InsightBoard: An Interactive Multi-Metric Visualization and Fairness Analysis Plugin for TensorBoard

Ray Zeyao Chen and **Christan Grant**
Department of Computer & Information Science & Engineering
University of Florida, Gainesville, FL, USA
{chenz1, christan}@ufl.edu

Abstract

Modern machine learning systems deployed in safety-critical domains require visibility not only into aggregate performance but also into how training dynamics affect subgroup fairness over time. Existing training dashboards primarily support single-metric monitoring and offer limited support for examining relationships between heterogeneous metrics or diagnosing subgroup disparities during training. We present InsightBoard, an interactive TensorBoard plugin that integrates synchronized multi-metric visualization with slice-based fairness diagnostics in a unified interface. InsightBoard enables practitioners to jointly inspect training dynamics, performance metrics, and subgroup disparities through linked multi-view plots, correlation analysis, and standard group fairness indicators computed over user-defined slices. Through case studies with YOLOX on the BDD100k dataset, we demonstrate that models achieving strong aggregate performance can still exhibit substantial demographic and environmental disparities that remain hidden under conventional monitoring. By making fairness diagnostics available during training, InsightBoard supports earlier, more informed model inspection without modifying existing training pipelines or introducing additional data stores.

Introduction

TensorBoard is widely used to monitor training, but many debugging questions require correlating heterogeneous signals (e.g., loss, learning rate schedules, gradient distributions) across runs and time. In parallel, responsible deployment requires slice-level auditing: strong aggregate scores can coexist with large disparities across demographic and environmental subgroups (Mehrabi et al. 2021). Existing fairness toolkits can compute disparities, but they are often disconnected from the training dashboard, often leading to fragmented workflows where fairness checks are performed separately from routine training monitoring(Bird et al. 2020; Bellamy et al. 2018).

We present **InsightBoard**, a TensorBoard plugin that integrates multi-run, multi-metric monitoring with interactive slice-based fairness analysis in a single interface. InsightBoard supports synchronized multi-view plots, correlation views, and subgroup fairness indicators, enabling iterative "slice–inspect–adjust–recheck" cycles during development. In a YOLOX case study on BDD100k, we show that models with high overall accuracy can still exhibit large subgroup gaps, and that InsightBoard makes these trade-offs explicit early in training.

Contributions. This paper makes two contributions: (1) System and diagnostic integration: We design and implement InsightBoard, a lightweight TensorBoard plugin that unifies synchronized multi-run, multi-metric visualization with slice-based fairness auditing (including standard disparity indicators and IN/OUT analysis), without requiring additional data stores or modifications to existing training workflows. (2) Empirical validation: Through case studies on YOLOX with BDD100k and an overhead analysis, we demonstrate the feasibility and practical value of integrating fairness diagnostics directly into the training process.

Related Work

Object detection and autonomous driving datasets have documented subgroup performance disparities that aggregate metrics can mask (Yu et al. 2020; Liu et al. 2021; Zhang et al. 2021; Buolamwini and Gebru 2018). Fairness objectives and the accuracy–fairness trade-off are well studied in classification (Hardt, Price, and Srebro 2016; Chouldechova 2017), but real-time monitoring during training is less explored for detection.

Fairness toolkits such as Fairlearn and AI Fairness 360 provide reporting and mitigation workflows (Bird et al. 2020; Bellamy et al. 2018). TensorFlow Fairness Indicators integrates sliced metrics into TensorBoard-style pipelines, while the What-If Tool supports interactive probing (Wexler et al. 2019). These tools support interactive fairness analysis, but do not provide synchronized multi-run training trace comparison or correlated multi-metric views within the standard TensorBoard workflow.

Experiment tracking platforms (e.g., Weights & Biases, MLflow) support multi-run comparisons but do not natively expose fairness indicators or linked correlation views. Conversely, fairness toolkits provide valuable post-hoc auditing but remain disconnected from routine training workflows. InsightBoard bridges this gap by integrating slice-based fairness diagnostics directly into the standard TensorBoard environment. Interactive visual diagnosis has recently proven

