Learning to Trigger: Reinforcement Learning at Large Hadron Collider

Zixin Ding [1] **Shaghayegh Emami** [2] **Giovanna Salvi** [2] **Cecilia Tosciri** [1] **Abhijith Gandrakota** [3] **Jennifer Ngadiuba** [3] **Nhan Tran** [3] **Christian Herwig** [2] **David Miller** [1] **Yuxin Chen** [1]

Abstract

High-throughput scientific facilities such as the Large Hadron Collider depend on real-time event filtering (*triggering*) under tight constraints on bandwidth, latency, and storage. In practice, trigger menus are largely static and hand-tuned and can become suboptimal as detector conditions, pileup, and background composition drift over time. We cast online threshold tuning as a sequential decision-making problem: a reinforcement learning agent ingests streaming summaries of recent rates and signal-sensitive features and updates trigger thresholds to maximize signal efficiency while tracking a target background rate within a tolerance band. We adapt Group-Filtered Policy Optimization (GFPO) to streaming control and introduce two variants (GFPO-F, GFPO-FR) that enforce background rate feasibility during training. On a benchmark that emulates realistic collider operation, we study two representative triggers: a total transverse energy (H_T) trigger sensitive to pileup variation, and an anomaly-detection (AD) trigger based on reconstruction loss for rare or non-standard signatures. On Monte Carlo streams, our agent increases the fraction of in-tolerance time intervals by 38% (H_T) and 30% (AD), with a cumulative gain of up to 5% in signal efficiency on those in-tolerance intervals. Transferring from simulation to *real* collision data (CMS Run 283408), the same agent, without fine-tuning, achieves a 40% (H_T) and 25% (AD) in-tolerance improvement over baselines, with an additional 1% signal-efficiency gain on both triggers. To our knowledge, this is the first demonstration of RL-based trigger control on real Large Hadron Collider collision data.

[1]University of Chicago [2]University of Michigan, Ann Arbor [3]Fermi National Accelerator Laboratory. Correspondence to: Zixin Ding <zixin@uchicago.edu>, Yuxin Chen <chenyuxin@uchicago.edu>.

Proceedings of the AI4Physics Workshop at the 43rd International Conference on Machine Learning (AI4Physics@ICML 2026), Seoul, South Korea. 2026. Copyright 2026 by the author(s).

1. Introduction

High-throughput discovery science relies on real-time event selection under strict bandwidth and latency constraints (Albrecht et al., 2025; Mahesh et al., 2021; collaboration, 2006; Aad et al., 2022). At the Large Hadron Collider (LHC), trigger menus are largely static and hand-tuned, requiring detailed simulation and repeated expert re-optimization as pileup and background composition drift (Aad et al., 2022; Emami et al., 2026). This workflow is costly and can quickly become miscalibrated under distribution shift, reducing efficiency for detecting rare phenomena. Only recently have efforts in dynamic bandwidth allocation (Evans et al., 2025) and low-latency menu adaptation (Mahesh et al., 2021) begun to explore learned policies that retune thresholds online. Yet whether such continuous adaptation is even *feasible* in the more common regime, where a fixed menu of thresholds must be retuned online under shifting conditions, remains largely unexplored. We study the design of a *self-driving* trigger: an autonomous, hardware-aware filtering system that adapts its selection thresholds online to maintain stable background rate with improved signal efficiency, in response to changing physical conditions.

Existing work (Emami et al., 2026) introduced an autonomous trigger-control framework extending a PID baseline (Bradu et al., 2018) to a cost-function formulation jointly optimizing rate stability, signal efficiency, and computational cost (CMS Collaboration, 2017). This exhaustive search over discrete threshold candidates is effective for small menus but becomes computationally prohibitive as the action space grows, and neither strategy conditions on streaming context such as pileup (collaboration et al., 2014). RL (Sutton & Barto, 1998; Hausknecht & Stone, 2015; Hafner et al., 2020) offers a principled alternative: a learned policy over feature-rich states can anticipate distribution shifts, scale to large action spaces, and navigate the multi-objective trade-offs that the cost-function framework formalizes but cannot efficiently solve.

We thus formulate adaptive thresholding as a sequential decision-making problem (Matchev & Shyamsundar, 2021; Oh & Iyengar, 2019; Yang et al., 2024) and train an RL policy (Sutton & Barto, 1998) that observes compact streaming summaries (e.g., recent rates and distributional statistics)

and updates thresholds to maximize long-horizon physics utility while satisfying hard background-rate constraints (Albrecht et al., 2025; Cerri et al., 2019; Cella, 2024). By learning directly from time-dependent data, the policy can capture nonlinear and delayed effects and allocate rate budget more effectively, reducing manual retuning and improving performance in non-stationary environments.

We study two representative trigger classes trained on simulated samples with deployment on **real** collision data recorded by the CMS experiment (CMS Collaboration, 2024b). Specifically, we ask: *Can an RL-based trigger controller improve signal efficiency while maintaining comparable (or better) background-rate stability than control-based baselines on real collision data?* If so, *is it preferable to train on physicist-generated simulation and deploy on real collision data, or to rely on test-time training?*

Contributions. Concretely, we make four contributions.

(1) We introduce a unified evaluation suite with metrics for **feasibility** (InBand fraction: the proportion of events whose triggered rate falls within the bandwidth tolerance) and **performance** (signal efficiency). We cast online trigger thresholding as an RL problem with streaming observations and show that even a simple DQN controller substantially improves over PID in background rate stability and signal efficiency (Section 3.1).

(2) We introduce a sequence-based observation model representing the state as a length-K event sequence augmented with physics-informed features. Unlike PID control (Emami et al., 2026), which reacts to a scalar error conflating distinct failure modes (a global luminosity shift versus a local change in the score distribution near threshold), our representation preserves distributional context, enabling the policy to diagnose why the rate has drifted (Hafner et al., 2020). A recurrent encoder compresses the K-event window into a fixed-size state vector (Table 11 and Algorithm 2, Appendix E.3). Algorithm 1 summarizes the full training and deployment pipeline.

(3) We adapt Group Filtered Policy Optimization (GFPO) (Shrivastava et al., 2026), a GRPO variant, to the streaming trigger setting and propose two new variants, GFPO-F and GFPO-FR, that filter rollouts (sampled candidate actions and their evaluated outcomes) by feasibility (accepted background rate) and signal efficiency. Our choice to build on GRPO rather than PPO is motivated by the drift-independent per-chunk bias and affine invariance of GRPO-style updates under streaming non-stationarity, formalized in Appendix B.Both variants Pareto-improve the stability–efficiency trade-off on Monte Carlo (MC) simulated events (Figure 1) and on **real** CMS collision data (Run 283408), transfer from simulation to CMS data with negligible performance loss (Table 2), eliminating the need for gradient-based fine-tuning at deployment, and remain robust to large variations in anomaly-detection score scale (Appendix G).

(4) Our framework extends beyond the LHC. We apply the framework to online anomaly-detection benchmarks and show the combination of sequence-based neural state representations with GFPO-F and GFPO-FR yields consistent improvements under streaming data with distribution shifts, suggesting that our methods generalize well beyond the particle-physics domain (Section 5).

Building on the self-driving-trigger framework of Emami et al. (2026), we demonstrate the *first* end-to-end RL-trained policy using real CMS collision data (Runs 283408 and 283876): one that learns its threshold update rule from streaming experience rather than minimizing a global cost function.

2. Related Work

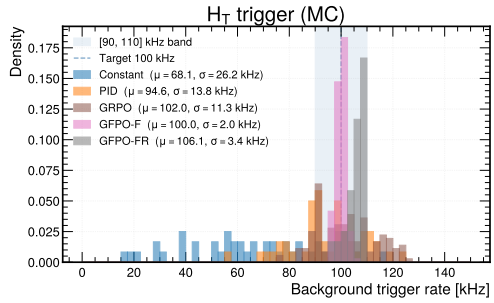
RL for scientific decision making. RL has been applied to self-driving labs (Volk et al., 2023), accelerator tuning (Kaiser et al., 2024), and tokamak regulation (Degrave et al., 2022). LHC trigger control differs in being explicitly *rate-constrained*: the controller must hold background acceptance within a strict tolerance while maximizing signal efficiency, under non-stationary pileup and detector drift.

Anomaly detection under drift. Active anomaly detection typically learns *querying* policies under a label budget (Zha et al., 2020). We instead adapt the *operating threshold* of an autoencoder-based score to satisfy rate constraints under drift. The closest formulation is ADT (Yang et al., 2024), which fixes a pretrained autoencoder and selects binary decision thresholds; we treat trigger thresholds as control variables in a real-time feedback loop. DSPOT (Siffer et al., 2017) sets adaptive thresholds via Extreme Value Theory, with its false-positive level q mapping to our target rate r_B^*; we include it as a baseline, though it has no mechanism for optimizing signal efficiency (Appendix E.1.2).

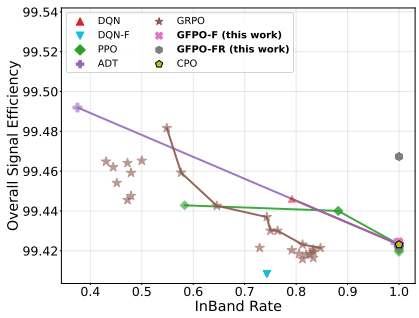
Selective updates. Rejection-style filtering stabilizes RL training, particularly for LLMs (Yu et al., 2025; Yue et al., 2026; Shrivastava et al., 2026). Most directly, GFPO (Shrivastava et al., 2026) samples larger candidate groups and filters by response length and token efficiency to control verbosity. We adapt this skeleton to streaming trigger control, where the filter must enforce rate feasibility under distribution shift, and develop two variants: **GFPO-F** keeps the top-K candidates with smallest rate deviation; **GFPO-FR** selects feasible candidates within an expanded tolerance band, then ranks by signal utility. Appendix A contains an extended account.

3. RL formulation in Adaptive Thresholding

We control a single scalar threshold c_t separately for two trigger paths: an H_T trigger that selects events with high total hadronic activity, $H_T = \sum_{\text{jets}} p_T^{jet}$, and an anomaly-detection (AD) trigger that scores events by autoencoder reconstruction loss (Govorkova et al., 2022). At each step, our agent chooses c_{t+1} to keep the background acceptance rate[1] near a target r_B^* while maximizing signal efficiency on a held-out signal sample. The same formulation applies to both triggers. We model adaptive thresholding as a Markov Decision Process (MDP) $(S, \mathcal{A}, T, R, \gamma)$ (Jayaraman et al., 2024). *State.* $s_t = (h_t, c_t)$, where $h_t \in \mathbb{R}^d$ is a fixed-size summary[2] of recent K events (See State representation design in Appendix E.3), i.e., trigger scores and $c_t = (c_t, \hat{r}_t, \Delta c_{t-1})$ holds the current threshold, the observed rate, and the last threshold change. *Action.* $a_t \in \mathcal{A}$ is a discrete threshold increment. The threshold updates as $c_{t+1} = c_t + a_t$ and is applied to next batch of events. *Reward.* R_t combines three terms designed to (a) maintain the background rate close to a target rate within tolerance; (b) maximizing signal efficiency for two representative signals, $t\bar{t}$ and $h \to 4b$ (Emami et al., 2026); (c) discouraging action update jitter for stability (Le et al., 2016; Aad et al., 2022) (Eq 3). We instantiate this MDP with three policies of increasing structure: DQN (Sec. 3.1), GRPO adapted to streaming control (Sec. 3.2), and feasibility-filtered GFPO (Sec. 4).



(a) Feasibility: Background rates for H_T (MC)



(b) Signal Efficiency: H_T $(t\bar{t})$

Figure 1. Our methods GFPO-F and GFPO-FR archieve highest inband rate and signal efficiency among all established baselines. Each symbol in (b) represents different λ in reward design (Eq 3) with lines connecting the points on the Pareto fronts. Full results on inband rates and signal efficiency are in App. F.2 and D.3.

3.1. Warmup: DQN for single trigger

Setting Following (Emami et al., 2026), we study a single-trigger that accepts an event if a scalar score exceeds a threshold c_t. The controller updates c_t once per chunk to keep the background acceptance rate at $r_B^* = 0.25\%$ - equivaleng to 100 kHz under $r_{kHz} = 400 \cdot r$ - within tolerance $|r_t - r_B^\star| \leq \tau$ with $\tau = 0.025\%$ (i.e., $r_t \in [r^-, r^+] = [90, 110]$ kHz), while maximizing signal efficiency on $t\bar{t}$ and $h \to 4b$ (Cella, 2024). We compare five controllers: (i) a constant menu threshold fixed from initial calibration, (ii) a PID controller updated per chunk on the background rate error (iii) DSPOT (Siffer et al., 2017), which fits a Generalized Pareto Distribution to the background score tail and sets thresholds at false-positive rate $q = r_B^*$ (App. A), (iv) a DQN (Mnih et al., 2015) controller, (v) DQN-F, an ablation in which the DQN is trained on a single calibration chunk and then frozen. Dataset description is in Appendix D.1.

RL formulation The state has three parts (i) a length-K window of recent background event scores together with the pileup context number of primary vertices N_{pv}; (ii) controller context $(c_t, \hat{r}_t, \hat{r}_{t-1}, \Delta c_{t-1})$ (iii) "near-threshold" occupancy features computed in narrow bands around the current cut, which estimate the local sensitivity $\frac{\partial r}{\partial c}$. The occupancy features carry information about distribution shape near the operating point that is unavailable to PID.

We present feature construction for state representations, sequential network architecture comparison (cf. Table 10) and the sequential network learning algorithm (Algorithm 2) in Appendix E.3.

Action and reward. The agent picks a discrete increment $a_t = \Delta c_t \in \mathcal{A}$ and updates

$$c_{t+1} = \text{clip}(c_t + \Delta c_t,\, c_{\min}, c_{\max})\,, \tag{1}$$

$$r_{t+1} = r_B(c_{t+1}) \tag{2}$$

where $r_B(\cdot)$ is the background acceptance and r_i the signal acceptance for $(i \in \{t\bar{t}\,, H \to AA \to 4b\})$. The reward R_t^* (capital R not to be confused with the background rate r) combines rate tracking, signal efficiency, and a smoothness

[1] We use *background rate* and *background acceptance rate* interchangeably throughout.

[2] $d = 19$ in our LHC setting; it is a physics-feature summary of recent K events.

