Gaussian Density Splatting Network

Miao Shang **Yabin Wang** **Xiaopeng Hong***
Faculty of Computing
Harbin Institute of Technology
miaos0522@gmail.com wang-yabin@outlook.com hongxiaopeng@ieee.org

Abstract

This paper proposes a novel crowd counting approach, the Gaussian Density Splatting Network (GDSNet). Unlike methods that rely on conventional, grid-based density maps and are sensitive to spatial resolution, GDSNet represents a crowd as a superposition of continuous 2D Gaussian primitives. Our approach is built upon two key contributions. First, a control-point-based fitting mechanism is introduced to structure the prediction of Gaussian parameters. A set of control points is adaptively allocated to define local regions, from which features are pooled to regress each primitive's parameters. Second, we adapt a differentiable Gaussian Splatting framework to the counting task by parameterizing each primitive with geometric parameters and a scalar density mass. This formulation allows the network to be trained end-to-end via spatial matching of differentiably rendered density maps, naturally providing both local density supervision and global count optimization. Extensive evaluations on four standard benchmarks show GDSNet consistently outperforms the state of the art. The code is available at https://github.com/infinite0522/GDSNet-Gaussian-Density-Splatting-Network.

1 Introduction

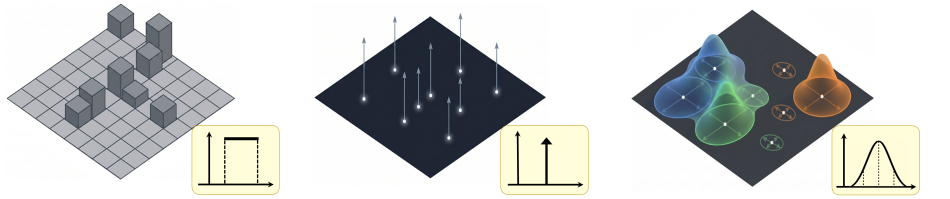
Crowd counting, the task of estimating the number and spatial distribution of people in an image, serves as a cornerstone of modern intelligent systems, with applications ranging from public safety monitoring to urban planning [1–3]. The core challenge of crowd counting lies in learning an expressive representation of the crowd density function, $f : \mathbb{R}^2 \to \mathbb{R}^+$, whose integral over the image domain yields the total count. However, existing paradigms for approximating this function suffer from fundamental representational limitations, as illustrated in Fig. 1.

The dominant paradigm, grid-based density map regression, discretizes the domain and models the density function on a rigid grid [4, 5]. This approach is fundamentally constrained by a *fixed* downsampling ratio of the network backbone, forcing the representation into a limited-resolution, piecewise-constant structure. This architectural rigidity not only introduces quantization artifacts but also struggles to model the continuous scale variations inherent in perspective-distorted scenes. A second paradigm, point-based detection, conceptualizes individuals as discrete points [6, 7]. While continuous in their coordinates, these methods reduce people to Dirac delta-like impulses, ignoring the essential geometric properties of individuals and groups (such as spatial extent and orientation). This oversimplification, combined with a reliance on unstable bipartite matching for supervision, may lead to training instability and limited performance in dense regions [8, 9].

To overcome these limitations, we propose to represent the crowd density field as a superposition of continuous, learnable 2D Gaussian primitives. Each primitive is explicitly parameterized by its *location*, anisotropic *covariance* that captures scale and orientation, and scalar *density mass*

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Grid-based Representation (b) Point-based Representation (c) Continuous Gaussian Representation

Figure 1: Comparison of grid-based, point-based, and our continuous Gaussian representation.

representing the local population count. This formulation elegantly unifies the localization-modeling problem (the geometric aspect) and the counting problem (the quantitative aspect) of crowd analysis.

However, this powerful continuous representation introduces a significant optimization challenge, *i.e.*, the learning problem becomes severely ill-posed. Unlike grid-based regression, where spatial support is predefined by pixels, learning a set of free-form Gaussians is prone to failure: First, without guidance, the network may waste modeling capacity in sparse areas while failing to represent high-density regions with sufficient fidelity. Second, the unconstrained optimization of covariance parameters can lead to degenerate solutions, where Gaussians collapse to points, stretch infinitely, or overlap chaotically, resulting in *physically implausible representations*. Third, the Gaussian optimization procedure usually learns each Gaussian independently and ignores the spatial correlations between neighboring people, leading to a disjointed and locally incoherent density field.

To address this ill-posed problem, we introduce a novel control-point guided approach that transforms unconstrained parameter regression into a structured geometric fitting problem. Our solution is built upon a hierarchy of components that directly counter the three challenges identified above. First, to satisfy the density-fidelity consistency, a structural complexity-guided control-point allocation strategy is introduced, which adaptively places points in structurally salient locations and further partitions the image into locally coherent regions. This directly prevents budget waste in homogeneous regions and perfectly aligns the primitive distribution with the non-uniform crowd density. Second, to prevent physically implausible representations, these control points are organized via Delaunay triangulation to form a mesh of non-overlapping simplices. Each simplex provides a stable geometric prior for the Gaussian primitive and defines an explicit spatial boundary for its center location. Finally, to ensure relational consistency, the final parameters of each Gaussian are predicted via a topology-aware feature aggregation scheme that considers the geometry and features of neighboring simplices. It substantially reduces optimization ambiguity and results in a stable Gaussian regression.

To parameterize each Gaussian primitive based on a simplex formed by the control points, we design a Gaussian Density Splatting Network (GDSNet) to predict the final Gaussian parameters, including the location, covariance, and the density mass, by aggregating features from the local simplex geometry and the underlying image. The entire network is optimized end-to-end by differentiably rendering a density map from the learned primitives and matching it to a ground-truth map. Moreover, we introduce a count consistency loss and a shape regularization loss to enforce mass conservation and geometric stability. Extensive experiments show that GDSNet sets a new state of the art on four challenging benchmarks and demonstrates superior robustness to variations in internal feature resolution at test time, validating the advantages of our continuous representation.

Our main contributions are summarized below:

- We present a unified basis function-based crowd representation. Under this formulation, crowd density is modeled as a superposition of continuous, anisotropic 2D Gaussian primitives, overcoming the intrinsic limitations of discrete density maps and point-based representations.
- We propose a structural complexity-guided control-point allocation method to construct a triangulated topology, which provides structured spatial supports for organizing and parameterizing Gaussian primitives.
- We develop an end-to-end differentiable Gaussian density splatting framework, where primitive parameters are predicted based on the control-point topology. This formulation jointly enables spatially dense supervision and accurate global count optimization, leading to consistent performance improvements across benchmarks.

2 Related Work

2.1 Crowd Density Representations

Existing crowd counting methods primarily rely on discrete representations, which can be further categorized into grid-based density regression and point-based coordinate localization.

Traditional **map-based methods** project continuous crowds onto rigid two-dimensional grids, inevitably creating severe scale mismatches and blurry visual overlaps. To alleviate these defects, researchers introduced multiscale architectures [1, 2, 4], hierarchical density experts [10], and scale-aware transformers [11, 12]. Moreover, intermediate targets evolved to prevent instance fusion through probability estimation [13], inverse distance maps [14], and regional integral supervision via Bayesian or spatial transport losses [15–19]. Among these methods, GST [18]and [20] fits appearance Gaussians offline to precompute a transport kernel for training supervision, while retaining a conventional network to regress the density map. Its Gaussians therefore serve as correspondence priors for supervision. Conversely, **point-based methods** treat crowd distributions as discrete Dirac impulses. Following the bipartite matching paradigm established by P2PNet [6], subsequent frameworks improved querying efficiency and stability via transformer processing [7], quadtree splitting [8], and localized constraints [9, 21]. Predicting isolated proposals intrinsically collapses crowds into zero-dimensional coordinates, entirely ignoring continuous spatial extents. This point-to-point matching paradigm also becomes exceptionally fragile in ultra-dense scenes, resulting in severe training instability and unreliable gradient assignments.

Bridging these representational extremes, GDSNet elevates crowd counting to a continuous functional approximation using latent Gaussian bases. Unlike GST [18], GDSNet directly predicts mass-carrying Gaussians to represent and render the density field. By explicitly modeling perspective geometry and mathematically decoupling global counts from spatial rendering grids, our framework establishes a matching-free and resolution-agnostic paradigm.

2.2 Image Representation and Perception via Continuous Gaussians

Gaussian splatting has significantly advanced visual representation by modeling scenes as continuous geometric mixtures in computer vision tasks such as compression [22], super-resolution [23], and inpainting [24]. Foundational optimization-based methods [22, 25, 26] achieve high-fidelity reconstruction and compression by leveraging multi-scale fitting and entropy-driven initialization. GaussianSR [23] and GSASR [27] further enable arbitrary-scale super-resolution via continuous field modeling. Nonetheless, the iterative optimization remains computationally expensive.

To accelerate inference, feedforward 2DGS approaches have been developed to directly predict Gaussian parameters from visual features [23, 27, 28], which leverage neural networks (*e.g.*, VAE, Transformer, UNet) to directly map image features to Gaussian fields. There are increasing studies on incorporating priors to enhance the quality of representations [29–31, 24]. Among them, Instant GaussianImage [30] represents observable RGB images through Gaussian-based interpolation.

These advances primarily target appearance reconstruction, leaving the formulation of task-oriented Gaussian fields for dense prediction less explored. GDSNet treats Gaussians as adaptive basis elements of an implicit continuous density field, with task-specific responses and spatial support. Through topology-aware primitive interaction and task-specific constraints, it adapts Gaussian representations to crowd density estimation.

3 Method

We begin by revisiting density estimation from a representation perspective and reformulate it as continuous Gaussian field reconstruction in Sec. 3.1. We then introduce GDSNet to learn this representation, as shown in Fig. 2 with its key components detailed in the following subsections.

3.1 A Unified Mathematical Perspective on Crowd Representation

Given an input image $I \in \mathbb{R}^{H\times W\times 3}$ and annotations $Y = \{y_m\}_{m=1}^{M}$, crowd counting aims to estimate a non-negative spatial density function $\hat{\mathcal{D}}(x)$ over the continuous domain $\Omega = \mathbb{R}^2$. From a functional approximation perspective, we unify existing paradigms as estimating a density field $\hat{\mathcal{D}}(x)$

Table 1: Comparison of crowd counting paradigms from a basis-function perspective.