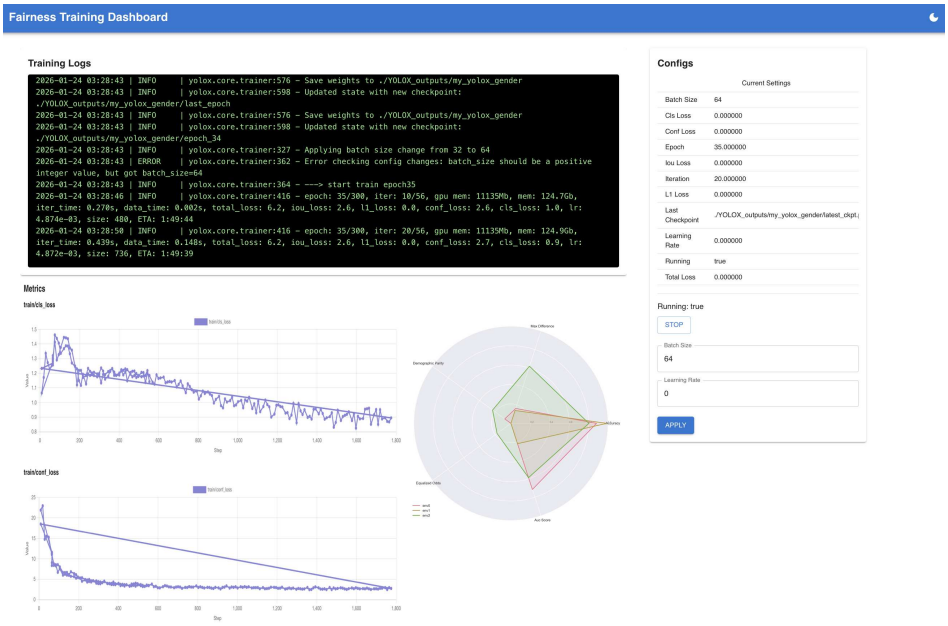

Figure 1: InsightBoard System Architecture. The backend plugin intercepts data requests to perform on-the-fly metric aggregation and fairness calculations, serving this processed data to a new frontend dashboard.

highly effective in surfacing multidimensional fairness issues (Chen and Grant 2026). However, these approaches are predominantly applied post-deployment. InsightBoard extends the visual diagnostic philosophies of RISE directly into the active training loop, bridging the gap between performance debugging and real-time responsible AI assessment.

System Architecture and Implementation

InsightBoard is a lightweight, extensible TensorBoard plugin designed to integrate directly into existing MLOps workflows for joint performance and fairness analysis. The key architectural novelty lies in its ability to perform on-the-fly multi-metric synchronization and slice-based fairness computation directly from standard event logs, without requiring secondary databases or custom logging frameworks. To foster community adoption and reproducible research, InsightBoard will be released as an open-source repository on GitHub upon publication. The system follows a modular backend–frontend design built entirely on standard TensorBoard infrastructure (Figure 1).

Backend Processing Pipeline

The backend operates entirely within the TensorBoard WSGI application state. We utilize the TensorBoard event processing API, specifically the EventMultiplexer, to asynchronously ingest scalar, histogram, and tensor summaries. Because computing intersectional fairness metrics across thousands of steps is memory-intensive, our Metric Aggregator implements a sliding-window reservoir sampling technique. It aligns heterogeneous logs (e.g., learning rate scalars) to a common monotonic axis (training step).

When a user requests a fairness slice, the Fairness Metric Calculator intercepts raw prediction tensors and ground-truth label bindings. It dynamically partitions these arrays based on the requested demographic attributes (e.g., gender, lighting) and computes standard indicators like Demographic Parity Difference and Equalized Odds (Hardt, Price, and Srebro 2016). This allows the system to serve complex, cross-referenced JSON payloads to the frontend REST endpoints in under 100ms, maintaining the low-latency requirement of interactive dashboards.

Frontend Design and Interactive Visualization

The frontend is a TypeScript and D3.js single-page application packaged as a TensorBoard plugin. It maintains its own client-side state, enabling rich interactions without page reloads. The interface is organized around two tightly integrated views:

Multi-Metric Analysis and Correlation. Users can select multiple experiment runs and arbitrary metric combinations. Visualized on a synchronized x-axis, this enables causal inspection (e.g., relating learning rates to gradient distributions). Interactive brushing highlights corresponding regions across plots, while a correlation heatmap surfaces salient dependencies.