Table 1. **Single-trigger control on MC.** Each RL policy is trained on the first 80% of the MC chunks, then frozen and evaluated on the held-out remaining 20%, on which all reported results are obtained. Background rates in percent units with target r_B^* and tolerance τ. Bold and underline mark the best and second-best per column with standard deviation shown in parentheses. We repeat each method for 3 times and 0 standard deviation denotes ≈ 0 standard deviation. Our methods place top-two for all metrics on both triggers.

Trigger	Method	MAE↓	P95$\lvert e\rvert$ ↓	InBand ↑	$\epsilon_{\text{ov}}^{t\bar{t}}$ ↑	$\epsilon_{\text{ov}}^{h\to 4b}$ ↑	$\epsilon_{\text{in}}^{t\bar{t}}$ ↑	$\epsilon_{\text{in}}^{h\to 4b}$ ↑
H_T **trigger**								
	Constant	0.083(0)	0.191(0)	0.250(0)	98;989(0)	28.062(0)	99.129(0)	29.418(0)
	PID (Emami et al., 2026)	0.029(0)	0.070(0)	0.521(0)	99.388(0)	33.289(0)	99.294(0)	31.444(0)
	DSPOT (Siffer et al., 2017)	0.031(0)	0.088(0)	0.604(0)	99.355(0)	32.612(0)	99.300(0)	31.718(0)
	DQN (Mnih et al., 2015)	0.010(0)	0.023(0)	**1.000(0)**	99.423(0)	34.255(0)	99.423(0)	34.255(0)
	DQN-F	0.018(0.007)	0.038(0.013)	0.743(0.223)	99.409(0.069)	34.141(1.085)	99.417(0.011)	34.184(0.186)
	PPO (Schulman et al., 2017)	0.013(0.005)	0.029(0.010)	0.889(0.192)	99.439(0.028)	34.544(0.5)	99.434(0.019)	34.492(0.411)
	ADT (Yang et al., 2024)	0.010(0)	0.023(0)	**1.000(0)**	99.423(0)	34.255(0)	99.423(0)	34.255(0)
	GRPO (Shao et al., 2024)	0.024(0.001)	0.052(0.006)	0.632(0.067)	99.422(0.21)	34.396(0.318)	99.401(0.054)	34.106(1.218)
	L-GRPO	0.006(0.001)	0.016(0.003)	**1.000(0)**	99.425(0.006)	34.293(0.1)	99.425(0.006)	34.293(0.100)
	CPO (Achiam et al., 2017)	0.010(0)	0.023(0)	**1.000(0)**	99.423(0)	34.255(0)	99.423(0)	34.255(0)
	GFPO-F (Ours)	**0.004(0)**	**0.010(0)**	**1.000(0)**	99.425(0.001)	34.273(0.008)	99.425(0.001)	34.273(0.008)
	GFPO-FR (Ours)	0.016(0)	0.024(0)	**1.000(0)**	**99.468(0)**	**35.057(0)**	**99.468(0)**	**35.057(0)**
AD trigger								
	Constant	0.042(0)	0.131(0)	0.479(0)	94.279(0)	25.404(0)	94.758(0)	25.285(0)
	PID (Emami et al., 2026)	0.020(0)	0.051(0)	0.729(0)	95.233(0)	27.298(0)	94.990(0)	26.287(0)
	DSPOT (Siffer et al., 2017)	0.022(0)	0.061(0)	0.667(0)	94.992(0)	26.767(0)	94.867(0)	26.050(0)
	DQN (Mnih et al., 2015)	0.011(0.001)	0.023(0.001)	0.993(0.012)	95.352(0.017)	27.717(0.033)	95.357(0.020)	27.739(0.055)
	DQN-F	0.016(0.009)	0.030(0.015)	0.847(0.265)	95.543(0.329)	28.185(0.796)	95.474(0.210)	27.948(0.386)
	PPO (Schulman et al., 2017)	0.011(0)	0.022(0)	**1.000(0)**	95.353(0)	27.726(0)	95.353(0)	27.726(0)
	ADT (Yang et al., 2024)	0.011(0)	0.022(0)	**1.000(0)**	95.353(0)	27.726(0)	95.353(0)	27.726(0)
	GRPO (Shao et al., 2024)	0.013(0.001)	0.025(0.001)	0.958(0.055)	95.347(0.027)	27.733(0.071)	95.356(0.011)	27.740(0.041)
	L-GRPO	0.010(0)	0.022(0)	**1.000(0)**	95.369(0.014)	27.764(0.031)	95.369(0.014)	27.764(0.031)
	CPO (Achiam et al., 2017)	0.011(0)	0.022(0)	**1.000(0)**	95.353(0)	27.726(0)	95.353(0)	27.726(0)
	GFPO-F (Ours)	**0.003(0)**	**0.007(0)**	**1.000(0)**	95.442(0)	27.926(0)	95.442(0)	27.926(0)
	GFPO-FR (Ours)	0.019(0)	0.024(0)	0.979(0)	**95.842(0)**	**28.875(0)**	**95.842(0)**	**28.868(0)**

penalty:

$$R_t^{\cdot} = \lambda_1 \underbrace{\begin{cases} 1 - \left(\frac{|r_{t+1} - r_B^\star|}{\tau}\right)^2 & \text{if } |r_{t+1} - r_B^\star| \le \tau \\ -\left(\frac{|r_{t+1} - r_B^\star|}{\tau} - 1\right) & \text{if } |r_{t+1} - r_B^\star| > \tau \end{cases}}_{\text{rate tracking}} \tag{3}$$

$$+ (1-\lambda_1) \underbrace{\left(\alpha\, \epsilon_{t+1}^{t\bar{t}} + (1-\alpha)\, \epsilon_{t+1}^{h\to 4b}\right)}_{\text{signal efficiency}} - \lambda_2 \underbrace{\left(\frac{|\Delta c_{t-1}|}{\Delta c_{\max}}\right)}_{\text{move penalty}}.$$

The notation compresses the trigger index into $\cdot$, which is identical for H_T and AD. $\Delta c_{\max}$ is the maximum threshold update the trigger can act on per step (i.e., safety shield), τ is the tolerance width, and $\lambda_{1,2} > 0$ trade off physics utility against smooth, rate-stable control; we select them via grid search over $\{0.0, 0.25, 0.5, 0.75, 1.0\}^2$. Ablation study and hyperparameter sensitivity are reported in Appendix D.3 (cf. Figure 10). In real LHC operations, trigger thresholds cannot be adjusted arbitrarily. They must satisfy strict operational constraints arising from detector readout bandwidth (Aaboud et al., 2017), computing resources (Collaboration et al., 2020), and detector safety limits (ATLAS Collaboration, 2017). For instance, the ATLAS Level-1 trigger rate must remain below approximately 100 kHz due to readout limitations, and trigger menus are designed such that thresholds and prescale factors maintain rates within these limits (Aaboud et al., 2017). Therefore, we introduce a safety shield that restricts the controller to threshold configurations that satisfy these operational constraints (Alshiekh et al., 2018; Dawood et al., 2025).

Metrics. Let $\hat{r}_t$ be the observed background rate at chunk/step $t \in \{1, \ldots, T\}$ and $e_t = \hat{r}_t - r_B^*$. We report the mean absolute tracking error (MAE) $= T^{-1} \sum_t |e_t|$, and the 95th percentile of the absolute error P95$|e|$, such that 95% of control steps satisfy $|e_t| \le$ P95$|e|$, capturing rare large deviations. We report a tolerance band round the target $r^- \le \hat{r}_t \le r^+$, where typically $r^- = r_B^* - \tau$ and $r^+ = r_B^* + \tau$ for tolerance τ. Define the in-band fraction InBand $= T^{-1} \sum_t \mathbf{1}[r^- \le \hat{r}_t \le r^+]$ For each signal process $i \in \{t\bar{t}, h \to 4b\}$, let $N_{\text{pass}}^i(t)$ be the number of signal events accepted at step t under the applied threshold c_t, and $N_{\text{tot}}^i(t)$ be the number of signal events available at step t. For each signal process $i \in \{t\bar{t}, h \to 4b\}$, let $N_{\text{pass}}^i(t)$ and $N_{\text{tot}}^i(t)$ be the number of accepted and total signal events

Table 2. **Single-trigger control on CMS Run 283408 (zero-shot transfer).** MC-trained policies from Table 1 are frozen and deployed on real collision data without fine-tuning. GFPO-FR (ours) attains the highest signal efficiency on both triggers and the highest InBand on AD; GFPO-F (ours) attains the lowest MAE on both. See Appendix F.4 for results on online training for CMS data.

Trigger	Method	MAE↓	P95$\lvert e\rvert$ ↓	InBand ↑	$\epsilon_{\text{ov}}^{t\bar{t}}$ ↑	$\epsilon_{\text{ov}}^{h\to 4b}$ ↑	$\epsilon_{\text{in}}^{t\bar{t}}$ ↑	$\epsilon_{\text{in}}^{h\to 4b}$ ↑
H_T **trigger**								
	Constant	0.118(0)	0.175(0)	0.041(0)	91.325(0)	22.365(0)	95.842(0)	15.198(0)
	PID (Emami et al., 2026)	0.033(0)	0.085(0)	0.432(0)	97.381(0)	33.347(0)	97.497(0)	**35.242**(0)
	DSPOT (Siffer et al., 2017)	0.163(0)	0.072(0)	0.432(0)	97.211(0)	33.436(0)	96.910(0)	33.278(0)
	DQN (Mnih et al., 2015)	0.0173(0)	0.037(0)	0.865(0)	97.510(0)	33.213(0)	97.447(0)	33.644(0)
	DQN-F	0.023(0)	0.061(0)	0.568(0)	97.487(0)	33.216(0)	97.360(0)	32.672(0)
	PPO (Schulman et al., 2017)	0.016(0.003)	0.034(0.013)	0.838(0.11)	97.491(0.15)	33.222(0.20)	97.358(0.09)	32.673(0.03)
	ADT (Yang et al., 2024)	0.019(0)	0.045(0)	0.743(0)	97.412(0)	33.286(0)	97.337(0)	32.954(0)
	GRPO (Shao et al., 2024)	0.016(0.001)	0.035(0.005)	0.838(0.02)	97.491(0.12)	33.397(0.11)	97.452(0.29)	33.698(0.90)
	L-GRPO	0.016(0.001)	0.032(0.003)	0.865(0.02)	97.486(0.04)	33.320(0.07)	97.406(0.07)	32.940(0.25)
	CPO (Achiam et al., 2017)	0.014(0)	**0.025(0)**	0.905(0)	97.498(0)	33.139(0)	97.392(0)	32.901(0)
	GFPO-F (Ours)	**0.012(0)**	0.042(0)	**0.99(0)**	97.587(0)	33.347(0)	97.508(0)	33.262(0)
	GFPO-FR (Ours)	0.015(0)	0.034(0)	0.95(0)	**97.734(0)**	**33.479(0)**	**97.821(0)**	33.451(0)
AD trigger								
	Constant	0.142(0)	0.195(0)	0.000(0)	62.548(0)	14.399(0)	NA	NA
	PID (Emami et al., 2026)	0.035(0)	0.080(0)	0.405(0)	75.053(0)	39.191(0)	76.210(0)	**44.573**(0)
	DSPOT (Siffer et al., 2017)	0.106(0)	0.087(0)	0.419(0)	74.622(0)	38.297(0)	75.387(0)	41.843(0)
	DQN (Mnih et al., 2015)	0.027(0)	0.060(0.001)	0.527(0.02)	75.093(0.06)	39.930(0.13)	76.013(0.51)	44.178(1.51)
	DQN-F	0.029(0)	0.064(0)	0(0)	62.55(0)	14.40(0)	NA	NA
	PPO (Schulman et al., 2017)	0.027(0)	0.061(0)	0.541(0)	75.059(0)	39.700(0)	75.601(0)	43.153(0)
	ADT (Yang et al., 2024)	0.028(0)	0.067(0)	0.541(0)	75.011(0)	39.510(0)	75.614(0)	42.930(0)
	GRPO (Shao et al., 2024)	0.027(0)	0.061(0.002)	0.541(0.01)	75.059(0.04)	39.700(0.02)	75.601(0.12)	43.153(0.18)
	L-GRPO	0.027(0)	0.063(0.002)	0.541(0.01)	75.059(0.06)	39.703(0.17)	75.597(0.28)	43.154(1.08)
	CPO (Achiam et al., 2017)	0.027(0)	0.061(0)	0.541(0)	75.059(0)	39.703(0)	75.597(0)	43.155(0)
	GFPO-F (Ours)	**0.022(0.001)**	**0.057(0.001)**	**0.689(0.04)**	75.562(0.26)	40.083(0.25)	76.239(0.25)	42.533(1.05)
	GFPO-FR (Ours)	0.025(0)	0.060(0)	0.608(0)	**75.582(0)**	**40.309(0)**	**76.515(0)**	44.477(0)

at step t under threshold c_t^i. We report two signal efficiencies, restricting to in-band steps or to all (i.e., overall) steps: $\epsilon_{\text{in}}^i = \frac{\sum_t \mathbf{1}[r^- \le \hat{r}_t \le r^+]\, N_{\text{pass}}^i(t)}{\sum_t \mathbf{1}[r^- \le \hat{r}_t \le r^+]\, N_{\text{tot}}^i(t)}$ and $\epsilon_{\text{ov}}^i = \frac{\sum_t N_{\text{pass}}^i(t)}{\sum_t N_{\text{tot}}^i(t)}$.

Results Tables 1 and 2 report all controllers on MC and on CMS Run 283408 with no fine-tuning; we discuss the non-learning baselines (constant menu and PID) and DQN here, and defer more variants to Section 3.2 and Section 4. For MC samples, DQN improves InBand from 0.52->1.00 (H_T) and 0.73->0.99 (AD) over PID, with comparable or higher signal efficiency on both $t\bar{t}$ and $h \to 4b$ (Table 1). The frozen ablation DQN-F degrades sharply on H_T as InBand drops to 0.74 and AD signal efficiency falls below the vanilla DQN controller (0.85 vs. 0.99). DQN-F shares the same network and inputs as DQN, with only the online updates removed. The gain thus comes from continued adaptation, not from a richer policy class. A basic DQN, however, does not suffice. Figures 22 and 23 show that the local slope $\frac{\partial r}{\partial c}$ is non-stationary across the run and exhibits heteroskedastic, occasionally high-sensitivity regimes where small threshold changes induce large rate excursions. Two consequences follow. First, non-stationarity favors on-policy learning over off-policy value estimation with stale targets. Second, DQN requires storing thousands of (s, a, r, s') transitions. For hardware implementations in LHC triggers, the memory is scarce and eliminating the replay buffer is a practical advantage. We present qualitative improvements on signal efficiency in Appendix F.1.

