

StreamFair: Online Fair Adaptation Under Temporal Intersectional Drift

Anonymous Author(s)

Abstract

A model that satisfies equalized odds on its training data does not stay fair once the input distribution starts to change, and the gap is the largest on intersectional subgroups whose sample counts may become smaller over time as the stream goes on. We can either let the equalized odds (EO) gap grow over time by doing nothing or apply online learning with cross-entropy which in turn can make the gap worse because the gradient is dominated by the majority intersection group. We present *StreamFair*, which tackles this issue by keeping the deployed model frozen, then attaching a small residual module on top of the frozen model, and training the residual module only when a fairness-aware drift detector detects a shift. The detector observes the worst-intersection EO gap with a cumulative sum and exponentially weighted moving average measurement. Once the detector is initiated and detects drift, the residual module is trained with cross-entropy along with per-intersection penalties on the true-positive rate and false-positive rate gaps and a supervised contrastive term that keeps rate estimates stable on small batches. Our experiments on the ACSIncome dataset spanning 2014 to 2019 in California, Texas and New York show that, *StreamFair* achieves same level fairness and accuracy compared to the always-adapt fair baseline while running 6 to 10x fewer updates.

Keywords

intersectional fairness, distribution shift, online learning, equalized odds

1 Introduction

A model that is fair on its training data does not stay fair once the input distribution starts to drift. This fairness gap is largest on intersectional subgroups which are defined by joint protected attributes (e.g., race x sex x age). Research shows model that uses single sensitive attribute for decision making system misses the intersectional bias [1]. Over time, as new data arrives, these intersectional groups are the ones whose number of samples becomes smaller. One way to mitigate this is to retrain the model with new data. However in this case, during the training, the gradient is dominated by the majority intersection while the minor one are pushed farther by drift. As a result, the per-intersection gap becomes larger even though the overall accuracy goes up.

Researchers have previously tried to mitigate these issues individually but have not combined them. Fair representation methods have been used to control the group gaps but they assume the data is static [2, 7]. Concept drift detectors act on changes in the data but they only observe the loss, so they cannot specify which subgroup is getting worse results [4, 5]. Existing intersectional fairness criteria measure the gap at one point in time and do not consider the continuous changes over time [6, 8]. So, there is a gap that monitors a fairness signal directly, retrains only when that signal worsens,

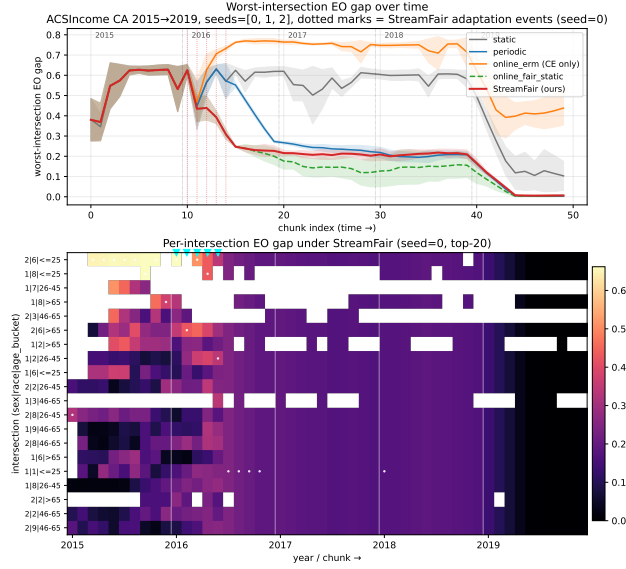


Figure 1: Top: $\Delta_{EO}(t)$ over the 50 stream chunks on CA, mean over three seeds with std bands; dotted verticals mark StreamFair’s adaptation events (seed 0). Bottom: per-intersection contributions to Δ_{EO} under StreamFair, rows ordered by peak; cyan triangles are adaptation events.

and stays stable on the small, imbalanced batches that streaming intersectional data produces.

In this work, we introduce STREAMFAIR with two contributions. (i) A frozen-base, residual-module design with a self-tuning fairness-aware drift detector that watches the worst-intersection EO gap using a Cumulative Sum (CUSUM) and an Exponentially Weighted Moving Average (EWMA). (ii) A training objective that combines cross-entropy with per-intersection penalties on the TPR and FPR gaps, and a supervised contrastive term that keeps the rate estimates stable on small batches. Experiments on ACSIncome 2014 to 2019 in CA, TX, and NY show that *StreamFair* reaches the same fairness level as a continuously adapting fair baseline while running 6 to 10x fewer updates, with the same method ordering on all three states.