Paradigm	**Magnitude** (α_i)	**Basis Function** (Ψ_i)	**Spatial Support** (ξ_i)	**State**	**Representation Formula** $\hat{\mathcal{D}}(x)$
Grid-based	Scalar density ($\hat{d}_u$)	Piecewise box (Π_{Ω_u})	Fixed grid cell (s)	Discrete	$\sum_{u\in\mathcal{G}_s} \hat{d}_u \Pi_{\Omega_u}(x)$
Point-based	Binary indicator ($\mathbb{I}$)	Dirac delta (δ)	Zero-volume point (μ_i)	Discrete	$\sum_i \mathbb{I}(p_i > \eta)\delta(x-\mu_i)$
GDSNet	**Density mass** (ρ_i)	**2D Gaussian** ($\mathcal{N}$)	**Explicit geometry** (μ_i, Σ_i)	**Continuous**	$\sum_i \rho_i \mathcal{N}(x;\mu_i,\Sigma_i)$

via a linear combination of *basis functions*:

$$\hat{\mathcal{D}}(x) = \sum\nolimits_i \alpha_i \Psi_i(x;\xi_i), \tag{1}$$

where Ψ_i denotes the basis function whose spatial support is parameterized by ξ_i, $\alpha_i \geq 0$ is its magnitude. The global integral yields the total count $\hat{M} = \int_\Omega \hat{\mathcal{D}}(x)\,dx$.

Limitations of existing paradigms. As summarized in Table 1, existing methods are constrained by their suboptimal basis choices [2]. Grid-based methods instantiate Ψ_i as a *zero-order box basis with a rigid grid-aligned support*. The grid support is dictated by the network's downsampling stride, limiting flexibility for continuous density variations or sub-pixel congestions. Point-based methods instantiate Ψ_i as a *Dirac delta basis with a zero-volume support*. The zero-volume support fails to explicitly model the spatial extent, local anisotropy, or continuous occupancy of crowds.

Our solution. To achieve a representation that is both geometrically explicit and spatially continuous, we instantiate the basis Ψ_i as an **anisotropic 2D Gaussian** and form a continuous representation:

$$\hat{\mathcal{D}}_{\text{gds}}(x) = \sum\nolimits_i \rho_i \mathcal{N}(x;\mu_i,\Sigma_i). \tag{2}$$

where $\mathcal{N}(x;\mu_i,\Sigma_i)$ is a normalized Gaussian distribution over $\mathbb{R}^2$. The magnitude α_i corresponds to a learnable density mass ρ_i. The count estimation is the summation of the density mass:

$$\hat{M} = \int_\Omega \hat{\mathcal{D}}_{\text{gds}}(x)\,dx = \int_\Omega \sum\nolimits_i \rho_i \mathcal{N}(x;\mu_i,\Sigma_i)\,dx = \sum\nolimits_i \rho_i \int_\Omega \mathcal{N}(x;\mu_i,\Sigma_i)\,dx = \sum\nolimits_i \rho_i. \tag{3}$$

The advantages are manifold. First, the infinitely differentiable smooth basis provides a smooth and flexible function family capable of approximating continuous crowd density variations to approximate the non-linear and smoothly decaying density distributions of real-world crowds under perspective variation and local occlusion. Second, count estimation is decoupled from the rasterization grid through mass summation, which reduces quantization errors. Gaussian geometry nevertheless defines the spatial support of the density field and is jointly learned with mass through spatial supervision. Third, the explicit primitive parameters make it possible to inject geometric and topological constraints into the learning process.

Despite its theoretical elegance, directly optimizing these Gaussian parameters is ill-posed due to non-identifiability and instability arising from unconstrained primitive locations and covariances. Without proper structural regulation, primitives easily collapse into singularities or migrate erratically. To impose structure on this otherwise unconstrained representation, inspired by the construction of free-form curves approximation like Bézier and B-spline curves [32], we propose to govern the continuous representation through a parametric modeling paradigm driven by structural control points. We reformulate the density estimation as a control-point-constrained parameterization problem, regulated by localized points that govern primitive centers and geometric priors.

This transforms free-form Gaussian estimation into a stable process of explicitly constrained parameterization. As in Fig. 2, we first perform structural complexity-guided control point allocation to construct a Delaunay triangulation $\mathcal{T}$ from the input image (Sec. 3.2), while the counting backbone extracts a dense feature map. Driven by this spatial structure, we then perform topology-aware aggregation to fuse intra- and inter-features. The features are utilized to regress the Gaussian parameters and density mass, governed by a constrained parameterization mechanism (Sec. 3.3). Finally, the primitives undergo differentiable density rendering for end-to-end supervision (Sec. 3.4).

3.2 Structural Complexity-Guided Control-Point Allocation

The effectiveness of our parameterization fundamentally relies on the spatial distribution of control points. A naive uniform sampling allocates excessive modeling capacity to homogeneous regions

[2] Detailed mathematical derivations and explanations of the formulas and notations in Table 1 are provided in *Suppl. A*.

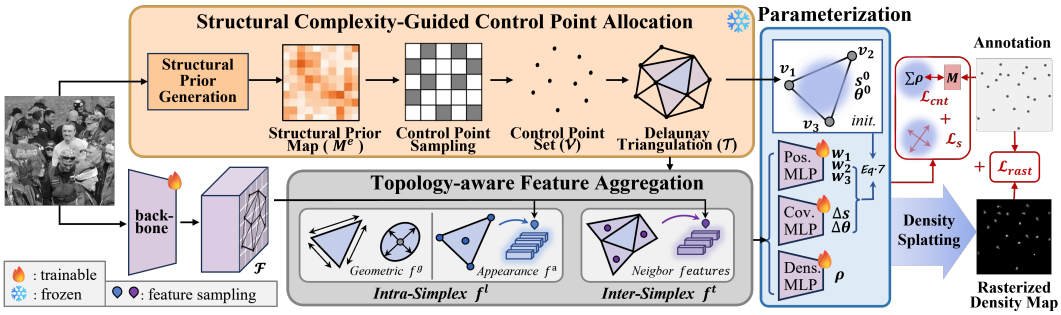


Figure 2: Pipeline of GDSNet.

while under-representing complex dense areas. To establish an adaptive spatial topology, we introduce a Structural Complexity-Guided Control-Point Allocation module.

Structural Guidance Sampling. Our design is motivated by the observation that local structural complexity reflects the underlying spatial layout of the scene. Regions exhibiting sharp appearance transitions, occlusion boundaries, or abrupt density variations typically present elevated structural complexity, indicating critical transitions in the underlying density field.

To capture critical scene transitions, we define a Structural Prior Map $\mathbf{M}^e \in [0,1]^{H\times W}$ that identifies regions of high structural complexity, such as sharp appearance transitions or occlusion boundaries. These regions require denser point allocation to ensure that the resulting interiors remain structurally coherent and ideally suited for compact Gaussian approximation. Thus, we generate M^e by leveraging the off-the-shelf spatial structural estimator from the 2DGS framework [30]. This network was trained to predict the optimal locations for placing Gaussians to minimize rendering error, which naturally correlates with high-frequency details, edges, and structural complexity. We provide a detailed discussion on alternative M^e in Sec. 4.2.

To convert the structural prior $\mathbf{M}^e$ into a discrete point set while maintaining explicit control over the representational granularity, we introduce a stride-controlled sampling mechanism, aiming to regulate both the total number of Gaussian primitives and their average spatial bounds across the continuous domain. We first apply non-overlapping max pooling with kernel size and stride s to $\mathbf{M}^e$, producing a coarse structural response map $\bar{\mathbf{M}}^e$. A Floyd-Steinberg dithering operator [33] is then applied to yield binary activations $z_{i,j} = \mathrm{Dither}(\bar{\mathbf{M}}^e_{i,j}) \in \{0,1\}$. The resulting control-point set is defined as:

$$\mathcal{V} = \left\{ (is + \tfrac{s}{2}, js + \tfrac{s}{2}) \mid z_{i,j} = 1 \right\}. \tag{4}$$

This strategy guarantees that the control points are adaptively distributed based on local structure, while their total quantity $|\mathcal{V}|$ is strictly bounded by $\lceil H/s \rceil \times \lceil W/s \rceil$.

The stride s explicitly dictates the spatial resolution of the continuous Gaussian field. Because this continuous paradigm fundamentally decouples density modeling from the fixed pixel grid, modulating s simply adjusts the representation's granularity. This intrinsic flexibility endows the framework with a strong robustness against varying representation granularities, thereby allowing GDSNet to perform stable, arbitrary test-time scaling without retraining, as validated experimentally in Sec. 4.3.

Delaunay simplex construction. Given the sampled control-point set $\mathcal{V}$, we perform planar Delaunay triangulation [34] to obtain a set of simplices $\mathcal{T} = \mathrm{Del}(\mathcal{V}) = \{\tau_i\}_{i=1}^{N}$, where each simplex $\tau_i = \mathrm{conv}(\{v_{i,1}, v_{i,2}, v_{i,3}\})$ is a convex hull defined by a triplet of vertices from $\mathcal{V}$.

This process yields triangular support domains that align with the structural boundaries identified in the sampling stage. The choice of 2D simplices over arbitrary polygons is motivated by three critical advantages for Gaussian parameterization. First, simplices *enable stable and unique geometric regression through barycentric coordinates [35]*, ensuring smooth gradient flow by restricting primitive centers within a convex feasible region. Second, the Delaunay *empty circumcircle* property [34] maximizes the minimum internal angle of the mesh, preventing spatially elongated primitives that would violate local density assumptions. Finally, by Euler's formula, $|\mathcal{V}|$ control points yield at most $2|\mathcal{V}|-5$ faces [36], *bounding the final number of generated Gaussians to* $\mathcal{O}(|\mathcal{V}|)$ while guaranteeing a completely hole-free, continuous representation of the crowd. By ensuring that each simplex encloses a region of relative informational stability, this geometric partitioning provides a locally coherent substrate for compact Gaussian approximation.

3.3 Topology-Aware Feature Aggregation and Constrained Parameterization

Given the control points $\mathcal{V}$ and the simplex set $\mathcal{T}$, the subsequent task is to determine the specific geometric configuration and density mass for each Gaussian primitive. We design a topology-aware feature aggregation module that operates over a topological graph to extract hierarchical features, followed by a formal geometric parameterization that enforces spatial constraints. To capture latent structural dependencies among primitives, we formalize the relational structure of the support domains as a topological graph $\mathcal{G} = (\mathcal{T}, \mathcal{E})$. The connectivity in $\mathcal{G}$ is governed by the shared edges $\mathcal{E}$ of the Delaunay triangulation, with the neighborhood for each simplex τ_i $\mathcal{B}(i) = \{\tau_j \mid \tau_i \cap \tau_j \in \mathcal{E}\}$ represents the set of adjacent simplices sharing a common edge.

