Local Spacing-Aware Hungarian Matching for Stable Point-Supervised Crowd Counting

Kai Jiang[1], Yiming Lin[1], and Zurui Ao[1]*

Beidou Research Institute, School of Geography, South China Normal University, Foshan 528000, China
aozurui1990@gmail.com

Abstract. Point-supervised crowd counting and localization enable end-to-end training by casting prediction as set matching with one-to-one Hungarian assignment. However, point annotations provide no explicit scale information, making distance-based assignment prone to neighbor-induced ambiguity in congested regions, where nearby ground-truth points compete for similar proposals and corrupt supervision. We propose Local Spacing-Aware Hungarian Matching (SAH-matcher), a drop-in replacement that derives a local spacing prior from k-nearest-neighbor distances and performs per-target rescaling of the geometric cost, inducing a more selective effective matching region in dense areas while remaining tolerant in sparse ones. To quantify assignment behavior, we introduce Competitive Ambiguity Score (CAS) and Hijacking Rate (HR) for within-epoch ambiguity and severe hijacking failures, and combine them with Instability Rate (IR) to measure cross-epoch consistency. Experiments on multiple benchmarks show improved assignment reliability and training stability, with particularly clear counting and localization gains in dense or locally ambiguous settings, modest overhead, and favorable transferability across the evaluated point-based frameworks. Code is available at `https://github.com/kaijiang77/SAHCC`.

Keywords: Crowd Counting · Point Supervision · Hungarian Matching · Assignment Stability

1 Introduction

Crowd counting and localization are fundamental problems in computer vision, with applications in public safety, smart-city analytics, and transportation management [1, 2, 15, 19, 28, 30, 38, 42, 45, 47]. Accurate crowd counting and individual localization in real-world scenes remain challenging under severe occlusion, background clutter, and imaging degradations [34, 40]. These difficulties are further compounded by perspective-induced scale variation and highly non-uniform crowd density, which increase ambiguity in point-level localization and matching [11, 46].

* Corresponding author.

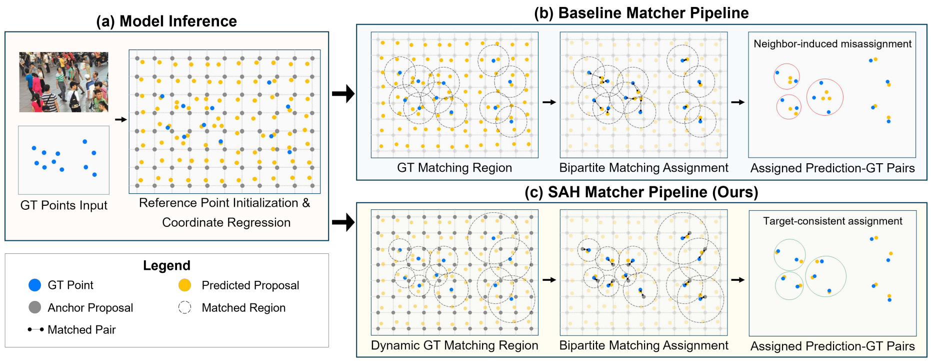

Fig. 1: Comparison of matching pipelines in point-based counting. Baseline Hungarian matching uses a fixed geometric scale, inducing a fixed GT matching region and ambiguous local competition in dense areas. SAH-matcher modulates the geometric cost by local spacing, yielding a target-specific matching region and more consistent one-to-one assignments.

Existing approaches are broadly categorized into map-based and localization-based paradigms. In this context, point-supervised methods have attracted increasing attention because of the low cost of point annotation and their ability to directly support crowd counting and individual localization without heuristic post-processing. In these frameworks, Hungarian matching pairs predicted points (proposals) with ground-truth points (targets), and the accuracy and stability of these matches significantly affect the final counting and end-to-end localization performance.

However, point annotations provide only center locations and contain no explicit scale or geometric information. Consequently, Hungarian assignment in existing point-based frameworks typically combines classification confidence with a fixed-form, scale-agnostic distance cost. In congested regions with high target density, such a distance cost may provide insufficient discriminative power to resolve competition among nearby proposal–target pairs, thereby increasing the likelihood of *neighbor-induced misassignment* and corrupting set-based supervision. This failure mode is illustrated in Fig. 1.

Motivated by this observation, we propose **Local Spacing-Aware Hungarian Matching** (SAH-matcher), a drop-in replacement for standard Hungarian assignment in point-based frameworks. Our key insight is that, although point annotations do not provide explicit scales, they still imply a reliable local spacing prior: the local density and effective scale around each ground-truth point can be approximated by its k-nearest-neighbor distances. SAH-matcher leverages this prior to adaptively rescale the geometric cost for each target, thereby reducing dense-region ambiguity while preserving global one-to-one matching. As also shown in Fig. 1, this spacing-aware modulation yields a target-specific matching range, making the assignment process more stable in congested regions without changing the global Hungarian formulation.

Beyond improving final MAE and RMSE, we further argue that assignment behavior itself should be explicitly diagnosed rather than only inferred from final metrics. To this end, we introduce process-level diagnostics for assignment dynamics. We use Competitive Ambiguity Score (CAS) and Hijacking Rate (HR) to characterize within-epoch ambiguity and *neighbor-hijacking* failures, respectively (i.e., cases where a matched proposal is closer to a neighboring target than to its assigned target), and combine them with Instability Rate (IR) [5] to capture cross-epoch assignment consistency. These diagnostics provide a more interpretable view of how assignment quality and stability evolve during training.

Our contributions are summarized as follows:

- We propose **SAH-matcher**, which incorporates target-wise local spacing into Hungarian geometric costs to better distinguish competing proposals in dense regions while preserving global one-to-one assignment.
- We introduce **CAS** and **HR**, together with **IR**, to diagnose assignment ambiguity, neighbor hijacking, and cross-epoch instability.
- Experiments across crowd benchmarks and point-based frameworks show improved counting, localization, and assignment reliability, particularly in dense or locally ambiguous settings.

2 Related Work

We review representative paradigms for crowd counting and localization, and then focus on prior efforts that improve assignment quality in point-based one-to-one frameworks.

