Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.077655

ARTICLE

FSS: Focusing on Suboptimal Samples for Detector-Agnostic Label Assignment in Object Detection

Lijuan Huang[1,2], **Zhixian Liu**[3], **Xinyu Zhou**[4], **Jinping Liu**[4,*], **Kunyi Zheng**[4] **and Yimei Yang**[2,4,*]

[1]Hunan Intelligent Rehabilitation Robot and Auxiliary Equipment Engineering Technology Research Center, Changsha, China
[2]School of Computer and Artificial Intelligence (School of Software), Huaihua University, Huaihua, China
[3]School of Business, Hunan Normal University, Changsha, China
[4]College of Information Science and Engineering, Hunan Normal University, Changsha, China

*Corresponding Authors: Jinping Liu. Email: ljp@hunnu.edu.cn; Yimei Yang. Email: yangym@hunnu.edu.cn

Received: 14 December 2025; Accepted: 11 March 2026; Published: 08 May 2026

ABSTRACT: Many occluded and ambiguous ground truths exist in object detection, making detectors unable to obtain optimal training samples. In this article, we revisit the suboptimal sample issue in label assignment for object detection and propose a novel detector-agnostic strategy, termed FSS, to address it. FSS reformulates label assignment as the process of selecting high-quality sub-optimal samples and progressively transforming them into optimal ones. Specifically, for each candidate, we estimate the probability of being an optimal sample by jointly considering localization quality and classification confidence, thereby constructing an instance-wise probability matrix. Based on the spatial distribution of potentially optimal samples, we introduce a Gaussian prior to adaptively determine the number of sub-optimal samples per instance. We then assign weights to these sub-optimal samples according to their optimality probabilities, enforcing consistent ranking between classification and localization and promoting the emergence of truly optimal samples. Extensive experiments on MS-COCO demonstrate the effectiveness and plug-and-play nature of FSS: when integrated into a modern one-stage detector, FSS achieves 50.8 AP under single-model, single-scale testing, without introducing any additional inference overhead.

KEYWORDS: Object detection; label assignment; suboptimal samples selection; Gaussian-prior dynamic-*k*

1 Introduction

Object detection, a fundamental yet still challenging aspect of computer vision, aims to localize and classify objects in images while suppressing irrelevant background interference. With the rapid development of deep learning, object detection has achieved remarkable progress. Current object detection approaches can be categorized into multi-stage and one-stage methods.

Multi-stage detectors typically follow a proposal-driven pipeline: candidate regions are generated to separate foreground from background, pruned to remove redundancy, and then refined by subsequent detection heads. Owing to progressive refinement and explicit control of the positive/negative (Pos/Neg) ratio, they often outperform one-stage methods, albeit with higher architectural complexity and computational cost. In contrast, *one-stage detectors* predict classification and box regression densely on feature maps, without an explicit proposal stage. Anchor-based variants use predefined anchors with a single refinement step, offering high efficiency. However, dense feature pyramid network (FPN) [1] predictions generate large numbers of candidates and induce severe class imbalance, with positives rare relative to negatives. This imbalance largely

accounts for the accuracy gap to multi-stage detectors: the latter explicitly regulates the Pos/Neg ratio via proposals, whereas one-stage methods must rely on loss design and sampling.

To alleviate the above-mentioned problem, RetinaNet introduces Focal Loss [2] to down-weight abundant negatives and emphasize hard positives. While effective, it does not resolve the fundamental scarcity of positive samples. Fully Convolutional One-Stage object detection (FCOS) [3] increases the number of positives by labeling points near each ground-truth center as positives across FPN levels, but the resulting set may include low-quality or ambiguous samples, potentially hindering convergence and final accuracy.

These observations raise a central question in dense detection: how to select informative candidates and assign them as positive or negative with respect to each ground-truth object, a process commonly termed *label assignment*. Recent studies [4,5] show that assignment design—spanning matching metrics, decision thresholds, and spatial/predictive priors—is a key, yet often overlooked, determinant of detection performance. Existing approaches broadly fall into two paradigms: static and dynamic label assignment. Static strategies [5,6] label anchors as positive using fixed IoU thresholds or hand-crafted spatial priors (e.g., grid cell centers), assigning the rest to negatives or discarding them. They are simple and efficient, but often brittle to variations in object scale, shape, and density, leading to suboptimal matches for irregular, small, or crowded targets. In contrast, dynamic label assignment methods [7] adapt criteria to the model's current predictions (e.g., classification confidence and localization quality), enabling more context-aware assignments in diverse scenes.

However, an important issue has received relatively little attention: in realistic detection scenarios, truly *optimal* samples may be absent or extremely rare. Due to occlusions, extreme aspect ratios, small object sizes, and cluttered backgrounds, many candidate samples may exhibit a mismatch between classification confidence and localization quality. In other words, the sample with the highest classification score is not necessarily the one with the best IoU, and vice versa. Moreover, it is often impossible to determine a priori whether a given sample is globally optimal.

To formalize this notion, we define a label-metric score s that jointly captures the classification and localization quality. Given a ground-truth object, the score of the i-th candidate is defined as

$$s = {cls_{score}}^{\alpha} \times IoU^{\beta} \tag{1}$$

where IoU measures the overlap between the predicted bounding box and the ground truth, and cls_{score} denotes the classification confidence score. The hyperparameters α and β balance the relative contributions of classification and localization, respectively. When $\alpha = 0$, the metric degenerates to the IoU-based metric. A larger s indicates a higher-quality sample. If an *optimal* sample $\hat{i}$ exists and is assigned to a valid ground-truth target, a sufficient condition for reaching the optimum is that there exists a positive sample that ranks first in both classification and localization, i.e.,

$$\hat{i} = \arg\max_{i} s_i \tag{2}$$

where i indexes the candidate samples. We refer to any sample that attains the maximum in Eq. (2) as a *potentially optimal* sample. In practice, an instance may not admit a uniquely optimal sample under the chosen metric; nevertheless, at least one potentially optimal candidate can always be identified by maximizing the score.

Fig. 1 illustrates the presence of uncertain samples in object detection, which can induce inconsistencies between the classification and localization rankings. We therefore collect these high-performing yet uncertain suboptimal samples and harmonize their task-specific rankings. It is worth noting that, if a truly optimal sample exists, it should maximize any reasonable label metric irrespective of the specific choices of α and β.

By contrast, most positive samples in dense detection are suboptimal: they are competitive under the chosen metric, yet fail to attain the best ranking in both classification and localization simultaneously. In this work, we focus on these suboptimal positives (hereafter *suboptimal samples*) and argue that they are critical for further improving detector performance.

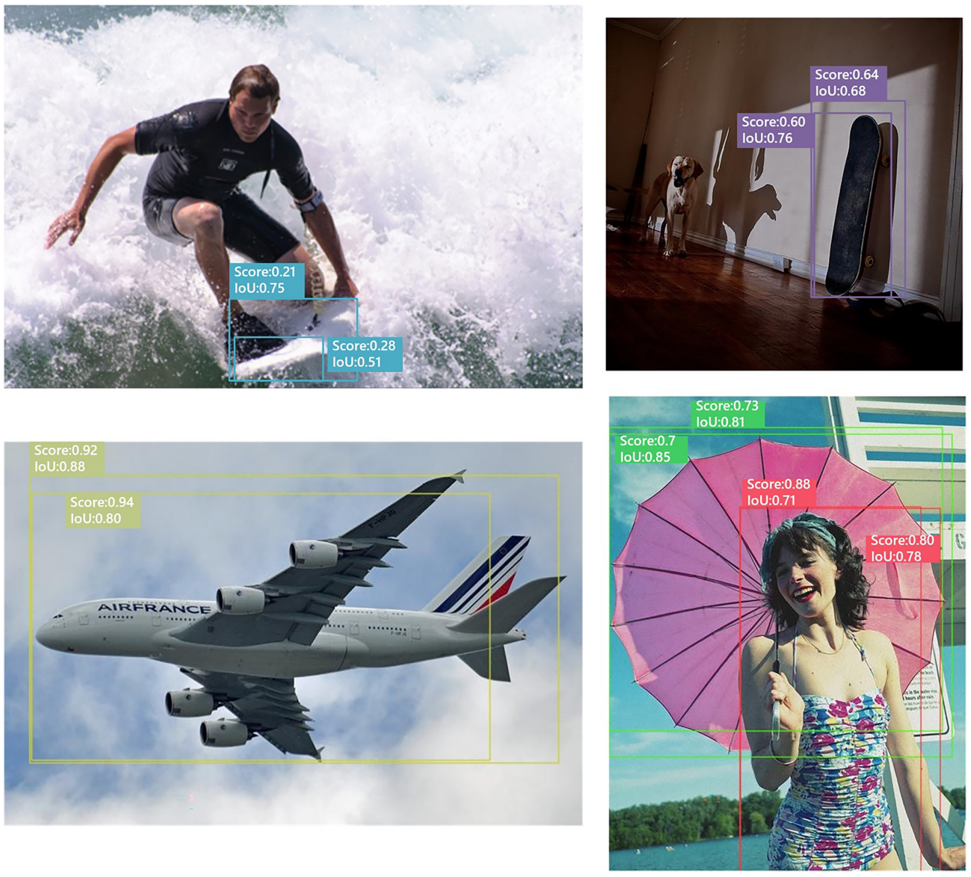

Figure 1: Illustrative uncertain samples existing in object detection, which lead to inconsistencies in classification and localization rankings. To address this, we collect these well-performing yet ambiguous suboptimal samples and align their rankings across tasks.