Topology-aware Feature Aggregation. We extract and aggregate features for each primitive over the simplicial graph $\mathcal{G}$ to encode both local and global contexts. Specifically, the feature f_i for the i-th Gaussian primitive consists of an intra- and inter-simplex representations, f_i^l and f_i^t, respectively.

Intra-simplex feature representation. The intra-simplex representation f_i^l fuses a geometric representation f_i^g with an appearance feature f_i^a. Specifically, f_i^g is explicitly constructed using the normalized coordinates of the triangle vertices alongside the parameters of the base ellipse e_i fitted to the simplex [37]. The appearance feature f_i^a is obtained by querying the dense feature map $\mathcal{F} \in \mathbb{R}^{C \times H' \times W'}$ extracted by the counting backbone. To capture the local appearance context across this support region, we sample $\mathcal{F}$ at the triangle barycenter c_i and its three vertices $v_{i,j}$:

$$f_i^a = \text{MLP}_{app}\left(\mathcal{F}(c_i) \oplus \mathcal{F}(v_{i,1}) \oplus \mathcal{F}(v_{i,2}) \oplus \mathcal{F}(v_{i,3})\right), \tag{5}$$

where MLP_{app} is designed to aggregate these queries.

Inter-simplex topological aggregation. Due to the spatial correlation inherent in crowd distributions, adjacent local regions commonly exhibit similar perspectives, scales, and density patterns. Treating each triangular receptive field independently would neglect this relational structure, leading to incoherent parameter predictions. Therefore, we perform topological context aggregation over the graph $\mathcal{G}$ to ensure consistency across neighboring primitives. For each triangle τ_i, we collect its one-hop adjacent neighbors $\mathcal{B}(i)$ and aggregate their graph-induced contextual cues to form f_i^t, which is then concatenated with the intra-simplex representation:

$$f_i = f_i^l \oplus f_i^t, \quad \text{where} \quad f_i^t = \bigoplus\nolimits_{b \in \mathcal{B}(i)} \text{MLP}_{nbr}(c_b \oplus \mathcal{F}(c_b)). \tag{6}$$

c_b denotes the barycenter of a neighboring triangle b, and MLP_{nbr} compresses the neighbor's feature channels. Eq. 6 allows the subsequent parameterization to incorporate local perspective and density variations, thereby reducing geometric and spatial inconsistencies across neighboring regions.

Control-Point-Constrained Parameterization. To facilitate stable Gaussian learning, we obtain the final state of each primitive $\mathcal{G}_i = \{\mu_i, \mathbf{s}_i, \theta_i, \rho_i\}$ through a constrained parameterization governed by the underlying support simplex. Rather than directly regressing these values in an unconstrained domain, we formulate the final parameters as regulated adjustments relative to a base prior.

Specifically, the reference ellipse e_i fitted over the simplex τ_i establishes a plausible base prior. Taking the aggregated feature representation f_i as input, parallel MLP heads predict a set of relative adjustments alongside the non-negative density mass ρ_i. These geometric adjustments include the barycentric weights $\mathbf{w}_i = [w_{i,1}, w_{i,2}, w_{i,3}]$, the scale offsets $\Delta\mathbf{s}_i$, and the rotation offset $\Delta\theta_i$.

Using the network predictions together with the initial rotation θ_i^0 and base aspect ratio $r_i^0 = s_{i,x}^0/(s_{i,y}^0 + \epsilon)$ derived from e_i, the final geometric parameters of the Gaussian primitive are:

$$\mu_i = \sum\nolimits_{j=1}^{3} w_{i,j} v_{i,j}, \quad \mathbf{s}_i = \Delta\mathbf{s}_i \cdot \left[r_i^0, 1\right]^\mathsf{T}, \quad \theta_i = \theta_i^0 + \Delta\theta_i. \tag{7}$$

Crucially, the optimization of the Gaussian center μ_i is explicitly anchored by the learned barycentric weights [35]. This restricts the center to a localized feasible region bounded by the convex hull of its associated control points $\mathcal{R}_i = \{x \mid x = \sum_{j=1}^{3} w_{i,j} v_{i,j}, \sum_{j=1}^{3} w_{i,j} = 1, w_{i,j} \geq 0\}$. This formulation establishes a rigid spatial boundary that prevents erratic primitive migration.

3.4 Continuous Density Rendering and Optimization

Building upon these parameterized Gaussian primitives, GDSNet adapts the differentiable Gaussian splatting framework from traditional RGB color synthesis to crowd density estimation. Specifically,

the continuous density field is formulated by aggregating all Gaussian primitives, each weighted by its corresponding scalar density mass ρ_i, as defined in Eq. 2.

For spatial density alignment, we rasterize the density map and apply Bayesian loss [15]:

$$\mathcal{L}_{\text{rast}} = \sum\nolimits_{m=1}^{M} \left| 1 - \int_{\Omega} \mathbf{P}_m(x) \hat{\mathcal{D}}(x)\, dx \right| + \left| 0 - \int_{\Omega} \mathbf{P}_{bk}(x) \hat{\mathcal{D}}(x)\, dx \right|, \tag{8}$$

$\mathbf{P}_m$ and $\mathbf{P}_{bk}$ are the foreground and background posterior probability maps as in [15].

To further enforce global counting consistency, we introduce a count conservation loss

$$\mathcal{L}_{\text{cnt}} = \left| M - \sum\nolimits_{i=1}^{N} \rho_i \right|. \tag{9}$$

To discourage excessively elongated Gaussian shapes, we apply a shape regularization term:

$$\mathcal{L}_{\text{s}} = \max_i \left[s_i^{\max} / (s_i^{\min} + \epsilon) - \gamma \right]_+, \tag{10}$$

where $s_i^{\max}$ and $s_i^{\min}$ denote the major and minor axis scales of the i-th Gaussian, respectively, and γ controls the maximum allowable aspect ratio. The overall loss function is:

$$\mathcal{L} = \mathcal{L}_{\text{rast}} + \lambda_1 \mathcal{L}_{\text{cnt}} + \lambda_2 \mathcal{L}_{\text{s}}. \tag{11}$$

Eq. 11 promotes density reconstruction, count preservation, and Gaussian geometry fitting jointly.

4 Experiment Results and Discussion

We evaluate GDSNet on four standard crowd benchmarks: ShanghaiTech A and B [38], JHU-Crowd++ [39], UCF-QNRF [40], and NWPU-Crowd [41], using MAE and MSE as the performance metrics. Following previous competitive methods [9, 19], we utilize the VGG-19 structure as our feature extractor. Default sampling stride s is 10. $\lambda_1 = \lambda_2 = 1$. AdamW optimizer is used, with a learning rate of 1×10^{-5}, a weight decay of 1×10^{-4}, and a batch size of 1. All experiments are conducted on NVIDIA RTX 4090 GPUs. Details and the implementation algorithm are in *Suppl. B*.

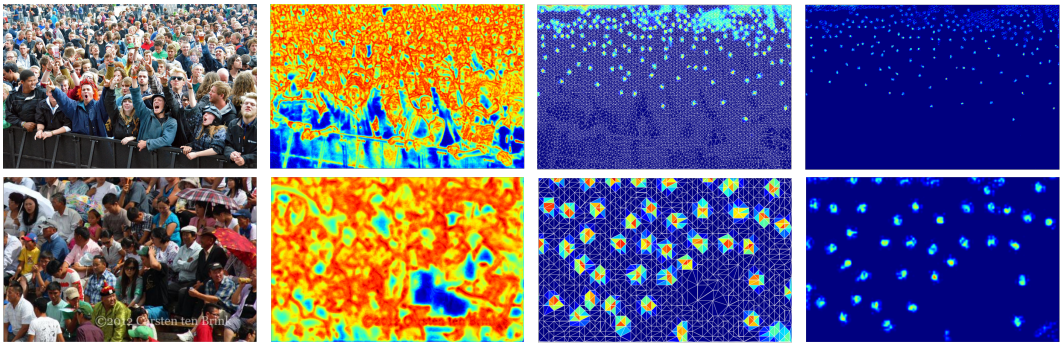
4.1 Evaluation Results

We compare GDSNet with representative map-based and point-based methods in Table 2. GDSNet achieves a state-of-the-art performance across *all* benchmarks. Specifically, GDSNet achieves a significant MAE reduction over the latest point-based method APGCC [9] on JHU++, UCF-QNRF, and NWPU. This is attributed to our continuous Gaussian representation. On ShanghaiTech Part B, GDSNet achieves an MAE of 5.4 and an MSE of 8.0, demonstrating its effectiveness in relatively sparse crowd scenes. By rasterizing Gaussian primitives into a structured density map, we leverage advanced dense supervision from map-based methods, circumventing the bipartite matching instabilities prevalent in ultra-dense regions. Moreover, GDSNet clearly outperforms recent map-based methods like PML [19] and GST [18], demonstrating the advantages of using continuous basis formulation over rigid grid-aligned scalar fields.

Visualization. Fig. 3 visualizes the intermediate representations of GDSNet across three crowd scenes. These examples exhibit diverse resolutions and significant inter-image and intra-image scale variations, ranging from dense congregations to sparser scenes with varying perspective distortions. The structural prior maps in Fig. 3 (b) identify structural boundaries and high transitions to guide the adaptive allocation of control points. In Fig. 3 (c), the resulting Delaunay triangulation demonstrates strong adaptability across diverse scenes. Specifically, the total number and spatial distribution of the triangular simplices naturally align with the underlying crowd density. Crucially, the triangle mass heatmap reveals a smooth density transition across adjacent simplices, which confirms that our neighbor context aggregation effectively maintains topological continuity and captures structural dependencies between neighboring primitives. Finally, the rasterized density maps in Fig. 3 (d) showcase the continuous density field formed by the superposition of Gaussian primitives, which accurately capture the local geometry and density of the scene.