2.1 Map-based Approaches

Map-based methods typically convert point annotations into pseudo supervisory signals, such as density maps or response maps, and train a regressor whose integral yields the crowd count [11, 14, 46]. Studies have improved this paradigm through stronger architectures and better learning objectives [9, 17, 20, 24, 27, 33, 39, 43]. These methods are often effective for counting but require heuristic post-processing, such as peak extraction or thresholding, for localization [1, 12]. As a result, their optimization objectives are not naturally aligned with end-to-end point localization, especially in highly congested regions where overlapping responses reduce separability.

2.2 Localization-based Approaches

Localization-based methods predict instance-level outputs explicitly. Detection-based methods cast crowd counting as object detection, regress bounding boxes or center-scale representations, and suppress duplicates via non-maximum suppression [29]. Under point supervision, pseudo box scales are estimated from local geometry and refined during training [25, 32]. However, heavy overlap in dense crowds may weaken duplicate suppression and complicate scale estimation [8, 41].

Point-based methods offer a more natural alternative by directly predicting point proposals and matching them one-to-one with ground-truth points via Hungarian assignment [5, 18, 23, 35]. This formulation eliminates the need for bounding boxes and non-maximum suppression, and its performance is highly dependent on assignment correctness and stability.

2.3 Point-based One-to-One Counting and Assignment Stabilization

P2PNet [35] introduces DETR-style set prediction [4] into crowd counting by matching point proposals to annotations through Hungarian assignment using a globally fixed point-distance cost regardless of local target spacing. Subsequent works improve point-based counting and localization from different perspectives: CLTR [18] augments Hungarian matching with neighborhood-context compatibility; PET [23] adaptively organizes point queries through data-dependent quadtree splitting; and APGCC [5] stabilizes proposal selection and optimization through auxiliary point guidance and implicit feature interpolation. In contrast, P2R [21] relaxes strict point-to-point correspondence into region-level supervision, offering a distinct alternative to conventional one-to-one assignment.

Our work focuses on an underexplored aspect within one-to-one matching: the geometric cost is typically governed by a globally shared scale and is not explicitly calibrated to the local spacing of each target. SAH intervenes at a distinct level by directly calibrating the geometric matching cost itself on a per-target basis, using local spacing to rescale the corresponding entries in the Hungarian cost matrix. It preserves the cardinality-constrained one-to-one formulation while inducing an effective target-specific matching region adapted to local crowd structure.

3 Preliminaries: Point-based One-to-One Counting

We briefly review the standard set-prediction pipeline used in point-based crowd counting, focusing on the assignment and supervision components most relevant to our method [4, 35].

3.1 Problem Setup

Given an image $I \in \mathbb{R}^{H\times W\times 3}$ with point annotations

$$\mathcal{G} = \{\mathbf{g}_i\}_{i=1}^{M}, \quad \mathbf{g}_i \in \mathbb{R}^2, \tag{1}$$

a point-based model predicts a set of Q proposals

$$\hat{\mathcal{Q}} = \{(\hat{c}_j, \hat{\mathbf{p}}_j)\}_{j=1}^{Q}, \tag{2}$$

where $\hat{c}_j \in [0, 1]$ denotes the foreground confidence and $\hat{\mathbf{p}}_j \in \mathbb{R}^2$ denotes the predicted point coordinate. Typically, $Q \geq M$, so not all proposals are expected to match a ground-truth point.

3.2 Hungarian Assignment

Training establishes one-to-one correspondences between proposals and ground-truth points through Hungarian matching. Let π be an injective mapping from GT indices to proposal indices. The optimal assignment is obtained by minimizing the total matching cost

$$\hat{\pi} = \arg\min_{\pi} \sum_{i=1}^{M} C_{\pi(i),i}, \tag{3}$$

where $C_{j,i}$ measures the cost of assigning proposal j to target i.

A common formulation combines classification confidence with geometric proximity:

$$C_{j,i} = \lambda_{\mathrm{cls}} C_{j,i}^{\mathrm{cls}} + \lambda_{\mathrm{geo}} C_{j,i}^{\mathrm{geo}}, \tag{4}$$

with

$$C_{j,i}^{\mathrm{cls}} = -\hat{c}_j \quad (\text{or } -\log \hat{c}_j), \qquad C_{j,i}^{\mathrm{geo}} = \|\mathbf{u}_j - \mathbf{g}_i\|_2 . \tag{5}$$

Here $\mathbf{u}_j$ denotes the reference point used for matching. In standard point-based formulations, Hungarian matching is usually defined on the regressed proposal coordinates $\hat{\mathbf{p}}_j$ [5, 35].

3.3 Set-based Supervision

Once the assignment $\hat{\pi}$ is obtained, matched proposals are treated as foreground and the remaining proposals are treated as background:

$$\mathcal{Q}^{+} = \{\hat{\pi}(i) \mid i = 1, \dots, M\}, \qquad \mathcal{Q}^{-} = \{1, \dots, Q\} \setminus \mathcal{Q}^{+}. \tag{6}$$

Training then applies binary classification supervision to both matched and unmatched proposals, together with point regression on matched pairs. A typical formulation is

$$\mathcal{L}_{\mathrm{cls}} = \frac{1}{Q}\Big(\sum_{j \in \mathcal{Q}^{+}} \mathrm{BCE}(\hat{c}_j, 1) + \lambda_{\mathrm{neg}} \sum_{j \in \mathcal{Q}^{-}} \mathrm{BCE}(\hat{c}_j, 0) \Big), \tag{7}$$

and

$$\mathcal{L}_{\mathrm{loc}} = \frac{1}{M} \sum_{i=1}^{M} \left\| \hat{\mathbf{p}}_{\hat{\pi}(i)} - \mathbf{g}_i \right\|_2^2 . \tag{8}$$

The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{cls}} + \lambda_{\mathrm{loc}} \mathcal{L}_{\mathrm{loc}}. \tag{9}$$

This formulation makes the quality of Hungarian assignment central to training: the assignment directly determines which proposals receive positive supervision and which point targets they are optimized toward.

4 Method

4.1 Overview

Under point supervision, Hungarian assignment typically relies on a fixed-form geometric cost shared by all targets, which becomes ambiguous in crowded scenes with large local scale variation. In dense areas, neighboring ground-truth targets often induce comparable costs for a given proposal, increasing neighbor-induced misassignment and destabilizing set-based supervision; similar matching instability has also been observed in DETR-style training [16].