Integrated Fairness Analysis. The Fairness Dashboard embeds fairness auditing directly into the training workflow (Bird et al. 2020; Wexler et al. 2019). Users can slice performance by sensitive attributes and inspect disaggregated metrics alongside group fairness indicators (Demographic Parity, Equalized Odds). Furthermore, InsightBoard supports real-time exploration of the accuracy–fairness tradeoff through interactive controls. Metrics update live as

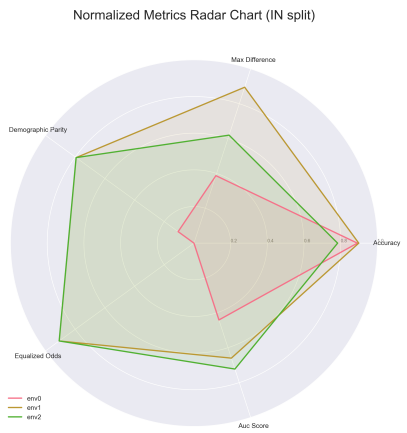

Figure 2: The Fairness Dashboard visualizing IN-distribution evaluation. The radar chart axes represent disaggregated mAP across specific subgroups (e.g., Male/Daytime vs. Female/Nighttime). The area of the polygon allows practitioners to visually compare the overall equity footprint of competing model runs.

Table 1: Feature comparison of visualization and fairness tools.

Feature	TensorBoard	Fairlearn	InsightBoard
Metric tracking	✓	—	✓
Multi-metric correlation	—	—	✓
Slice-based fairness	—	✓	✓
In-training monitoring	—	—	✓

hyperparameter controls (e.g., fairness weighting terms) change, enabling rapid what-if analysis to identify Pareto-optimal operating points. Group fairness metrics, including Demographic Parity Difference and Equalized Odds Difference, are computed and visualized to make inequities immediately apparent (Bellamy et al. 2018; Hardt, Price, and Srebro 2016).

InsightBoard also supports real-time exploration of the accuracy–fairness trade-off through interactive controls (Figure 3). In our experimental setup, we expose selected training hyperparameters, including a fairness weighting term implemented externally to the model, to support controlled what-if analysis. InsightBoard visualizes the resulting metric changes but does not prescribe or automate optimization strategies. Metrics update live as controls change, enabling rapid what-if analysis. In BDD100k experiments with YOLOX, this capability revealed persistent gender-based AP gaps even as overall mAP improved, allowing users to identify Pareto-optimal operating points by tuning fairness constraints during training.

Fairness Metrics and Interaction Design

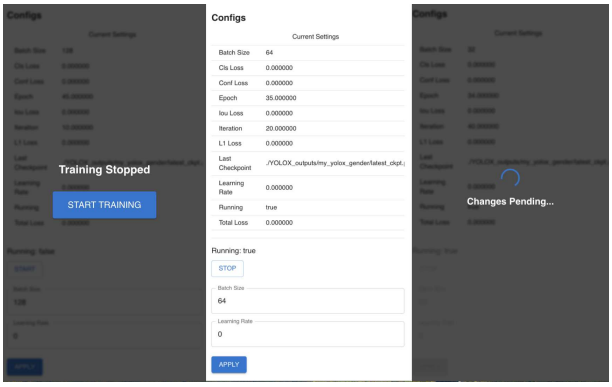

Figure 3: The configuration interface. The left panel shows specific hyperparameter deltas (e.g., fairness penalty weights), while the linked timeline on the right displays the corresponding real-time impact on the Equalized Odds gap during the training progression.

Disaggregated performance. InsightBoard supports task-dependent performance metrics. For classification, we track accuracy, precision/recall, and AUC; for detection, we track AP under user-selected IoU thresholds (Ge et al. 2021). All metrics can be computed per-slice (e.g., gender, age, lighting, domain), enabling users to identify where aggregate improvements hide subgroup regressions.

Fairness metrics. We include commonly used group disparity summaries such as Demographic Parity (DP) gap and Equalized Odds (EO) gaps (Hardt, Price, and Srebro 2016). For binary classification, DP gap can be expressed as: $\Delta_{\mathrm{DP}} = |\Pr(\hat{y} = 1 \mid a = 0) - \Pr(\hat{y} = 1 \mid a = 1)|$. EO is captured by disparities in error rates, such as differences in true positive rates and false positive rates across groups. These indicators are intended as diagnostic signals rather than final judgments, consistent with survey guidance on fairness evaluation (Mehrabi et al. 2021).

Distribution stability. InsightBoard reports metrics across IN (train/val/test) and OUT environments when available, enabling users to distinguish fairness improvements that persist from those that are brittle under distribution shift (Figure 2).

Linked interaction model. InsightBoard uses cross-view linking, where selecting a run, epoch range, or subgroup in one view highlights the corresponding elements across all views (timelines, subgroup bars, and correlation heatmaps), supporting rapid identification of patterns such as fairness–performance trade-offs. A complementary text pane summarizes configuration deltas (e.g., optimizer, learning rate schedule, batch size) for the selected run.

