. American Mathematical Soc.

Wan, J.; Liu, Z.; and Chan, A. B. 2021. A generalized loss function for crowd counting and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1974–1983.

Wan, J.; Wang, Q.; and Chan, A. B. 2020. Kernel-based density map generation for dense object counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3): 1357–1370.

Wang, B.; Liu, H.; Samaras, D.; and Hoai, M. 2020a. Distribution matching for crowd counting. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 1595–1607.

Wang, L.; Shi, Y.; and Ooi, W. T. 2025. GSVC: Efficient Video Representation and Compression Through 2D Gaussian Splatting. *arXiv preprint arXiv:2501.12060*.

Wang, Q.; Gao, J.; Lin, W.; and Li, X. 2020b. NWPU-crowd: A large-scale benchmark for crowd counting and localization. *IEEE transactions on pattern analysis and machine intelligence*, 43(6): 2141–2149.

Ye, V.; Li, R.; Kerr, J.; Turkulainen, M.; Yi, B.; Pan, Z.; Seiskari, O.; Ye, J.; Hu, J.; Tancik, M.; and Kanazawa, A. 2024. gsplat: An Open-Source Library for Gaussian Splatting. *arXiv preprint arXiv:2409.06765*.

Zhang, Q.; Zhang, K.; Chan, A. B.; and Huang, H. 2024a. Mahalanobis Distance-Based Multi-view Optimal Transport for Multi-view Crowd Localization. In *European Conference on Computer Vision*, 19–36. Springer.

Zhang, X.; Ge, X.; Xu, T.; He, D.; Wang, Y.; Qin, H.; Lu, G.; Geng, J.; and Zhang, J. 2024b. GaussianImage: 1000 FPS Image Representation and Compression by 2D Gaussian Splatting. In *European Conference on Computer Vision*.

Zhang, Y.; Zhou, D.; Chen, S.; Gao, S.; and Ma, Y. 2016. Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 589–597.

Zhu, L.; Lin, G.; Chen, J.; Zhang, X.; Jin, Z.; Wang, Z.; and Yu, L. 2025. Large Images are Gaussians: High-Quality Large Image Representation with Levels of 2D Gaussian Splatting. *arXiv preprint arXiv:2502.09039*.