To address this issue, we propose SAH-matcher, a matcher-level calibration mechanism that rescales the geometric entries of the Hungarian cost matrix according to the local spacing of each ground-truth target. As illustrated in Figure 1, the same proposal-to-target displacement is penalized more strongly for spatially close targets and more tolerantly for isolated ones, thereby inducing an effective target-specific matching region.

SAH preserves both the supervision target and the cardinality-constrained one-to-one assignment formulation, while requiring no modification to the inference pipeline. Its intervention is confined to target-specific calibration of the geometric matching cost, rather than changing the underlying prediction or supervision formulation.

4.2 Local Spacing Prior via k-Nearest Neighbors

We use local point spacing as a lightweight proxy for local density and effective object scale [3,25,46]. Given the ground-truth set $\mathcal{G} = \{\mathbf{g}_i\}_{i=1}^{M}$, let $\mathcal{N}_k(i)$ denote the k nearest neighbors of $\mathbf{g}_i$ (excluding itself). We define the local spacing of target i as

$$d_i = \frac{1}{k} \sum_{\mathbf{g} \in \mathcal{N}_k(i)} \|\mathbf{g} - \mathbf{g}_i\|_2 \,. \tag{10}$$

Smaller d_i indicates a more crowded neighborhood (stronger competition), while larger d_i corresponds to relatively sparse regions. We transform d_i consistently under geometric augmentations so it stays in the same coordinate system as the matching reference.

4.3 SAH-matcher: Spacing-Aware Hungarian Matching

Reference for matching. In standard point-based formulations, Hungarian matching is typically defined on regressed proposal coordinates. We unify both settings by denoting the matching reference as $\mathbf{u}_j$: $\mathbf{u}_j = \hat{\mathbf{p}}_j$ for pred-ref matching, and $\mathbf{u}_j = \mathbf{a}_j$ for anchor-ref matching. In our SAH-based implementation, we compute geometric matching costs on the fixed anchor reference ($\mathbf{u}_j = \mathbf{a}_j$), while the localization loss is still applied to the regressed coordinate $\hat{\mathbf{p}}_j$.

Algorithm 1 SAH-matcher

Require: Proposals $\{(\hat{c}_j, \mathbf{u}_j)\}_{j=1}^{Q}$, GT points $\{\mathbf{g}_i\}_{i=1}^{M}$, spacing prior $\{d_i\}_{i=1}^{M}$
Ensure: Matched indices $(I_{\text{pred}}, I_{\text{gt}})$
for $i = 1$ to M **do**
  $\alpha_i \leftarrow \max(\alpha_{\min}, \kappa/(d_i + \epsilon))$
for $j = 1$ to Q **do**
  for $i = 1$ to M **do**
    $C_{j,i}^{\text{cls}} \leftarrow -\hat{c}_j$
    $C_{j,i}^{\text{geo}} \leftarrow \alpha_i \cdot \|\mathbf{u}_j - \mathbf{g}_i\|_2$
    $C_{j,i} \leftarrow \lambda_{\text{cls}} C_{j,i}^{\text{cls}} + C_{j,i}^{\text{geo}}$
$(I_{\text{pred}}, I_{\text{gt}}) \leftarrow \text{Hungarian}(C)$
return $(I_{\text{pred}}, I_{\text{gt}})$

Spacing-aware geometric cost. We define the raw geometric error as $e_{j,i} = \|\mathbf{u}_j - \mathbf{g}_i\|_2$. In standard Hungarian matching, this term is globally weighted and therefore imposes the same effective matching scale for all targets. SAH instead introduces a target-specific modulation weight

$$\alpha_i = \max\left(\alpha_{\min}, \frac{\kappa}{d_i + \epsilon}\right), \tag{11}$$

where κ is a global scale factor and $\epsilon > 0$ avoids numerical instability when d_i approaches zero. We impose only a lower bound $\alpha_{\min}$ to prevent the geometric term from becoming excessively weak in sparse regions, where large spacing values may otherwise enlarge the feasible matching range and destabilize assignment.

The spacing-aware geometric cost is then defined as

$$C_{j,i}^{\text{geo}} = \alpha_i \, \|\mathbf{u}_j - \mathbf{g}_i\|_2 \,. \tag{12}$$

Smaller spacing yields larger α_i and stronger geometric selectivity, while larger spacing preserves tolerance in sparse regions.

The final SAH-matcher cost keeps the same additive form, with the spacing-aware modulation already absorbed into the geometric term:

$$C_{j,i} = \lambda_{\text{cls}} C_{j,i}^{\text{cls}} + C_{j,i}^{\text{geo}}, \tag{13}$$

where the geometric scaling factor is absorbed into κ for simplicity.

4.4 Architecture Overview

Our framework is built upon P2PNet [35], with the output head adapted to more tightly couple classification and point regression. As illustrated in Fig. 2, SAH-matcher replaces the baseline Hungarian matcher during training for proposal–GT assignment. Apart from this matcher replacement, the baseline and SAH variants use the same network architecture and set-based supervision.

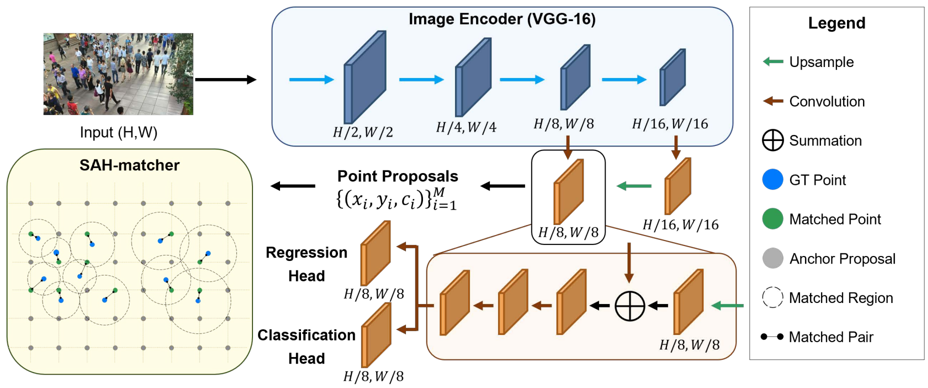

Fig. 2: Architecture overview of the P2PNet-based training pipeline. The output head is adapted to more tightly couple classification and point regression, while SAH-matcher replaces the baseline Hungarian matcher for proposal–GT assignment.