Dataset Method	Venue & Year	ShTech A		ShTech B		JHU-Crowd++		UCF-QNRF		NWPU	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
BL [15]	ICCV 2019	62.8	101.8	7.7	12.7	75.0	299.9	88.7	154.8	105.4	454.2
DMCount [16]	NeurIPS 2020	59.7	95.7	7.4	11.8	61.6	256.1	85.6	148.3	88.4	388.6
UOT [17]	AAAI 2021	58.1	95.9	6.5	10.2	60.5	252.7	83.3	142.3	87.8	387.5
GL [42]	CVPR 2021	61.3	95.4	7.3	11.7	59.5	259.5	84.3	147.5	79.3	346.1
P2PNet [6]	ICCV 2021	52.7	85.1	6.3	9.9	–	–	85.3	154.5	72.6	331.6
MAN [11]	CVPR 2022	56.8	90.3	–	–	53.7	**209.9**	77.3	131.5	76.5	323.0
CLTR [7]	ECCV 2022	56.9	95.2	6.5	10.6	59.5	240.6	85.8	141.3	74.3	333.8
ChfL [43]	CVPR 2022	57.5	94.3	6.9	11.0	57.0	235.7	80.3	137.6	76.8	343.0
PET [8]	ICCV 2023	49.3	78.8	6.2	9.7	58.5	238.0	79.5	144.3	74.4	328.5
APGCC [9]	ECCV 2024	48.8	76.7	5.6	8.7	54.3	225.9	80.1	136.6	71.7	**284.4**
Gramformer [12]	AAAI 2024	54.7	87.1	–	–	53.1	228.1	76.7	**129.5**	72.5	316.4
PML [19]	ICLR 2025	55.5	89.0	6.0	9.3	57.4	227.4	76.6	132.2	73.6	338.6
P2R [21]	CVPR 2025	51.0	79.7	6.2	9.8	58.8	253.1	83.3	138.1	–	–
GST [18]	AAAI 2026	–	–	–	–	53.9	225.4	80.7	131.1	74.4	306.2
GDSNet	–	**48.8**	**76.5**	**5.4**	**8.0**	**52.8**	227.6	**75.9**	130.5	**68.7**	294.2

Table 2: Counting performance on ShTech A and B, JHU-Crowd++, UCF-QNRF, and NWPU.



(a) Input (b) Structural Prior Map (c) Simplex Heatmap (d) Rasterized Density Map

Figure 3: Visualization of the structural prior, simplex heatmap, and rasterized density map. The simplex heatmap is obtained by assigning each Delaunay simplex the mass predicted by its associated Gaussian primitive, revealing how responses are spatially distributed over the triangulated support.

4.2 Ablation Study

Contribution of Components. We investigate the contribution of the structural complexity-guided Control Point Allocation (CPA), Simplex-Constrained Parameterization (SCP), and Topology-aware Feature Aggregation (TFA) in GDSNet on the UCF-QNRF dataset in Table 3. Removing the topological context aggregation (*Row 2*) leads to a performance drop, confirming that explicitly modeling local neighborhood interactions is important for preserving relational consistency among adjacent Gaussian primitives and producing spatially coherent density estimation. Moreover, the substantial degradation in (*Rows 3&4*) highlights the critical role of our simplex-constrained geometric formulation. Without the explicit geometric boundaries imposed by local simplices, Gaussian parameter learning becomes severely under-constrained, allowing the regression head to fit training samples through arbitrary scale and shape configurations that lack structural regularity. Although these unconstrained variants can achieve satisfactory fitting on the training set, they exhibit significantly weaker generalization on the test set, indicating that simplex support serves as an effective geometric prior that regularizes parameter optimization toward stable and transferable Gaussian representations. Furthermore, *Row 3* outperforms *Row 4*, further suggesting that structural complexity-guided sampling provides more informative primitive allocation than uniform grid sampling. Details are provided in *Suppl. D.1*.

Gaussian Geometry Ablation. We investigate the role of Gaussian geometry on UCF-QNRF in Table 6. Triangle mass regression uniformly distributes the predicted mass over each triangle. Fixed Gaussians use the center, scale, and orientation of the reference ellipse without learning geometric adjustments. Isotropic Gaussians restrict spatial support to a single scale shared across both axes, whereas the full model learns anisotropic scales and orientation with simplex-constrained centers. All variants share the same primitive allocation, feature aggregation, mass prediction head, and rasterization supervision. As shown in Table 6, compared with triangle mass regression, Gaussian

Table 3: Ablation of key components.

CPA	SCP	TFA	MAE	MSE
✓	✓	✓	**75.86**	**130.52**
✓	✓		77.89	132.59
✓			82.94	135.62
			89.46	140.85

Table 4: Ablation of losses.

$\mathcal{L}_{\text{rast}}$	$\mathcal{L}_{\text{cnt}}$	$\mathcal{L}_{\text{geo}}$	MAE	MSE
–	✓	✓	94.87	154.24
✓			77.54	133.17
✓	✓		76.87	131.58
✓		✓	76.20	130.74
✓	✓	✓	**75.86**	**130.52**

Table 5: CPA Strategies.

Strategy	MAE	MSE
Uniform	81.65	133.97
Sobel	78.44	131.58
Structural	**75.86**	**130.52**

bases provide smooth density responses rather than piecewise-constant distributions. Learnable geometry allows spatial support to adapt to local crowd structure instead of remaining fixed to the initial triangulation. Anisotropic geometry further captures directional variations and elongated density patterns that isotropic support cannot represent. Implementation details are in *Suppl. D.3*.

Table 6: Gaussian geometry ablation on UCF-QNRF.

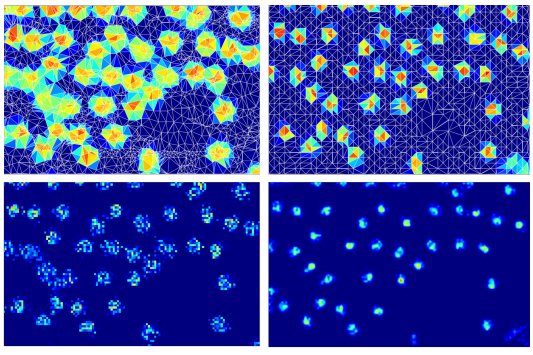
Configuration	MAE	MSE
Triangle mass regression	82.21	133.85
Fixed Gaussian	81.42	133.24
Isotropic Gaussian	78.79	132.86
Full anisotropic Gaussian	**75.86**	**130.52**

Loss Function Ablation. The optimization of GDSNet is guided by a joint objective that addresses the unique characteristics of our Gaussian representation. Table 4 shows that every loss term contributes by providing constraints from different perspectives. The rasterization loss serves as the primary driver by providing supervision on the rendered density field to ensure spatial fidelity. Furthermore, the count loss further refines the estimation by supervising the total density masses of Gaussian primitives. More importantly, geometric regularization plays a vital role in stabilizing primitive shapes by penalizing extreme anisotropies. The synergy of these objectives allows GDSNet to achieve the best accuracy by balancing spatial fidelity, counting precision, and geometric stability.

Impact of Control Point Allocation Strategies. We evaluate three control point allocation strategies: *baseline uniform sampling*, *sampling according to the Sobel filtering responses [26]*, and our *structural complexity-guided sampling*. Details and more prior map visualization are in *Suppl. D.2*.

As in Table 5, *uniform sampling* yields the highest errors as it allocates control points indiscriminately across the image domain. The support simplex obtained from these control points becomes homogeneous because they have the same geometric characteristics. Compared with ours, *sampling by Sobel filtering responses* is less effective because it is overly sensitive to sharp boundaries and fine textures. This over-sensitivity results in an excessive allocation of primitives in semantically irrelevant regions, which hurts the counting task.

Our *structural complexity guided sampling* performs best. Taking the 3rd sample from Fig. 3 as an example, Fig. 4 further visualizes the simplex heatmap and rasterized density maps. Under the same sampling stride, *sampling by Sobel filtering responses* activates a larger number of broad-response simplices around each head instance. To satisfy the unit-mass constraint, optimization compensates by aggressively shrinking individual Gaussian scales, resulting in fragmented and pulse-like density responses that undermine local spatial continuity. In contrast, our *structural complexity guided sampling* produces more compact and spatially focused simplex activations, enabling Gaussian primitives to maintain appropriate support extent and generate smoother, more coherent density reconstruction. Enhanced alignment between simplex geometry and crowd structure ultimately leads to more stable Gaussian parameterization and improved counting accuracy.



(a) Sobel Filter Response Prior (b) Structural Prior

Figure 4: Simplex heatmap (top) and rasterized density map (bottom) of different CPA priors.

4.3 Discussion on Sampling Stride Sensitivity

Robustness. To evaluate robustness under varying sampling resolutions, we conduct a train-test stride cross-ablation on UCF-QNRF by training GDSNet with strides 8, 10, and 12, and evaluating each across test strides from 6 to 16 without finetuning. As shown in Fig. 5, all configurations maintain MAE below 80 for test strides less than 15, demonstrating strong robustness across a broad sampling range. This verifies that the continuous Gaussian representation effectively reduces dependence on a fixed discrete sampling resolution. Performance drops consistently at test strides 15 and 16 for all settings, revealing an intrinsic undersampling limit rather than train-test mismatch. At such sparse resolutions, Gaussian primitives become excessively coarse, leading to merged local structures and over-smoothed density reconstruction. Training with strides 8 and 10 achieves the strongest overall performance, with stride 10 performing best in most cases. Notably, it even surpasses its matched-stride result when evaluated at neighboring strides (*e.g.*, 6, 7, and 11), indicating that GDSNet can compensate for moderate sampling mismatch through its continuous geometric parameterization. Although stride 12 yields slightly lower accuracy, it still stably achieves an MAE of 77.23 under matched-stride evaluation, highlighting the strong tolerance of the proposed representation to coarser discretization.

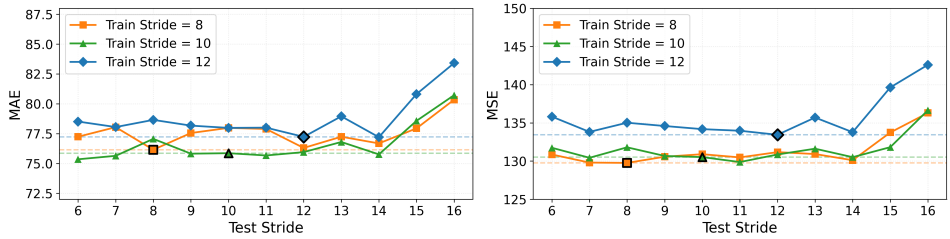


Figure 5: Performance *w.r.t.* train and test sampling strides.