In summary, truly optimal samples rarely occur in real-world scenes, yet whenever they do, they are almost surely labeled as positives. Consequently, the detector is largely shaped by the abundant *suboptimal samples*. Our goal is to **focus on suboptimal samples (FSS)**, which consists of two steps: (1) **Selecting suboptimal samples.** Rather than using static heuristics, FSS builds an instance-wise probability matrix to estimate how likely each candidate is to be optimal, jointly accounting for classification confidence and regression quality. (2) **Transforming suboptimal samples into optimal ones.** FSS assigns each selected candidate an instance-specific weight derived from its probability score. These weights encourage consistent ordering between classification and localization by amplifying the learning signal for higher-ranked candidates. The main contributions of this article are summarized as follows:

- The underexplored role of *suboptimal positives* in dense label assignment is identified and formalized, with emphasis placed on regimes where truly optimal samples are absent or unreliable.
- A unified probability score coupling classification confidence and localization quality is introduced, based on which a Gaussian-prior-guided dynamic-k selection strategy and an instance-wise weighting

scheme are devised to select high-quality suboptimal candidates and preserve ranking consistency between classification and regression.
- The detector-agnostic applicability of the proposed FSS with no additional inference overhead is validated on MS-COCO and DOTA benchmarks, where consistent gains over representative baselines and competitive performance are achieved.

The remainder of this article is organized as follows. Section 2 reviews related work on one-stage object detection and label assignment. Section 3 presents the proposed label assignment strategy, FSS. Section 4 reports extensive experiments on the MS-COCO and DOTA benchmark datasets and compares our method with state-of-the-art approaches. Section 5 concludes the paper and outlines directions for future research.

2 Related Works

This section briefly reviews related works on one-stage object detection and label assignment strategies for dense detectors.

2.1 One-Stage Object Detection

Depending on the design, one-stage detectors can be anchor-based or anchor-free. Anchor-based detectors rely on a set of predefined anchor boxes (priors) with different scales and aspect ratios as regression references, which are often designed using statistics (e.g., clustering) over the training set. Anchor-free methods dispense with explicit anchors and instead predict bounding boxes from points or keypoints on feature maps, leading to simpler designs and often better robustness to extreme aspect ratios and small objects. OverFeat [8] is among the earliest deep learning-based one-stage detectors, introducing a unified framework for joint classification, localization, and detection. YOLO [5] formulates object detection as a single regression problem: it partitions the final feature map into a 7×7 grid, where each cell predicts class scores and bounding-box coordinates. SSD [6] further extends this paradigm by exploiting multi-scale feature maps and a set of default boxes with diverse aspect ratios, thereby discretizing the bounding-box space and enabling multi-scale detection within a single forward pass.

Anchor-free detectors further reduce reliance on handcrafted priors. CornerNet [9] casts detection as paired keypoint prediction by producing heatmaps for the top-left and bottom-right corners. FCOS [3] treats each pixel on feature maps as a candidate location and regresses distances to the four box sides, while an additional centerness branch suppresses low-quality predictions. DETR [10] introduces Transformers and reformulates detection as a set-prediction problem, using Hungarian matching between a fixed set of object queries and ground-truth boxes, thereby removing the need for anchor design and non-maximum suppression.

These developments have significantly improved the accuracy and simplicity of one-stage detectors, enabling their benchmarking and application across diverse dense detection scenarios. However, they also highlight a key bottleneck: how to effectively assign labels to the large number of dense candidates.

2.2 Label Assignment

During training, each candidate (anchor or point) must be assigned to a ground-truth instance or to the background prior to loss computation; this positive/negative assignment shapes optimization and largely determines detector performance.

Early detectors such as Faster R-CNN [11] and RetinaNet mainly rely on anchor ground-truth Intersection over Union (IoU): candidates above a preset threshold are labeled as positives, whereas those below a lower threshold are labeled as negatives. In contrast, FCOS [3] and YOLO [5] incorporate spatial constraints.

YOLO assigns responsibility to anchors whose centers fall in the grid cell containing the ground-truth center, while FCOS and FoveaBox [12] expand the positive set by treating points within a region around each ground truth as positives.

Despite their differences, these approaches share a key limitation: positives and negatives are separated by a single hand-crafted criterion (e.g., IoU threshold, scale, or spatial rule). Such fixed heuristics can yield noisy or ambiguous supervision and fail to exploit richer context among candidates, thereby limiting adaptivity and attainable performance.

To improve adaptivity, a range of dynamic assignment strategies has been proposed. ATSS [13] derives instance-specific IoU thresholds from the mean and standard deviation of candidate IoUs. FreeAnchor [14] casts assignment as maximum-likelihood estimation, allowing anchors to select ground truths via learned likelihood scores. Zhang et al. [15] model quality scores with a Gaussian mixture and use EM to probabilistically separate positives from negatives. AutoAssign [16] introduces instance-wise labeling via central/confidence weighting modules, and DW [17] assigns task-aware weights to both positive and negative samples.

More recently, several methods explicitly combine classification confidence and localization quality for matching and/or weighting. SimOTA (and OTA) [18] performs dynamic-k selection and minimum-cost matching based on classification and regression losses. TOOD [19] designs a task-aligned score to guide sample selection and weighting, while GFL/GFLv2 [20,21] injects localization quality into classification/regression via quality-aware objectives and distributional box modeling. However, while improving task consistency, current methods typically treat label assignment as an instantaneous matching or alignment problem. They often fail to account for the scarcity of truly optimal samples in complex scenes and overlook the potential of explicitly modeling the abundant *suboptimal* positives to drive a progressive evolution toward optimality.

3 Proposed FSS Framework

The overall pipeline of the proposed FSS framework is illustrated in Fig. 2. For each spatial location, the detector outputs a classification score, an objectness score, and a bounding-box offset. Based on the label metric s, an instance-wise probability matrix is devised to estimate the likelihood of each candidate being optimal. We then introduce a Gaussian-prior dynamic-k scheme to adaptively determine how many suboptimal samples should be selected for each ground-truth instance. This design encourages the selection to concentrate around the potentially optimal sample, yielding a more reasonable and spatially coherent set of suboptimal positives. Finally, it converts the probability scores of the selected samples into instance-specific weights to preserve their relative ranking and progressively promote suboptimal samples toward optimal ones during training.

3.1 Choosing Suboptimal Samples

3.1.1 Probability Matrix: The Likelihood of Being an Optimal Sample

Conventional sample-quality metrics typically rely on IoU thresholds or spatial constraints as a proxy for geometric alignment with the assigned ground truth. In dynamic label assignment, each ground-truth instance is usually matched to multiple candidates (n:1), meaning that most matched candidates are inherently suboptimal. Selecting them indiscriminately can introduce noisy supervision and force the detector to fit poorly located or shaped anchors/points. In contrast, a truly optimal sample should be consistently favored across reasonable metrics, motivating us to focus on *high-quality* suboptimal samples.