2 StreamFair

In this section, we present StreamFair. The main idea is to keep the deployed model frozen so its behavior stays easy to inspect. We attach a small correction module (residual module) on top of the frozen model, observe the fairness signal, and train only the correction module when the signal gets worse.

Frozen base, residual module: Let ϕ be an encoder and w a linear head trained on year Y_0 . At deployment we freeze (ϕ, w) and

add a small residual module g_θ :

$$\text{logit}(x) = w(\phi(x)) + g_\theta(\phi(x)) \quad (1)$$

The adapter g_θ is a two-layer MLP with LayerNorm and it fixes the fairness gap that comes up as the distribution shifts. We only update θ online, and rolling back to the previous version of the model just takes a single weight swap.

Fairness monitor: We divide the data into chunks for each year, and for every chunk we update a sliding window of size W over sub-group's (I) TPR and FPR and compute the worst-intersection equalized odds difference.

$$\Delta_{EO}(t) = \max_I \max(|TPR_I - TPR_{pop}|, |FPR_I - FPR_{pop}|) \quad (2)$$

where pop is the population-level rate. The $\Delta_{EO}(t)$ is zero when equalized odds hold on every well-populated intersection, and the value rises with the size of the largest violation.

Fairness drift detector: The $\Delta_{EO}(t)$ from the monitor phase gives us the status of fairness violation at time t . However, a single number is not enough to decide when to retrain. If we train for every small bump, we will end up with constant re-training, and if we wait for a big one, we will react too late. So, we want to capture a long, slow rise where the gap increases chunk after chunk, and a short burst where the gap suddenly stays high for a few chunks in a row. To capture these, we use Cumulative Sum (CUSUM) [10] and Exponentially Weighted Moving Average (EWMA) [11] together. CUSUM keeps track of the Δ_{EO} over time and the detector is initiated once the CUSUM statistic crosses the threshold h , capturing the slow rise. EWMA tracks the smoothed average of Δ_{EO} and initiates the detector when it exceeds h_E , catching the short burst. Usually a CUSUM needs a hand-set reference based on prior knowledge of the in-control gap, we address this by running the model for T_{warm} chunks and using the mean of Δ_{EO} over that window, so the detector tunes itself from the data. The residual module is trained only on chunks where at least one of the two detectors is initiated.

Final Objective: Once the detector decides that the fairness gap has drifted, we adapt the model by training only the residual module on the labeled examples kept in memory. Since only a small part of the data is labeled, we store labeled examples in a per-intersection reservoir with capacity K using Vitter sampling [12]. This way each intersection keeps a representative sample at fixed memory and rare subgroups are not crowded by majority. When the detector is initialized and the reservoir holds at least K_{min} rows, we update the residual module by minimising

$$\begin{aligned} \mathcal{L} = & CE + \lambda_T \sum_{I: n_I^+ \geq 1} |\tilde{T}_I - \tilde{T}| \\ & + \lambda_F \sum_{I: n_I^- \geq 1} |\tilde{F}_I - \tilde{F}| + \lambda_{SC} \mathcal{L}_{SupCon}^{(y, I)}, \end{aligned} \quad (3)$$

where \tilde{T}_I, \tilde{F}_I are soft per-intersection TPR and FPR estimates and \tilde{T}, \tilde{F} are the corresponding population values. The CE keeps track of the model accuracy, the two TPR and FPR terms pull each intersection's TPR and FPR toward the population rate, and the final part is a supervised contrastive loss [9] that brings points sharing both the label and the intersection close together in the embedding so the rate estimates stay stable on small batches.

Table 1: ACSIncome from 2014 to 2019, we present mean \pm std across three seeds. $\overline{\Delta_{EO}}$ is the chunk-averaged worst-intersection EO gap. Δ_{EO}^{final} is the same quantity at the last chunk (2019). Lower EO, higher acc are better. Residual-module training count is the median across seeds. Best result per state per column in bold, second best underlined.

