Bandit-Based Sample Reweighting for YOLO Screw Detection

Siyao Yu [*1] **Xintong Li** [*1] **Anyongyong Zhao** [*1]

Abstract

We investigate how to improve YOLO-based screw detection by enhancing the model's ability to learn from challenging samples. From the initial training stage, we extract per-image statistics, including false positives, false negatives, and confidence scores, to quantify the difficulty of each sample. These difficulty estimates are then used within a bandit-inspired sample reweighting framework to assign adaptive weights for a second-stage fine-tuning process. Experiments demonstrate that this reweighting strategy leads to more reliable detection performance compared to standard fine-tuning, without modifying the underlying YOLO architecture.

1. Introduction

Screw detection in cluttered industrial images poses a significant challenge for YOLO models, which often produce numerous false detections when screws are densely packed. A major limitation is that all training samples are treated uniformly, even though difficult images contribute disproportionately to errors.

To address this, we develop an idea to compute per-image difficulty in terms of three metrics, including false positives, false negatives, and confidence, from an initial YOLO model, and use these metrics to guide adaptive sample weighting. We formulate this weighting process using a bandit-inspired sample reweighting approach, which assigns higher weights to more challenging samples during a second fine-tuning stage.

This two-stage pipeline improves detection reliability without modifying the YOLO architecture.

[*]Equal contribution [1]Department of Electrical Engineering, Columbia University, New York, USA. Correspondence to: Siyao Yu <sy3342@columbia.edu>, Xintong Li <xl3601@columbia.edu>, Anyongyong Zhao <az2932@columbia.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

2. Background and Related Work

2.1. Bandit Learning and Sample Reweighting

Multi-armed bandit (MAB) and contextual bandit frameworks have been widely studied as lightweight reinforcement learning paradigms for decision-making under uncertainty, where an agent selects actions based on partial feedback to maximize cumulative reward (Langley, 2000; Lattimore & Szepesvári, 2020). Compared to full Markov Decision Processes (MDPs), bandit methods offer lower complexity and faster convergence, making them suitable for problems where long-horizon dynamics are not explicitly modeled.

In parallel, sample reweighting and hard example mining have been extensively explored in computer vision and object detection. Prior works such as Online Hard Example Mining (OHEM) and focal loss emphasize difficult samples by increasing their contribution to the training objective(Shrivastava et al., 2016; Lin et al., 2017). However, these approaches typically rely on heuristic criteria or fixed loss reshaping, without explicitly framing sample selection as a learning problem. Despite their effectiveness, existing reweighting strategies lack an explicit decision-making formulation. In particular, most object detection pipelines treat the assignment of sample importance as a static design choice rather than a learning-driven process. This limits their ability to adaptively emphasize samples based on evolving error patterns and risk profiles.

2.2. Novel Contribution

In this work, we propose a bandit-based sample reweighting framework for YOLO training. Unlike prior heuristic reweighting methods, we explicitly formulate sample weighting as a contextual bandit problem, where image-level detection statistics serve as context and training weights correspond to discrete actions.

Our approach bridges reinforcement learning and object detection by introducing a lightweight RL formulation that does not require modifying the YOLO architecture or training objective. Instead, the bandit policy influences the effective data distribution through importance sampling, allowing the model to focus on high-regret samples during fine-tuning.

3. Approach

3.1. Stage 1: Baseline Training and State Extraction

A baseline YOLO model is first trained on the screw dataset using standard configurations, producing the baseline model that serves as the foundation for subsequent analysis.

Using this baseline model, per-image inference is performed across the entire training set, generating detailed detection statistics recorded in `state.csv`. These statistics quantify model behavior in terms of ground-truth counts, predictions, true/false detections, and confidence scores, and serve as the input to the difficulty estimation module.

3.2. Stage 2: Contextual Bandit Formulation for Sample Reweighting

We formulate the sample reweighting problem as an offline contextual bandit task. Each training image x_i is treated as an independent decision instance, where the learner observes a context s_i, selects an action a_i, and aims to minimize the expected detection regret associated with that sample.

The context s_i is defined by the image-level error profile extracted from the baseline YOLO model, including the number of false positives, false negatives, and the average prediction confidence. The action space $\mathcal{A}$ consists of a finite set of discrete training weights, corresponding to different levels of importance assigned to each sample during fine-tuning.

Unlike online reinforcement learning, the reward signal is not obtained through environment interaction. Instead, we approximate the *negative reward* using offline error statistics derived from the baseline detector, allowing the bandit policy to be learned in a single-pass, non-interactive manner.