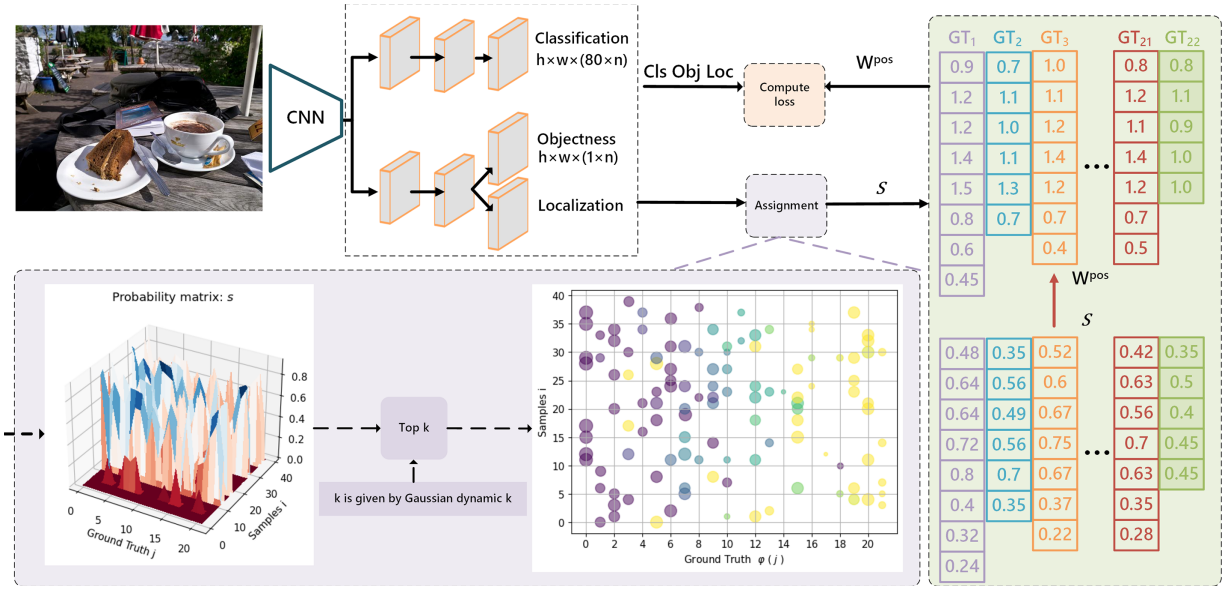

Figure 2: The pipeline of the proposed FSS. The model consists of a CNN-based backbone and a detection head. The classification score (h × w × 80) and the confidence score ($h \times w \times 1$) are obtained as the final classification score (cls_{score}), and n denotes the number of anchors associated with each grid cell. The lower (purple) module illustrates the label assignment process, where the scatter plot visualizes the selected indices, and the marker size reflects the corresponding probability score s. The right (green) module converts s into instance-specific weights, thereby preserving the ranking structure used for optimization.

For such suboptimal samples, the rankings induced by IoU and by classification confidence should be as consistent as possible with respect to the corresponding ground truth. From the viewpoint of the joint label metric s, we interpret s as the (unnormalized) probability that a candidate is the optimal sample; thus, it should faithfully capture overall prediction quality so as to prioritize better suboptimal candidates. Accordingly, we construct an instance-wise probability matrix that jointly encodes classification and localization quality,

$$s_{i,j} = \left(p_i^{cls} \cdot p_i^{obj}\right)^{\alpha} \cdot IoU\left(p_i^{box}, gt_j\right)^{\beta} \tag{3}$$

where p_i^{cls}, p_i^{obj}, and p_i^{box} denote the predicted classification score, objectness, and bounding-box offsets of sample i, respectively, and gt_j denotes the j-th ground-truth instance. $IoU(p_i^{box}, gt_j)$ denotes the IOU ratio between the predicted box decoded from p_i^{box} and gt_j.

The resulting score $s_{i,j}$ is therefore defined by a higher-order combination of classification confidence and localization quality. Maximizing $s_{i,j}$ encourages the network to favor suboptimal candidates that are jointly strong in both tasks, rather than candidates that excel in only one.

Training stability. Eq. (3) is used only to generate supervision (sample selection and loss reweighting) and is *not* treated as a differentiable objective. In implementation, we compute $s_{i,j}$ under a stop-gradient operation (i.e., we detach p_i^{cls}, p_i^{obj}, and the decoded boxes used by $IoU(\cdot)$), so no gradients are back-propagated through the IoU computation. Therefore, although the predicted boxes can be noisy in early epochs, this noise affects only the temporary assignment/weighting decision and does not directly destabilize optimization via IoU gradients. In addition, our instance-wise normalization and the smoothing factor u in the weighting function (Section 3.2) further prevent overly sharp weights at the early stage.

3.1.2 Gaussian Prior Dynamic k

Determining how many suboptimal samples should be assigned to each instance is crucial for stable and effective training. Many existing methods control this quantity with a fixed hyperparameter or a static threshold, overlooking substantial instance-level variability: heavily occluded objects may provide only a few reliable candidates, whereas large, well-defined objects can support many. Since this factor is difficult to model analytically, OTA [18] proposed a simple yet effective heuristic, termed *dynamic k*, to adaptively estimate the number of positive samples for each ground-truth instance.

Given the j-th ground-truth object gt_j, dynamic k first selects the top-q candidates with the largest overlaps,

$$\{i^*\} = \mathrm{TOP}_q \, IoU(p_i^{box}, gt_j), \tag{4}$$

where $\{i^*\}$ denotes the index set of these q candidates. It then computes an instance-specific value k by summing their IoUs, i.e.,

$$k = \sum_{i \in \{i^*\}} IoU(p_i^{box}, gt_j). \tag{5}$$

Consequently, candidates with larger overlaps contribute more to k and are more likely to be selected as positives.

However, the localization-quality landscape over feature locations can be discrete and irregular: many regions inside an object may yield high IoU yet remain weakly discriminative. As a result, IoU-only dynamic k may admit noisy positives—candidates that localize well but have low classification confidence—which do not faithfully reflect the overall matching quality. Ideally, the selected candidates should form a compact neighborhood around the *potentially optimal sample* (Eq. (2)), as illustrated in Fig. 3.

Figure 3: Visualization of Gaussian-prior dynamic k. (**a**) input image; (**b**) classification-score map; (**c**) IoU map; (**d**) Gaussian prior. For easy objects, (**b**,**c**) are well aligned; for hard cases, they diverge. The Gaussian prior suppresses distracting regions and prioritizes candidates near the potentially optimal sample.

To encourage such compactness, we introduce a 2-D Gaussian prior $\mathcal{N}_j(\mu_j, \Sigma_j)$ centered at the potentially optimal sample for gt_j. Let $\hat{i}_j = \arg\max_i s_{i,j}$ denote the index of the potentially optimal candidate for gt_j, and let $(x_{\hat{i}_j}, y_{\hat{i}_j})$ be its center coordinates. With (w_j, h_j) denoting the width and height of gt_j, we define

$$\mu_j = \begin{bmatrix} x_{\hat{i}_j} \\ y_{\hat{i}_j} \end{bmatrix}, \quad \Sigma_j = \begin{bmatrix} w_j^2 & 0 \\ 0 & h_j^2 \end{bmatrix}. \tag{6}$$

We emphasize that the 2-D Gaussian prior is a lightweight heuristic that regularizes the spatial distribution of *candidate centers* rather than assuming that an object itself strictly follows a Gaussian shape. As a soft, instance-wise re-ranking term, it is primarily intended to suppress spatially distant candidates

and encourage compact selection around the potentially optimal sample. Potential failure cases include highly elongated or irregular objects, fragmented instances under heavy occlusion, and crowded scenes with overlapping objects, where the optimal candidate region may be non-elliptical. In such cases, the prior may be less accurate, but its effect remains bounded because the final selection is still jointly governed by localization and classification quality (via IoU and $s_{i,j}$).

To visually analyze potential failure cases, we present an asymmetric airplane case in Fig. 4. The airplane in the image has an irregular shape with an incomplete left wing. The classification peak (b) is localized on the fuselage, causing the center-focused prior (c) to neglect the wide wings. Consequently, the detection box (red) is suppressed and clipped by the rigid prior weights, even when the manual ground truth (green) is accurately defined. This confirms that unimodal priors struggle with non-convex or protruding geometries.

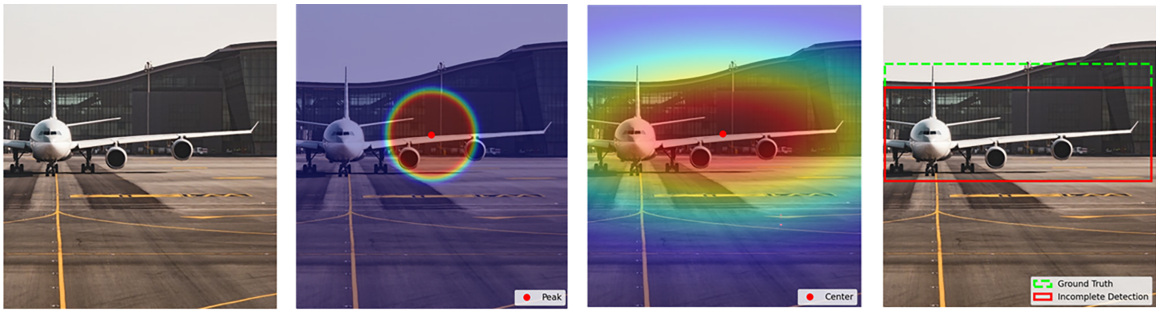

Figure 4: Failure analysis on an irregular object. (**a**) Input; (**b**) Classification heatmap; (**c**) Gaussian prior; (**d**) Ground truth (GT) vs. suppressed detection.

This size-dependent covariance yields a scale-adaptive prior: small (large) objects naturally induce a narrower (broader) spatial support, which is consistent with the typical extent of reliable candidates. Moreover, since the prior is only used to re-rank candidates within each instance, it mitigates sensitivity to absolute object scale.