3.2. Critic Free Policy Optimization

DQN's gain over PID isolates the value of online adaptation, but two properties of the trigger environment make value-based control fragile in deployment. First, background score distributions shift continuously with pileup and detector conditions (Fig. 5 in Appendix D.1), so replay data quickly becomes off-distribution. Second, the local sensitivity $\frac{\partial r}{\partial c}$ is non-stationary and occasionally large (Figures 22, 23), forcing a single Q-function to extrapolate across rapidly changing local dynamics, a known failure mode for TD bootstrapping (Andrychowicz et al., 2020). PPO (Schulman et al., 2017; Li et al., 2026) avoids replay but inherits the same critic-staleness problem under drift (Pardo et al., 2018). Appendix B formalizes this: under per-chunk drift, PPO's gradient bias scales with the realized drift and compounds across chunks through critic bootstrapping.

We adapt Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Guo et al., 2025) to the streaming control setting, a memory-efficient PPO variant that removes the learned critic and instead uses *within-group reward normal-*

ization. By comparing a small group of candidate threshold updates under the current operating context, GRPO provides a relative learning signal that is less sensitive to reward-scale shifts and regime-dependent sensitivity, yielding more reliable updates under run drift. At each step t, GRPO samples G candidate actions $a_t^{(g)} \sim \pi_{\theta_{\text{old}}}(\cdot|s_t)$, evaluates each on a window W_t of recent events, and forms advantages by within-group normalization:

$$A_t^{(g)} \triangleq \frac{R_t^{(g)} - \mu_t}{\sigma_t + \varepsilon}, \mu_t \triangleq \frac{1}{G}\sum_g R_t^{(g)},\ \sigma_t^2 \triangleq \frac{1}{G}\sum_g (R_t^{(g)} - \mu_t)^2. \tag{4}$$

The reward $R_t^{(g)}$ uses Eq. 3 with window-level aggregates only, so each candidate evaluation costs a single pass over W_t with no learned critic, no inference-time generation, and no auxiliary verifier model. The policy is updated by maximizing

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_t\left[\frac{1}{G}\sum_g A_t^{(g)} \log \pi_\theta(a_t^{(g)} \mid s_t)\right] - \beta\, \mathbb{E}_t[\text{KL}(\pi_\theta \,\|\, \pi_{\theta_{\text{old}}})] \tag{5}$$

As advantages depend only on the *relative* ranking of candidates within each group, GRPO is robust to absolute reward-scale shifts that destabilize critic-based methods.

The constrained-RL response is insufficient. The standard reaction to a constraint-violation problem is to make the constraint penalty adaptive. Indeed, a natural constrained RL approach to reward shaping is to replace the fixed λ_1 (hyperparameter for background rate penalty) in Eq. 3 with an adaptive dual variable λ_t following the standard primal-dual optimization (Achiam et al., 2017; Stooke et al., 2020; Tessler et al., 2019):

$$\lambda_{t+1} = \left[\lambda_t + \alpha_{\text{step}}(|r_t - r_B^\star| - \tau)\right]^+ \tag{6}$$

This yields *Lagrangian GRPO* (L-GRPO). On MC, L-GRPO roughly matches GRPO (same InBand 0.984 on H_T; 0.957 vs. 0.995 on AD). Under distribution shift to CMS Run 283408, however, L-GRPO collapses to InBand 0.541 for AD trigger (Table 2), and enabling test-time training provides no improvement in AD trigger (0.541 in Table 12). The failure mode is structural, not a hyperparameter issue (see Appendix E.1.3 for details). In $\sim 14\%$ (H_T) and $\sim 20\%$ (AD) of MC steps, the sampled group of G candidates contains *zero* rate-feasible actions (Figure 2 and 16). When this happens, every candidate is rate-infeasible, but group-relative normalization still assigns positive advantage to the above-mean candidates, all of which violate the rate budget, and the policy update reinforces a rate violation.

This limitation is not specific to the adaptive penalty. The natural response is to reach for a more principled constrained method, and Constrained Policy Optimization (CPO) (Achiam et al., 2017) is the canonical one: rather than fold the rate constraint into the reward, it enforces the constraint directly, solving a linearized, KL-constrained program at each update that approximately satisfies the expected-cost constraint with a bounded per-iteration violation (Appendix E.1.4). Does moving the constraint out of the reward and into the update fix the failure?

It turns out that it does not. To isolate why, we give CPO the same critic-free, group-relative advantage estimation used by GRPO, so that the value-estimation step (unreliable in the rare-signal, high-imbalance regime) cannot confound the comparison. CPO and GRPO then differ in exactly one place: where the background rate constraint is enforced (See Eqs. 18 and 19). CPO's guarantee is genuinely stronger than L-GRPO's. But the guarantee concerns the *expected* cost under a step in parameter space, not the feasibility of the candidates the policy actually proposes, and here it is the latter that matters.

Consider what happens when the current policy concentrates on the (background) rate-infeasible region, the same condition that produced the all-infeasible groups above. CPO's linearized constraint becomes infeasible, and the algorithm falls back to a recovery step (Achiam et al., 2017, Eq. 14) that descends purely along the cost direction and discards the reward gradient. Under distribution shift the feasible region moves, and this fallback activates on roughly half of the CMS chunks (Table 2). Intuitively, CPO spends half its updates merely climbing back towards feasibility, never improving inband rate.

In our trigger the consequence is sharper still. Because the executed action is screened from the policy (the shield out of band, and the dominant zero-move action in band), these updates cannot change behavior at all. Indeed, test-time training reproduces the frozen-policy rollout to floating-point precision (the CPO rows of Tables 2 and 12 are nearly identical; see Appendix F.5). Like the adaptive penalty, CPO acts on the objective and the parameter update, but *never* on the proposal distribution: the agent is asked to choose among bad options, and the choice it makes is irrelevant to whether good options exist. The fix is to enforce feasibility *before* advantage normalization, not via the reward. We develop this in Section 4.

4. Group-Filtered Policy Optimization

The previous section showed that GRPO and L-GRPO fail under distribution shift because the sampled candidate group frequently contains zero feasible actions. We address this by enforcing feasibility *before* advantage normalization. This is the same algorithmic skeleton as Group Filtered Policy Optimization (GFPO) (Shrivastava et al., 2026), originally proposed for LLM reasoning, where candidates are filtered by response length and token efficiency. We adapt this skeleton to streaming trigger control, where the natural filter is rate feasibility, and develop two variants (GFPO-F,

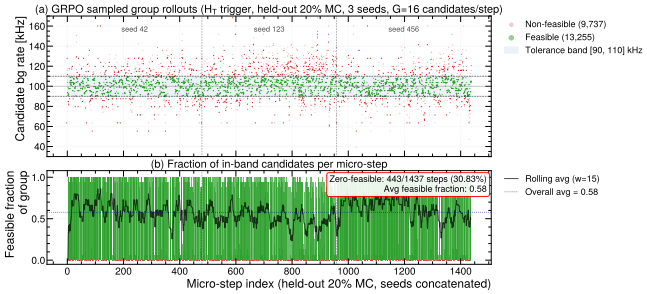


Figure 2. **GRPO's group-feasibility failure on H_T (latter 20% of MC, $G = 16$) averaged across 3 seeds.** (a) Candidate background rates; green/red denote feasible/infeasible w.r.t. the tolerance band. (b) Per-step feasible fraction $f_t = n_{\text{feas}}/G$ (run mean $\langle f_t \rangle = 0.58$). f_t=0 on 30.8% of steps, so the group has no in-band sample and GRPO reinforces the least-infeasible (still out-of-band) action. AD trigger is in Figure 16.

GFPO-FR) tailored to the rate–signal trade-off. The resulting method has two design choices: (i) which candidates to keep, and (ii) how to handle steps where too few feasible candidates exist.[3]

Feasibility set and kept set. At step t, the feasibility set is

$$\mathcal{F}_t(\tau) \triangleq \{g \in [G] : |\hat{e}_t(c_t^{(g)})| \le \tau\}, \tag{7}$$

where $\hat{e}_t = r_{t,W_t} - r_B^*$ is the window-aggregated rate error. GFPO selects a kept set $\mathcal{K}_t \subseteq [G]$ with $|\mathcal{K}_t| = K$ and performs GRPO-style normalization over $\mathcal{K}_t$ with the same reward as Eq. 3:

$$A_t^{(g)} \triangleq \frac{R_t^{(g)} - \mu_t}{\sigma_t + \varepsilon}, \mu_t \triangleq \frac{1}{|\mathcal{K}_t|} \sum_{g \in \mathcal{K}_t} R_t^{(g)}, \tag{8}$$

$$\sigma_t^2 \triangleq \frac{1}{|\mathcal{K}_t|} \sum_{g \in \mathcal{K}_t} \left(R_t^{(g)} - \mu_t\right)^2. \tag{9}$$

and updates the policy with the GRPO objective restricted to $\mathcal{K}_t$ (Eq. 5 with $g \in \mathcal{K}_t$). When $\mathcal{F}_t = \emptyset$ for either variant, the update is *shielded*: skipped entirely to prevent reinforcing infeasible actions.

GFPO-F: rate-error ranking. GFPO-F keeps the K candidates with smallest absolute rate error, regardless of feasibility, biasing learning toward the band boundary and stabilizing training when feasible candidates are scarce:

$$\mathcal{K}_t = \text{TopK}\big(\{-|\hat{e}_t(c_t^{(g)})|\}_{g \in [G]}\big). \tag{10}$$

GFPO-FR: feasible first then ranked by signal efficiency. GFPO-FR strictly enforces feasibility when possible, and then ranks by signal efficiency $\hat{\epsilon}_t^{\text{mix}}(c_t^g) = \alpha \hat{\epsilon}_t^{t\bar{t}} + (1 - \alpha)\hat{\epsilon}_t^{h \to 4b}$:

$$\mathcal{K}_t \triangleq \begin{cases} \text{TopK}\Big(\hat{\epsilon}_t^{\text{mix}}(c_t^{(g)})\Big) \subseteq \mathcal{F}_t(\tau), & \text{if } |\mathcal{F}_t(\tau)| \ge K, \\ \mathcal{F}_t(\tau) \cup \text{TopK}\Big(-|\hat{e}_t(c_t^{(g)})|\Big) \subseteq [G] \setminus \mathcal{F}_t(\tau), & \text{if } 0 < |\mathcal{F}_t(\tau)| < K, \\ \text{TopK}\Big(-|\hat{e}_t(c_t^{(g)})|\Big) \subseteq [G], & \text{if } \mathcal{F}_t(\tau) = \emptyset. \end{cases} \tag{11}$$

When $0 < |\mathcal{F}_t| < K$, the kept set is *padded* with the closest-to-feasible out-of-band candidates. GFPO-FR seeks signal efficiency within the safe region; GFPO-F trades signal for tighter rate control.

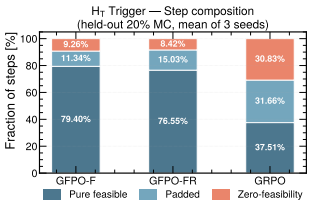


Figure 3. Average Step composition for H_T (K=16, G=64 for GFPO, G=K=16 for GRPO). Each step is classified by $|F_t|$ relative to K: Pure feasible ($|F_t| \ge K$, kept set $\mathcal{K}_t \subseteq \mathcal{F}_t$); Padded ($0 < |F_t| < K$, kept set mixes feasible and out-of-band candidates); Zero feasibility ($\mathcal{F}_t = \emptyset$). On Zero-feasibility steps, GFPO shields the update; GRPO has no shield and performs a degenerate update on a fully infeasible group. The failure mode is characterised in Fig. 2. AD trigger is in Fig. 17.

Diagnostics. Fig.1 shows our methods concentrate background rates inside the $[90, 110]$ kHz band (a) and dominate the in-band background rate –signal efficiency Pareto front across the (λ_1, λ_2) sweep (b). Fig.3 compares both filters with vanilla GRPO on H_T (20% MC).[4] The two filters share an architecture and differ only in their keep rule, which is what induces the *accuracy–coverage* trade-off we observe. GFPO-F keeps the candidates closest to the target rate, collapsing the policy onto a tight, target-centered action distribution; this yields the highest pure-feasible update rate (79.40%), at the cost of a slightly higher zero-feasibility rate (9.26%), since a narrow spread occasionally misses the band when the background drifts. GFPO-FR instead ranks by signal, biasing the policy toward looser, higher-signal cuts; its executed rate deviation is about $4\times$ larger (MAE 0.016 vs. 0.004 on H_T; Tab. 1), reflecting operation nearer the band edge. Crucially, zero-feasibility groups, those with no in-band candidate and hence no learning signal, remain rare for both (8.42% for GFPO-FR vs. 9.26% for GFPO-F): GFPO-FR trades a few perfectly-feasible groups for marginally fewer infeasible ones, buying signal efficiency at a small in-band cost (97.9% vs. 100% on AD).

[3] We use temperature = 1.0 for rollouts generation throughout the paper.

[4] Vanilla GRPO does not pad; We report its *Padded* rate as the fraction of steps with any out-of-band candidate.

5. Beyond HEP: Anomaly Detection Benchmarks

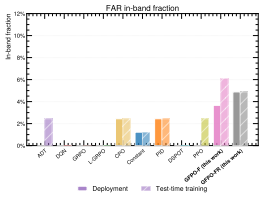
The LHC trigger problem is an instance of a more general one: a streaming controller must hold an aggregate false-positive rate inside a hard tolerance band while retaining rare signal events under drifting score distributions. *Is the GRPO failure mode of Section 4 specific to particle physics, or a generic property of group-sampled policy optimization under hard rate constraints?*

We argue the latter. The failure structure is rate-constrained threshold control under drift, which appears wherever (i) decisions reduce to thresholding a scalar score, (ii) the background distribution drifts, and (iii) a hard, non-tunable rate budget governs allowable positives. The CMS trigger chain instantiates (iii) sharply: a $40\,\mathrm{MHz}$ collision rate to $\sim 2-3\mathrm{kHz}$ for permanent storage, bounded by tape write speed and buffer capacity (Arsuaga-Rios et al., 2021). Bandwidth is zero-sum, collisions are irreversible, and every false positive displaces a signal event, so violations induce detector deadtime. None of (i), (ii), (iii) is physics-specific. The zero-feasibility failure mode arises in any similar setting once the policy drifts off the feasible manifold.