3.2.1. Difficulty Estimation (Negative Reward Modeling)

Based on the aggregated error analysis, which revealed a high false-positive rate (approximately 11%) alongside rare but critical false negatives, we design a difficulty score D_i to serve as a proxy for the *negative expected reward* of sample i. The difficulty score is computed as a weighted sum of error metrics:

$$D_i = \alpha \cdot \text{FP}_i + \beta \cdot \text{FN}_i + \gamma \cdot (1 - \text{conf}_i) \tag{1}$$

where FP_i and FN_i represent the counts of false positives and false negatives for image i, and conf_i denotes the average prediction confidence, representing epistemic uncertainty.

We employ a risk-sensitive parameter setting to define the coefficients:

- $\alpha = 2.0$: Penalizes the dominant error mode (False Positives) to suppress over-detection.
- $\beta = 3.0$: Although False Negatives are rare, they present critical safety risks, necessitating a higher penalty weight.
- $\gamma = 1.0$: Account for samples where the model is uncertain, encouraging exploration.

3.2.2. Policy Definition and Importance Sampling

Based on the scalar difficulty D_i, we define a discrete deterministic policy $\pi(D_i)$ to assign the training weight w_i. This acts as a greedy policy designed to maximize the model's exposure to high-regret samples:

$$w_i = \pi(D_i) = \begin{cases} 3.0 & \text{if } D_i \geq 2.5 \quad \text{(Severe Failure / High Regret)} \\ 2.0 & \text{if } 1.0 \leq D_i < 2.5 \quad \text{(Hard Sample)} \\ 1.0 & \text{otherwise} \quad \text{(Normal Sample)} \end{cases} \tag{2}$$

To implement these weights without modifying the internal gradient calculation of the YOLO architecture, we adopt a sampling strategy via physical oversampling. The probability of sampling image x_i is adjusted so that $P(x_i) \propto w_i$. In practice, images with $w_i \in \{2.0, 3.0\}$ are duplicated in the training manifest.

This physical oversampling strategy effectively implements importance sampling, ensuring that samples with higher bandit-assigned weights are more frequently observed during training. As a result, a reweighted training distribution $\mathcal{D}_{\text{weighted}}$ is constructed, forcing the optimizer to minimize the weighted loss expectation:

$$\mathcal{L}_{\text{total}} = \mathbb{E}_{x \sim \mathcal{D}_{\text{weighted}}}[\mathcal{L}_{\text{YOLO}}(x)] \approx \frac{1}{N} \sum_{i=1}^{N} w_i \cdot \ell(x_i) \tag{3}$$

Since the bandit policy is derived from offline error statistics rather than online interaction, we adopt a deterministic greedy policy, as exploration-based strategies such as ϵ-greedy are not applicable in this offline setting.

3.3. Stage 3: Weighted Fine-Tuning

Finally, the weighted training procedure is executed to evaluate how the proposed reweighting scheme influences detection performance during fine-tuning. This completes the full experimental pipeline involving baseline training, difficulty estimation, weight generation, and weighted fine-tuning.

4. Experiment Results

4.1. Baseline Training and Initial Verification

A baseline YOLO model was first trained on the screw dataset using the standard Ultralytics configuration. The resulting checkpoint (`best.pt`) serves as the basis for all downstream analyzes. Before large-scale processing, the baseline `best.pt` model is tested on a single image to verify that the inference pipeline functions correctly in the local environment. Figure 1 shows an example inference result, where the model successfully detects multiple screws with reasonable confidence scores.

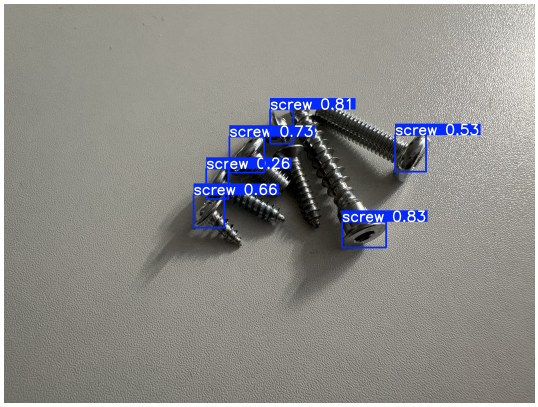

Figure 1. Sanity check of inference using the baseline model on a single image.

4.2. Per-Image Inference and State Generation

After verification, the baseline model was applied to all 1252 training images. For each image, the following statistics were recorded:

- ground-truth object count (GT)
- number of predictions
- true positives (TP)
- false positives (FP)
- false negatives (FN)
- average prediction confidence.

These results were compiled into `state.csv`, which forms the data set used to compute the difficulty of the sample.

4.3. Aggregated Error Statistics

The per-image statistics obtained during state generation were further aggregated into a global error summary stored in `error_summary.csv`. This aggregation provides a dataset-level view of the baseline model's performance characteristics and reveals systematic error patterns that are not evident from individual images alone.