Let (x_i, y_i) denote the center coordinates of candidate i and define the Gaussian prior weight as $g_{i,j} = \mathcal{N}_j\left([x_i, y_i]^\top \middle| \mu_j, \Sigma_j\right)$. We incorporate this prior into Eq. (4) and obtain *Gaussian-prior dynamic* k by re-ranking candidates using the product of $g_{i,j}$ and IoU:

$$\{i^*\} = \mathrm{TOP}_q \, g_{i,j} \cdot IoU(p_i^{box}, gt_j). \tag{7}$$

By down-weighting spatially distant candidates, this formulation suppresses distracting regions and reduces the chance of selecting noisy samples, thereby concentrating the assignment around more plausible locations.

Overall, the Gaussian-prior dynamic k provides each instance with an adaptive yet compact set of foreground samples. A larger k suggests that the local neighborhood contains many well-aligned candidates and thus warrants more positives, while a smaller k indicates the opposite. Finally, for gt_j, we select the k candidates with the largest scores $s_{i,j}$ as its suboptimal samples.

3.2 Transforming Optimal Samples

Intuitively, each ground-truth object should correspond to one and only one optimal sample. Therefore, preserving a meaningful ranking among suboptimal samples is crucial: only when their task rankings are properly ordered can the truly optimal sample emerge during training.

In FSS, we explicitly focus on the suboptimal samples of each instance and aim to "excavate" the optimal one from these suboptimal candidates. To this end, the probability associated with each suboptimal sample is used to guide the allocation of its learning weight, which in turn reweights the classification and localization losses to encourage consistent task rankings.

FSS mitigates cross-task inconsistency among suboptimal samples by assigning instance-aware weights and gradually transforming high-quality suboptimal samples into optimal ones. The weighting design follows three principles:

1. **Preserve intra-instance ranking.** For suboptimal samples of the same instance, a larger score s should lead to a larger learning weight and predictions closer to the ground truth.
2. **Maintain inter-instance fairness.** Potentially optimal samples across different instances should have comparable weight scales, preventing the detector from overfitting to a few instances.
3. **Respect score gaps.** Within an instance, larger gaps in s should translate into larger gaps in learning weights, so that training concentrates more on clearly better candidates.

Formally, for the j-th ground-truth object, let $\psi(j)$ denote the index set of its assigned suboptimal samples. To normalize the ranking among these samples, we derive their learning weights from the probability scores:

$$w_i = \left(\frac{s_i}{\bar{s}}\right)^{\frac{1}{u(\alpha+\beta)}}, \quad \text{where} \quad \bar{s} = \frac{\sum_{i\in\psi(j)} s_i}{|\psi(j)|} \tag{8}$$

where i is the sample index. However, when α, β take the larger values, the variance of s tends to increase, which can lead to overly sharp or unstable weights. To moderate this effect, we introduce a reweighting factor u as a hyperparameter that smooths the weight differences between low-ranked and high-ranked suboptimal samples within the same group. By dividing the exponent by $u(\alpha+\beta)$, we effectively constrain the variance of w_i, allowing the detector to focus more on high-quality suboptimal samples while avoiding excessive emphasis on extremely hard cases.

Eq. (8) can be viewed as a mean-normalized power transform of the optimality score. It satisfies the desired properties: (i) *monotonicity and intra-instance ranking preservation*: for any fixed instance j, w_i is strictly increasing in s_i (when $s_i > 0$), and for any two suboptimal samples $a, b \in \psi(j)$ we have $\frac{w_a}{w_b} = \left(\frac{s_a}{s_b}\right)^{\frac{1}{u(\alpha+\beta)}}$, so the ranking induced by s is preserved; (ii) *inter-instance fairness*: dividing by $\bar{s}$ makes the weights invariant to a global rescaling of scores within the same instance (e.g., due to instance difficulty), keeping weight magnitudes comparable across instances; (iii) *controlled sharpness*: taking logarithms yields $\log w_i = \frac{1}{u(\alpha+\beta)}(\log s_i - \log \bar{s})$, showing that u acts as a temperature that smooths weight differences and prevents overly peaked weights when the score distribution becomes sharp (e.g., for larger α, β).

Finally, we apply the learned weights to both classification and localization losses to promote the emergence of optimal samples. The overall FSS loss is defined as:

$$\begin{aligned}
cls &= \frac{1}{N_{pos}} \sum_{i=1}^{N_{pos}} w_i \cdot \mathcal{L}_{cls}(p_i^{cls}, IoU_i) \\
obj &= \frac{1}{N_{pos}} \sum_{i=1}^{N_{pos}} w_i \cdot \mathcal{L}_{cls}(p_i^{obj}, 1) + \frac{1}{N_{neg}} \sum_{q=1}^{N_{neg}} \mathcal{L}_{cls}(p_q^{obj}, 0) \\
box &= \frac{1}{N_{pos}} \sum_{i=1}^{N_{pos}} w_i \cdot \mathcal{L}_{reg}(p_i^{box}, GT_i^{box})
\end{aligned} \tag{9}$$

where i indexes positive samples and q indexes negative samples. $\mathcal{L}_{cls}$ denotes the binary cross entropy (BCE) loss for classification, while $\mathcal{L}_{reg}$ denotes the bounding-box regression loss. In our implementation, $\mathcal{L}_{reg}$ is instantiated as an IoU-based regression loss (e.g., α-CIoU loss [22]) computed between the decoded predicted box and the assigned ground truth; for brevity, the decoding operation is omitted in the notation. We use the *foreground IoU* as a soft target for classification, thereby integrating localization quality into the classification task under the unified weighting scheme of FSS. Concretely, for a positive sample i assigned to ground truth gt_j with class c_j, we decode its predicted box $\hat{B}_i$ from p_i^{box} and define $IoU_i \triangleq IoU(\hat{B}_i, gt_j)$.

4 Experimental Validation and Result Discussions

This section reports the confirmatory and comparative experimental results on the MS-COCO and DOTA datasets.

4.1 Dataset Description

(1) MS-COCO dataset

The MS-COCO benchmark [23] contains approximately 118k training images, 5k validation images, and 20k test-dev images. Following standard practice, we adopt the *trainval135k*/*minival* split: models are trained on *trainval135k* (the union of the train set and a 35k subset of the val set, totaling 135k images), and validated on the remaining 5k images (*minival*). Final results are reported on *test-dev* by submitting predictions to the official MS-COCO evaluation server, ensuring fair comparisons with state-of-the-art (SOTA) detectors.

(2) DOTA dataset (a large-scale dataset of object detection in aerial images)

The DOTA dataset [24] is an open-source benchmark for object detection in remote-sensing imagery. Unlike natural-image datasets, objects in aerial images appear with arbitrary orientations due to the overhead viewing geometry. DOTA-v1.5 extends DOTA-v1.0 by expanding the label space from 10 to 16 categories. It contains over 2800 images collected from diverse platforms and online sources, each with a resolution of approximately 4000 × 4000 pixels. Using oriented bounding boxes to capture objects with varying orientations, scales, and shapes, DOTA provides annotations for 16 categories with 188,282 instances.

Given the extremely high resolution of DOTA images, we adopt a tiling-based preprocessing strategy inspired by YOLT (You Only Look Twice) [25], which splits each image into overlapping tiles and merges tile-level predictions with non-maximum suppression (NMS) to remove duplicates. While tiling is effective for small, densely packed objects, performing it online during inference substantially increases runtime because each high-resolution image must be processed into many patches. To avoid additional inference cost, we apply tiling offline as a data-augmentation procedure, thereby improving sensitivity to small/occluded objects without sacrificing inference efficiency. After extensive experiments, we crop each image into 1024 × 1024 patches with a stride of 800 pixels. This setting largely preserves boundary cues for large objects while increasing the effective number of small-object instances. Statistics on the tiled set indicate that small objects dominate across categories and that the dataset exhibits pronounced imbalance both between large and small objects and among categories, making it a challenging testbed for object detection. After tiling, the training and validation sets contain 11,046 and 3615 patches, respectively.

4.2 Implementation Details and Evaluation Criteria

The experiments are trained on Ubuntu 18.04.3 with an NVIDIA Tesla T4 GPU (16 GB) and an Intel(R) Xeon(R) Silver 4110 CPU @ 2.10 GHz (16 cores). We use SGD with an initial learning rate of 0.01, momentum of 0.937, a 3-epoch learning-rate warmup, and a weight decay of 0.0005, and we further adopt an exponential moving average (EMA) of model parameters. The test environment is Microsoft Windows 10 (19043.1348)

with an NVIDIA GeForce RTX 3060 Laptop GPU (80 W) and an 11th Gen Intel(R) Core(TM) i7-11800H CPU @ 2.30 GHz (8 cores).