4.5 Assignment Quality Diagnostics

In point-based set-prediction frameworks, proposal–target assignment directly determines the supervision received by each proposal during training. Despite its central role, assignment behavior is usually evaluated only indirectly through final counting or localization performance. We therefore introduce process-level diagnostics that explicitly quantify within-epoch ambiguity and its severe failure cases.

Competitive Ambiguity Score (CAS). Given Hungarian assignment $\hat{\pi}$, define the matched distance and the nearest-competitor distance as

$$d_i^+ = \left\|\mathbf{u}_{\hat{\pi}(i)} - \mathbf{g}_i\right\|_2, \qquad d_i^- = \min_{k \neq i} \left\|\mathbf{u}_{\hat{\pi}(i)} - \mathbf{g}_k\right\|_2, \tag{14}$$

and compute

$$\mathrm{CAS}_i = \frac{2\, d_i^+}{d_i^+ + d_i^- + \epsilon}, \qquad \mathrm{CAS} = \frac{1}{M} \sum_{i=1}^{M} \mathrm{CAS}_i, \tag{15}$$

where $\epsilon > 0$ is a small constant. CAS approaches 0 when the matched proposal is well separated from competing targets, is around 1 when the assigned and competing targets are at comparable distances, and exceeds 1 when the proposal is closer to a competing target than to its assigned target.

Hijacking Rate (HR). To quantify severe failures, we summarize the case $\mathrm{CAS}_i > 1$ as

$$\mathrm{HR} = \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}\left[d_i^+ > d_i^-\right] = \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}\left[\mathrm{CAS}_i > 1\right], \tag{16}$$

which measures the fraction of targets that suffer from severe neighbor hijacking in an epoch. Here, a neighbor hijacking event refers to the case where the proposal matched to target i is actually closer to a competing neighboring target than to $\mathbf{g}_i$ itself, *i.e.*, $d_i^+ > d_i^-$. We regard this as a severe failure because the matched proposal is no longer even geometrically closest to its assigned target, indicating a cross-target takeover under local competition.

CAS and HR are computed from the same Hungarian matches used for optimization and with the same matching reference $\mathbf{u}_j$, so they reflect assignment behavior itself rather than downstream regression noise. Together, CAS captures continuous within-epoch ambiguity, while HR focuses on severe assignment failures.

5 Experiments

5.1 Experimental Setup

Datasets and Metrics. We evaluate on SHHA and SHHB [46], UCF-QNRF [11], JHU-Crowd++ [34], and NWPU-Crowd [40]. We report MAE/RMSE for counting and precision, recall, and F_1 for localization under one-to-one point matching. A predicted point is considered correct if its distance to the matched ground-truth point is within σ pixels. We set σ to 4 and 8 on SHHA [5]. For NWPU-Crowd, we use $\sigma_s = \min(w, h)$ and $\sigma_l = \sqrt{w^2 + h^2}$ as the small and large thresholds, respectively, following the official evaluation protocol [40].

Training Configuration. Our implementation follows the point-proposal framework of P2PNet [35], with several architectural adaptations. We use $s = 8, K = 4$ reference points on ShanghaiTech and $s = 4, K = 1$ otherwise; we use reference anchors for matching and predicted points for regression. Models are trained using AdamW [26] with a batch size of 8 and learning rates of 10^{-4} and 10^{-5} for non-backbone and backbone parameters, respectively. Unless stated otherwise, the spacing prior uses $k = 4$ neighbors, with $\lambda_{\text{cls}} = 1$, $\lambda_{\text{loc}} = 2 \times 10^{-4}$, $\kappa = 1.2$, and $\alpha_{\min} = 0.03$.

Data Processing. For UCF-QNRF, JHU-Crowd++, and NWPU-Crowd, the longer image side is limited to 2048 pixels while preserving the aspect ratio. Training applies random scaling in $[0.7, 1.3]$, horizontal flipping, photometric augmentation [6], and random crops of 128×128 on ShanghaiTech or 256×256 otherwise. At inference, each resized image is processed in full without cropping or tiling.

5.2 Main Results

Crowd counting. Table 1 reports MAE/RMSE of representative detection-based [1, 32, 36], map-based [9, 17, 20, 24, 27, 33, 39, 43], and point-based [5, 18, 21, 23, 35] methods. SAH-matcher achieves competitive results across the evaluated

Table 1: Main results on crowd counting (MAE/RMSE). Best is in bold and second best is underlined for each metric.

Method	SHHA		SHHB		UCF-QNRF		JHU-Crowd++		NWPU(T)	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Detection-based										
LSC-CNN [32]	66.40	117.00	8.10	12.70	120.50	218.20	–	–	–	–
NoisyCC [36]	61.90	99.60	7.40	11.30	85.80	150.60	67.70	258.50	96.90	534.20
TopoCount [1]	61.20	104.60	7.80	13.70	89.00	159.00	60.90	267.40	107.80	438.50
Map-based										
CSRNet [17]	68.20	115.00	10.60	16.00	–	–	85.90	309.20	121.30	387.80
CAN [24]	62.30	100.00	7.80	12.20	107.00	183.00	100.10	314.00	106.30	386.50
S-DCNet [43]	58.30	95.00	6.70	10.70	104.40	176.10	–	–	90.20	370.50
DM-Count [39]	59.70	95.70	7.40	11.80	85.60	148.30	–	–	88.40	388.60
UOT [27]	58.10	95.90	6.50	10.20	83.30	142.30	60.50	252.70	87.80	387.50
ChfL [33]	57.50	94.30	6.90	11.00	80.30	137.60	57.00	235.70	76.80	343.00
MAN [20]	56.80	90.30	–	–	<u>77.30</u>	**131.50**	**53.40**	**209.90**	76.50	323.00
HMoDE+REL [9]	54.40	87.40	6.20	9.80	–	–	55.70	<u>214.60</u>	73.40	331.80
Point-based										
P2PNet [35]	52.74	85.06	6.25	9.90	85.32	154.50	–	–	83.28	553.92
CLTR [18]	56.90	95.20	6.50	10.60	85.80	141.30	59.50	240.60	74.30	333.80
PET [23]	49.34	78.77	6.19	9.69	79.53	144.32	58.50	238.00	74.40	328.50
APGCC [5]	<u>48.80</u>	<u>76.70</u>	**5.60**	**8.70**	80.10	136.60	<u>54.30</u>	225.90	<u>71.40</u>	**284.40**
P2R [21]	51.02	79.68	6.17	9.84	83.26	138.11	58.83	253.10	–	–
Ours	**47.23**	**75.17**	<u>6.14</u>	<u>9.50</u>	**76.91**	<u>135.92</u>	58.32	250.62	**68.90**	<u>306.70</u>

benchmarks, with its clearest improvements on dense or strongly scale-varying datasets such as SHHA, UCF-QNRF, and NWPU, where comparable geometric costs amplify local competition in Hungarian assignment.