Efficiency. The sampling stride s controls the allocation budget. For an $H \times W$ image, the number of sampled control points $V = |\mathcal{V}|$ satisfies $V \leq \lceil H/s \rceil \lceil W/s \rceil$. A planar triangulation with h convex-hull vertices contains $N = 2V - h - 2 \leq 2V - 5$ triangular faces, each associated with one Gaussian primitive. Thus, increasing s reduces the upper bound on Gaussian allocation, while the actual allocation remains image-adaptive. Delaunay construction takes $\mathcal{O}(V \log V)$ time, and feature aggregation over the bounded-degree simplex dual graph takes $\mathcal{O}(N)$, yielding **quasi-linear** $\mathcal{O}(N \log N)$ complexity for topology construction and primitive processing. Detailed runtime, memory, and scaling measurements, including the dominant cost of topology construction, are provided in *Suppl. C*.

5 Conclusion

In this paper, we propose GDSNet, which formulates crowd counting as a continuous density reconstruction problem using Gaussian basis functions, moving beyond traditional grid- and point-based representations. To address the ill-posed nature of unconstrained continuous modeling, we introduce a control-point-guided parameterization that imposes structured geometric constraints on Gaussian primitives. This design enables stable learning while preserving spatial continuity and flexibility. Extensive experiments on challenging benchmarks demonstrate consistent improvements, with strong robustness to variations in test-time sampling density. How to further accelerate the Delaunay triangulation will be a future direction.

6 Acknowledgement

This work was funded in part by the National Key R&D Program of China (2025YFC3811300), the National Natural Science Foundation of China (62376070, 62606119, 62076195), the China Postdoctoral Science Foundation under Grant (2026M794773), and the Fundamental Research Funds for the Central Universities (AUGA5710011522).

References

[1] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016.

[2] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1091–1100, 2018.

[3] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[4] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5099–5108, 2019.

[5] Tao Han, Lei Bai, Lingbo Liu, and Wanli Ouyang. Steerer: Resolving scale variations for counting and localization via selective inheritance learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

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

[7] Dingkang Liang, Wei Xu, and Xiang Bai. An end-to-end transformer model for crowd localization. In *ECCV*, 2022.

[8] Chengxin Liu, Hao Lu, Zhiguo Cao, and Tongliang Liu. Point-query quadtree for crowd counting, localization, and more. In *ICCV*, pages 1676–1685, 2023.

[9] I-Hsiang Chen, Wei-Ting Chen, Yu-Wei Liu, Ming-Hsuan Yang, and Sy-Yen Kuo. Improving point-based crowd counting and localization based on auxiliary point guidance. In *European Conference on Computer Vision*, pages 428–444. Springer, 2024.

[10] Zhipeng Du, Miaojing Shi, Jiankang Deng, and Stefanos Zafeiriou. Redesigning multi-scale neural network for crowd counting. *IEEE Transactions on Image Processing*, 32:3664–3678, 2023.

[11] Hui Lin, Zhiheng Ma, Rongrong Ji, Yaowei Wang, and Xiaopeng Hong. Boosting crowd counting via multifaceted attention. In *CVPR*, 2022.

[12] Hui Lin, Zhiheng Ma, Xiaopeng Hong, Qinnan Shangguan, and Deyu Meng. Gramformer: Learning crowd counting via graph-modulated transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 3395–3403, 2024.

[13] Jian Cheng, Haipeng Xiong, Zhiguo Cao, and Hao Lu. Decoupled two-stage crowd counting and beyond. *IEEE Transactions on Image Processing*, 30:2862–2875, 2021.

[14] Dingkang Liang, Wei Xu, Yingying Zhu, and Yu Zhou. Focal inverse distance transform maps for crowd localization. *IEEE Transactions on Multimedia*, 25:6040–6052, 2023.

[15] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6142–6151, 2019.

[16] Boyu Wang, Huidong Liu, Dimitris Samaras, and Minh Hoai. Distribution matching for crowd counting. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 1595–1607, 2020.

[17] Zhiheng Ma, Xing Wei, Xiaopeng Hong, Hui Lin, Yunfeng Qiu, and Yihong Gong. Learning to count via unbalanced optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2319–2327, 2021.

[18] Miao Shang and Xiaopeng Hong. 2d gaussians spatial transport for point-supervised density regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 8824–8832, 2026.

[19] Wei Lin, Jia Wan, and Antoni B. Chan. Proximal mapping loss: Understanding loss functions in crowd counting & localization. In *The Thirteenth International Conference on Learning Representations*, 2025.

[20] Miao Shang, Xiaopeng Hong, Xiujuan Chai, and Wangmeng Zuo. Untangling context: A gaussian splatting-mediated bayesian framework for crowd counting. *Pattern Recognition*, page 114725, 2026.

[21] Wei Lin, Chenyang Zhao, and Antoni B. Chan. Point-to-region loss for semi-supervised point-based crowd counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29363–29373, June 2025.

[22] Xinjie Zhang, Xingtong Ge, Tongda Xu, Dailan He, Yan Wang, Hongwei Qin, Guo Lu, Jing Geng, and Jun Zhang. Gaussianimage: 1000 fps image representation and compression by 2d gaussian splatting. In *European Conference on Computer Vision*, 2024.

[23] Jintong Hu, Bin Xia, Bin Chen, Wenming Yang, and Lei Zhang. Gaussiansr: High fidelity 2d gaussian splatting for arbitrary-scale image super-resolution. *arXiv preprint arXiv:2407.18046*, 2024.

[24] Hongyu Li, Chaofeng Chen, Xiaoming Li, and Guangming Lu. 2d gaussian splatting with semantic alignment for image inpainting. *arXiv preprint arXiv:2509.01964*, 2025.

[25] Lingting Zhu, Guying Lin, Jinnan Chen, Xinjie Zhang, Zhenchao Jin, Zhao Wang, and Lequan Yu. Large images are gaussians: High-quality large image representation with levels of 2d gaussian splatting. *arXiv preprint arXiv:2502.09039*, 2025.

[26] Yunxiang Zhang, Bingxuan Li, Alexandr Kuznetsov, Akshay Jindal, Stavros Diolatzis, Kenneth Chen, Anton Sochenov, Anton Kaplanyan, and Qi Sun. Image-gs: Content-adaptive image representation via 2d gaussians. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–11, 2025.

[27] Du Chen, Liyi Chen, Zhengqiang Zhang, and Lei Zhang. Generalized and efficient 2d gaussian splatting for arbitrary-scale super-resolution. *arXiv preprint arXiv:2501.06838*, 2025.

[28] Jiajun Dong, Chengkun Wang, Wenzhao Zheng, Lei Chen, Jiwen Lu, and Yansong Tang. Gaussiantoken: An effective image tokenizer with 2d gaussian splatting. *arXiv preprint arXiv:2501.15619*, 2025.

[29] Long Peng, Anran Wu, Wenbo Li, Peizhe Xia, Xueyuan Dai, Xinjie Zhang, Xin Di, Haoze Sun, Renjing Pei, Yang Wang, et al. Pixel to gaussian: Ultra-fast continuous super-resolution with 2d gaussian modeling. *arXiv preprint arXiv:2503.06617*, 2025.

[30] Zhaojie Zeng, Yuesong Wang, Tao Guan, Chao Yang, and Lili Ju. Instant gaussianimage: A generalizable and self-adaptive image representation via 2d gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27896–27905, 2025.

[31] Zhengqiang Zhang, Rongyuan Wu, Lingchen Sun, and Lei Zhang. Gpstoken: Gaussian parameterized spatially-adaptive tokenization for image representation and generation. *arXiv preprint arXiv:2509.01109*, 2025.

[32] Wolfgang Böhm, Gerald Farin, and Jürgen Kahmann. A survey of curve and surface methods in CAGD. *Computer Aided Geometric Design*, 1(1):1–60, 1984.

[33] Giorgia Franchini, Roberto Cavicchioli, and Jia Cheng Hu. Stochastic floyd-steinberg dithering on gpu: image quality and processing time improved. In *2019 Fifth International Conference on Image Information Processing (ICIIP)*, pages 1–6. IEEE, 2019.

[34] N Boris. Delaunay. sur la sphere vide. *Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7:793–800, 1934.

[35] Brian Cabral, Marc Olano, and Philip Nemec. Reflection space image based rendering. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 165–170, 1999.

[36] Raimund Seidel. The upper bound theorem for polytopes: an easy proof of its asymptotic version. *Computational Geometry*, 5(2):115–116, 1995.

[37] Andrew W Fitzgibbon, Robert B Fisher, et al. *A buyer's guide to conic fitting*. University of Edinburgh, Department of Artificial Intelligence Edinburgh, UK, 1996.

[38] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 589–597, 2016.

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

[40] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–546, 2018.

[41] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. Nwpu-crowd: A large-scale benchmark for crowd counting and localization. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):2141–2149, 2020.

[42] Jia Wan, Ziquan Liu, and Antoni B Chan. A generalized loss function for crowd counting and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1974–1983, 2021.

[43] Weibo Shu, Jia Wan, Kay Chen Tan, Sam Kwong, and Antoni B Chan. Crowd counting in the frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19618–19627, 2022.

[44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[45] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for Gaussian splatting. *arXiv preprint arXiv:2409.06765*, 2024.

[46] N. Amini-Naieni and A. Zisserman. Countgd++: Generalized prompting for open-world counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[47] Xiaowen Zhang, Zijie Yue, Yong Luo, Cairong Zhao, Qijun Chen, and Miaojing Shi. Bootstrapping mllm for weakly-supervised class-agnostic object counting. In *The Fourteenth International Conference on Learning Representations*, 2026.