We evaluate both accuracy and efficiency. For accuracy, we report the standard metrics, with Average Precision (AP) as the primary measure. The model resolution input is 1024 × 1024 pixels for DOTA dataset images and 640 × 640 for MS-COCO dataset images, with the remaining regions padded with zero if necessary. For the COCO dataset, all *test-dev* accuracy results are obtained by submitting predictions to the official COCO evaluation server. For efficiency, we report inference throughput (FPS) with a batch size of 1, and inference latency to reflect resource-constrained deployment.

We use stochastic gradient descent (SGD) with a momentum of 0.937 and a weight decay of 5e−4. The base learning rate is set to 0.01 with a cosine annealing schedule, preceded by 3 warmup epochs. Unless otherwise specified, models are trained from scratch for 200 epochs on 4 GPUs with a mini-batch size of 16 per GPU, using a strong data augmentation pipeline (without pre-training). The hyperparameter u in the weighting function is set to 2 by default.

Since FSS is a detector-agnostic label assignment strategy, it can be seamlessly integrated into a wide range of modern detectors. In our experiments, we instantiate two representative architectures based on commonly used backbones: ResNet-101 [26] and CSPDarkNet [27]. In line with mainstream object detection frameworks, both architectures comprise a backbone, a neck, and a detection head. For the CSPDarkNet backbone, we adopt a PANet-style neck and a decoupled detection head; for the ResNet-101 backbone, we use an FPN neck coupled with the same decoupled head. The primary difference between the two instantiations thus lies in the choice of backbone and neck, while the head and the proposed FSS label assignment remain identical.

Although FSS introduces no additional inference overhead, it adds extra computation during training due to the Gaussian-prior dynamic k selection and probability-to-weight conversion. To quantify this cost, we report the training-time overhead and memory footprint of the training stage in Table 1. All measurements were conducted under identical hardware and training settings. As shown in Table 1, integrating FSS incurs only a minor training-time overhead on both detector instantiations. Specifically, the per-iteration latency increases by ~2.1%–3.1% and the per-epoch time increases by ~2.0%–2.3%, while the peak GPU memory rises by less than 5% (~3.8%–4.2%). These results indicate that the additional computations introduced by the Gaussian-prior dynamic k and the probability-to-weight conversion are lightweight, making FSS a practical drop-in label assignment strategy with negligible impact on training efficiency and no inference-time cost.

Table 1: Training-time overhead of FSS.

Detector Setting		Iter Time (ms/iter)	Epoch Time (min/epoch)	Peak Memory (GB)
CSPDarkNet-based	w/o FSS	~191	~24.5	~4.8
	with FSS	~197	~25.0	~5.0
Overhead (%)		~+3.1% ↑	~+2.0% ↑	~+4.2% ↑
CSPDarkNet-based	w/o FSS	~240	~31.0	~5.2
	with FSS	~245	~31.7	~5.4
Overhead (%)		~+2.1% ↑	~+2.3% ↑	~+3.8% ↑

4.3 Ablation Studies

All ablation experiments are conducted on the COCO *minival* set. Unless otherwise specified, we use CSPDarkNet [27] as the backbone, train for 42 epochs under the same settings as in Section 4.2, and keep all other implementation details identical to ensure fair comparisons.

4.3.1 Effectiveness of FSS for Label Assignment

To verify that FSS selects higher-quality suboptimal samples and progressively promotes them toward the optimal ones, we compare it with five representative label assignment strategies under the same baseline. Results are reported in Table 2.

Table 2: Comparison between FSS and other label assignment methods on COCO *minival*. All methods use CSPDarkNet as a backbone and are trained under the same settings. We report mean ± std over 10 runs with different random seeds.

Method	Max-IoU	SimOTA [18]	ATSS [13]	TOOD [19]	GFL [20]	FSS
AP (%)	43.74 ± 0.44	44.35 ± 1.47	42.43 ± 0.48	44.73 ± 1.08	42.18 ± 0.83	46.48 ± 0.34
AP_{50} (%)	63.81 ± 0.73	65.00 ± 1.23	62.98 ± 0.85	64.61 ± 1.33	63.1 ± 0.94	66.2 ± 0.63

All results in Table 2 are averaged over 10 independent runs with different random seeds and reported as mean ± standard deviation. As shown in Table 2, averaged over 10 runs, FSS achieves the best performance on COCO *minival* under the same CSPDarkNet backbone and training settings, reaching 46.48 ± 0.34 AP and 66.20 ± 0.63 AP_{50}. Compared with the static Max-IoU baseline (43.74 ± 0.44 AP and 63.81 ± 0.73 AP_{50}), FSS improves the mean AP by 2.74 and AP_{50} by 2.39. It also consistently surpasses strong dynamic assignment methods, exceeding SimOTA by 2.13 AP and 1.20 AP_{50}, TOOD by 1.75 AP and 1.59 AP_{50}, and ATSS by 4.05 AP and 3.22 AP_{50}. In addition, FSS outperforms the quality-aware GFL baseline by 4.30 AP and 3.10 AP_{50}. Notably, FSS shows relatively low variance across runs (0.34 AP std), indicating stable improvements. Since FSS only modifies the training-time assignment procedure, it introduces no additional inference overhead, making it a plug-and-play enhancement for dense detectors.

4.3.2 Contribution Analysis of Suboptimal Selection and Transformation

It is worth noting that FSS consists of two key stages: suboptimal sample selection (*SubOptSel*) and optimal sample transformation (*SubOptTrans*). To quantify the contribution of these two stages, we adopt Max-IoU as the baseline label assignment strategy and progressively introduce suboptimal selection and optimal transformation. As reported in Table 3, replacing Max-IoU with suboptimal selection improves AP by 2.0. Adding optimal transformation on top of suboptimal selection yields an additional 0.3 AP gain. Notably, these improvements are achieved purely through training-time assignment and reweighting, introducing no extra inference overhead; hence, they can be regarded as "free" performance gains for the existing detector.

4.3.3 Effectiveness of Hyperparameters α, β

The hyperparameters α and β control the relative contributions of classification confidence and localization quality in the probability score that measures how likely a candidate is to be an optimal sample (see Eq. (3)). Intuitively, a larger β increases the emphasis on localization quality, while a larger α strengthens the impact of classification confidence.

Table 3: Ablation on the contributions of SubOptSel and SubOptTrans over the Max-IoU baseline.

Method	AP	AP_{50}
Baseline:MAX-IoU	43.9%	63.7%
Baseline+SubOptSel	45.9%	65.8%
Baseline+SubOptSel+SubOptTrans	46.2%	66.2%

To evaluate the sensitivity of FSS to these hyperparameters, we vary α and β within a reasonable range and report the corresponding performance in Table 4. As shown in Table 4, the detection performance is stable across a wide range of configurations: AP varies only from 45.9% to 46.2%, and AP_{50} from 65.9% to 66.3%. This indicates that FSS is insensitive to the precise choice of α and β and does not require careful hyperparameter tuning to achieve strong results. Unless otherwise specified, we use $\alpha = 0.5$ and $\beta = 6$ as the default settings, which provides a balanced contribution of classification confidence and localization quality while maintaining consistently high accuracy.

Table 4: Verify the impact of different values of α, β on COCO *minival*.

α	β	AP	AP_{50}
1	10	46.1%	66.0%
1	8	45.9%	65.9%
0.5	8	46.2%	66.2%
0.5	6	46.2%	66.1%
0.5	4	46.2%	66.3%

4.4 Comparison with State-of-the-Art Methods

The comparative quantitative results of FSS with representative label assignment strategies on detectors on the MS-COCO *test-dev* dataset are summarized in Table 5. The compared strategies include ATSS [13], which calculates adaptive IoU thresholds based on statistical properties of object fits; GFL [20], which optimizes a generalized focal loss to jointly model classification and localization quality; PAA [15], which separates positive and negative anchors using a probabilistic Gaussian mixture model; TOOD [19], which aligns tasks via an explicit task-aligned head and learning mechanism; and DW [17], which introduces dual weighting to dynamically refine label importance. FSS, in contrast, focuses on progressively promoting high-quality suboptimal samples through unified probability scoring.

Table 5: Performance comparison of different label assignment strategies on representative detectors on MS-COCO *test-dev*. For fair comparisons, performance should be compared with the same backbone family (as indicated in the *Backbone* column). FPS is measured on the same hardware with a batch size of 1 under FP32 precision, using the default inference configuration for each method. DDH: Decoupled detection head.