This pattern is consistent with our motivation: SAH mainly helps in dense regions, where competing neighboring targets induce comparable geometric costs and make assignments fragile. In sparser scenes, such competition is weaker and the gain correspondingly shrinks, as observed on SHHB. When distribution shift and degradations dominate the error budget, refining assignment alone offers limited headroom, as observed on JHU-Crowd++.

Crowd localization. We further evaluate localization on SHHA and NWPU using Precision, Recall, and F_1 (Tables 2 and 3). On SHHA, SAH-matcher achieves the best F_1 under $\sigma = 4$ and remains competitive with existing methods under $\sigma = 8$. On NWPU, SAH-matcher improves both Precision and Recall more substantially under the stricter σ_s setting, indicating more reliable proposal-to-target alignment. Overall, the gains increase as the localization thresholds become stricter, possibly because stricter thresholds are more sensitive to cross-target misassignment and dense-region ambiguity [40].

5.3 Generalization as a Plug-in Matcher

To evaluate the portability of SAH-matcher, we integrate it into several representative point-based frameworks based on their official repositories and report locally reproduced results without additional hyperparameter retuning, while keeping the remaining training pipeline unchanged. For CLTR [18], we retain

Table 2: Evaluation of crowd localization on SHHA [46] dataset.

Method	$\sigma = 4$			$\sigma = 8$		
	F(%)	P(%)	R(%)	F(%)	P(%)	R(%)
Map-based						
LOBB [13]	25.9	34.9	20.7	53.9	67.6	44.8
Detection-based						
LCFCN [31]	32.5	43.3	26.0	56.3	75.1	45.1
LSC-CNN [32]	32.6	33.4	31.9	62.4	63.9	61.0
TopoCount [1]	41.1	41.7	40.6	73.6	74.6	72.7
Point-based						
P2PNet [35]	40.6	41.5	39.8	74.6	76.2	73.1
CLTR [18]	43.2	43.6	42.7	74.2	74.9	73.5
APGCC [5]	48.7	49.2	48.3	**78.4**	**79.1**	**77.7**
Ours	**48.9**	**49.3**	**48.6**	78.1	78.6	77.6

Table 3: Evaluation of crowd localization on NWPU [40] dataset.

Method	σ_l (large)			σ_s (small)		
	F(%)	P(%)	R(%)	F(%)	P(%)	R(%)
Detection-based						
TinyFaces [10]	56.7	52.9	61.1	52.6	49.1	56.6
TopoCount [1]	63.7	65.1	62.4	–	–	–
Map-based						
RAZ [22]	59.8	66.6	54.3	57.6	47.0	51.7
GL [37]	66.0	**80.0**	56.2	58.7	71.1	50.0
D2CNet [7]	70.0	74.1	66.2	63.2	67.0	59.8
AutoScale [44]	67.3	57.4	62.0	–	–	–
Point-based						
P2PNet [35]	71.2	72.9	69.5	67.5	68.4	66.6
CLTR [18]	68.5	69.4	67.6	59.1	59.9	58.3
PET [23]	74.2	75.2	73.2	67.5	68.4	66.6
APGCC [5]	76.4	79.2	73.6	68.9	71.5	66.5
Ours	**77.7**	77.8	**77.6**	**72.1**	**72.1**	**72.0**

Table 4: Portability of SAH-matcher across point-based frameworks (MAE/RMSE). Avg. Δ is computed as (+SAH − Baseline) over available benchmarks; negative values indicate improvements. Within each method, the better result between Baseline and +SAH is highlighted in bold (lower is better).

Method	Variant	SHHA		SHHB		QNRF		JHU		Avg. Δ	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	ΔMAE	ΔRMSE
P2PNet [35]	Baseline	54.70	89.06	6.29	10.18	93.81	164.21	62.23	269.70	–	–
	+SAH	**50.30**	**81.86**	**6.11**	**9.97**	**85.81**	**151.43**	**60.85**	**264.30**	-3.49	-6.40
CLTR [18]	Baseline	66.53	114.74	7.53	13.16	93.49	166.72	64.17	**257.45**	–	–
	+SAH	**65.20**	**108.89**	**7.49**	**12.62**	**91.11**	**154.90**	**62.74**	258.86	-1.29	-4.20
PET [23]	Baseline	51.90	83.64	**6.56**	**10.23**	95.66	**162.57**	59.98	252.52	–	–
	+SAH	**49.76**	**80.99**	6.76	10.95	**88.99**	164.32	**59.05**	**250.93**	-2.39	-0.44
P2R [21]	Baseline	51.02	79.68	6.95	11.54	93.50	168.48	**62.23**	272.66	–	–
	+SAH	**50.67**	**78.50**	**6.68**	**10.51**	**88.75**	**154.74**	63.45	**267.91**	-1.03	-5.18

its KMO-based Hungarian matcher and context-based auxiliary cost, and apply SAH only to the distance-related term. For P2R [21], we evaluate only its fully supervised branch to avoid confounding effects from pseudo-labeling and teacher–student components.

Results are reported in Table 4. This comparison is intended to assess cross-framework transferability under a controlled setting rather than the best achievable performance of each method after full retuning; the corresponding best-reported results are provided in Table 1. Overall, SAH-matcher improves MAE in most settings, with larger gains on denser benchmarks where neighboring targets induce comparable geometric costs and stronger local competition.