Supplementary Material for Gaussian Density Splatting Network

A Discussion on Mathematical Presentation of Different Counting Paradigms

Given an input image $I \in \mathbb{R}^{H\times W\times 3}$ and point annotations $Y = \{y_m\}_{m=1}^{M}$, existing methods estimate the crowd distribution via grid-based density regression or point-based detection. From a functional approximation perspective, different counting paradigms can be unified as estimating $\mathcal{D}(x)$ over the continuous image domain $\Omega = \mathbb{R}^2$ via a linear combination of specific **basis functions** with varying **spatial supports**:

$$\hat{\mathcal{D}}(x) = \sum\nolimits_i \alpha_i \Psi_i(x; \xi_i), \tag{12}$$

where Ψ_i denotes the basis function whose spatial support is parameterized by ξ_i, $w_i \geq 0$ is the discrete magnitude, and the global integral yields the total count $\hat{M} = \int_\Omega \mathcal{D}(x)\, dx$. In the following, we formalize grid-based and point-based methods under this unified view, and subsequently introduce GDSNet as a continuous mathematical approximation representation.

grid-based density estimation. grid-based methods [1, 15] discretize the image plane into a regular grid $\mathcal{G}_s$, where s denotes the output stride, and predict a scalar density value for each grid cell. The corresponding continuous density can be viewed as a piecewise-constant approximation:

$$\hat{\mathcal{D}}_{\text{map}}(x) = \sum\nolimits_{u\in\mathcal{G}_s} \hat{d}_u \Pi_{\Omega_u}(x), \qquad \Pi_{\Omega_u}(x) = \begin{cases} 1, & \|x - x_u\|_\infty \leq s/2, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Here $\hat{d}_u$ is the scalar density predicted for the u-th cell centered at x_u, and Π_{Ω_u} is a **zero-order box basis with a rigid grid-aligned support**. Under our unified framework, the magnitude α_i corresponds to the predicted scalar density $\hat{d}_u$ for the u-th grid cell. The basis Ψ_i is instantiated as a zero-order box function $\Pi_{\Omega_u}(x)$. The spatial support parameter ξ_i is strictly locked to the rigid grid with fixed cell size s. While simple and effective, its spatial support is tightly coupled with the network architecture, and thus rigidly locked to the grid cell size prescribed by the downsampling stride. This rigidity severely limits representational flexibility, barring handling extreme sub-pixel congestions and arbitrary-resolution density sampling.

Point-based prediction. Point-based methods regress on a discrete set of candidate proposals, predicting a confidence score p_i and a coordinate μ_i for each proposal [6, 21]. After confidence thresholding, the crowd distribution degenerates into a sparse measure supported on isolated points:

$$\hat{\mathcal{D}}_{\text{point}}(x) = \sum\nolimits_i \mathbb{I}(p_i > \eta)\delta(x - \mu_i), \tag{14}$$

where $\mathbb{I}(\cdot)$ is the indicator function and $\delta(\cdot)$ is the **Dirac delta basis function**, which mathematically possesses **a zero-volume isolated support**. Under the unified framework, the magnitude α_i degenerates into a binary indicator $\mathbb{I}(p_i > \eta)$, and the basis Ψ_i collapses into a Dirac delta function δ. The support parameter ξ_i comprises solely the isolated coordinate μ_i. Although point coordinates are continuous, the resulting density has zero-volume support and fails to explicitly model the spatial extent, anisotropy, or smooth occupancy of crowded regions.

GDSNet: continuous Gaussian basis representation. To facilitate a representation that is both geometrically explicit and spatially continuous, we introduce the **exponential basis function** and formulate the density distribution as a superposition of 2D Gaussian primitives:

$$\hat{\mathcal{D}}_{\text{gds}}(x) = \sum\nolimits_i \rho_i \mathcal{N}(x; \mu_i, \Sigma_i), \tag{15}$$

where $\mathcal{N}$ represents the bivariate Gaussian basis function, parameterized by a learnable continuous center μ_i, an anisotropic covariance matrix Σ_i, and a non-negative density mass scalar ρ_i. Here,

the magnitude w_i is represented by a learnable non-negative density mass ρ_i and the basis ϕ_i is an exponential Gaussian function. The spatial support parameter $\theta_i = \{\mu_i, \Sigma_i\}$ explicitly models both the continuous center μ_i and the anisotropic covariance matrix Σ_i. Compared with box or Dirac bases, Gaussian bases provide smooth spatial support, explicit geometry, and analytical mass conservation:

$$\hat{M} = \int_{\Omega} \hat{\mathcal{D}}_{\text{gds}}(x)\, dx = \int_{\Omega} \sum\nolimits_i \rho_i \mathcal{N}(x; \mu_i, \Sigma_i)\, dx = \sum\nolimits_i \rho_i \int_{\Omega} \mathcal{N}(x; \mu_i, \Sigma_i)\, dx = \sum\nolimits_{i=1}^{N} \rho_i, \tag{16}$$

where the approximation becomes exact when the Gaussian mass outside the image boundary is negligible or handled by the rasterization domain.

B Implementation Details

All experiments are conducted on one NVIDIA RTX 4090 GPU, implemented in PyTorch 2.0.1 with CUDA 11.8. We use VGG19 [44] with ImageNet-pretrained weights as the backbone. Gaussian Splatting is implemented using gsplat [45]. The sampling stride for control point allocation is set to $s = 10$, the loss balancing temperatures are $\lambda_1 = \lambda_2 = 1$, and the shape regularization threshold is $\gamma = 2$. The structural prior is generated by an off-the-shelf, frozen structural estimator from [30], without additional training or task-specific fine-tuning, as detailed in *Suppl. D.2*. The reference ellipse fitted on each triangle simplex is obtained by least-squares fitting to the six points consisting of the three vertices and the three edge midpoints [37]. The Adam optimizer (batch size = 1 and learning rate = 10^{-5}) is used with image crop sizes adjusted per dataset: 256×256 for ShanghaiTech, 384×384 for NWPU, and 512×512 for the remaining benchmarks.

The pseudo code of model training and inference is provided in Algorithm 1. Rasterization is required during training for spatial density supervision. At inference, count-only evaluation returns the analytical mass sum, while density-map evaluation additionally renders the learned Gaussian field using its geometry and mass parameters.

C Complexity and Scalability Analysis

A critical concern for crowd counting in extremely dense scenarios is the scalability of the representation. We analyze the theoretical complexity of GDSNet with respect to the number of allocated Gaussians N, which corresponds to the number of triangular simplices. Unlike transformer-based architectures that typically suffer from quadratic $\mathcal{O}(N^2)$ or $\mathcal{O}(H^2W^2)$ complexity due to global self-attention, GDSNet maintains a highly efficient scalability profile.

The process begins with the Delaunay triangulation of V control points, which operates in quasi-linear $\mathcal{O}(V \log V)$ time. Since the number of triangular faces N in a planar triangulation is linearly proportional to the number of vertices ($N \approx 2V$), this stage effectively scales as $\mathcal{O}(N \log N)$. The subsequent *Topology-Aware Aggregation* is performed over the dual graph of the triangulation, where each Gaussian (simplex) acts as a node. In such a graph, every node is adjacent to at most three neighbors. This **bounded degree** ensures that the total number of edges in the topological graph is strictly linear, $\mathcal{O}(N)$. Consequently, the message-passing complexity for feature aggregation remains strictly linear, $\mathcal{O}(N)$.

Furthermore, the memory footprint of our continuous representation scales linearly with N ($\mathcal{O}(N)$), contrasting favorably with dense volumetric or high-resolution grid representations. During the rasterization phase, tile-based rendering is bounded by the image resolution $\mathcal{O}(H \times W)$ and is completely **decoupled** from the crowd density. This $\mathcal{O}(N \log N)$ overall bound ensures that GDSNet scales robustly even on challenging datasets like JHU-CROWD++ containing tens of thousands of individuals, without encountering out-of-memory (OOM) failures or exponential inference delays.

C.1 Component-wise Efficiency

Since GDSNet contains multiple components, we provide a detailed component-wise analysis to identify the computational cost distribution of each module. Table 7 reports the parameter, GPU memory, and runtime costs of the main components when processing a 256×256 input image. Values prefixed by "+" denote the additional cost of each component.

Algorithm 1: Training and Inference Pipeline of GDSNet

Input: Input image I, sampling stride s, annotations Y (train only), density-map output flag `return_density`.

Output: Updated parameters Θ (train), estimated count $\hat{M}$ (test), rendered map $\hat{\mathcal{D}}$ (optional for test).

`/* Phase 1: Structural Simplexity-Guided Control-Point Allocation (Sec. 3.2) */`

Extract dense semantic feature map $\mathcal{F} \leftarrow \text{Backbone}_{\Theta}(I)$;

Generate structural prior map $\mathbf{M}^e \leftarrow \text{Prior}(I)$;

Apply max-pooling (stride s) and dithering [33] to $\mathbf{M}^e$ to yield binary activations $z_{i,j}$;

Allocate discrete control points $\mathcal{V} \leftarrow \left\{(is + \frac{s}{2}, js + \frac{s}{2}) \mid z_{i,j} = 1\right\}$;

Construct simplicial mesh $\mathcal{T}$ and topological graph $\mathcal{G}(\mathcal{T}, \mathcal{E}) \leftarrow \text{DelaunayTriangulation}(\mathcal{V})$;

`/* Phase 2: Topology-Aware Feature Aggregation (Sec. 3.3) */`

foreach *simplex* $\tau_i \in \mathcal{T}$ **do**

  Fit base reference ellipse [37] e_i and extract its geometric priors $\{r_i^0, \theta_i^0\}$;

  Construct **geometric feature** f_i^g using normalized vertex coordinates and e_i;

  Sample $\mathcal{F}$ at center c_i and vertices $v_{i,j}$ to extract **appearance feature** f_i^a (Eq. 5);

  Fuse intra-simplex representation $f_i^l \leftarrow f_i^g \oplus f_i^a$;

  Aggregate graph-induced contextual features from neighbors $b \in \mathcal{B}(i)$ to form f_i^t (Eq. 6);

  Concatenate to form the comprehensive primitive feature: $f_i \leftarrow f_i^l \oplus f_i^t$;

end

`/* Phase 3: Constrained Parameterization (Sec. 3.3) */`

foreach *feature* f_i **do**

  Regress relative adjustments and density mass via parallel MLPs: $\mathbf{w}_i, \Delta\mathbf{s}_i, \Delta\theta_i, \rho_i \leftarrow \text{MLPs}_{\Theta}(f_i)$;

  Explicitly restrict center μ_i within the simplex convex hull via $\mathbf{w}_i$: $\mu_i \leftarrow \sum_{j=1}^{3} w_{i,j} v_{i,j}$ (Eq. 7);

  Map scale and rotation relative to base priors: $\mathbf{s}_i \leftarrow \Delta\mathbf{s}_i \cdot [r_i^0, 1]^\top$, $\theta_i \leftarrow \theta_i^0 + \Delta\theta_i$ (Eq. 7);