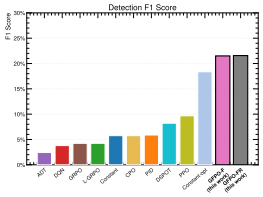
We test this on two benchmarks. **UNSW-NB15** (Moustafa & Slay, 2015) preserves the full rate-constrained protocol and tests whether the failure mode and GFPO's resolution transfer to a non-physics domain. **NAB** (Lavin & Ahmad, 2015) drops the rate constraint and reports standard anomaly-detection metrics, isolating whether our components (sequence-based state, adaptive thresholding) are competitive on more standard problems.

5.1. Experimental Setup

Common protocol. We fix every component except the dataset and context variable: an autoencoder trained on normal-class data provides the scalar score $s(x_i)$; thresholds are updated once per temporal chunk; a target false-positive rate $r^\star$ and tolerance τ with $\tau/r^\star \approx 10\%$ are pre-registered to match LHC stringency. We report InBand fraction, MAE, $\mathrm{P95}|e|$, and signal true-positive rate (TPR) using metrics under Section 3.1. Physics-specific features (pileup N_{PV}) are replaced by domain-appropriate context variables. **UNSW-NB15.** UNSW-NB15 preserves the full rate-constrained protocol, providing a sim-to-real transfer test analogous to MC $\rightarrow$ CMS: we train on Period 1 and deploy the frozen policy on Period 2. Normal traffic is the background. *Exploits* serve as the easy signal and *Backdoors* as the hard signal (analogous to $t\bar{t}$ and $h \rightarrow 4b$). The context variable is the per-window connection rate, which tracks operating-regime shifts analogously to pileup. In the LHC setting we call this the background rate; in UNSW-NB15 the corresponding quantity is the *false alert rate* (FAR), the per-chunk fraction of benign records flagged as anomalies. Preprocessing details and training setup are in App. H.1. **NAB.** NAB drops the rate constraint and reports standard detection metrics (precision, recall, F1). The agent optimizes a pure detection-quality reward (App. H.2). The purpose is to isolate our sequence-based state and adaptive thresholding from feasibility-filtered policy optimization: if our gains came solely from the latter, NAB results would be flat. We define *Constant-opt* as the oracle static threshold maximizing F1 in hindsight and methods underperforming it do not meaningfully leverage adaptivity.



(a) **GFPO dominates rate control on UNSW-NB15.** Period 1-trained policies deployed frozen on Period 2 (solid) or with online training (hatched).



(b) **GFPO improves detection quality on NAB even without a rate constraint.** F1 across methods. GFPO-F and GFPO-FR reach the highest F1 (0.215, 0.216), a 3% gain over the oracle Constant-opt (F1=0.184).

Figure 4. Anomaly detection results on UNSW-NB15 and NAB.

5.2. Experimental Results

UNSW-NB15 results. Both predictions hold (Fig. 4a). The zero-feasibility failure mode transfers: GRPO, L-GRPO, DQN, and DSPOT collapse to 0% in-band FAR (False Alert Rate), with no recovery under online training. GFPO's resolution transfers: GFPO-F and GFPO-FR reach 3.9% and 5.1% in-band frozen, $1-2\times$ the best classical baseline (2.5% PID). Signal TPR on Backdoors, the hard class, is dominated by GFPO-F/FR (App. H.1). **NAB without rate regulation.** Does GFPO's advantage depend on an active rate constraint? NAB removes the constraint. GFPO-F and GFPO-FR attain the highest F1 (0.215, 0.216; Fig. 4b), roughly a 3% gain over the oracle Constant-opt. DQN, GRPO, and L-GRPO underperform Constant-opt despite training on the same reward. The contribution is structural: group-based policy optimization with feasibility filtering, not RL itself, drives the gain. Precision and recall are dominated by GFPO-F/FR (App. H.2).

Impact Statement

This paper advances the application of machine learning, specifically reinforcement learning, to trigger design in high-energy physics experiments at the LHC. Trigger systems govern which collision events are recorded for downstream analysis, and their efficiency directly shapes the sensitivity of searches for rare processes such as $h \to 4b$. By adaptively regulating thresholds to balance signal retention against bandwidth constraints, our framework has the potential to accelerate discovery in underexplored regions of phase space and to reduce the manual effort required to retune triggers as detector and beam conditions evolve.

We do not foresee direct negative societal consequences. We do note, however, that deploying learned controllers inside scientific instruments raises concerns around reproducibility, interpretability, and robustness under distribution shift (e.g. sim-to-real transfer from Monte Carlo to CMS data) that the science community will need to address carefully to preserve the integrity of downstream physics results.

We also wish to scope what our framework does and does not achieve for physics. It adjusts a single scalar threshold per representative trigger to regulate background rate; because all signal processes share this threshold, their achievable efficiencies are coupled. The optimum for $t\bar{t}$ generally differs from that for $h \to 4b$, yet the controller must pick a single operating point. In our experiments, $t\bar{t}$ consistently achieves $> 95\%$ efficiency while $h \to 4b$ is limited to $\sim$ 28–35% under the same threshold (Hall & Konstantinidis, 2025). The mixing parameter α in the reward offers *partial* control over this trade-off but cannot overcome the *fundamental* constraint that a single cut cannot simultaneously optimize across signals with differing detector-level distributions. Extending to multi-dimensional trigger menus, where each signal class has its own threshold or where multiple paths are combined via logic gates (Boggia et al., 2025), would decouple these efficiency ceilings, and is a natural direction for future work.

6. Acknowledgements

This work is supported by the University of Chicago Joint Task Force Initiative (JTFI) – Partnerships for Emerging Technologies Seed Funding. A.G, J.N, and N.T are supported by FermiForward Discovery Group, LLC under Contract No. 89243024CSC000002 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics. Z.D and Y.C are supported by the U.S. National Science Foundation (NSF) under grants 2037026, 2313131, 2332475 and 2543755.

References

Aaboud, M., Aad, G., Abbott, B., Abdallah, J., Abeloos, B., Aben, R., AbouZeid, O., Abraham, N., Abramowicz, H., Abreu, H., et al. Performance of the atlas trigger system in 2015. *The European Physical Journal C*, 77(5):1–53, 2017.

Aaboud, M., Aad, G., Abbott, B., Abeloos, B., Abhayasinghe, D., Abidi, S., AbouZeid, O., Abraham, N., Abramowicz, H., Abreu, H., et al. Search for the higgs boson produced in association with a vector boson and decaying into two spin-zero particles in the $h \to aa \to 4b$ channel in pp collisions at $\sqrt{s} = 13$ tev with the atlas detector. *Journal of High Energy Physics*, 2018(10):1–48, 2018.

Aad, G., Abbott, B., Abbott, D. C., Abud, A. A., Abeling, K., Abhayasinghe, D. K., Abidi, S. H., AbouZeid, O., Abraham, N., Abramowicz, H., et al. Performance of the atlas level-1 topological trigger in run 2. *The European Physical Journal C*, 82(1):7, 2022.

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. Pmlr, 2017.

Albrecht, J., Bozianu, L., Calefice, L., Cella, S., Toapaxi, C. C., Doglioni, C., Gligorov, V., Gooding, J. A., Iversen, K. E., Inkaew, P., et al. Summary of the trigger systems of the large hadron collider experiments alice, atlas, cms and lhcb. *Journal of Physics G: Nuclear and Particle Physics*, 52(3):030501, 2025.

Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., and Topcu, U. Safe reinforcement learning via shielding. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.

Andrychowicz, M., Raichuk, A., Stańczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., et al. What matters in on-policy reinforcement learning? a large-scale empirical study. *arXiv preprint arXiv:2006.05990*, 2020.

Arsuaga-Rios, M., Bahyl, V., Batalha, M., Caffy, C., Cano, E., Capitoni, N., Contescu, C., Davis, M., Alvarez, D. F., Guenther, J., et al. Lhc data storage: Preparing for the challenges of run-3. In *EPJ Web of Conferences*, volume 251, pp. 02023. EDP Sciences, 2021.

ATLAS Collaboration. Technical Design Report for the Phase-II Upgrade of the ATLAS Tile Calorimeter. Technical report, CERN, Geneva, 2017. URL https://cds.cern.ch/record/2285583.

Bertelsen, H., Montoya, G. C., Deviveiros, P.-O., Eifert, T., Galster, G., Glatzer, J., Haas, S., Marzin, A., Oliveira,

M. S., Pauly, T., et al. Operation of the upgraded atlas central trigger processor during the lhc run 2. *Journal of Instrumentation*, 11(02):C02020–C02020, 2016.

Boggia, L., Cocha, C., Giasemis, F., Hansen, J., Inkaew, P., Iversen, K. E., Jawahar, P., Monteagudo, H. P., Olocco, M., Astrand, S., et al. Review of machine learning for real-time analysis at the large hadron collider experiments alice, atlas, cms and lhcb. *arXiv preprint arXiv:2506.14578*, 2025.

Bradu, B., Blanco Viñuela, E., Marti, R., and Tilaro, F. Automatic pid performance monitoring applied to lhc cryogenics. In *16th International Conference on Accelerator and Large Experimental Physics Control Systems*, pp. WEAPL02, 2018. doi: 10.18429/JACoW-ICALEPCS2017-WEAPL02.

Cella, S. The atlas run-3 trigger menu. Technical report, ATL-COM-DAQ-2024-077, 2024.

CERN Open Data Portal. About cms. https://opendata.cern.ch/docs/about-cms, 2026. Accessed: 2026-01-03.

Cerri, O., Nguyen, T. Q., Pierini, M., Spiropulu, M., and Vlimant, J.-R. Variational autoencoders for new physics mining at the large hadron collider. *Journal of High Energy Physics*, 2019(5):1–29, 2019.

CMS Collaboration. The CMS trigger system. *JINST*, 12: P01020, 2017. doi: 10.1088/1748-0221/12/01/P01020.

CMS Collaboration. 2024 Data Collected with AXOL1TL Anomaly Detection at the CMS Level-1 Trigger. Technical report, CERN, 2024a. URL https://cds.cern.ch/record/2904695.

CMS Collaboration. About CMS Open Data, 2024b. URL https://opendata.cern.ch/docs/about-cms. Accessed: March 9, 2025.

CMS Collaboration. Simulated dataset tttohadronic_tunecp5_13tev-powheg-pythia8 in miniaodsim format for 2016 collision data. CERN Open Data Portal, 2024c. URL https://opendata.cern.ch/record/67840. Data recorded in 2016 and published in 2024.

Collaboration, A. et al. Operation of the atlas trigger system in run 2. *Journal of Instrumentation*, 15(10):P10004–P10004, 2020.

collaboration, C. The cms high level trigger. *The European Physical Journal C-Particles and Fields*, 46(3):605–667, 2006.

collaboration, C. et al. Description and performance of track and primary-vertex reconstruction with the cms tracker. *Journal of Instrumentation*, 9(10):P10009–P10009, 2014.

Daw, A., Karpatne, A., Watkins, W. D., Read, J. S., and Kumar, V. Physics-guided neural networks (pgnn): An application in lake temperature modeling. In *Knowledge guided machine learning*, pp. 353–372. Chapman and Hall/CRC, 2022.

Dawood, M., Shokry, A., and Bennewitz, M. A dynamic safety shield for safe and efficient reinforcement learning of navigation tasks. In *7th Annual Learning for Dynamics & Control Conference*, pp. 686–697. PMLR, 2025.

de Florian, D., Grojean, C., Maltoni, F., Mariotti, C., Nikitenko, A., Pieri, M., Savard, P., Schumacher, M., Aggleton, R., Ahmad, M., et al. *CERN Yellow Reports: Monographs, Vol 2 (2017): Handbook of LHC Higgs cross sections: 4. Deciphering the nature of the Higgs sector*, volume 2. Cern, 2017.

Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de Las Casas, D., et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

Dey, R. and Salem, F. M. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pp. 1597–1600. IEEE, 2017.

Emami, S., Tosciri, C., Salvi, G., Ding, Z., Chen, Y., Gandrakota, A., Herwig, C., Miller, D. W., Ngadiuba, J., and Tran, N. Towards a self-driving trigger at the LHC: Adaptive response in real time. *Machine Learning: Science and Technology*, 2026. doi: 10.1088/2632-2153/ae631f. URL https://iopscience.iop.org/article/10.1088/2632-2153/ae631f. Accepted manuscript.

Evans, L. and Bryant, P. Lhc machine. *Journal of instrumentation*, 3(08):S08001–S08001, 2008.

Evans, T., Fitzpatrick, C., and Horswill, J. An automated bandwidth division for the lhcb upgrade trigger. *Computing and Software for Big Science*, 9(1):7, 2025.

Gandrakota, A. Real-time anomaly detection at the l1 trigger of cms experiment. *arXiv preprint arXiv:2411.19506*, 2024.

Gavrikov, A., García Pardiñas, J., and Garfagnini, A. Dinamo: Dynamic and interpretable anomaly monitoring for large-scale particle physics experiments. *Machine Learning: Science and Technology*, 6(3):035050, 2025.

Govorkova, E., Puljak, E., Aarrestad, T., James, T., Loncar, V., Pierini, M., Pol, A. A., Ghielmetti, N., Graczyk, M., Summers, S., et al. Autoencoders on field-programmable gate arrays for real-time, unsupervised new physics detection at 40 mhz at the large hadron collider. *Nature Machine Intelligence*, 4(2):154–161, 2022.

Graves, A. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Hall, N. C. and Konstantinidis, N. Decade: Decorrelated anomaly detection triggers to enhance the low-mass discovery potential of the lhc. *arXiv preprint arXiv:2508.10224*, 2025.

Hausknecht, M. J. and Stone, P. Deep recurrent q-learning for partially observable mdps. In *AAAI fall symposia*, volume 45, pp. 141, 2015.

Jayaraman, P., Desman, J., Sabounchi, M., Nadkarni, G. N., and Sakhuja, A. A primer on reinforcement learning in medicine for clinicians. *NPJ digital medicine*, 7(1):337, 2024.