5.4 Assignment Dynamics Diagnostics

CAS/HR by local spacing and IR over epochs. Fig. 3 shows that SAH mainly benefits small-d_i bins (congested regions): CAS drops from 0.63 to 0.44 and HR

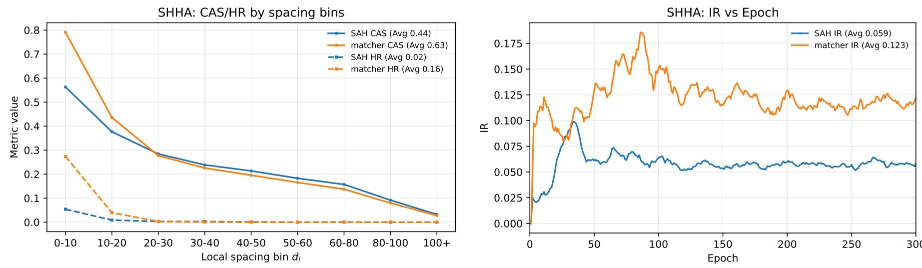

Fig. 3: Assignment diagnostics on SHHA. Left: CAS/HR over spacing bins (averaged over the last 10 epochs). Right: IR over epochs (moving average).

from 0.16 to 0.02, indicating reduced ambiguity caused by comparable distances to competing targets and far fewer severe hijacking events. IR is also consistently lower throughout training and decreases from 0.123 to 0.059, reflecting more stable proposal–target assignments across epochs.

Overall, these diagnostics show that SAH improves assignment quality in densely packed regions with strong local competition. It also makes proposal–target assignments more consistent throughout training. Fig. 4 further illustrates two representative failures, neighbor hijacking and neighbor-induced misassignment, where SAH restores target-consistent one-to-one assignments.

5.5 Ablation Study

Effectiveness of SAH-matcher. We isolate SAH by replacing only the matcher and keeping the backbone, proposal generation, point head, and losses unchanged. Table 5 shows that SAH's per-target rescaling consistently outperforms a global sweep of the geometric-cost weight, indicating that the gain comes from spacing-aware modulation rather than tuning a single coefficient.

The gain is not explained by either switching the matching reference or retuning a globally shared geometric weight. Combining Tables 5 and 6, the predicted-reference baseline obtains 50.89 MAE. Switching to the anchor reference yields only a marginal improvement to 50.70 MAE (-0.19), while retuning the global geometric weight under the same anchor reference further reduces it to 50.06 MAE (-0.83 cumulatively). With the anchor reference fixed, SAH achieves an MAE of 47.23, corresponding to a 2.83 reduction that exceeds the improvements obtained from reference switching and global-weight retuning. These results support the use of anchors as a stable reference for cost evaluation and demonstrate the effectiveness of the proposed target-specific local-spacing calibration.

Hyper-parameters. We ablate the neighborhood size k (Eq. (10)) and the lower bound $\alpha_{\min}$ (Eq. (11)). Table 7 shows robust performance across a reasonable range, with the best setting at $k = 4$ and $\alpha_{\min} = 0.03$.

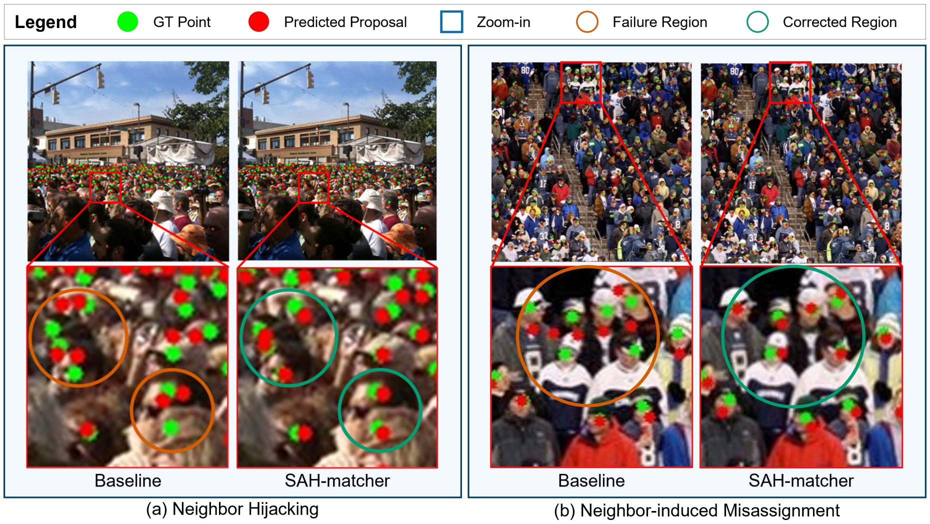

Fig. 4: Qualitative assignment failures and SAH corrections. (a) Neighbor hijacking and (b) neighbor-induced misassignment: baseline matching causes cross-target takeover and ambiguous pairing, whereas SAH restores target-consistent one-to-one assignments.

Table 5: Global geometric-cost weight sweep ($\lambda_{\text{geo}} = w$) vs. SAH dynamic per-target rescaling on SHHA (MAE/RMSE).

Variant	MAE ↓	RMSE ↓
Fixed $w = 0.01$	51.82	85.64
Fixed $w = 0.04$	50.44	79.87
Fixed $w = 0.05$ (baseline)	50.06	81.57
Fixed $w = 0.07$	50.53	81.68
Dynamic per-target (SAH)	**47.23**	**75.17**

Table 6: Effect of matching reference $\mathbf{u}_j$ (anchor points vs. predicted points) on SHHA (MAE/RMSE).

Variant	MAE ↓	RMSE ↓
SAH + predicted points	51.63	86.84
SAH + anchor points	**47.23**	**75.17**
Baseline matcher + predicted points	50.89	87.27
Baseline matcher + anchor points	50.70	81.76

Computation and scalability. With the spacing prior precomputed offline, SAH introduces $O(M)$ computation for evaluating the target-specific factors α_i and $O(QM)$ elementwise rescaling of the geometric-cost matrix, while leaving the cost-matrix dimensions and the Hungarian solver unchanged. As shown in Table 8(a), this results in a 10.50% increase in end-to-end training time on SHHA and no inference-time overhead.

In practice, matching is performed on training crops, so the assignment size depends on the number of visible targets M per crop rather than the annotation count of the full image. The UCF-QNRF profiling results in Table 8(b) show that, although matcher time increases with M for both methods, the relative overhead of SAH decreases from 15.89% to 3.82%. Thus, the additional cost modulation does not become a dominant bottleneck over the observed crop-level assignment range.

Table 7: Hyper-parameter sensitivity of SAH-matcher on SHHA (MAE/RMSE): neighborhood size k and lower bound $\alpha_{\min}$.