  Instantiate continuous Gaussian primitive: $\mathcal{G}_i \leftarrow \{\mu_i, \mathbf{s}_i, \theta_i, \rho_i\}$;

end

`/* Phase 4: State Execution & Optimization (Sec. 3.4) */`

Compute analytical total count by summing the density mass: $\hat{M} \leftarrow \sum_{i=1}^{|\mathcal{T}|} \rho_i$;

if *is Training Stage* **then**

  Rasterize primitives $\{\mathcal{G}_i\}$ to continuous density map $\hat{\mathcal{D}}$ 2 via Gaussian splatting;

  Compute spatial alignment Bayesian loss $\mathcal{L}_{\text{rast}}(\hat{\mathcal{D}}, Y)$ (Eq. 8);

  Compute global count loss $\mathcal{L}_{\text{cnt}}$ (Eq. 9) and geometric shape loss $\mathcal{L}_s$ (Eq. 10);

  Optimize overall objective $\mathcal{L} = \mathcal{L}_{\text{rast}} + \lambda_1 \mathcal{L}_{\text{cnt}} + \lambda_2 \mathcal{L}_s$ (Eq. 11) via backpropagation to update Θ;

else

  if *`return_density`* **then**

    Rasterize $\{\mathcal{G}_i\}$ to obtain $\hat{\mathcal{D}}$;

    return $\hat{M}, \hat{\mathcal{D}}$;

  else

    return $\hat{M}$;

  end

end

Table 7: Component-wise parameter, GPU memory, and runtime measurements.

Component	Params (M)	GPU memory (MB)	Runtime (ms)
Baseline: backbone + regression head	21.50	130.50	12.54
Backbone	20.02	48.64	11.02
+ Topology computation	+0	+0.65	+9.00
+ Feature aggregation	+1.66	+11.04	+0.29
+ Constrained Gaussian parameterization	+0.04	+1.70	+0.20
+ Gaussian field rasterization	+0	+0.21	+0.11

The analysis shows that the Gaussian field learning and rendering modules themselves introduce only marginal computational overhead. The feature aggregation, constrained Gaussian parameterization, and Gaussian field rasterization together require only 0.60 ms and 12.95 MB memory, demonstrating that the Gaussian parameter learning and differentiable rendering modules are lightweight.

The main additional overhead comes from the topology computation. This overhead is mainly caused by the Delaunay triangulation step that explicitly constructs spatial relationships among adaptive control points. Importantly, this operation is non-parametric and does not introduce additional trainable parameters.

In future work, more advanced GPU-accelerated triangulation tools or learned primitive organization strategies can be investigated to further reduce the overhead of topology construction.

C.2 Scaling with Primitive Allocation

We vary the sampling stride on the same input image under an identical inference setting and exclude image loading time. Table 8 separately reports topology construction to isolate the principal primitive-dependent cost. Smaller strides allocate more Gaussians and increase topology construction time. GPU memory varies little over the measured range.

Table 8: Runtime and GPU memory measurements with varying Gaussian primitive allocation.

Stride	Gaussian	Runtime (ms)	GPU memory (MB)	Topology time (ms)	Topology memory (MB)
6	1814	16.93	48.95	14.37	1.06
8	1002	13.65	48.93	11.15	0.56
10	639	12.27	48.92	9.72	0.55
12	451	10.94	48.92	8.42	0.55
16	244	10.07	48.91	7.59	0.51

The results show that the number of allocated Gaussian primitives can be flexibly adjusted through the stride s. The topology computation dominates the primitive-dependent scaling cost and increases consistently as more simplices are allocated. Importantly, as the number of simplices/Gaussians increases, the topology time and overall runtime scale approximately linearly, which empirically validates our claim of quasi-linear complexity. The other costs remain nearly unchanged. These results verify that the computational overhead of the proposed Gaussian representation is controllable with respect to the number of allocated primitives.

D Implementation Details of Ablation Studies

To investigate the modules in the proposed GDSNet, we implement module ablation and Control Point Allocation (CPA) strategy ablation studies, as shown in Table 9.

D.1 Implementation of Module Ablations

Removing Topology-aware Feature Aggregation (TFA). This variant removes the inter-simplex neighbor feature while keeping control point allocation, dither sampling, and simplex-constrained parameterization. The final feature for Gaussian becomes:

$$f_i = f_i^l, \tag{17}$$

Table 9: Implementation comparison of the full model and ablation variants. CPA, SCP, and TFA are the control point allocation, simplex-constrained parameterization, and topology-aware feature aggregation modules, respectively.

	Method	CPA Strategy	Sampling	Support Region	SCP	TFA
Module Ablations	Full GDSNet	Structural Prior	Dither	Triangle Simplex	✓	✓
	w/o TFA	Structural Prior	Dither	Triangle Simplex	✓	–
	w/o (TFA & SCP)	Structural Prior	Dither	Fixed grid patch	–	–
	w/o (TFA & SCP & CPA)	Structural Prior	Uniform grid	Fixed grid patch	–	–
CPA Ablation	Uniform Grid	Uniform	Dither	Triangle Simplex	✓	✓
	Sobel Filter Response Prior	Sobel Filter Response Prior	Dither	Triangle Simplex	✓	✓
	Structural Prior	Structural Prior	Dither	Triangle Simplex	✓	✓

instead of concatenating the context feature of one-hop Delaunay neighbors. This setting evaluates the contribution of local topological context to scale, rotation, and density mass prediction.

Removing Topology-aware Feature Aggregation (TFA) and Simplex-Constrained Parameterization (SCP). This variant also removes neighbor modeling of the Gaussian feature, and replaces the adaptive triangular receptive field with a fixed square grid patch. For each sampled anchor v_i, four patch corners are defined as:

$$q_{i,1} = v_i + (-\Delta_x, -\Delta_y), \quad q_{i,2} = v_i + (+\Delta_x, -\Delta_y), \tag{18}$$

$$q_{i,3} = v_i + (-\Delta_x, +\Delta_y), \quad q_{i,4} = v_i + (+\Delta_x, +\Delta_y), \tag{19}$$

where $\Delta_x = s/W$ and $\Delta_y = s/H$ in normalized coordinates. The local feature is obtained by sampling the backbone feature map at the four patch corners and the center

$$f_i^l = \mathcal{F}(q_{i,1}) \oplus \mathcal{F}(q_{i,2}) \oplus \mathcal{F}(q_{i,3}) \oplus \mathcal{F}(q_{i,4}) \oplus \mathcal{F}(v_i). \tag{20}$$

Without the constraint of the triangle simplex, this variant feeds the Gaussian feature into MLP heads and gets $\{\Delta\mu, \Delta\mathbf{s}_i, \Delta\theta_i, \rho_i\}$. The parameters of the Gaussian primitive are:

$$\mu_i = v_i + \Delta\mu_i, \quad \mathbf{s}_i = \Delta\mathbf{s}_i \cdot [1,1]^\mathsf{T}, \quad \theta_i = 0 + \Delta\theta_i, \quad \rho_i = \rho_i. \tag{21}$$

Removing Topology-aware Feature Aggregation (TFA), Simplex-Constrained Parameterization (SCP), and Structural Complexity Guided Control Point Allocation (CPA). This is the most reduced variant. It removes the structural prior and instead places Gaussian anchors uniformly on the image plane:

$$\mathcal{V}_{\text{grid}} = \{(k/2 + ik,\ k/2 + ik)\}_{i,j}. \tag{22}$$

Each anchor uses the same fixed grid-patch receptive field as described above. The feature extraction and Gaussian parameterization follows Eq. 20 and Eq. 21. This experiment evaluates whether uniform spatial allocation with local patch features is sufficient for Gaussian density rendering.

D.2 Implementation and Visualization of Control Point Allocation Strategy Ablations

Uniform Grid Allocation. In this setting, the structural prior map is replaced by a constant map:

$$\mathbf{M}^e(x) = 1, \quad \forall x \in \Omega. \tag{23}$$

The remaining pipeline is unchanged. This variant isolates the influence of learned complexity guidance from the rest of the topology-aware Gaussian feature aggregation pipeline.

Sobel Filter Response Prior. This variant constructs a prior map from handcrafted image structural prior as in Image-gs [26]. The input image is first converted to grayscale:

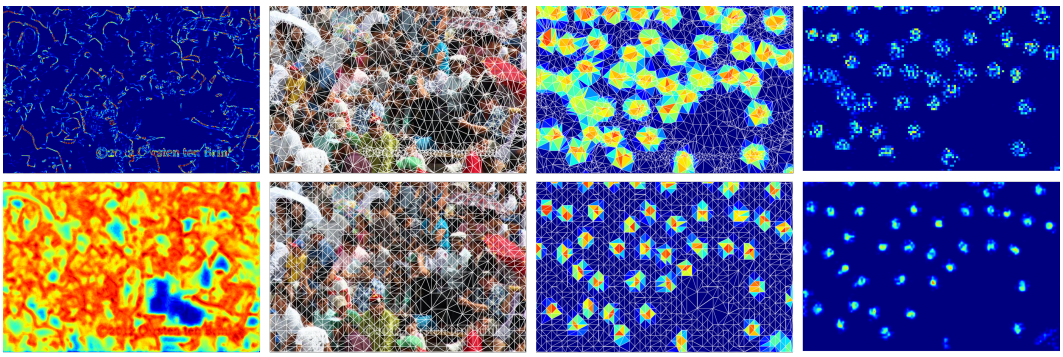
$$Y = 0.299R + 0.587G + 0.114B. \tag{24}$$

Then, Sobel filters are applied:

$$G_x = K_x * Y, \qquad G_y = K_y * Y, \tag{25}$$

and the response magnitude is computed as:

$$G = \sqrt{G_x^2 + G_y^2}. \tag{26}$$



(a) Prior Map (b) Triangulation Mesh (c) Simplex Heatmap (d) Rasterized Density Map

Figure 6: Comparison of Sobel filter response prior (top) and our structural prior (bottom).

After max normalization, the response is squared to emphasize strong edges:

$$\mathbf{M}^e_{\text{grad}} = \left(\frac{G}{\max(G) + \epsilon} \right)^2 . \tag{27}$$

To avoid overly sparse sampling in flat regions, a small lower bound is applied:

$$\mathbf{M}^e_{\text{grad}} = \max(\mathbf{M}^e_{\text{grad}}, \rho), \tag{28}$$

where ρ is a small constant. The subsequent pipeline is identical to the full model.