Method	Backbone	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv3	DarkNet-53	37	33.0	57.9	34.4	18.3	35.4	41.9
YOLOv3+ATSS	DarkNet-53	37	34.8	59.0	36.0	19.5	36.2	43.2
YOLOv3+GFL	DarkNet-53	37	34.5	60.5	36.3	20.0	37.0	43.4
YOLOv3+PAA	DarkNet-53	37	36.0	62.0	36.4	18.3	35.4	41.9
YOLOv3+TOOD	DarkNet-53	35	36.0	62.0	36.5	18.3	35.4	44.2

(Continued)

Table 5 (continued)

Method	Backbone	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv3+DW	DarkNet-53	37	34.9	59.4	36.1	19.2	37.2	43.3
YOLOv3+FSS	DarkNet-53	37	36.3	63.0	38.0	20.5	37.5	44.6
Faster R-CNN	Resnet-50	12	40.2	61.0	43.8	24.2	43.5	52.0
Faster R-CNN +ATSS	Resnet-50	12	42.5	62.5	47.0	26.2	44.5	54.4
Faster R-CNN+GFL	Resnet-50	12	44.2	63.0	48.2	27.2	45.8	55.2
Faster R-CNN+PAA	Resnet-50	12	44.0	63.8	48.2	27.2	47.2	55.6
Faster R-CNN+TOOD	Resnet-50	10.8	43.9	62.8	49.2	27.8	46.5	56.2
Faster R-CNN+DW	Resnet-50	12	41.8	62.4	45.8	24.2	44.9	53.2
Faster R-CNN+FSS	Resnet-50	12	44.5	63.2	49.3	28.4	48.0	55.9
RetinaNet	ResNet-101	8.9	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet+ATSS	ResNet-101	8.9	42.4	60.1	45.9	22.6	44.2	52.8
RetinaNet+GFL	ResNet-101	8.9	41.2	61.5	45.0	23.4	44.0	51.6
RetinaNet+PAA	ResNet-101	8.9	42.0	62.2	45.2	22.9	42.7	52.5
RetinaNet+TOOD	ResNet-101	8.0	42.8	60.8	45.6	23.8	42.7	54.3
RetinaNet+DW	ResNet-101	8.9	42.9	62.3	45.8	23.6	42.7	52.4
RetinaNet+FSS	ResNet-101	8.9	44.1	63.4	46.6	24.6	45.0	55.4
CSPDarkNet+PANet +DDH	CSPDarkNet	37.1	44.8	63.5	48.4	26.5	49.3	57.5
CSPDarkNet+PANet +DDH+ATSS	CSPDarkNet	37.1	50.8	65.6	50.2	28.5	52.2	59.8
CSPDarkNet+PANet +DDH+GFL	CSPDarkNet	37.1	50.8	67.4	52.8	29.6	51.6	60.5
CSPDarkNet+PANet +DDH+PAA	CSPDarkNet	37.1	50.8	66.9	53.2	28.7	53.4	61.5
CSPDarkNet+PANet +DDH+TOOD	CSPDarkNet	33.4	50.8	67.8	54.2	29.8	52.6	62.3
CSPDarkNet+PANet +DDH+DW	CSPDarkNet	37.1	49.6	66.8	51.2	28.6	50.8	60.0
CSPDarkNet+PANet +DDH+FSS	CSPDarkNet	37.1	50.8	69.1	55.3	31.9	54.8	63.8
ResNet-101+FPN+DDH	ResNet-101	37.6	44.5	62.0	48.0	24.6	46.8	57.4
ResNet-101+FPN+DDH+ATSS	ResNet-101	37.6	48.2	64.9	51.2	25.6	49.3	59.4
ResNet-101+FPN+DDH+GFL	ResNet-101	37.6	48.6	65.5	49.8	27.2	50.2	61.5
ResNet-101+FPN+DDH+PAA	ResNet-101	37.6	47.3	64.8	50.4	26.8	51.8	60.4
ResNet-101+FPN+DDH+TOOD	ResNet-101	34.4	49.0	65.9	52.4	28.0	51.5	62.8

(Continued)

Table 5 (continued)

Method	Backbone	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-101+FPN+DDH+DW	ResNet-101	37.6	47.4	64.3	50.3	27.6	50.2	60.4
ResNet-101+FPN+DDH+FSS	ResNet-101	37.6	49.0	67.8	53.5	29.7	53.4	62.0

Since detection accuracy is strongly influenced by model capacity (e.g., the backbone/neck), we primarily draw conclusions from backbone-matched comparisons and report results under multiple backbone settings to demonstrate generality. As shown in Table 5, FSS consistently improves performance across diverse detector architectures (YOLOv3, Faster R-CNN, RetinaNet, and modern Decoupled detection head (DDH)-based models) while maintaining high inference efficiency.

Improvements on Standard Detectors. On the classic YOLOv3 (DarkNet-53), FSS boosts the baseline from 33.0 to 36.3 AP (+3.3 AP), outperforming advanced assignment methods such as TOOD [19] (36.0 AP) and PAA [15] (36.0 AP) without the speed drop observed in TOOD (35 vs. 37 FPS). Similarly, on the two-stage Faster R-CNN (ResNet-50), FSS achieves the highest AP of 44.5, surpassing GFL (44.2) and TOOD (43.9). Notably, on the anchor-based RetinaNet (ResNet-101), FSS delivers a substantial gain of +5.0 AP (39.1 → 44.1 AP), significantly outperforming DW (42.9 AP) and ATSS (42.4 AP).

Validation on Modern Decoupled Heads. To verify effectiveness on stronger baselines, we apply FSS to detectors with Decoupled Detection Heads (DDH). With a ResNet-101 backbone, FSS achieves 49.0 AP, matching the top-performing TOOD but with a distinct speed advantage (37.6 FPS vs. 34.4 FPS). More importantly, FSS yields superior localization quality, achieving higher AP_{75} (53.5 vs. 52.4 for TOOD) and AP_{50} (67.8 vs. 65.9). When instantiated with a CSPDarkNet backbone (YOLOX-style), FSS reaches 50.8 AP, equaling the best competitor, but again demonstrates a superior accuracy-speed trade-off (37.1 FPS) compared to TOOD (33.4 FPS) and higher AP_{50}/AP_{75} metrics.

These results confirm that (i) the proposed suboptimal-sample–focused label assignment can be effectively integrated into different detector architectures (ResNet-101+FPN and CSPDarkNet+PANet), and (ii) FSS provides a competitive or superior accuracy–speed trade-off compared with strong SOTA detectors, without introducing any additional inference overhead.

4.5 Generalization on Remote Sensing Benchmark

To further evaluate the generalization of FSS in scenarios dominated by tiny, densely packed, and arbitrarily oriented objects, we conduct additional experiments on the DOTA dataset. DOTA is a large-scale aerial-image benchmark featuring extreme aspect ratios, cluttered backgrounds, and dense object layouts, where "perfectly aligned" optimal samples are often scarce. This makes DOTA a particularly suitable testbed for assessing whether our strategy can reliably mine high-quality suboptimal positives and improve label assignment under challenging conditions.

The quantitative results are summarized in Table 6. Since the table contains multiple detectors and assignment variants, we highlight the main takeaway here: *FSS consistently improves AP and AP_{50} across diverse detector families while keeping model complexity essentially unchanged*. In particular, FSS yields +4.2 AP/+3.0 AP_{50} on YOLOX-L (47.4 → 51.6 AP; 71.3 → 74.3 AP_{50}) at the same Params/FLOPs/FPS, +6.2 AP/+4.0 AP_{50} on Mobile-Former (46.5/70.3 → 52.7/74.3), and +6.0 AP/+3.4 AP_{50} on SSD512 (38.6/62.2 → 44.6/65.6). These results indicate that focusing on high-quality suboptimal positives is particularly

beneficial for remote-sensing scenes with dense layouts and large scale variations, and they provide direct evidence of the *detector-agnostic* property of FSS. We observe that DOTA images contain many densely packed, tiny instances, frequent partial occlusions, and complex backgrounds. Under these conditions, many candidates around an object exhibit inconsistent classification confidence and localization quality, making "clean" positive assignments difficult and causing noisy gradients, especially for lightweight/compact detectors (e.g., Mobile-Former) or older architectures with weaker feature representations (e.g., SSD512). FSS explicitly mines high-quality suboptimal positives and forms a spatially coherent positive set (Gaussian-prior dynamic-k) around the potentially optimal candidate, thereby reducing ambiguity in crowded regions. Moreover, the probability-derived weighting encourages consistent ranking between classification and localization, providing more reliable supervision for small and adjacent objects; this effect is amplified when the baseline model capacity is limited, leading to the larger relative gains observed on Mobile-Former and SSD512.

Table 6: Performance comparison of different label assignment strategies on representative detectors on DOTA dataset.