Jia, X., Willard, J., Karpatne, A., Read, J., Zwart, J., Steinbach, M., and Kumar, V. Physics guided rnns for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM international conference on data mining*, pp. 558–566. SIAM, 2019.

Kaiser, J., Xu, C., Eichler, A., Santamaria Garcia, A., Stein, O., Bründermann, E., Kuropka, W., Dinter, H., Mayet, F., Vinatier, T., et al. Reinforcement learning-trained optimisers and bayesian optimisation for online particle accelerator tuning. *Scientific reports*, 14(1):15733, 2024.

Kazemi, S. M., Goel, R., Eghbali, S., Ramanan, J., Sahota, J., Thakur, S., Wu, S., Smyth, C., Poupart, P., and Brubaker, M. Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*, 2019.

Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2):209–232, 2002.

Khanda, R., Baqar, M., Chakrabarti, S., and Changdar, S. Extending group relative policy optimization to continuous control: A theoretical framework for robotic reinforcement learning. *arXiv preprint arXiv:2507.19555*, 2025.

Lavin, A. and Ahmad, S. Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, pp. 38–44. IEEE, 2015.

Le, H., Kang, A., Yue, Y., and Carr, P. Smooth imitation learning for online sequence prediction. In *International Conference on Machine Learning*, pp. 680–688. PMLR, 2016.

Li, J., Zhou, P., Meng, R., Vadera, M. P., Li, L., and Li, Y. Turn-ppo: Turn-level advantage estimation with ppo for improved multi-turn rl in agentic llms. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 6227–6243, 2026.

Lim, B., Arık, S. Ö., Loeff, N., and Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International journal of forecasting*, 37(4):1748–1764, 2021.

Lutz, I. D., Wang, S., Norn, C., Courbet, A., Borst, A. J., Zhao, Y. T., Dosey, A., Cao, L., Xu, J., Leaf, E. M., et al. Top-down design of protein architectures with reinforcement learning. *Science*, 380(6642):266–273, 2023.

Mahesh, C., Dona, K., Miller, D. W., and Chen, Y. Towards an interpretable data-driven trigger system for high-throughput physics facilities. *arXiv preprint arXiv:2104.06622*, 2021.

Matchev, K. T. and Shyamsundar, P. Thickbrick: optimal event selection and categorization in high energy physics. part i. signal discovery. *Journal of High Energy Physics*, 2021(3):291, 2021.

Mei, J., Chung, W., Thomas, V., Dai, B., Szepesvari, C., and Schuurmans, D. The role of baselines in policy gradient optimization. *Advances in Neural Information Processing Systems*, 35:17818–17830, 2022.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Moustafa, N. and Slay, J. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 Military Communications and*

Information Systems Conference (MilCIS), pp. 1–6. IEEE, 2015.

Ngairangbam, V. S., Rozwoda, B., Sakurai, K., and Spannowsky, M. Enhancing anomaly detection with topology-aware autoencoders. *Machine Learning: Science and Technology*, 6(4):045051, 2025.

Oh, M.-h. and Iyengar, G. Sequential anomaly detection using inverse reinforcement learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & data mining*, pp. 1480–1490, 2019.

Pardo, F., Tavakoli, A., Levdik, V., and Kormushev, P. Time limits in reinforcement learning. In *International Conference on Machine Learning*, pp. 4045–4054. PMLR, 2018.

Racz, A. Trigger throttling system for cms daq. Technical report, CERN, 2000. URL https://cds.cern.ch/record/479701.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Shrivastava, V., Awadallah, A. H., Balachandran, V., Garg, S., Behl, H., and Papailiopoulos, D. Sample more to think less: Group filtered policy optimization for concise reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=UKOqoULbZS.

Siffer, A., Fouque, P.-A., Termier, A., and Largouet, C. Anomaly detection in streams with extreme value theory. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1067–1075, 2017.

Stooke, A., Achiam, J., and Abbeel, P. Responsive safety in reinforcement learning by pid lagrangian methods. In *International conference on machine learning*, pp. 9133–9143. PMLR, 2020.

Strehl, A. L. and Littman, M. L. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

Tang, Y., Fan, J., Li, X., Ma, J., Qi, M., Yu, C., and Gao, W. Physics-informed recurrent neural network for time dynamics in optical resonances. *Nature computational science*, 2(3):169–178, 2022.

Tessler, C., Mankowitz, D. J., and Mannor, S. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkfrvsA9FX.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Volk, A. A., Epps, R. W., Yonemoto, D. T., Masters, B. S., Castellano, F. N., Reyes, K. G., and Abolhasani, M. Alphaflow: autonomous discovery and optimization of multi-step chemistry using a self-driven fluidic lab guided by reinforcement learning. *Nature Communications*, 14 (1):1403, 2023.

Xu, J., Wu, H., Wang, J., and Long, M. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=LzQQ89U1qm_.

Yang, X. Adt: Agent-based dynamic thresholding for time series anomaly detection, 2026. URL https://github.com/XueYang0130/ADT. GitHub repository, accessed 2026-01-02.

Yang, X., Howley, E., and Schukat, M. Adt: Time series anomaly detection for cyber-physical systems via deep reinforcement learning. *Computers & Security*, 141: 103825, 2024.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Yue, Y., Song, S., and Huang, G. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=4OsgYD7em5.

Zha, D., Lai, K.-H., Wan, M., and Hu, X. Meta-aad: Active anomaly detection with deep reinforcement learning. In *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 771–780. IEEE, 2020.

Zhou, H., Ye, K., Xu, E., Zhu, J., Yang, Y., Gong, S., and Shi, C. Demystifying group relative policy optimization: Its policy gradient is a u-statistic. *arXiv preprint arXiv:2603.01162*, 2026.

Zipper, N. and collaboration, C. Testing a neural network for anomaly detection in the cms global trigger test crate during run 3. *Journal of Instrumentation*, 19(03):C03029, 2024.

A. Extended Related Work

RL for scientific decision making. RL is increasingly used across scientific domains, including experiment planning in self-driving laboratories (Volk et al., 2023), online control of particle accelerators (Kaiser et al., 2024), and tokamak plasma regulation (Degrave et al., 2022). These settings share the structural challenges of streaming data and non-stationary dynamics, but they target a desired physical state. Our objective is different: *rate-constrained operation*, maintaining a fixed background acceptance rate (e.g., 0.25%) within tolerance while maximizing physics utility (signal efficiency). Search-heavy approaches such as MCTS-based design (Lutz et al., 2023) are also a poor fit at our deployment latency, which requires a single forward pass per threshold update; training can nonetheless reuse logged or streaming experience via off-policy learning.

The closest GRPO-adjacent work is Khanda et al. (2025), which extends GRPO to continuous control via trajectory-based clustering and state-aware advantage estimation. Our setting differs in two ways: the action is one-dimensional (a threshold update) rather than a robotics-style continuous trajectory, and the deployment is compute- and latency-constrained, so trajectory clustering (k-clustering in particular) is incompatible with the per-chunk update budget. The rate constraint itself motivates a different family of methods: constrained policy optimization via Lagrangian relaxation (Stooke et al., 2020; Tessler et al., 2019) replaces fixed penalty weights with adaptive dual variables. We include Lagrangian-GRPO (L-GRPO) as a direct baseline (Section 3.2, Appendix E.1.3) and show that dual adaptation alone cannot resolve the structural sampling failure under distribution shift, motivating our feasibility-filtering approach (GFPO-F and GFPO-FR).

Anomaly detection under drift and threshold control. Two families of work are most directly comparable. Active anomaly detection (Zha et al., 2020) learns acquisition policies under a label-query budget; we instead adapt the operating threshold of an autoencoder-based detector online, so the agent's interface is a continuous threshold update rather than a discrete query decision. The closest sequential-decision formulation is ADT (Yang et al., 2024), but its deployment assumptions are too permissive for trigger operation: ADT assumes anomaly scores in $[0, 1]$ with known global extrema and pre-normalizes scores before inference. We train the autoencoder on *background-only* data, cannot bound score scale *a priori* (Figure 5), and must adapt under non-stationary shifts where any fixed threshold fails. We include ADT as a baseline with the modifications described in Appendix E.1.

We deliberately exclude the Anomaly Transformer (Xu et al., 2022). Its core assumption is that anomalies in a time series

can be detected because their self-attention concentrates on nearby time steps, while normal points attend broadly across the whole series, i.e., the Association Discrepancy criterion. This only works when neighboring points are informative about each other, i.e. when the series is regularly sampled and autocorrelated. LHC bunch crossings are not such a series: each collision is an independent event (Evans & Bryant, 2008), so the previous event tells us nothing about the current one and "neighbors" in the input window have no special status. The Association Discrepancy criterion therefore has no signal to exploit on this data. Beyond this mismatch, our task is rate regulation over streaming windows rather than point-wise classification of individual events, which makes threshold-based RL methods the appropriate comparison class.

A complementary non-learning baseline is Extreme Value Theory: DSPOT (Siffer et al., 2017) sets streaming thresholds by fitting a Generalized Pareto distribution to the tail of the score distribution, with false-positive level q as the sole parameter. Identifying q with the target background rate r_B^* maps DSPOT directly onto our setting, and its online tail re-estimation gives a principled statistical response to score drift without any learned policy. However, DSPOT is *rate-agnostic*: it tracks observed score quantiles rather than closing the loop on the measured background acceptance rate, and offers no mechanism for signal-efficiency optimization. We include DSPOT by using its adaptive threshold z_q directly as the per-chunk trigger cut (Appendix E.1); it is competitive with PID in some regimes but falls short of RL-based controllers in both rate stability and signal efficiency under pileup drift.

Data filtering and selective updates in RL. A line of recent RL methods for LLMs uses rejection sampling and selective updates to stabilize training. GFPO (Shrivastava et al., 2026) oversamples candidates per prompt and updates only on a filtered subset; DAPO (Yu et al., 2025) discards prompts whose rollouts carry trivial learning signal; analyses by Yue et al. (2026) confirm that training on a selected subset of trajectories can outperform updates on the full sample. In all of these cases, filtering is task-agnostic: the criterion is signal quality (informative reward, non-trivial rollouts), not problem-side feasibility.

We adapt GFPO to the streaming trigger setting and replace its task-agnostic filter with constraint-aware selection rules. GFPO-F samples G candidate threshold updates per control step and retains the top-K by smallest rate deviation $|r - r_B^*|$. GFPO-FR applies a feasibility-first rule: it selects candidates inside an expanded tolerance band, ranks the feasible set by signal utility (a weighted mixture of signal efficiencies), and pads with closest-to-feasible candidates when the feasible set is too small. The shared move across both variants is that the filter becomes a soft realization of the rate constraint rather than a generic data-quality heuristic for LLM training.

Physics-informed sequential state representations. A related line of work improves robustness in non-stationary physical systems by embedding domain structure into sequence models (Hausknecht & Stone, 2015; Jia et al., 2019; Daw et al., 2022; Tang et al., 2022). For example, Jia et al. (2019) propose physics-guided recurrent architectures that inject physical priors into hidden-state evolution for more data-efficient and physically consistent time-series prediction. Similarly, Tang et al. (2022) develop a physics-guided, physics-explainable recurrent model for optical resonance dynamics using multi-fidelity training with synthetic physics-model data and a smaller set of real measurements. PGNN-style approaches also integrate mechanistic knowledge via simulator-augmented inputs and physics-consistency regularization in supervised forecasting (Daw et al., 2022). In contrast, we study *online threshold control* under explicit rate constraints. Rather than forecasting system states under a known (or approximate) dynamical law, our policy operates a real-time trigger where the action is an operating threshold that must satisfy a hard background-rate tolerance under pileup-driven drift. We therefore construct a physics- and control-informed sliding-window observation: a length-K sequence of recent event summaries augmented with threshold-relative features (distance-to-cut, pass/near-cut occupancy) and chunk-level control context (rate error, drift, last action, feasibility). This representation exposes the local "mass near the cut" that governs rate sensitivity (Figure 23), enabling a sequence policy to anticipate distribution shifts and retune thresholds online without assuming a closed-form governing equation for the evolving score distributions.

Choice of policy optimizer. PPO with a learned critic and GAE-bootstrapped advantages (Schulman et al., 2017; 2016) is the standard choice for continuous-control policy optimization under stationary state-action distributions. Our streaming LHC trigger setting violates this assumption: the background score distribution and the local rate–threshold sensitivity drift across chunks, so value targets fit on past chunks are miscalibrated on the current chunk, and multi-epoch reuse compounds the error. GRPO (Shao et al., 2024) removes the source of this compounding by replacing the critic with a within-chunk group baseline over G on-policy rollouts, which is recomputed at every update. Because our control loop already evaluates multiple candidate threshold updates per chunk, GRPO's algorithmic structure aligns naturally with the deployment loop. Appendix B formalizes the comparison: the critic carries stale value estimates across chunks, giving a PPO gradient bias of $\Theta(\delta_k/((1-\gamma)(1-\gamma\lambda)))$ in the realized drift, while the per-chunk group baseline yields a GRPO bias of $\mathcal{O}(G^{-1})$ that

is invariant under per-chunk affine rescaling of returns. The same robustness extends to the rate-constrained Lagrangian formulation.

B. Theoretical Justification

Overview. Here, we formalize the claim that GRPO-style updates remain calibrated under the streaming, non-stationary distribution shifts characteristic of LHC trigger control, while PPO with a learned critic acquires a drift-dependent bias that compounds across chunks. Lemma B.1 bounds the PPO advantage error in terms of critic mismatch; Lemma B.2 establishes within-chunk unbiasedness of the GRPO gradient at rate $\mathcal{O}(G^{-1})$; Lemma B.3 shows that the GRPO learning signal is invariant under per-chunk affine rescaling of returns; Corollary B.4 combines these into per-chunk gradient bias bounds for the two estimators.

