1 Paper strengths

R1:"Performance boosts on many whole-dataset metrics." "FreeAnchor performs notably better than the baseline..." **R2:**"Overall, the introduced method is interesting and novel..." "This paper could have impact if the overall presentation would be improved." **R3:** "Overall this is a nice paper to resolve the anchor assignment based on the Maximum likelihood...".

We thank reviewers for their valuable comments and support of the novelty and results of our work. Major concerns are mainly about presentation and explanation, which are addressed point-to-point below.

2 Response to review comments

R1: Could the anchor bag be constructed once before the training loop and cached? Yes. Nevertheless, we can call it in each iteration for negligible computational cost and simplicity of implementation.

R1: Figure 4. Yes you are right. As suggested, we improved the contrast and explanation of Figure 4, as below.

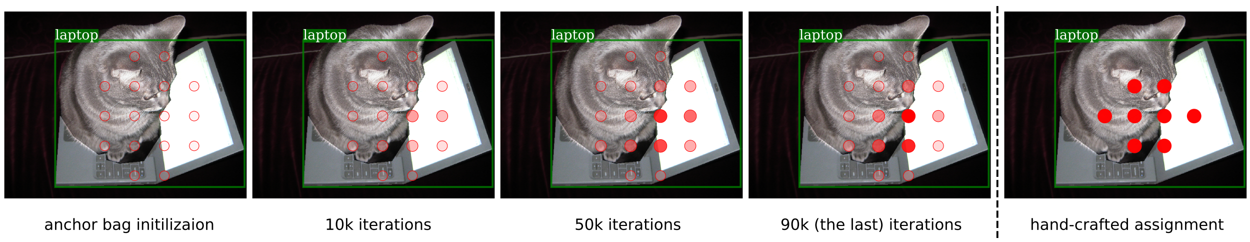

Figure 4: Comparison of learning-to-match anchors (left) with hand-crafted anchor assignment (right) for the "laptop" object. Red dots denote anchor centers. Darker (redder) dots denote higher confidence to be matched. For clarity, we select 16 anchors of aspect-ratio 1:1 from all 40 anchors for illustration. (Best viewed in color)

R1: L172, why this formula is used? & why b is used for a bias initialization According to Focal Loss [12], the formula (b) is the inverse function of Sigmoid. b, as a bias initialization, can alleviate the class imbalance issue.

R1&R3: Minor comments. L16, [5] and [6] were removed; Fig. 1, "up" $\rightarrow$ "top", "down" $\rightarrow$ "bottom"; L145, "be" is removed; Algorithm 1, "super-" $\rightarrow$ "hyper-" and $\theta^{t+1} = \theta^t - \lambda \nabla_{\theta^t} L(\theta^t)$ is used instead; L172, $\pi \rightarrow \rho$.

R2: Third and fourth paragraph of Section 1, more clear motivation requires. As suggested, we added: "It is hard to design a generic rule which can optimally match anchors/features with objects of various geometric layouts. The widely used hand-crafted assignment could fail when facing acentric, slender, and/or crowded objects. A learning-based approach requires to be explored to solve this problem in a systematic way."

R2: Section 2, how the presented approach differs from existing methods (FoveaBox and MetaAnchor)? FoveaBox used hand-crafted boxes to match features while MetaAnchor dynamically generated anchors from prior boxes. They remain using hand-crafted configuration to match objects with anchors/features, and thereby differs from our approach, essentially.

R2: Section 3.1, what for was the IoU criterion used? What is similar/different to existing approaches. & What is the conclusion here (L110-113)? & Is this a limitation? Section 3.1 is not our approach but a revisit of the baseline method which used the IoU criterion. The conclusion is that the matching matrix C_{ij} in the baseline method is hand-crafted, not learned. In Section 3.2, we propose the learning-to-match approach to solve C_{ij}, and break the hand-crafted assignment limitation.

R2: Why will it be close to the Mean when training is insufficient? When training is insufficient, the "Mean" function assigns each anchor equal opportunity to be matched, which prevents the algorithm getting stuck into local minimum in early epochs.

R2: The average performance across all categories & Why "couch" is a slender object? & Limitations The average mAPs of square and slender objects are: 40.5%/40.3% and 22.7%/27.1% (baseline/ours), which show that FreeAnchor significantly improved the mAP on slender objects, while achieving comparable mAP on square objects. The average aspect ratio of couch is larger than 2.0. Limitation: FreeAnchor has no advantage on square objects as the IoU criterion can also find proper anchors for them.

R2: Minor comments. L92, "R^4 denotes {x, y, w, h}"; SmoothL1 was introduced in L103; "bg" indicates "background"; L121, "we require to guarantee..."; L156,"using the parameters..."; Bag construction were described in L118-120.

R3: CenterNet is slightly better for large objects. When using the same ResNet backbone, FreeAnchor outperforms state-of-the-art methods. CenterNet slightly outperforms FreeAnchor for large objects, as it uses a larger backbone (Hourglass) and a smaller input image size: 210.1M vs 96.9M parameters and 511×511 vs 1333×800 pixels (CenterNet vs Ours).

R3: FreeAnchor and Anchor-Free. "FreeAnchor" means that instances freely match anchors, without IoU restriction. To avoid the confusion with "Anchor-free", we revised "FreeAnchor" to "CatchAnchor": each instance can catch a proper anchor after training.

R3: It would be better to see such a plug-and-play module could also help for the two-stage detectors. During rebuttal, we have applied the learning-to-match module to a two-stage detector, FPN, and achieved significant (2.5%) performance gain.