Method	Parama(M)	GFLOPs	FPS	AP(%)	AP_{50}(%)
YOLOXL(with SimOTA) [28]	54.16	155.6	13	47.4	71.3
ATSS+YOLOXL	54.16 (≈)	155.6 (≈)	13 (≈)	49.5 (↑ 2.1)	73.1 (↑1.8)
Max-IoU+YOLOXL	54.16 (≈)	155.6 (≈)	13 (≈)	49.1 (↑ 1.7)	72.3 (↑ 1.0)
TOOD+YOLOXL	55.20 (↑)	159.6 (↑)	12 (↓)	50.1 (↑ 2.7)	73.8 (↑ 2.5)
FSS+YOLOXL	54.16 (≈)	155.6 (≈)	13 (≈)	51.6 (↑4.2)	74.3 (↑3.0)
Mobile-Former [29]	61.00	161.0	34	46.5	70.3
ATSS+Mobile-Former	61.20 (≈)	162.0 (≈)	33 (≈)	50.3 (↑ 3.8)	72.9 (↑ 2.6)
Max-IoU+Mobile-Former	61.20 (≈)	161.0 (≈)	34 (≈)	48.9 (↑ 2.4)	71.9 (↑ 1.6)
SimOTA+Mobile-Former	61.00 (≈)	161.0 (≈)	34 (≈)	51.3 (↑ 4.8)	73.2 (↑ 2.9)
TOOD+Mobile-Former	65.20 (↑)	169.4 (↑)	30 (↓)	51.5 (↑ 5.0)	75.3 (↑ 3.2)
FSS+Mobile-Former	61.00 (≈)	161.0 (≈)	34 (≈)	52.7 (↑6.2)	74.3 (↑4.0)
SSD512 [6]	257.5	181.7	53	38.6	62.2
ATSS+SSD512	257.5 (≈)	181.7 (≈)	53 (≈)	40.6 (↑ 2.0)	63.3 (↑ 1.1)
Max-IoU+SSD512	257.5 (≈)	181.7 (≈)	53 (≈)	40.2 (↑ 1.6)	63.3 (↑ 1.1)
SimOTA+SSD512	257.5 (≈)	181.7 (≈)	53 (≈)	41.1 (↑ 2.5)	63.8 (↑ 1.6)
TOOD+SSD512	262.0 (↑)	188.8 (↑)	47 (↓)	42.2 (↑ 3.6)	64.5 (↑ 2.3)
FSS+SSD512	257.5 (≈)	181.7 (≈)	53 (≈)	44.6 (↑6.0)	65.6 (↑3.4)
SADet [30]	4.87	11.3	95	42.8	67.6
ATSS+SADet	4.87 (≈)	11.3 (≈)	95 (≈)	46.0 (↑3.2)	69.6 (↑2.0)
Max-IoU+SADet	4.87 (≈)	11.3 (≈)	95 (≈)	45.6 (↑ 2.8)	69.0 (↑ 1.4)
SimOTA+SADet	4.87 (≈)	11.3 (≈)	95 (≈)	44.8 (↑ 2.0)	68.7 (↑ 1.1)
TOOD+SADet	5.05 (↑)	12.0 (↑)	90 (↓)	47.0 (↑ 4.2)	70.5 (↑ 2.9)
FSS+SADet	4.87 (≈)	11.3 (≈)	95 (≈)	48.2 (↑5.4)	71.8 (↑4.2)

Furthermore, by integrating our proposed FSS strategy (denoted as SADet+FSS), the detection performance is boosted to 48.2 AP and 71.8 AP_{50}, representing a consistent improvement of 5.4 AP and 4.2 AP_{50}.

Furthermore, the high inference speed of 95 FPS remains. This indicates that mining high-quality suboptimal samples is particularly effective for small and crowded objects.

To intuitively demonstrate this capability, Fig. 5 presents the qualitative visualization of FSS on the DOTA dataset compared with ground-truth annotations. As observed, even in extreme scenarios with densely packed vehicles and varying-scale ships, FSS maintains exceptional localization precision. The predicted bounding boxes closely align with the ground truth, effectively distinguishing adjacent tiny instances without introducing significant false positives. The successful application on DOTA confirms that FSS is not limited to natural scenes but generalizes well to complex remote sensing tasks, further validating its cross-domain robustness.

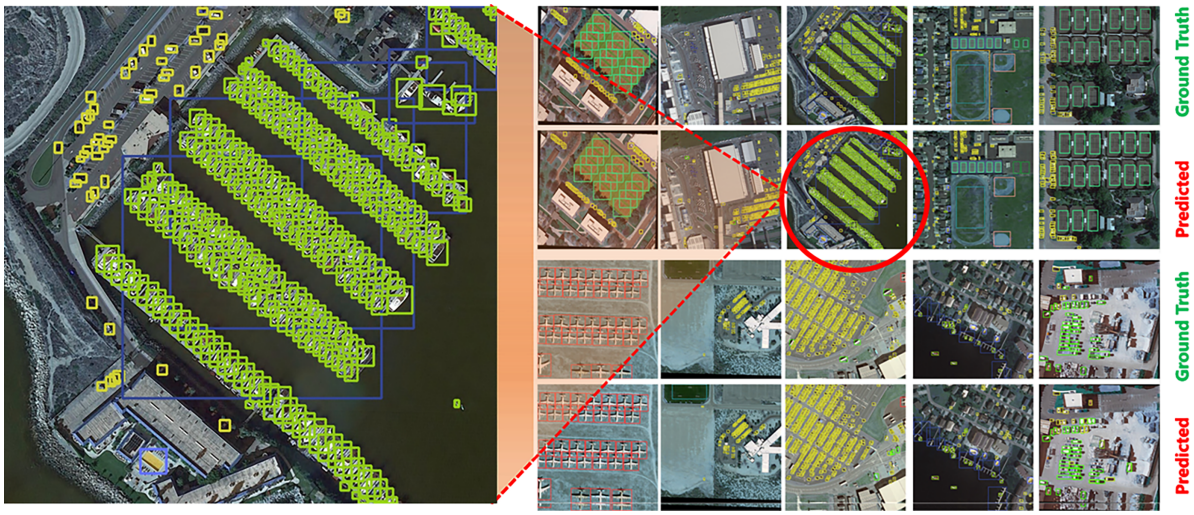

Figure 5: Visualization of detection results on the DOTA dataset. FSS achieves high localization accuracy on densely packed objects, such as vehicles and ships, generating bounding boxes that closely match ground truths.

4.6 Discussions

Conceptual comparison with related methods. Although several recent methods also couple classification confidence with localization quality, their primary focus differs from ours. ATSS [13] adaptively sets an IoU threshold by exploiting the statistics of candidate IoUs, whereas FSS explicitly models the learning value of suboptimal candidates via a unified probability score and then uses a Gaussian-prior-guided dynamic k to form instance-adaptive positives around the potentially optimal sample. PAA [15] assigns labels by fitting a mixture model to separate positives and negatives based on training signals, while FSS treats label assignment as a progressive promotion process: it first selects high-quality suboptimal candidates and then reweights them by optimality probability to preserve ranking consistency across classification and localization. SimOTA/OTA [18] performs dynamic matching by minimizing a cost composed of classification and regression losses, whereas FSS directly models candidate optimality with a probability score and uses this score consistently for both selection and transformation. AutoAssign [16] generates soft labels with center-based priors and learned confidence weighting, whereas FSS centers the selection on the *potentially optimal* candidate and enforces ranking consistency between classification and localization within each instance through probability-derived weights. GFL [20] (and its variants) improves localization-quality estimation via distributional modeling and quality-aware classification, whereas FSS is orthogonal to head design and can be integrated as a training-time assignment/reweighting strategy to reduce the mismatch between classification confidence and localization quality. TOOD [19] improves task alignment through a dedicated

head and learning objective, while FSS addresses alignment from the perspective of supervision generation: by enforcing consistent ranking through instance-wise weights, it encourages the emergence of truly optimal samples from informative suboptimal ones.

Limitations and applicability. FSS leverages the correlation between classification confidence and localization quality to construct the probability score. In the early epochs, both classification confidence and IoU can be poorly calibrated, which may lead to unstable score rankings and fluctuating selected samples/weights. To mitigate this early-stage instability, the score is used only for sample selection/weighting via stop-gradient, and it is further stabilized by instance-wise normalization and smoothing, thereby bounding its impact on optimization. In extremely sparse scenes, the pool of informative suboptimal candidates may be limited, and in extremely dense/crowded scenes, multiple nearby instances can produce highly ambiguous candidates, potentially increasing assignment uncertainty. In practice, we find FSS is stable across a wide range of α and β (Section 4.2). We compute Eq. (3) with stop-gradient (detach) so that gradients do not propagate through the IoU term during score computation, avoiding a noisy feedback loop between localization quality and the assignment weights. We further improve stability in early training via standard learning-rate warmup and EMA, together with instance-wise normalization/smoothing in the weighting function.