Structural Complexity Prior. This is the default setting of GDSNet. The prior map is generated by the off-the-shelf, frozen structural estimator from in instantGI [30]. Benefiting from its exceptional generalization across diverse scenes, we *directly employ this plug-and-play component to generate* $\mathbf{M}^e$ *for all experiments*. It is important to note that the InstantGI model is used *purely for inference in our pipeline to produce the structural prior*, and is *not fine-tuned or trained* on any target dataset. By utilizing this task-agnostic substrate, GDSNet can focus exclusively on task-specific density regression, effectively avoiding optimization instabilities while ensuring robust performance across varied crowd counting benchmarks.

As for the architecture of this estimator, $\mathbf{M}^e$ is predicted by a ConvNeXt-UNet followed by a pixel-wise MLP:

$$\mathbf{F}_e = f_{\text{SPM}}(I), \qquad \mathbf{M}^e = \sigma(\phi(\mathbf{F}_e)), \tag{29}$$

where $\mathbf{F}_e \in \mathbb{R}^{64\times H\times W}$ and ϕ is a three-layer MLP with channel dimensions $64 \to 64 \to 64 \to 1$. σ is the Sigmoid activation function.

More Visual Comparison of CPA Strategies. To further elucidate the impact of different priors, we visualize the intermediate stages of GDSNet using Sobel filtering versus our structural prior in Fig. 6. As shown in the top row, the Sobel-generated map is strictly contour-focused, capturing high-frequency edges and irrelevant background noise (e.g., the watermark at the bottom). This leads to an overly fragmented mesh that forces the Gaussian primitives to collapse into thin, erratic shapes, resulting in the noisy and disjointed density field. In contrast, our structural complexity-guided prior (bottom row) identifies geometric manifolds and salient clusters rather than just edges. This provides a more balanced triangulation substrate, allowing the network to regress smooth, ellipsoid-like primitives that accurately represent the spatial occupancy of individuals. This qualitative evidence confirms that our choice of structural prior is essential for maintaining the topological continuity of the crowd density field. Ultimately, this stable initialization effectively regularizes the learning process by anchoring the primitive topology, enabling the model to prioritize precise density mass estimation over topological searching. This synergy translates observed visual coherence into superior counting accuracy.

D.3 Implementation Details of Gaussian Geometry Ablation

All variants share the same control points, primitive allocation, feature aggregation module, density mass prediction head, and rasterization supervision. They differ in the spatial basis and its geometric parameterization.

Full anisotropic Gaussian parameterization. After Delaunay triangulation, a reference ellipse e_i is fitted to each simplex τ_i to establish its geometric prior. From the aggregated feature representation, the network predicts barycentric weights $w_{i,j}$, scale adjustments $\Delta\mathbf{s}_i$, a rotation adjustment $\Delta\theta_i$, and density mass ρ_i. Following Eq. 7, the final geometry is

$$\mu_i = \sum_{j=1}^{3} w_{i,j} v_{i,j}, \qquad \mathbf{s}_i = \Delta\mathbf{s}_i \odot [r_i^0, 1]^\mathsf{T}, \qquad \theta_i = \theta_i^0 + \Delta\theta_i, \tag{30}$$

where r_i^0 and θ_i^0 are the axis ratio and orientation of the fitted ellipse. The barycentric weights constrain the center to its simplex. The following variants modify this formulation while retaining the shared mass prediction and supervision.

Triangle mass regression. This variant uses a unit-integral, piecewise-constant triangle basis:

$$\Psi_i(x) = \frac{\mathbf{1}[x \in \tau_i]}{|\tau_i|}, \qquad \hat{\mathcal{D}}(x) = \sum_i \rho_i \Psi_i(x), \tag{31}$$

where $|\tau_i|$ is the triangle area. The predicted mass is uniformly distributed over each triangle.

Fixed Gaussian. This variant removes learnable geometric adjustments: μ_i, $\mathbf{s}_i$, and θ_i are directly set to the center, axis scales, and orientation of the fitted ellipse e_i and remain fixed during training. The density mass ρ_i is still predicted by the mass head.

Isotropic Gaussian. This variant removes anisotropic aspect-ratio modeling by setting $r_i^0 = 1$ and tying the two scale dimensions:

$$\mathbf{s}_i = \Delta s_i [1, 1]^\mathsf{T}. \tag{32}$$

The center retains the full model's learnable barycentric parameterization, and Δs_i is a learnable scalar scale adjustment. Thus, the variant preserves adaptive center and scale estimation while removing directional anisotropy. Rotation does not affect an isotropic Gaussian's spatial support.

These variants provide a progressive comparison from a piecewise-constant triangle basis to fixed Gaussian support, learnable isotropic support, and full anisotropic geometry. The quantitative results and analysis are provided in Table 6 in the main paper.

E Comparison with MLLMs

MLLMs and specialized crowd counting networks leverage distinct strengths and cater to different application paradigms.

MLLMs excel at high-level semantic understanding, open-world reasoning, and object-centric counting in sparse scenarios with distinct instances, such as the FSC-147 benchmark which averages approximately 56 objects per image. Specialized networks like GDSNet are purpose-built for ultra-dense human scenes characterized by extreme scale variations, severe occlusion, and perspective distortion, where benchmarks like ShanghaiTech Part A average 501 persons per image and UCF-QNRF averages 815 persons per image.

Directly applying MLLMs to dense crowd benchmarks remains highly challenging due to visual tokenization resolution limits and auto-regressive numeric regression bottlenecks. Table 10 compares reported ShanghaiTech Part A results, highlighting this performance gap. External model results are taken from CountGD++ [46] and WS-COC [47].

Table 10: Reported MLLM and GDSNet results on ShanghaiTech Part A.

Model / source	Setting	MAE
Molmo [46]	Direct prompting	7.26×10^8
Gemini-2.5 [46]	Direct prompting	517.17
LLaVA-OneVision-7B [47]	Direct prompting	336.80
LLaVA-OneVision-7B / WS-COC [47]	Counting-specific tuning	128.90
GDSNet	Specialized density estimation	48.80

Table 10 indicate a substantial gap between general-purpose MLLMs and specialized models on dense crowd counting. Visual tokenization can limit access to small, heavily occluded heads, while

numerical count prediction introduces an additional challenge. Counting-specific tuning improves MLLM performance, but specialized density estimation remains valuable in these dense scenes.

F Failure Cases and Limitations

In extremely congested scenes, individual heads may occupy only a few pixels. GDSNet can produce a reasonably accurate total count while rendering smooth high-response regions rather than sharply localized peaks, as shown in Fig. 7. Constraining primitives against excessively small support improves training stability but limits response sharpness. The current supervision emphasizes density alignment and count consistency without a strong individual-head localization constraint. Two directions for improvement are extracting centers of high-response primitives or sharpening responses at inference, and jointly optimizing a localization-oriented objective to learn sharper spatial responses alongside accurate counting.

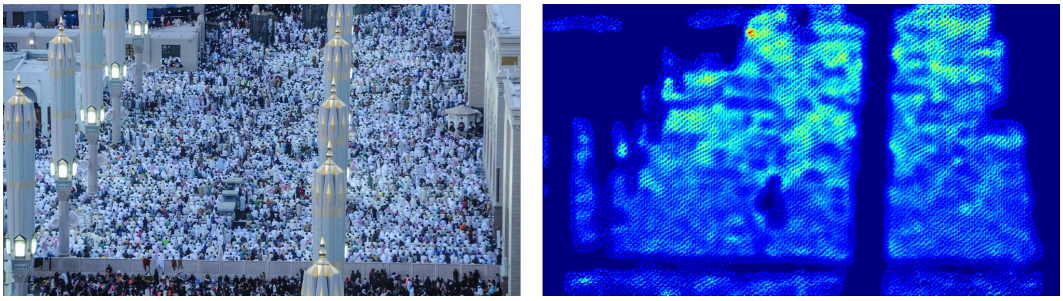


Figure 7: Representative failure case of GDSNet on a highly congested scene. The model produces a smooth density map which lacks sharply localized peaks for individual heads.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the paper's contributions (Sec. 1), and all claims are substantiated by the experimental results in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the Conclusion section (Sec. 5) and supplementary material (*Suppl. F*).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper does not include theoretical results. It is an empirical contribution focused on algorithm design and experimental evaluation.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The method is fully described in Sec. 3 with mathematical formulations for all the components. Implementation details (optimizer, learning rates, batch size, hyperparameters) and implementation algorithm are specified in *Suppl. B*.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code and model checkpoints are not included with this submission. We will release the full codebase and trained model weights upon acceptance to facilitate reproducibility. All datasets used are publicly available (see Sec. 4).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 4 specifies datasets, metrics (MAE, MSE), optimizer (AdamW), learning rates ($1e-5$), batch size (1), hyperparameters (sampling stride $s = 10$, loss temperatures $\lambda_1 = \lambda_2 = 1$, shape threshold $\gamma = 2$). The training and test details can be found in Sec. 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not reported because it would be too computationally expensive.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Hardware specifications are reported in the *Suppl. B*. We provide the complexity and scalability analysis in *Suppl. C*

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have reviewed the NeurIPS Code of Ethics. This research uses only publicly available datasets for crowd counting and does not involve human subjects, sensitive data, or applications with direct harmful potential. Anonymity is preserved in this submission.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

Justification: This work focuses on fundamental algorithmic improvements in crowd density estimation, without immediate deployment scenarios or specific applications. While the proposed continuous density representation theoretically addresses the limitations of existing discrete representations, we currently lack concrete evidence of its societal impacts, either positive or negative, as it represents a general-purpose methodology not tied to particular use cases. We acknowledge this limitation and will address potential applications' societal implications in future applied research.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not release new high-risk models or datasets. It uses standard publicly available datasets and a standard pretrained VGG backbone. No generative models, language models, or scraped datasets are involved.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: The codes and datasets we used is open-source and are cited in this paper.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: No new datasets or models are released with this submission. The codebase and trained model weights will be made publicly available upon acceptance with appropriate documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects. All experiments use publicly available datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This paper does not involve research with human subjects. It is a computational study using publicly available benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, a declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as a component of the core methodology in this research. The proposed method is a vision-based learning framework that does not involve language models.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.