Streaming non-stationary setting. We model the LHC trigger as a sequence of chunks $k = 1, 2, \ldots$, each inducing an MDP $\mathcal{M}_k = (\mathcal{S}, \mathcal{A}, P_k, r_k, \gamma)$ with chunk-specific value V_k^π, advantage A_k^π, and policy objective $J_k(\theta) = \mathbb{E}_{\pi_\theta, \mathcal{M}_k}[\sum_{t\geq 0} \gamma^t r_t]$; we assume $r_k \in [0, R_{\max}]$. Successive chunks are not identical: both the reward and the transition kernel shift as luminosity, pileup, and detector conditions evolve (Figs. 5 and 6). We summarize the gap between $\mathcal{M}_{k-1}$ and $\mathcal{M}_k$ by the single scalar

$$\delta_k \triangleq \underbrace{\sup_{s,a} \left| r_k(s,a) - r_{k-1}(s,a) \right|}_{\text{reward drift}} + \underbrace{\tfrac{\gamma R_{\max}}{1-\gamma} \sup_{s,a} \left\| P_k(\cdot|s,a) - P_{k-1}(\cdot|s,a) \right\|_{\mathrm{TV}}}_{\text{transition drift}}, \tag{12}$$

chosen so that one Bellman-difference step controls the induced value-function gap. The prefactor $\gamma R_{\max}/(1-\gamma)$ equals an upper bound on the span of any value function with $r \in [0, R_{\max}]$, so a TV-norm perturbation in the kernel translates into a value perturbation of matching units; the two terms of Equation 12 then sum cleanly into the simulation-lemma bound (Kearns & Singh 2002, Lemma 4; Strehl & Littman 2008, Lemma 1): for any policy π,

$$\left\| V_{k-1}^\pi - V_k^\pi \right\|_\infty \leq \frac{\delta_k}{1-\gamma}. \tag{13}$$

We sketch the argument for self-containedness. Let T_k^π denote the chunk-k policy-evaluation Bellman operator, $(T_k^\pi V)(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[r_k(s,a) + \gamma\, \mathbb{E}_{s'\sim P_k(\cdot|s,a)} V(s')]$, of which V_k^π is the unique fixed point. Adding and subtracting $T_{k-1}^\pi V_k^\pi$ and applying the triangle inequality,

$$\begin{aligned} \|V_{k-1}^\pi - V_k^\pi\|_\infty &= \|T_{k-1}^\pi V_{k-1}^\pi - T_k^\pi V_k^\pi\|_\infty \\ &\leq \underbrace{\|T_{k-1}^\pi V_{k-1}^\pi - T_{k-1}^\pi V_k^\pi\|_\infty}_{\leq \gamma \|V_{k-1}^\pi - V_k^\pi\|_\infty \ (\gamma\text{-contraction})} \\ &\quad + \underbrace{\|T_{k-1}^\pi V_k^\pi - T_k^\pi V_k^\pi\|_\infty}_{\leq \delta_k}. \end{aligned}$$

The bound on the second term uses $\sup_{s,a}|r_{k-1} - r_k|$ on the reward channel and the standard inequality $|\mathbb{E}_P V_k^\pi - \mathbb{E}_Q V_k^\pi| \le \mathrm{span}(V_k^\pi)\,\|P-Q\|_{\mathrm{TV}}$ together with $\mathrm{span}(V_k^\pi) \le R_{\max}/(1-\gamma)$ on the transition channel, recovering exactly δ_k. Solving $\|V_{k-1}^\pi - V_k^\pi\|_\infty \le \gamma\|V_{k-1}^\pi - V_k^\pi\|_\infty + \delta_k$ for the left-hand side yields Equation 13.

This is not a worst-case abstraction. Figs. 5 and 6 show that the background H_T and AD-score distributions drift monotonically across a run in both Monte Carlo and CMS Open Data, exhibiting the affine location-and-scale shifts invoked in Lemma B.3.

Critic drift biases the PPO advantage. Let V_ϕ be the critic fit on chunks $\{1,\dots,k-1\}$, and let $\hat{A}_t^{\mathrm{GAE}} = \sum_{l\ge 0}(\gamma\lambda)^l\delta_{t+l}$, where $\lambda \in [0,1]$ is the GAE parameter, with $\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$. The following bound follows directly from the GAE estimator of Schulman et al. (2016).

Lemma B.1 (Critic-induced advantage bias). *For any π and any chunk k,*

$$\left|\mathbb{E}\left[\hat{A}_t^{\mathrm{GAE}} \,\middle|\, s_t, a_t\right] - A_k^\pi(s_t,a_t)\right| \;\le\; \frac{1+\gamma}{1-\gamma\lambda}\,\|V_\phi - V_k^\pi\|_\infty.$$

Proof. Let $e(s) = V_\phi(s) - V_k^\pi(s)$. For the chunk-$k$ MDP,

$$\mathbb{E}[\delta_t \mid s_t, a_t] = A_k^\pi(s_t,a_t) + \gamma\mathbb{E}[e(s_{t+1}) \mid s_t, a_t] - e(s_t).$$

For $l \ge 1$, since $a_{t+l} \sim \pi(\cdot \mid s_{t+l})$,

$$\mathbb{E}[A_k^\pi(s_{t+l}, a_{t+l}) \mid s_t, a_t] = 0.$$

Therefore,

$$\mathbb{E}[\hat{A}_t^{\mathrm{GAE}} \mid s_t, a_t] - A_k^\pi(s_t,a_t) = \sum_{l\ge 0}(\gamma\lambda)^l\mathbb{E}[\gamma e(s_{t+l+1}) - e(s_{t+l}) \mid s_t, a_t]\,.$$

Taking absolute values and using $\|e\|_\infty = \|V_\phi - V_k^\pi\|_\infty$ gives

$$\left|\mathbb{E}[\hat{A}_t^{\mathrm{GAE}} \mid s_t, a_t] - A_k^\pi(s_t,a_t)\right| \le \sum_{l\ge 0}(\gamma\lambda)^l(1+\gamma)\|e\|_\infty = \frac{1+\gamma}{1-\gamma\lambda}\|V_\phi - V_k^\pi\|_\infty.$$

□

A simulation-lemma argument (Kearns & Singh, 2002) bounds $\|V_{k-1}^\pi - V_k^\pi\|_\infty \le \delta_k/(1-\gamma)$, so even an oracle critic on past chunks satisfies $\|V_\phi - V_k^\pi\|_\infty = \mathcal{O}(\delta_k/(1-\gamma))$, with a matching lower bound under monotone drift (Remark B.5). Through bootstrapping in δ_t and multi-epoch reuse of trajectories, this bias propagates and *compounds* across chunks.

Group baseline as a within-chunk control variate. On chunk k, draw $a_t^{(g)} \sim \pi_\theta(\cdot|s_t)$ and execute under $\mathcal{M}_k$ to obtain returns $R_t^{(g)}$, $g = 1,\dots,G$.

Define the within-chunk preconditioned policy gradient

$$\widetilde{\nabla} J_k(\theta) \triangleq \mathbb{E}_{s\sim\pi_\theta,\mathcal{M}_k}\left[\frac{1}{\sqrt{\Sigma_k(s)}+\varepsilon} \cdot \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[A_k^\pi(s,a)\nabla_\theta \log\pi_\theta(a \mid s)]\right], \tag{14}$$

where $\Sigma_k(s) \triangleq \mathrm{Var}_k(R \mid s)$. The state-dependent factor $1/(\sqrt{\Sigma_k(s)}+\varepsilon) > 0$ rescales each state's contribution but preserves the per-state ascent direction; in the sense of Mei et al. (2022) the within-chunk z-score acts as an adaptive step-size.

Lemma B.2 (Group-normalized advantage consistency of GRPO)**.** *Let $\mu_t = \frac{1}{G}\sum_g R_t^{(g)}$, $\sigma_t^2 = \frac{1}{G}\sum_g (R_t^{(g)} - \mu_t)^2$, and $\hat{A}_t^{(g)} = (R_t^{(g)} - \mu_t)/(\sigma_t + \varepsilon)$ (with $\varepsilon > 0$ a small numerical-stability constant). Then $\mathbb{E}[\mu_t \mid s_t] = V_k^\pi(s_t)$, and the group-normalized advantage is asymptotically unbiased in conditional expectation,*

$$\mathbb{E}\left[\hat{A}_t^{(g)} \,\middle|\, s_t, a_t^{(g)}\right] \xrightarrow[G\to\infty]{} \frac{A_k^\pi(s_t, a_t^{(g)})}{\sqrt{\mathrm{Var}_k(R \mid s_t)}+\varepsilon}.$$

Consequently, the empirical per-state GRPO estimator

$$\widehat{g}_{k,G}^{\mathrm{GRPO}}(s_t) \triangleq \frac{1}{G}\sum_{g=1}^{G}\hat{A}_t^{(g)}\,\nabla_\theta\log\pi_\theta(a_t^{(g)} \mid s_t)$$

is biased by $\mathcal{O}(G^{-1})$ relative to the population group-normalized update

$$g_{k,\mathrm{norm}}^\star(s_t) \triangleq \mathbb{E}_{a\sim\pi_\theta(\cdot|s_t)}\left[\frac{A_k^\pi(s_t,a)}{\sqrt{\Sigma_k(s_t)}+\varepsilon}\,\nabla_\theta\log\pi_\theta(a \mid s_t)\right],$$

with conditional MSE of order $\mathcal{O}(G^{-1})$, depending only on within-chunk statistics of $\mathcal{M}_k$. Averaging $g_{k,\mathrm{norm}}^\star(s_t)$ over $s_t \sim \pi_\theta, \mathcal{M}_k$ recovers the preconditioned gradient $\widetilde{\nabla} J_k(\theta)$ in equation 14.

Proof. Conditional on s_t, the returns $\{R_t^{(g)}\}_{g=1}^G$ are i.i.d. over the sampling $a_t^{(g)} \sim \pi_\theta(\cdot \mid s_t)$ and the chunk-k rollout, with mean $V_k^\pi(s_t)$ and variance $\Sigma_k(s_t) \triangleq \mathrm{Var}_k(R \mid s_t)$. Returns are bounded, $R_t^{(g)} \in [0, R_{\max}/(1-\gamma)]$.

(i) Unbiasedness of μ_t. By linearity,

$$\mathbb{E}[\mu_t \mid s_t] = \frac{1}{G}\sum_{g=1}^{G}\mathbb{E}[R_t^{(g)} \mid s_t] = V_k^\pi(s_t).$$

(ii) Conditional numerator. Fix g and condition on $(s_t, a_t^{(g)})$. For the realized index, $\mathbb{E}[R_t^{(g)} \mid s_t, a_t^{(g)}] = Q_k^\pi(s_t, a_t^{(g)})$. For $g' \neq g$, $a_t^{(g')} \sim \pi_\theta(\cdot \mid s_t)$ is sampled independently of $a_t^{(g)}$, so $\mathbb{E}[R_t^{(g')} \mid s_t, a_t^{(g)}] = V_k^\pi(s_t)$. Hence

$$\mathbb{E}[\mu_t \mid s_t, a_t^{(g)}] = \tfrac{1}{G}\, Q_k^\pi(s_t, a_t^{(g)}) + \tfrac{G-1}{G}\, V_k^\pi(s_t),$$
$$\mathbb{E}\Big[R_t^{(g)} - \mu_t \,\Big|\, s_t, a_t^{(g)}\Big] = \tfrac{G-1}{G}\, A_k^\pi(s_t, a_t^{(g)}).$$

(iii) Asymptotic limit. The strong law gives $\mu_t \xrightarrow{a.s.} V_k^\pi(s_t)$ and $\sigma_t^2 \xrightarrow{a.s.} \Sigma_k(s_t)$ as $G \to \infty$, so $\sigma_t \xrightarrow{a.s.} \sqrt{\Sigma_k(s_t)}$. Boundedness of returns yields the deterministic bound $|\hat{A}_t^{(g)}| \leq R_{\max}/[(1-\gamma)\varepsilon]$, hence uniform integrability. Slutsky's theorem applied to the continuous map $(x, y) \mapsto (R_t^{(g)} - x)/(y + \varepsilon)$ then gives

$$\mathbb{E}\Big[\hat{A}_t^{(g)} \,\Big|\, s_t, a_t^{(g)}\Big] \xrightarrow[G\to\infty]{} \frac{A_k^\pi(s_t, a_t^{(g)})}{\sqrt{\Sigma_k(s_t)} + \varepsilon}.$$

(iv) Bias rate. The conditional numerator carries the multiplicative factor $(G-1)/G = 1 - \mathcal{O}(G^{-1})$ from (ii). For the denominator, the sample variance with divisor G has bias $\mathbb{E}[\sigma_t^2 \mid s_t] = \Sigma_k(s_t)\,(G-1)/G$, and a first-order Taylor expansion of $x \mapsto 1/(\sqrt{x} + \varepsilon)$ around $\Sigma_k(s_t)$, valid by smoothness on $(0, \infty)$ and bounded moments of σ_t^2, gives

$$\mathbb{E}\left[\frac{1}{\sigma_t + \varepsilon} \,\middle|\, s_t\right] = \frac{1}{\sqrt{\Sigma_k(s_t)} + \varepsilon} + \mathcal{O}(G^{-1}).$$

The remaining covariance term $\mathrm{Cov}(R_t^{(g)} - \mu_t,\, 1/(\sigma_t + \varepsilon) \mid s_t, a_t^{(g)})$ vanishes to leading order after integration against $\nabla_\theta \log \pi_\theta(a_t^{(g)} \mid s_t)$ by the score-function identity, contributing $\mathcal{O}(G^{-1})$ to g_k^{GRPO} (Zhou et al., 2026). Combining the numerator and denominator expansions,

$$\mathbb{E}\Big[\hat{A}_t^{(g)} \,\Big|\, s_t, a_t^{(g)}\Big] = \frac{A_k^\pi(s_t, a_t^{(g)})}{\sqrt{\Sigma_k(s_t)} + \varepsilon} + \mathcal{O}(G^{-1}).$$

Multiplying by $\nabla_\theta \log \pi_\theta(a_t^{(g)} \mid s_t)$ and integrating: since $\Sigma_k(s_t)$ does not depend on a, the denominator factors out of the action expectation, leaving the per-state preconditioning $1/(\sqrt{\Sigma_k(s_t)} + \varepsilon)$. Averaging $g_{k,\mathrm{norm}}^\star(s_t)$ over $s_t \sim \pi_\theta, \mathcal{M}_k$ recovers the preconditioned policy gradient $\widetilde{\nabla} J_k(\theta)$ defined in Equation 14, so $\big\|\mathbb{E}[g_k^{\mathrm{GRPO}}] - \widetilde{\nabla} J_k(\theta)\big\| = \mathcal{O}(G^{-1})$.

(v) Conditional MSE. Each summand $\hat{A}_t^{(g)} \nabla_\theta \log \pi_\theta(a_t^{(g)} \mid s_t)$ is bounded; the $\mathcal{O}(G^{-1})$ MSE on chunk k then follows from the exchangeable-sample analysis of the GRPO estimator in Zhou et al. (2026). Every quantity above depends only on $\mathcal{M}_k$, so no information from chunks $\{1, \ldots, k-1\}$ enters the gradient. □

Affine equivariance.