Table 1 reports the aggregated statistics across the entire training set. The results show that false negatives are nearly absent, indicating that the model rarely misses true screw instances. In contrast, false positives occur frequently, with an average of more than two false detections per image. This behavior indicates a strong tendency toward over-detection, particularly in densely cluttered scenes.

Total Images	1252
Total Ground-Truth Objects	20247
Total Predictions	22735
True Positives (TP)	20195
False Positives (FP)	2540
False Negatives (FN)	52
Average Confidence Score	0.848
False Negatives per Image	0.0415
False Positives per Image	2.03
False Negative Rate	0.0026
False Positive Rate	0.11

Table 1. Aggregated error statistics computed from `error_summary.csv`.

4.4. Second-Stage Fine-Tuning with a Controlled Baseline

To evaluate whether bandit-based reweighting improves performance beyond simply training longer, we compare three models under the same validation protocol (conf=0.25, IoU=0.5) on the same validation split.

M0 (Baseline) is the stage-1 checkpoint `best.pt` without any second-stage training.

M1 (Unweighted Fine-Tune, control) continues training from the same M0 initialization for $K = 30$ epochs. To ensure a fair comparison in terms of optimization steps, M1 uses a *uniform-expanded* training set whose size matches the weighted training set, so that each epoch has the same number of iterations as in the weighted setting.

M2 (Weighted Fine-Tune, proposed) also fine-tunes from the same M0 initialization for $K = 30$ epochs with identical hyperparameters and training budget as M1, differing only in that samples are reweighted using the bandit-derived weights.

Compared to M0, M1 yields only modest improvements, indicating that additional fine-tuning alone provides limited gains. Under the same budget, M2 improves recall substantially while also increasing mAP, suggesting that em-

Model	P	R	mAP50	mAP50-95
M0 Baseline (stage-1)	0.865	0.832	0.888	0.751
M1 Unweighted FT (uniform-expanded, $K = 30$)	0.865	0.838	0.891	0.753
M2 Weighted FT (bandit weights, $K = 30$)	0.861	0.867	0.898	0.758

Table 2. Second-stage fine-tuning comparison under matched training budget and matched training-set size. All results are evaluated with conf=0.25 and IoU=0.5 on the same validation split.

phasizing difficult samples helps reduce missed detections. The small precision decrease indicates a mild sensitivity–precision trade-off.

5. Conclusion

We proposed a bandit-inspired sample reweighting pipeline for YOLO screw detection that uses per-image FP/FN/confidence statistics to estimate difficulty and oversample high-regret images during fine-tuning. Under a controlled comparison with matched training budget and matched training-set size, the weighted fine-tuning model improves recall from 0.832 to 0.867 while also improving mAP50 from 0.888 to 0.898 and mAP50-95 from 0.751 to 0.758, with only a small precision drop from 0.865 to 0.861. These gains align with the dataset's error profile where false positives are common (FPR 0.11), suggesting that emphasizing difficult samples can improve robustness without changing the YOLO architecture. Future work will explore learning the weighting policy instead of using a fixed greedy rule and validating on additional datasets and operating points.

Team Contribution Statement

Siyao Yu was responsible for the data preparation and baseline analysis of the project. This included training the initial YOLO model, verifying the correctness and stability of both the inference and training pipelines, and performing large-scale per-image inference across the entire training set to collect detection statistics for the construction of `state.csv`. Siyao Yu further conducted aggregated error analysis based on these statistics and produced `error_summary.csv`, which provided quantitative insights that supported the subsequent sample reweighting strategy and fine-tuning experiments.

Xintong Li was responsible for the reinforcement learning formulation and algorithmic design of the proposed method. This included formulating sample reweighting as an offline contextual bandit problem, designing the image-level difficulty function as a proxy negative reward, and defining the discrete action space of training weights. Xintong Li also implemented the bandit-based weighting strategy, generated the `weights.csv` file, and designed the importance sampling mechanism via physical oversampling that integrates the bandit policy into the YOLO fine-tuning process.

Anyongyong Zhao contributed to the overall project framework and experiment pipeline setup, including organizing the second-stage fine-tuning workflow and ensuring a consistent, reproducible evaluation protocol. Anyongyong Zhao was responsible for the second-stage fine-tuning experiments and controlled evaluation design, including constructing the unweighted fine-tuning control with matched training-set size (uniform-expanded) and running the bandit-weighted fine-tuning under identical hyperparameters. Anyongyong Zhao also extracted the final metrics used in Table 2 and contributed to the results presentation and conclusion writing.

References

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lattimore, T. and Szepesvári, C. *Bandit Algorithms*. Cambridge University Press, 2020.

Lin, T., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

Shrivastava, A., Gupta, A., and Girshick, R. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 761–769, 2016.