Effect of k in the spacing prior

k	MAE ↓	RMSE ↓
1	50.10	81.81
2	48.12	78.98
3	48.85	82.70
4	**47.23**	**75.17**
5	48.73	75.43
6	50.35	79.67
7	49.74	81.14

Effect of lower bound $\alpha_{\min}$

$\alpha_{\min}$	MAE ↓	RMSE ↓
0.01	51.82	85.64
0.02	50.79	82.40
0.03	**47.23**	**75.17**
0.04	48.42	78.81
0.05	48.12	78.98
0.06	49.28	78.86
0.07	48.64	79.38

Table 8: Training overhead and crop-level scalability of SAH-matcher. The upper block reports end-to-end and matcher-only time on SHHA. The lower block reports matcher time over 2,711 non-empty UCF-QNRF crops collected from 500 iterations, where M is the number of visible targets per crop and reaches 1,801.

(a) Overall training-time overhead on SHHA

Matcher	End-to-end (ms/iter) ↓	Matcher-only (ms/iter) ↓	E2E overhead (%) ↓
Baseline matcher	104.44	21.46	–
SAH-matcher	115.41	33.28	10.50

(b) Density-binned matcher time on UCF-QNRF

M	1–4	5–29	30–95	96–431	> 431
Number of crops	782	947	584	358	40
Baseline (ms)	0.472	0.752	1.820	5.311	26.585
SAH (ms)	0.547	0.830	1.913	5.559	27.601
Overhead (%)	15.89	10.33	5.15	4.66	3.82

5.6 Robustness, Failure Modes, and Scope

SAH derives its target-specific calibration from a kNN local-spacing prior and therefore assumes that the annotations preserve reasonably reliable local neighborhood geometry. To evaluate this sensitivity, we perturb the SHHA training annotations by randomly removing points (Drop 10%/20%) or independently displacing each coordinate by an offset sampled from $[-r, r]$ pixels, where $r \in \{2, 4, 6, 8\}$. These perturbations affect both the supervision targets and the derived spacing prior. The baseline and SAH are trained with the same corrupted annotations and evaluated on the original test set.

As shown in Table 9, SAH retains an MAE gain of roughly 2.8 under Drop 10% and jitter up to 4 px. The gain decreases to 1.02 under Drop 20% and nearly vanishes at 6 px as the spacing prior becomes less reliable. Under the severe 8 px perturbation, distorted spacing estimates can induce excessively large calibration factors, causing SAH to underperform the baseline by 12.12 MAE. Capping α_i at 0.08 recovers baseline-level accuracy but not the clean-data gain,

Table 9: Robustness on SHHA under annotation perturbations (MAE / RMSE). ΔMAE denotes SAH minus baseline; negative is better. The capped variant uses $\alpha_i \leq 0.08$ as a diagnostic intervention under the severe jitter setting.

Regime	Setting	Baseline	SAH	ΔMAE
Clean	—	50.06 / 81.57	47.23 / 75.17	−2.83
Mild	Drop 10%	50.95 / 81.14	48.15 / 76.94	−2.80
	Jitter 2 px	50.70 / 82.26	47.83 / 76.50	−2.87
	Jitter 4 px	50.97 / 82.92	48.17 / 77.35	−2.80
Moderate	Drop 20%	52.95 / 87.36	51.93 / 84.42	−1.02
	Jitter 6 px	51.33 / 83.79	51.34 / 83.32	+0.01
Stress	Jitter 8 px	53.38 / 84.66	65.50 / 107.82	+12.12
	+ $\alpha_i \leq 0.08$	53.38 / 84.66	53.14 / 82.62	−0.24

identifying excessive calibration as the primary failure mechanism. SAH therefore relies on sufficiently informative local spacing and faces limitations under severe annotation corruption.

6 Conclusion

We address assignment ambiguity in point-supervised crowd counting, where globally weighted, scale-agnostic matching costs can become unreliable in congested regions. SAH-matcher is a lightweight drop-in replacement that uses a kNN local-spacing prior to rescale the geometric cost for each target while preserving global one-to-one assignment. We further introduce CAS, HR, and IR to explicitly characterize assignment ambiguity and instability.

Across crowd benchmarks and point-based frameworks, SAH-matcher generally improves counting and localization, with the clearest gains in dense and strongly scale-varying scenes. These results show that incorporating local spacing into Hungarian matching provides a practical way to improve the reliability of point-based assignment.

Acknowledgements

This work was supported by the Guangdong Basic and Applied Basic Research Foundation of China (grant nos. 2023A1515110192, 2025A1515010399, and 2024TQ08A345) and the Science and Technology Projects of Xizang Autonomous Region, China (grant no. XZ202501ZY0091).

References

1. Abousamra, S., Hoai, M., Samaras, D., Chen, C.: Localization in the crowd with topological constraints. In: Proceedings of the AAAI conference on artificial intelligence. pp. 872–881 (2021)

2. Alhothali, A., Balabid, A., Alharthi, R., Alzahrani, B., Alotaibi, R., Barnawi, A.: Anomalous event detection and localization in dense crowd scenes. Multimedia Tools and Applications **82**(10), 15673–15694 (2023)
3. Babu Sam, D., Surya, S., Venkatesh Babu, R.: Switching convolutional neural network for crowd counting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5744–5752 (2017)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
5. Chen, I.H., Chen, W.T., Liu, Y.W., Yang, M.H., Kuo, S.Y.: Improving point-based crowd counting and localization based on auxiliary point guidance. In: European Conference on Computer Vision. pp. 428–444. Springer (2024)
6. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning. pp. 1597–1607. PMLR (2020)
7. Cheng, J., Xiong, H., Cao, Z., Lu, H.: Decoupled two-stage crowd counting and beyond. IEEE Transactions on Image Processing **30**, 2862–2875 (2021)
8. Chu, X., Zheng, A., Zhang, X., Sun, J.: Detection in crowded scenes: One proposal, multiple predictions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12214–12223 (2020)
9. Du, Z., Shi, M., Deng, J., Zafeiriou, S.: Redesigning multi-scale neural network for crowd counting. arXiv preprint arXiv:2208.02894 (2022)
10. Hu, P., Ramanan, D.: Finding tiny faces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 951–959 (2017)
11. Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N., Shah, M.: Composition loss for counting, density map estimation and localization in dense crowds. In: Proceedings of the European conference on computer vision (ECCV). pp. 532–546 (2018)
12. Kang, D., Ma, Z., Chan, A.B.: Beyond counting: comparisons of density maps for crowd analysis tasks—counting, detection, and tracking. IEEE Transactions on Circuits and Systems for Video Technology **29**(5), 1408–1422 (2018)
13. Laradji, I.H., Rostamzadeh, N., Pinheiro, P.O., Vazquez, D., Schmidt, M.: Where are the blobs: Counting by localization with point supervision. In: Proceedings of the european conference on computer vision (ECCV). pp. 547–562 (2018)
14. Lempitsky, V., Zisserman, A.: Learning to count objects in images. Advances in neural information processing systems **23** (2010)
15. Lengvenis, P., Simutis, R., Vaitkus, V., Maskeliunas, R.: Application of computer vision systems for passenger counting in public transport. Elektronika ir elektrotechnika. **19**(3), 69–72 (2013)
16. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: Dn-detr: Accelerate detr training by introducing query denoising. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13619–13627 (2022)
17. Li, Y., Zhang, X., Chen, D.: Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1091–1100 (2018)
18. Liang, D., Xu, W., Bai, X.: An end-to-end transformer model for crowd localization. In: European Conference on Computer Vision. pp. 38–54. Springer (2022)
19. Lin, C., Hu, X.: Efficient crowd density estimation with edge intelligence via structural reparameterization and knowledge transfer. Applied Soft Computing **154**, 111366 (2024)