Lemma B.3 (Per-chunk affine equivariance)**.** *If $R_t^{(g)\prime} = \alpha_k R_t^{(g)} + \beta_k$ with $\alpha_k > 0$, then*

$$\hat{A}_t^{(g)\prime} = \frac{\alpha_k\big(R_t^{(g)} - \mu_t\big)}{\alpha_k \sigma_t + \varepsilon},$$

$$\hat{A}_t^{(g)\prime} - \hat{A}_t^{(g)} = (R_t^{(g)} - \mu_t)\,\frac{(\alpha_k - 1)\,\varepsilon}{(\alpha_k \sigma_t + \varepsilon)(\sigma_t + \varepsilon)}.$$

In particular, $\hat{A}_t^{(g)\prime} = \hat{A}_t^{(g)}$ exactly when $\varepsilon = 0$, and otherwise $\big|\hat{A}_t^{(g)\prime} - \hat{A}_t^{(g)}\big| = \mathcal{O}\big(\varepsilon\,|\alpha_k - 1|/\sigma_t^2\big)$.

Proof. Under $R_t^{(g)\prime} = \alpha_k R_t^{(g)} + \beta_k$ with $\alpha_k > 0$, the group statistics transform as

$$\mu_t' = \alpha_k \mu_t + \beta_k, \qquad R_t^{(g)\prime} - \mu_t' = \alpha_k\big(R_t^{(g)} - \mu_t\big),$$

$$\sigma_t^{2\prime} = \alpha_k^2 \sigma_t^2, \qquad \sigma_t' = \alpha_k \sigma_t,$$

so $\hat{A}_t^{(g)\prime} = \alpha_k\big(R_t^{(g)} - \mu_t\big)/(\alpha_k \sigma_t + \varepsilon)$. Subtracting $\hat{A}_t^{(g)} = (R_t^{(g)} - \mu_t)/(\sigma_t + \varepsilon)$ and combining over the common denominator,

$$\begin{aligned}
\hat{A}_t^{(g)\prime} - \hat{A}_t^{(g)} &= (R_t^{(g)} - \mu_t)\left[\frac{\alpha_k}{\alpha_k \sigma_t + \varepsilon} - \frac{1}{\sigma_t + \varepsilon}\right] \\
&= (R_t^{(g)} - \mu_t) \cdot \frac{\alpha_k(\sigma_t + \varepsilon) - (\alpha_k \sigma_t + \varepsilon)}{(\alpha_k \sigma_t + \varepsilon)(\sigma_t + \varepsilon)} \\
&= (R_t^{(g)} - \mu_t) \cdot \frac{(\alpha_k - 1)\,\varepsilon}{(\alpha_k \sigma_t + \varepsilon)(\sigma_t + \varepsilon)},
\end{aligned}$$

which is the displayed identity. For $\varepsilon = 0$ the difference vanishes identically, giving exact equivariance. For $\varepsilon > 0$, boundedness of returns yields $|R_t^{(g)} - \mu_t| = \mathcal{O}(R_{\max}/(1-\gamma))$, and the elementary inequality $(\alpha_k \sigma_t + \varepsilon)(\sigma_t + \varepsilon) \geq \alpha_k \sigma_t^2$ gives

$$\big|\hat{A}_t^{(g)\prime} - \hat{A}_t^{(g)}\big| = \mathcal{O}\left(\frac{\varepsilon\,|\alpha_k - 1|}{\sigma_t^2}\right). \qquad \square$$

The PPO advantage in Lemma B.1 admits no analogous equivariance because V_ϕ is not re-scaled at chunk k. From the viewpoint of Mei et al. (2022), the within-chunk z-score acts as an *adaptive step-size*.

Per-chunk bias bounds.

Corollary B.4 (Per-chunk bias of PPO and GRPO estimators)**.** *Let $g_k^{\mathrm{PPO}}, g_k^{\mathrm{GRPO}}$ be the gradient estimators on chunk k, $g_k^\star = \nabla J_k(\theta)$ the vanilla policy gradient under $\mathcal{M}_k$, and $\widetilde{\nabla} J_k(\theta)$ the preconditioned target equation 14. Combining Lemmas B.1 and B.2, for any fixed group size G,*

$$\big\|\mathbb{E}[g_k^{\mathrm{PPO}}] - g_k^\star\big\| = \mathcal{O}\Big(\tfrac{\delta_k}{(1-\gamma)(1-\gamma\lambda)}\Big),$$

$$\left\|\mathbb{E}[g_k^{\mathrm{GRPO}}] - \widetilde{\nabla} J_k(\theta)\right\| = \mathcal{O}(G^{-1}).$$

By Lemma B.3, the GRPO bias is additionally invariant under per-chunk affine rescaling of r_k.

Proof. The PPO bound follows by combining Lemma B.1 with $\|V_\phi - V_k^\pi\|_\infty = \mathcal{O}(\delta_k/(1-\gamma))$ from Equation 13, then integrating against the bounded score $\nabla_\theta \log \pi_\theta(a_t \mid s_t)$ over π_θ. The GRPO bound follows by integrating the per-state bias of Lemma B.2 over $s_t \sim \pi_\theta, \mathcal{M}_k$; the integration step uses that $\Sigma_k(s_t)$ does not depend on a, so the per-state denominator factors out of the action expectation and the result matches the definition equation 14. Affine invariance is immediate from Lemma B.3. □

Remark B.5 (Tightness under monotone drift). Suppose the drift is monotone in the sense that there exists $(s^\star, a^\star)$ with $r_j(s^\star, a^\star) - r_{j-1}(s^\star, a^\star) \geq c\,\delta_j$ for all $j \leq k$ and some $c > 0$. Then for any critic V_ϕ fit on chunks $\{1, \ldots, k-1\}$, $\|V_\phi - V_k^\pi\|_\infty \geq c\,\delta_k/(2(1-\gamma))$, since no convex combination of past values can match the chunk-k value at $(s^\star, a^\star)$. The monotonic background-distribution drift in Figs. 5, 6 satisfies this condition.

Remark B.6 (Rate-constrained objective). Recall in Section 3, we model the trigger problem as a constrained MDP. We maximize $J_k(\theta)$ subject to $\mathbb{E}_{\pi_\theta, \mathcal{M}_k}[\rho_t] \leq \rho_{\max}$, with Lagrangian $\tilde{R}_t = r_t - \lambda_k(\rho_t - \rho_{\max})$. Luminosity drift scales r_t and ρ_t jointly, so $\tilde{R}_t$ inherits the affine drift of Lemma B.3. The GRPO update absorbs $\mathcal{L}_k$-rescaling (per-chunk luminosity) without re-tuning λ_k; the PPO critic acquires the Lemma B.1 bias on both reward and rate channels. This matches the experiments: GRPO holds the rate–efficiency Pareto front across luminosity sweeps, while PPO with GAE drifts off it.

Discussion and limitations. Corollary B.4 characterizes the per-chunk bias of the two estimators against their respective targets: PPO's bias against the vanilla policy gradient $\nabla J_k(\theta)$ scales as $\mathcal{O}(\delta_k)$ in the per-chunk drift, while GRPO's bias against the preconditioned target $\widetilde{\nabla} J_k(\theta)$ scales as $\mathcal{O}(G^{-1})$ in the group size, drift-independent. The preconditioning factor $1/(\sqrt{\Sigma_k(s)} + \varepsilon) > 0$ preserves per-state ascent direction, so $\widetilde{\nabla} J_k$ shares ascent direction with ∇J_k at every state. The two bounds measure different things and we draw no direct comparative conclusion.

In summary, on chunk k the PPO estimator has bias $\mathcal{O}(\delta_k/((1-\gamma)(1-\gamma\lambda)))$ against $\nabla J_k(\theta)$, while the GRPO estimator has bias $\mathcal{O}(G^{-1})$ against the preconditioned target $\widetilde{\nabla} J_k(\theta)$ — drift-independent and additionally invariant under per-chunk affine rescaling of r_k. The cost is that GRPO targets a state-preconditioned gradient rather than the vanilla one, and requires G on-policy samples per state.

C. GFPO for Streaming Trigger Control: Pseudocode

Algorithm 1 GFPO for streaming trigger control at LHC: MC training and CMS deployment

Input: Data Streams $\mathcal{D}_{\mathrm{MC}}, \mathcal{D}_{\mathrm{CMS}}$; variant $v \in \{\text{GFPO-F}, \text{GFPO-FR}\}$; group size G, kept size K; target background rate r_B^*, tolerance τ; MC passes E; deployment mode $\in \{\text{FROZEN/DEPLOYMENT}, \text{TTT (TEST-TIME TRAINING)}\}$.

Output: Trained policy π_θ; threshold trajectory $\{c_t\}_{t \in \mathcal{D}_{\mathrm{CMS}}}$.

Procedure GFPO-STEP(s_t, c_t) Sample actions $\{a_t^{(g)}\}_{g=1}^G \sim \pi_{\theta_{\mathrm{old}}}(\cdot \mid s_t)$; form candidates $c_t^{(g)} \leftarrow c_t + a_t^{(g)}$ Evaluate rate errors $\hat{e}_t(c_t^{(g)})$ and rewards $\{R_t^{(g)}\}$; `// Eq. 3`

$\mathcal{F}_t(\tau) \leftarrow \{g \in [G] : |\hat{e}_t(c_t^{(g)})| \leq \tau\}$ $\mathcal{K}_t \leftarrow$ kept set per variant v ; `// Eqs. 10/11`

if $\mathcal{K}_t \neq \emptyset$ **then**

  $\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{L}_{\mathrm{GRPO}}(\theta; \mathcal{K}_t)$; `// Eqs. 9, 5`

return Δc_t *from* $\mathcal{K}_t$ *(or sampled from* π_θ *if* $\mathcal{K}_t = \emptyset$*)*

Initialize π_θ **Phase 1: MC training for** $e \leftarrow 1$ **to** E, *chunk* $t \in \mathcal{D}_{MC}$ **do**

  $s_t \leftarrow$ BUILDSEQUENTIALSTATE ; `// Alg. 2`

  $\Delta c_t \leftarrow$ GFPO-STEP(s_t, c_t) $c_{t+1} \leftarrow c_t + \mathrm{clip}(\Delta c_t, [-\Delta c_{\max}, \Delta c_{\max}])$

Phase 2: CMS deployment for *chunk* $t \in \mathcal{D}_{CMS}$ **do**

  $s_t \leftarrow$ BUILDSEQUENTIALSTATE **if** *mode* = FROZEN/DEPLOYMENT **then**

    $\Delta c_t \leftarrow \arg\max_a \pi_\theta(a \mid s_t)$; `// no gradient update`

  else

    $\Delta c_t \leftarrow$ GFPO-STEP(s_t, c_t) ; `// TTT: one update per chunk`

  $c_{t+1} \leftarrow c_t + \mathrm{clip}(\Delta c_t, [-\Delta c_{\max}, \Delta c_{\max}])$

D. Experimental Setup for LHC

D.1. Dataset Summary

Datasets. We use the datasets of Emami et al. (2026). Sample sizes are listed in Table 3. $t\bar{t}$ and $h \to 4b$ are shared for both MC and real CMS collision data setting.

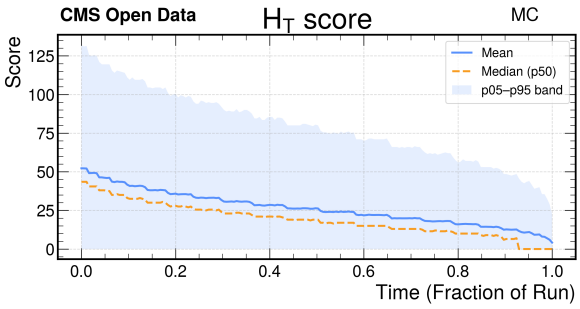
Score drift. Figures 5and 6 report the running mean, median, and central 5–95% band of background H_T and AD scores over MC and CMS Run 283408. Both streams drift downward over the run; MC drifts more strongly toward zero, while CMS Run 283408 declines mildly and retains a heavy upper tail. CMS Run 283408 is also globally higher-scoring than MC, a domain shift on top of the within-run drift.

Autoencoder protocol. Following Emami et al. (2026), we use latent dimension $d = 2$. For real data, we train on CMS Run 283876 (CMS Collaboration, 2024c) (the second-longest 2016 run (CERN Open Data Portal, 2026)) and evaluate AD scores on CMS Run 283408 (the largest 2016 run; 1.99M background events). For MC, we train on one background sample and evaluate on an independent background sample.

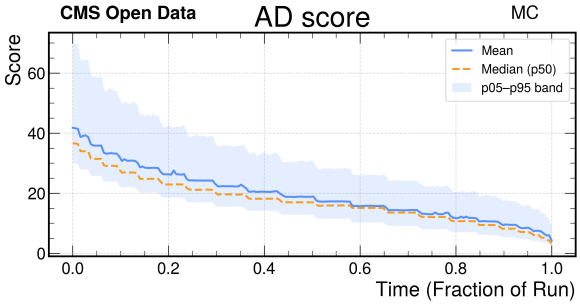
Signal/background overlap (Figure 7). $t\bar{t}$ events are kinematically well-separated from MinBias background under both H_T and the AD score, lying predominantly to the right of the oracle cut. $h \to 4b$ events overlap heavily with the background: each b-quark carries $\sim m_H/4 \approx$ 31 GeV (de Florian et al., 2017; Aaboud et al., 2018), so H_T and AD scores fall in the bulk of the background distribution and a large fraction of signal events sits below the oracle cut at any operating point. Notably, this ceiling is *intrinsic* to the signals, not the controllers: every method in Tables 1 and 2 reaches $\sim$98% on $t\bar{t}$ but stays below $\sim$30% on $h \to 4b$ for H_T (and analogously for AD), tracking the oracle bounds in Table 4.

Oracle cut. The oracle cut r^+ is the threshold c for which the background rate equals $r^+ = r_B^* + \tau = 0.275\%$, the tightest cut still inside the upper edge of the tolerance band. Its signal efficiency is the maximum achievable by any deterministic threshold rule. Table 4 reports oracle efficiencies at both r_B^* and r^+ across triggers and signals. Same as Emami et al. (2026), we exclude the first 10 chunks (initial threshold calibration) and tabulate only events past the first 500K (MC) and 200K (CMS Run 283408).