Performance-efficiency trade-off and deployment feasibility. FSS modifies only the *training-time* label assignment and reweighting procedure, and it does not introduce any additional layers or computations in the inference graph. Therefore, the deployed detector preserves the same model complexity (Params/FLOPs) as the corresponding baseline, and the inference throughput/latency reported in our experiments reflects practical deployment behavior under fixed hardware and precision settings (e.g., batch size 1 and FP32 as stated in the table captions). From the perspective of computational and energy constraints, this means that inference-time compute and the associated energy cost remain essentially unchanged when adopting FSS. The only extra cost occurs during training; as quantified in Table 1, the added overhead is minor, indicating that FSS is feasible for practical training and deployment pipelines.

Complexity and scaling. The dominant additional computation in FSS comes from constructing the instance-wise score/matching statistics between candidate locations and ground-truth instances. Let N denote the number of candidates (which grows with input resolution and feature-map density) and M denote the number of ground-truth objects; the associated matrix-style operations scale roughly with $N \times M$, similar in order to many dynamic assignment methods that build per-instance matching costs. In practice, these computations are fully vectorized on the GPU and account for only a small fraction of training time in our settings (Table 1). For extremely high-resolution inputs or extreme object counts, the cost increases proportionally with N and M, and it can be controlled in implementation by restricting candidates using spatial priors and/or computing the score matrix in chunks, without affecting inference-time efficiency.

5 Conclusions

In this article, we proposed FSS, an adaptive and detector-agnostic label assignment scheme that explicitly focuses on suboptimal samples. For each instance, FSS identifies high-quality suboptimal candidates using a unified probability score that couples classification confidence with localization quality. It then combines IoU with a Gaussian prior centered at the potentially optimal sample to adaptively determine the number of positives for each ground truth. The resulting optimality probability is further mapped to instance-wise weights applied to both classification and localization heads, preserving the ranking structure and progressively promoting truly optimal samples during training. Extensive experiments on MS-COCO and DOTA datasets demonstrate that FSS is effective and generalizes well across detectors, achieving a competitive accuracy-speed trade-off with no additional inference overhead. In future work, we will integrate

FSS into fully end-to-end detection pipelines by eliminating hand-crafted NMS and explore its extension to broader detection and instance-level recognition tasks.

Acknowledgement: Not applicable.

Funding Statement: This research was funded by the National Natural Science Foundation of China under Grant No. 62371187 and the Open Program of Hunan Intelligent Rehabilitation Robot and Auxiliary Equipment Engineering Technology Research Center under Grant No. 2024JS101.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Jinping Liu and Kunyi Zheng; methodology, Lijuan Huang, Kunyi Zheng, Xinyu Zhou, Jinping Liu and Yimei Yang; software, Yimei Yang and Zhixian Liu; validation, Lijuan Huang and Zhixian Liu; investigation, Lijuan Huang, Zhixian Liu and Jinping Liu; resources, Zhixian Liu; writing—original draft preparation, Jinping Liu and Yimei Yang; writing—review and editing, Yimei Yang and Jinping Liu; visualization, Lijuan Huang, Xinyu Zhou, Zhixian Liu and Yimei Yang; supervision, Jinping Liu; funding acquisition, Jinping Liu. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The public accessed datasets-MS-COCO and DOTA are used in this study.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ, USA: IEEE; 2017. p. 936–44. doi:10.1109/CVPR.2017.106.
2. Lin TY, Goyal P, Girshick R, He K, Dollar P. Focal loss for dense object detection. IEEE Trans Pattern Anal Mach Intell. 2020;42(2):318–27. doi:10.1109/TPAMI.2018.2858826.
3. Tian Z, Shen C, Chen H, He T. FCOS: fully convolutional one-stage object detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway, NJ, USA: IEEE; 2019. p. 9626–35. doi:10.1109/ICCV.2019.00972.
4. Dai H, Gao S, Huang H, Mao D, Zhang C, Zhou Y. An adaptive sample assignment network for tiny object detection. IEEE Trans Multimed. 2024;26:2918–31. doi:10.1109/TMM.2023.3305120.
5. Khanam R, Hussain M. YOLOv11: an overview of the key architectural enhancements. arXiv:2410.17725. 2024.
6. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot MultiBox detector. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2016. p. 21–37. doi:10.1007/978-3-319-46448-0_2.
7. Zhao Z, Du J, Li C, Fang X, Xiao Y, Tang J. Dense tiny object detection: a scene context guided approach and a unified benchmark. IEEE Trans Geosci Remote Sens. 2024;62:5606913. doi:10.1109/TGRS.2024.3357706.
8. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y. OverFeat: integrated recognition, localization and detection using convolutional networks. arXiv:1312.6229. 2013.
9. Law H, Deng J. CornerNet: detecting objects as paired keypoints. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2018. p. 734–50.
10. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S, et al. End-to-end object detection with transformers. In: European Conference on Computer Vision. Cham, Switzerland: Springer International Publishing; 2020.
11. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2017;39:1137–49.
12. Kong T, Sun F, Liu H, Jiang Y, Li L, Shi J. FoveaBox: beyound anchor-based object detection. IEEE Trans Image Process. 2020;29:7389–98. doi:10.1109/TIP.2020.3002345.

13. Zhang S, Chi C, Yao Y, Lei Z, Li SZ. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ, USA: IEEE; 2020. p. 9756–65. doi:10.1109/CVPR42600.2020.00978.
14. Zhang X, Wan F, Liu C, Ji X, Ye Q. Learning to match anchors for visual object detection. IEEE Trans Pattern Anal Mach Intell. 2022;44(6):3096–109. doi:10.1109/TPAMI.2021.3050494.
15. Zhang F, Zhou S, Wang Y, Wang X, Hou Y. Label assignment matters: a Gaussian assignment strategy for tiny object detection. IEEE Trans Geosci Remote Sens. 2024;62:5633112. doi:10.1109/TGRS.2024.3430071.
16. Zhu B, Wang J, Jiang Z, Zong F, Liu S, Li Z, et al. AutoAssign: differentiable label assignment for dense object detection. arXiv:2007.03496. 2020.
17. Li S, He C, Li R, Zhang L. A dual weighting label assignment scheme for object detection. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ, USA: IEEE; 2022. p. 9377–86. doi:10.1109/CVPR52688.2022.00917.
18. Ge Z, Liu S, Li Z, Yoshie O, Sun J. OTA: optimal transport assignment for object detection. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ, USA: IEEE; 2021. p. 303–12. doi:10.1109/CVPR46437.2021.00037.
19. Xu C, Zhang R, Yang W, Zhu H, Xu F, Ding J, et al. Oriented tiny object detection: a dataset, benchmark, and dynamic unbiased learning. IEEE Trans Pattern Anal Mach Intell. 2026;48(3):3167–84. doi:10.1109/TPAMI.2025.3634161.
20. Li X, Wang W, Wu L, Chen S, Hu X, Li J, et al. Generalized focal loss: learning qualified and distributed bounding boxes for dense object detection. Adv Neural Inf Process Syst. 2020;33:21002–12.
21. Li X, Wang W, Hu X, Li J, Tang J, Yang J. Generalized focal loss V2: learning reliable localization quality estimation for dense object detection. arXiv:2011.12885. 2020.
22. He J, Erfani S, Ma X, Bailey J, Chi Y, Hua X. α -IoU: a family of power intersection over union losses for bounding box regression. Adv Neural Inf Process Syst. 2021;34:20230–42.
23. Chen X, Fang H, Lin T, Vedantam R, Gupta S, Dollar P, et al. Microsoft COCO captions: data collection and evaluation server. arXiv:1504.00325. 2015.
24. Ding J, Xue N, Xia GS, Bai X, Yang W, Yang M, et al. Object detection in aerial images: a large-scale benchmark and challenges. IEEE Trans Pattern Anal Mach Intell. 2022;44(11):7778–96. doi:10.1109/tpami.2021.3117983.
25. Li W, Li W, Yang F, Wang P. Multi-scale object detection in satellite imagery based on YOLT. In: IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium. Piscataway, NJ, USA: IEEE; 2019. p. 162–5. doi:10.1109/IGARSS.2019.8898170.
26. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ, USA: IEEE; 2016. p. 770–8. doi:10.1109/CVPR.2016.90.
27. Wang CY, Liao HYM, Wu YH, Chen PY, Hsieh JW, Yeh IH. CSPNet: a new backbone that can enhance learning capability of CNN. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Piscataway, NJ, USA: IEEE; 2020. p. 1571–80. doi:10.1109/CVPRW50498.2020.00203.
28. Ge Z, Liu S, Wang F, Li Z, Sun J. YOLOX: exceeding YOLO series in 2021. arXiv:2107.08430. 2023.
29. Chen Y, Dai X, Chen D, Liu M, Dong X, Yuan L, et al. Mobile-former: bridging mobilenet and transformer. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2022. p. 5270–9.
30. Liu J, Zheng K, Liu X, Xu P, Zhou Y. SDSDet: a real-time object detector for small, dense, multi-scale remote sensing objects. Image Vis Comput. 2024;142:104898.