state	mode	$\overline{\Delta_{EO}}$	Δ_{EO}^{final}	Acc	Residual.
CA	STATIC	0.501 \pm .013	0.103 \pm .072	<u>0.745 \pm .009</u>	0
CA	PERIODIC	0.306 \pm .010	0.003 \pm .003	0.676 \pm .005	<u>8</u>
CA	ONLINE_ERM	0.638 \pm .024	0.438 \pm .084	0.771 \pm .015	40
CA	ONLINE_FAIR_STATIC	0.242 \pm .028	0.007 \pm .002	0.643 \pm .006	40
CA	STREAMFAIR	<u>0.271 \pm .005</u>	<u>0.006 \pm .004</u>	0.655 \pm .007	7
TX	STATIC	0.349 \pm .028	0.149 \pm .086	<u>0.716 \pm .003</u>	0
TX	PERIODIC	0.220 \pm .018	<u>0.072 \pm .036</u>	0.674 \pm .006	<u>8</u>
TX	ONLINE_ERM	0.558 \pm .010	0.322 \pm .018	0.756 \pm .001	40
TX	ONLINE_FAIR_STATIC	0.141 \pm .020	0.006 \pm .006	0.652 \pm .004	40
TX	STREAMFAIR	<u>0.187 \pm .015</u>	0.078 \pm .021	0.667 \pm .001	4
NY	STATIC	0.463 \pm .019	0.240 \pm .076	<u>0.731 \pm .005</u>	0
NY	PERIODIC	0.268 \pm .013	0.015 \pm .020	0.661 \pm .003	<u>8</u>
NY	ONLINE_ERM	0.568 \pm .025	0.437 \pm .040	0.759 \pm .006	40
NY	ONLINE_FAIR_STATIC	0.185 \pm .010	<u>0.005 \pm .004</u>	0.626 \pm .003	40
NY	STREAMFAIR	<u>0.202 \pm .008</u>	0.003 \pm .002	0.637 \pm .003	5

3 Experiments

For our experiments, we use the ACSIncome dataset [3] with a three-layer MLP base model and trained on $Y_0 = 2014$. Then we stream the data from 2015 to 2019 on CA, TX, and NY at 10 chunks per year and train across three seeds. We use sex, race, and age bucket (up to 72 intersections) as protected attributes. Baselines: STATIC (no adaptation), PERIODIC (every five chunks), ONLINE_ERM (always adapt, CE only), and ONLINE_FAIR_STATIC (always adapt with Eq. (3)).

Results: Table 1 reports the average equalized odds gaps over the chunks, accuracy and residual module training count. Figure 1 shows $\Delta_{EO}(t)$ on CA. We observe from the table and the figure that StreamFair achieves almost the same averaged EO and accuracy in all three states, compared to the Online_Fair_Static baseline. However, we achieve almost 6 to 10 times fewer residual training count due to our fairness drift detector.

4 Conclusion

We presented STREAMFAIR, a frozen-base, residual-module design that adapts only when a self-tuning fairness-aware drift detector activates. On ACSIncome 2014 to 2019, it reaches the same fairness level as a continuously adapting fair baseline at roughly one tenth of the compute, while plain cross-entropy retraining *amplifies* intersectional bias even as accuracy goes up. Since the deployed weights stay frozen, only the residual module changes and rollback is a single weight swap, which makes the adaptation mechanism transparent. In future work, we will aim to extend the detector to non-tabular streams.

References

- [1] Robert Bartlett, Adair Morse, Richard Stanton, and Nancy Wallace. 2022. Consumer-lending discrimination in the FinTech era. *Journal of Financial Economics* 143, 1 (2022), 30–56.
- [2] Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. 2023. Leace: Perfect linear concept erasure in closed form. *Advances in Neural Information Processing Systems* 36 (2023), 66044–66063.
- [3] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. 2021. Retiring adult: New datasets for fair machine learning. *Advances in neural information processing systems* 34 (2021), 6478–6490.
- [4] Gregory Ditzler, Manuel Roveri, Cesare Alippi, and Robi Polikar. 2015. Learning in nonstationary environments: A survey. *IEEE Computational intelligence magazine* 10, 4 (2015), 12–25.
- [5] João Gama, Indrè Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM computing surveys (CSUR)* 46, 4 (2014), 1–37.
- [6] Avijit Ghosh, Lea Genuit, and Mary Reagan. 2021. Characterizing intersectional group fairness with worst-case comparisons. In *Artificial Intelligence Diversity, Belonging, Equity, and Inclusion*. PMLR, 22–34.
- [7] Nikola Jovanović, Mislav Balunovic, Dimitar Iliev Dimitrov, and Martin Vechev. 2023. Fare: Provably fair representation learning with practical certificates. In *International Conference on Machine Learning*. PMLR, 15401–15420.
- [8] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International conference on machine learning*. PMLR, 2564–2572.
- [9] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems* 33 (2020), 18661–18673.
- [10] Ewan S Page. 1954. Continuous inspection schemes. *Biometrika* 41, 1/2 (1954), 100–115.
- [11] Stuart W Roberts. 2000. Control chart tests based on geometric moving averages. *Technometrics* 42, 1 (2000), 97–101.
- [12] Jeffrey S Vitter. 1985. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)* 11, 1 (1985), 37–57.