20. Lin, H., Ma, Z., Ji, R., Wang, Y., Hong, X.: Boosting crowd counting via multifaceted attention. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19628–19637 (2022)
21. Lin, W., Zhao, C., Chan, A.B.: Point-to-region loss for semi-supervised point-based crowd counting. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 29363–29373 (2025)
22. Liu, C., Weng, X., Mu, Y.: Recurrent attentive zooming for joint crowd counting and precise localization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1217–1226 (2019)
23. Liu, C., Lu, H., Cao, Z., Liu, T.: Point-query quadtree for crowd counting, localization, and more. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1676–1685 (2023)
24. Liu, W., Salzmann, M., Fua, P.: Context-aware crowd counting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5099–5108 (2019)
25. Liu, Y., Shi, M., Zhao, Q., Wang, X.: Point in, box out: Beyond counting persons in crowds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6469–6478 (2019)
26. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
27. Ma, Z., Wei, X., Hong, X., Lin, H., Qiu, Y., Gong, Y.: Learning to count via unbalanced optimal transport. In: Proceedings of the AAAI conference on artificial intelligence. pp. 2319–2327 (2021)
28. Mccarthy, C., Ghaderi, H., Marti, F., Jayaraman, P., Dia, H.: Video-based automatic people counting for public transport: On-bus versus off-bus deployment. Computers in Industry **164**, 104195 (2025)
29. Neubeck, A., Van Gool, L.: Efficient non-maximum suppression. In: 18th international conference on pattern recognition (ICPR'06). vol. 3, pp. 850–855. IEEE (2006)
30. Nurseitov, D., Bostanbekov, K., Toiganbayeva, N., Zhalgas, A., Yedilkhan, D., Amirgaliyev, B.: Vision-based people counting and tracking for urban environments. Journal of Imaging **12**(1), 27 (2026)
31. Ribera, J., Guera, D., Chen, Y., Delp, E.J.: Locating objects without bounding boxes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6479–6489 (2019)
32. Sam, D.B., Peri, S.V., Sundararaman, M.N., Kamath, A., Babu, R.V.: Locate, size, and count: accurately resolving people in dense crowds via detection. IEEE transactions on pattern analysis and machine intelligence **43**(8), 2739–2751 (2020)
33. Shu, W., Wan, J., Tan, K.C., Kwong, S., Chan, A.B.: Crowd counting in the frequency domain. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19618–19627 (2022)
34. Sindagi, V.A., Yasarla, R., Patel, V.M.: Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. IEEE transactions on pattern analysis and machine intelligence **44**(5), 2594–2609 (2020)
35. Song, Q., Wang, C., Jiang, Z., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Wu, Y.: Rethinking counting and localization in crowds: A purely point-based framework. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3365–3374 (2021)
36. Wan, J., Chan, A.: Modeling noisy annotations for crowd counting. Advances in Neural Information Processing Systems **33**, 3386–3396 (2020)

37. Wan, J., Liu, Z., Chan, A.B.: A generalized loss function for crowd counting and localization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1974–1983 (2021)
38. Wan, J., Wu, Q., Lin, W., Chan, A.: Robust zero-shot crowd counting and localization with adaptive resolution sam. In: European Conference on Computer Vision. pp. 478–495. Springer (2024)
39. Wang, B., Liu, H., Samaras, D., Nguyen, M.H.: Distribution matching for crowd counting. Advances in neural information processing systems **33**, 1595–1607 (2020)
40. Wang, Q., Gao, J., Lin, W., Li, X.: Nwpu-crowd: A large-scale benchmark for crowd counting and localization. IEEE transactions on pattern analysis and machine intelligence **43**(6), 2141–2149 (2020)
41. Wang, X., Xiao, T., Jiang, Y., Shao, S., Sun, J., Shen, C.: Repulsion loss: Detecting pedestrians in a crowd. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7774–7783 (2018)
42. Wang, Y., Hou, J., Hou, X., Chau, L.P.: A self-training approach for point-supervised object detection and counting in crowds. IEEE Transactions on Image Processing **30**, 2876–2887 (2021)
43. Xiong, H., Lu, H., Liu, C., Liu, L., Cao, Z., Shen, C.: From open set to closed set: Counting objects by spatial divide-and-conquer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 8362–8371 (2019)
44. Xu, C., Liang, D., Xu, Y., Bai, S., Zhan, W., Bai, X., Tomizuka, M.: Autoscale: Learning to scale for crowd counting. International Journal of Computer Vision **130**(2), 405–434 (2022)
45. Zahra, A., Ghafoor, M., Munir, K., Ullah, A., Ul Abideen, Z.: Application of region-based video surveillance in smart cities using deep learning. Multimedia Tools and Applications **83**(5), 15313–15338 (2024)
46. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 589–597 (2016)
47. Zhao, C., Zhang, Z., Hu, J., Wang, D., Fan, B., Pan, Q., He, Q.: Crowd anomaly event detection in surveillance video based on the evolution of the spatial position relationship feature. In: 2018 IEEE 14th International Conference on Control and Automation (ICCA). pp. 247–252. IEEE (2018)