Experiments and Case Studies

Our experiments serve as diagnostic case studies to evaluate whether InsightBoard enables practitioners to detect and interpret performance–fairness trade-offs that remain hidden under standard dashboards. We use YOLOX (Ge et al. 2021), a widely adopted real-time object detector, trained on BDD100k, a large-scale autonomous driving dataset con-

Table 2: Pilot dataset composition reported during development.

Attribute	Category	Count
Skin tone	$skin_0$ (unknown)	6614
	$skin_1$ (light)	2883
	$skin_2$ (dark)	847
Age	age_0 (adult)	6293
	age_1 (young)	3051

taining demographic and environmental annotations (e.g., lighting, weather).

Dataset. BDD100k is a large-scale autonomous driving dataset with demographic and environmental annotations relevant for fairness analysis (Yu et al. 2020). Its diversity across lighting and weather conditions enables slice-level evaluation of detection performance. **Model.** We use YOLOX (Ge et al. 2021), a widely adopted real-time object detector, as a representative modern detection model."

Implementation Details

InsightBoard is implemented as a Python backend plugin and a TypeScript frontend application. The backend leverages TensorBoard's plugin architecture to register custom routes that intercept data requests, perform on-the-fly metric computation, and serve processed data via REST APIs. The frontend uses D3.js for visualization rendering and React for state management, enabling complex interactions such as linked brushing and real-time updates without page reloads.

For YOLOX integration, we extended the training loop to log fairness metrics alongside standard performance metrics. Specifically, we compute subgroup-specific metrics by filtering predictions and ground truth annotations based on demographic attributes from BDD100k. These metrics are logged as TensorBoard scalar summaries, which InsightBoard then aggregates and visualizes. The fairness weight parameter is implemented as a hyperparameter that scales the contribution of fairness loss terms in the training objective, allowing real-time exploration of the accuracy-fairness trade-off.

System Overhead Evaluation

We evaluated InsightBoard's computational overhead across different scales. For a typical YOLOX training run with 100 epochs, parsing event logs takes approximately 2-3 seconds. Metric computation time per epoch is negligible ($<$10ms) for standard metrics, but increases to 50-100ms when computing fairness metrics across multiple slices (e.g., gender $\times$ lighting condition). Frontend responsiveness remains acceptable (refresh latency $<$500ms) for up to 50 concurrent runs, degrading gracefully beyond this scale. These overheads are minimal compared to training time, making real-time fairness monitoring feasible without significant performance impact.

We demonstrate InsightBoard's capabilities through two core scenarios using YOLOX models trained on the BDD100k dataset. These scenarios illustrate how InsightBoard enables practitioners to identify and analyze fairness issues that are invisible when examining only aggregate performance metrics.

Table 3: Evaluation tasks and outcome measures for a controlled comparison study.

Task	Outcome measure
Find worst slice	correctness; time
Explain trade-off	explanation quality; time
Pick best run	correctness; confidence
Locate config cause	correctness; time

Case Study: Correlation-Driven Debugging and Fairness Diagnosis

We demonstrate InsightBoard on YOLOX trained on BDD100k, focusing on how integrated multi-metric and slice-based analysis reveals both training instabilities and hidden subgroup disparities.

First, comparing two runs with different learning rates (0.01 vs. 0.001), InsightBoard's synchronized multi-metric views (mAP, loss, learning rate, and gradient norm distributions) reveal that the higher learning rate induces unstable gradient behavior during critical training phases. This instability explains why the model converges faster but achieves lower final mAP (78% vs. 85%), providing a direct link between hyperparameter choice, internal dynamics, and performance.

Second, slice-based analysis exposes substantial subgroup disparities masked by aggregate metrics. While a baseline model achieves 85.2% overall mAP, InsightBoard reveals a large gender–environment gap (94.8% for male/daytime vs. 59.3% for female/nighttime). Notably, this divergence becomes visible early in training (around epoch 20). By adjusting the fairness penalty and applying targeted augmentation, we observe a trade-off in real time: aggregate mAP decreases slightly (to 83.1%), while the subgroup gap is substantially reduced (from 35% to 18%).

Together, these results show that InsightBoard enables early detection of both optimization issues and fairness failures, supporting more informed and efficient model iteration.

Key Findings

Our results demonstrate that aggregate metrics consistently mask intersectional failures in safety-critical object detection. InsightBoard advances the subject domain by collapsing the "train, then audit" lifecycle into a single, cohesive workflow. By exposing these gaps dynamically, practitioners can halt biased models early, saving compute resources, and actively steer the model toward a more equitable Pareto frontier during hyperparameter tuning rather than attempting post-hoc mitigation.

Precision vs. Recall Trade-off: In safety-critical applications like autonomous driving, false negatives (missed pedestrians) are typically more costly than false positives

(false alarms). Therefore, practitioners may prioritize recall over precision, accepting more false alarms to ensure pedestrian safety. InsightBoard enables real-time exploration of this trade-off by visualizing precision and recall disaggregated by demographic groups, revealing whether precision-recall imbalances disproportionately affect certain subgroups.

Accuracy-Fairness Trade-off: Perhaps the most fundamental trade-off is between aggregate accuracy and fairness. Optimizing solely for accuracy (e.g., maximizing $cocoAP5$) may lead models to exploit demographic correlations present in training data, resulting in biased outcomes. Conversely, enforcing strict fairness constraints may reduce aggregate performance. InsightBoard's real-time parameter adjustment allows practitioners to explore the accuracy-fairness Pareto frontier, identifying configurations that achieve acceptable balances for their specific deployment context.

Effectiveness of Visual Representations Rather than introducing novel chart types, InsightBoard's value stems from the tight spatial integration of standard visual tools. The correlation matrix serves as a standard utility to quickly surface inverse relationships between aggregate accuracy and fairness gaps. By linking these standard views (timelines, radar charts, and matrices) to a shared state, users can immediately correlate a spike in learning rate to a degradation in a specific demographic slice's performance. Furthermore, radar charts (Figure 2) enable users to assess whether fairness improvements persist across IN/OUT data distributions—a critical consideration for real-world deployment.

Conclusion

We presented InsightBoard, an interactive TensorBoard plugin that integrates multi-metric correlation analysis with slice-based fairness diagnostics during training. Through case studies on YOLOX and BDD100k, we showed that strong aggregate performance can coexist with substantial demographic and environmental disparities that remain invisible under conventional monitoring.

By embedding fairness diagnostics directly into the training workflow, InsightBoard enables earlier and more informed inspection of performance–fairness trade-offs. This capability is particularly valuable in safety-sensitive domains where late discovery of subgroup failures carries serious consequences. Future work includes extending InsightBoard to additional model classes (e.g., language models), such as MultiScript30k (Driggers-Ellis et al. 2025), evaluating its applicability on multi-script datasets to study non-vision fairness diagnostics, and conducting comprehensive user studies to empirically validate its interaction design.

References

Bellamy, R. K. E.; Dey, K.; Hind, M.; Hoffman, S. C.; Houde, S.; Kannan, K.; Lohia, P.; Martino, J.; Mehta, S.; Mojsilovic, A.; et al. 2018. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*.

Bird, S.; Dudik, M.; Edgar, R.; Horn, B.; Lutz, R.; Milan, V.; Sameki, M.; Wallach, H.; Walker, K.; et al. 2020. Fairlearn: A toolkit for assessing and improving fairness in AI. Technical Report MSR-TR-2020-32, Microsoft.

Buolamwini, J., and Gebru, T. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 77–91.

Chen, R., and Grant, C. 2026. Rise: Interactive visual diagnosis of fairness in machine learning models. *arXiv preprint arXiv:2602.04339*.

Chouldechova, A. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data* 5(2):153–163.

Driggers-Ellis, C.; Brinkley, D.; Chen, R.; Dhawan, A.; Wang, D. Z.; and Grant, C. 2025. Multiscript30k: Leveraging multilingual embeddings to extend cross script parallel data. *arXiv preprint arXiv:2512.11074*.

Ge, Z.; Liu, S.; Wang, F.; Li, Z.; and Sun, J. 2021. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*.

Hardt, M.; Price, E.; and Srebro, N. 2016. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*.

Liu, L. T.; Dean, S.; Rolf, E.; Simchowitz, M.; and Hardt, M. 2021. Fairness in deep learning: A computational perspective. *IEEE Security & Privacy* 19(5):81–89.

Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; and Galstyan, A. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys* 54(6):1–35.

Wexler, J.; Pushkarna, M.; Bolukbasi, T.; Wattenberg, M.; Viegas, F.; and Wilson, J. 2019. The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics* 26(1):56–65.

Yu, F.; Xian, W.; Chen, Y.; Liu, F.; Liao, M.; Madhavan, V.; and Darrell, T. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2636–2645.

Zhang, H.; Lu, Y.; Wang, Y.; et al. 2021. Fairness in object detection: A survey. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.