Reading the gap tables. Tables 5 and 6 report the gap between our methods and the oracle at r^+. On CMS Run 283408 $h \to 4b$, both GFPO variants *exceed* the oracle in-band efficiency: in-band chunks are not uniformly distributed but concentrate in the high-efficiency tail of the run (Figure 9).



(a) H_T trigger



(b) AD trigger

Figure 5. Background scores drift over time for both triggers for MC. Running mean (solid), median (dashed), and the central 5-95% band (shaded) of the background H_T score (left) and AD score (right) as a function of run time (fraction of run).

D.2. Training setup

Training pass. Single chronological pass over the 148-chunk training partition (80% of the 185-chunk MC stream after the 10-chunk calibration window). Each chunk: 50,000 events; micro-step: 5,000 events.

Optimizer. All policies use Adam. Learning rates: GRPO and L-GRPO 2×10^{-4}; PPO 3×10^{-4}; DQN and DQN-F 1×10^{-4} with mini-batch size 32.

Group sizes. GRPO, L-GRPO: $G = 16$. GFPO-F, GFPO-FR: $G = 64$ candidates per micro-step, $K = 16$ kept (matching Figure 3).

Reward. $\lambda_1 = 0.25$, $\lambda_2 = 1.00$, $\alpha = 0.7$. Recall in Section 3, we define target background rate $r_B^* = 0.25\%$, tolerance $\tau = \pm0.025\%$ ($\approx \pm10$ kHz).

Reporting. Mean over 5 random seeds per method.

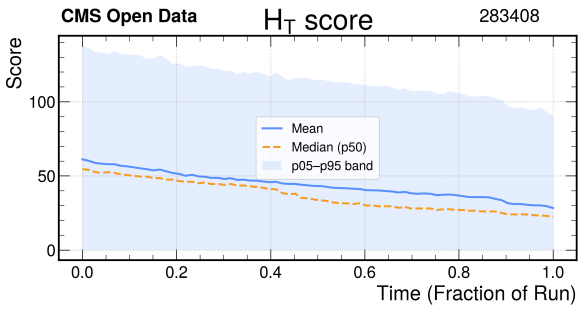
Compute. Single Apple M4 CPU (10 cores, 32 GB unified memory; no GPU). Wall-clock $\approx$ 10 minutes per method per seed for the full pass.

Table 3. Dataset summary

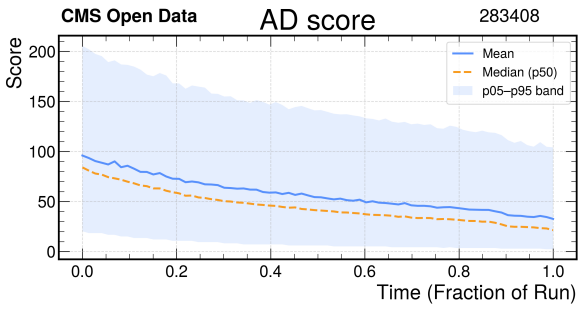
Sample	Events
MC Background	9,794,099
$t\bar{t}$ Signal	2,233,999
$h \to 4b$ Signal	1,102,412
CMS Run 283876 Background (AD Trigger Training)	664,475
CMS Run 283408 Background (Evaluation)	1,987,943

Table 4. Overall oracle signal efficiencies for MC and CMS Run 283408 at $r_B^* = 0.250\%$ and the loosest feasible rate $r^+ = r_B^* + \tau = 0.275\%$. For oracle performance, the default way to calculate signal efficiency is to assume every threshold is in-band (i.e., the background rate stays within the tolerance band), and dashed lines represent the same values as overall signal efficiency.

		MC				CMS Run 283408			
		$\epsilon_{\text{overall}}$		ϵ_{inband}		$\epsilon_{\text{overall}}$		ϵ_{inband}	
Trigger	Method	$t\bar{t}$	$h \to 4b$	$t\bar{t}$	$h \to 4b$	$t\bar{t}$	$h \to 4b$	$t\bar{t}$	$h \to 4b$
H_T	Oracle ($r^+ = r_B^* + \tau$)	**98.286**	**27.537**	—	—	**98.049**	**33.693**	—	—
	PID	97.989	25.990	97.912	25.304	97.381	33.347	97.497	35.242
	GFPO-F (Ours)	98.076	26.463	98.083	26.393	97.587	33.347	97.508	33.262
	GFPO-FR (Ours)	98.232	27.231	98.214	27.003	97.734	33.479	97.821	33.451
AD	Oracle ($r^+ = r_B^* + \tau$)	**94.414**	**23.739**	—	—	**76.664**	**41.028**	—	—
	PID	93.597	22.441	93.588	22.277	75.053	39.191	76.210	44.573
	GFPO-F (Ours)	93.727	22.549	93.813	22.800	75.562	40.083	76.209	42.533
	GFPO-FR (Ours)	94.193	23.221	94.310	23.556	75.582	40.309	76.515	44.477



(a) H_T trigger



(b) AD trigger

Figure 6. Background scores drift over time for both triggers for CMS Run 283408. Running mean (solid), median (dashed), and the central 5-95% band (shaded) of the background scores as a function of run time (fraction of run).

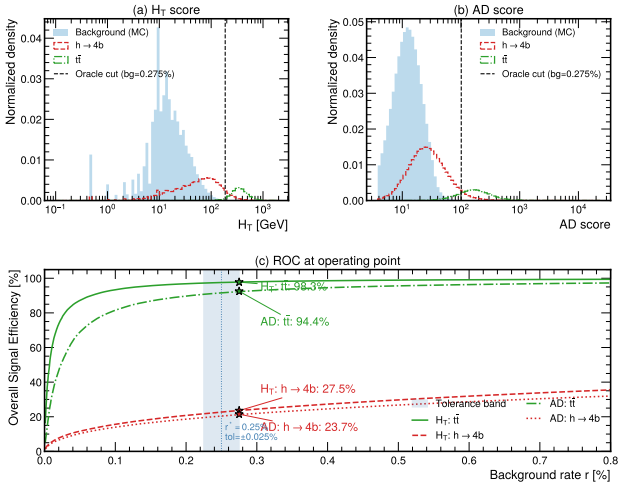


Figure 7. Signal-background discrimination for the H_T and AD triggers (MC). (a) H_T score and (b) AD score distributions for background (MinBias, shaded blue), $h \to 4b$ signal (red solid), and $t\bar{t}$ signal (green dashed). The dashed black line indicates the oracle cut corresponding to the upper edge of the tolerance band ($r^+ = 0.275\%$). (c) ROC curves showing signal efficiency versus background rate r for both triggers and both signals. The light blue shaded region denotes the tolerance band ($r^* \pm \tau = 0.25\% \pm 0.025\%$). Markers at the upper band edge show the maximum achievable signal efficiency: H_T trigger yields 98.3% for $t\bar{t}$ and 27.5% for $h \to 4b$, while the AD trigger yields 94.4% for $t\bar{t}$ and 23.7% for $h \to 4b$. The large gap between the two signals reflects the inherent overlap of the $h \to 4b$ distribution with the MinimumBias background in both feature spaces, setting a physics-limited upper bound on achievable signal efficiency at this operating point.

Table 5. Oracle signal efficiencies (MC) at $r_B^* = 0.250\%$ and $r^+ = r_B^* + \tau = 0.275\%$. Δ columns show improvement over Oracle(r^+). GFPO-FR has smaller signal efficiency gap with Oracle (r^+).

		$\epsilon_{\text{overall}}$		Δ_{overall} (Oracle (r^+) - Ours)		ϵ_{inband}		Δ_{inband} (Oracle (r^+) - Ours)	
Trigger	Method	$t\bar{t}$	$h \to 4b$	$t\bar{t}$	$h \to 4b$	$t\bar{t}$	$h \to 4b$	$t\bar{t}$	$h \to 4b$
H_T	Oracle (r^+)	**98.286**	**27.537**			**98.286**	**27.537**		
	GFPO-F	98.076	26.463	+0.210	+1.074	98.083	26.393	+0.203	+1.144
	GFPO-FR	98.232	27.231	+0.054	+0.306	98.214	27.003	+0.072	+0.534
AD	Oracle (r^+)	**94.414**	**23.739**			**94.414**	**23.739**		
	GFPO-F	93.727	22.549	+0.687	+1.190	93.813	22.800	+0.601	+0.939
	GFPO-FR	94.193	23.221	+0.221	+0.518	94.310	23.556	+0.104	+0.183

Table 6. Oracle signal efficiencies (CMS Run 283408) at $r_B^* = 0.250\%$ and $r^+ = r_B + \tau = 0.275\%$. Δ columns show improvement over Oracle (r^+). GFPO-FR has smaller signal efficiency gap with Oracle (r^+) except for $h \to 4b$ inband. Figure 9 presents why inband $h \to 4b$ signal efficiency is higher than oracle (r^+).

		$\epsilon_{\text{overall}}$		Δ_{overall} (Oracle (r^+) - Ours)		ϵ_{inband}		Δ_{inband} (Oracle (r^+) - Ours)	
Trigger	Method	$t\bar{t}$	$h \to 4b$	$t\bar{t}$	$h \to 4b$	$t\bar{t}$	$h \to 4b$	$t\bar{t}$	$h \to 4b$
H_T	Oracle (r^+)	**98.049**	**33.693**			**98.049**	**33.693**		
	GFPO-F (Ours)	97.587	33.347	+0.462	+0.346	97.508	33.262	+0.541	+0.431
	GFPO-FR (Ours)	97.734	33.479	+0.315	+0.214	97.821	33.451	+0.228	+0.242
AD	Oracle (r^+)	**76.664**	**41.028**			**76.664**	41.028		
	GFPO-F (Ours)	75.562	40.083	+1.102	+0.945	76.209	42.533	+0.455	−1.505
	GFPO-FR (Ours)	75.582	40.309	+1.082	+0.719	76.515	**44.477**	+0.149	−3.449

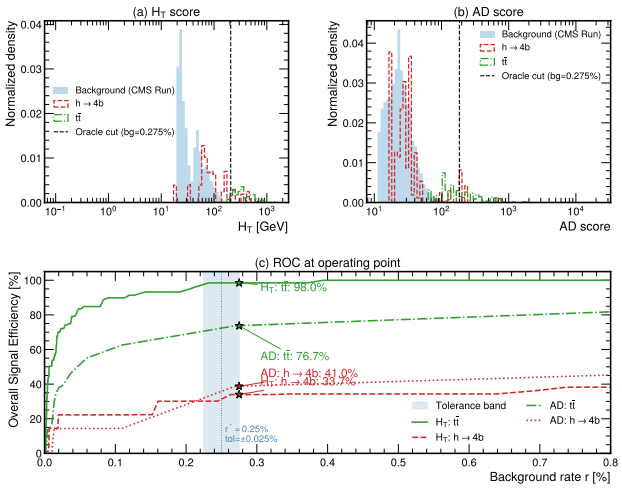


Figure 8. Signal-background discrimination for the H_T and AD triggers (CMS Run 283408). (a) H_T score and (b) AD score distributions for background (MinBias, shaded blue), $h \to 4b$ signal (red solid), and $t\bar{t}$ signal (green dashed). The dashed black line indicates the oracle cut corresponding to the upper edge of the tolerance band ($r^+ = 0.275\%$). (c) ROC curves showing signal efficiency versus background rate r for both triggers and both signals. The light blue shaded region denotes the tolerance band ($r^* \pm \tau = 0.25\% \pm 0.025\%$). Markers at the upper band edge show the maximum achievable signal efficiency: H_T trigger yields 98.0% for $t\bar{t}$ and 33.7% for $h \to 4b$, while the AD trigger yields 76.7% for $t\bar{t}$ and 41.0% for $h \to 4b$.

D.3. Models and Hyperparameters

Hyperparameters in reward design. We sweep $\lambda_1, \lambda_2 \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$ in Equation 3 (25 configurations per trigger). Figure 10 plots in-band rate against signal efficiency across all configurations and all four trigger$\times$signal pairs. GFPO-F and GFPO-FR collapse to a tight cluster in the upper-right (high in-band rate, high signal efficiency) regardless of (λ_1, λ_2); baselines (DQN, PPO, GRPO, ADT) exhibit a pronounced trade-off, with hulls spanning a wide region of the objective space. The optimal one that has highest signal efficiency and inband rate is $\lambda_1 = 0.25$, $\lambda_2 = 1.00$.

D.4. Standard anomaly detection metrics for LHC

Our method achieves strong signal efficiency while maintaining the highest InBand rate. In this subsection, we examine standard anomaly detection metrics commonly used in the literature, including precision, recall, and F1 score, within the context of the LHC trigger setting. We emphasize that these conventional anomaly detection metrics primarily evaluate signal-selection performance and do not account for operational constraints such as background-rate requirements, which are central to realistic LHC trigger deployment. Tables 7, 8, and 9 summarize the anomaly detection results for the three experimental settings corresponding to Tables 1, 12, and 2, respectively, which report the LHC-specific operational metrics discussed in the main text.

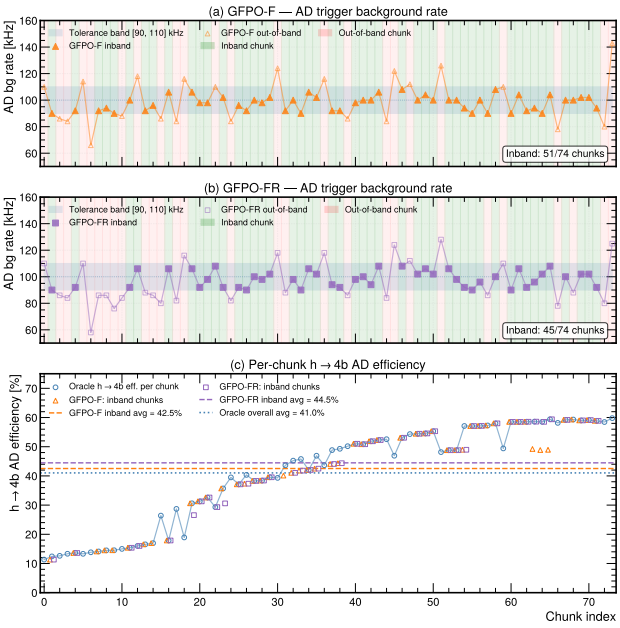
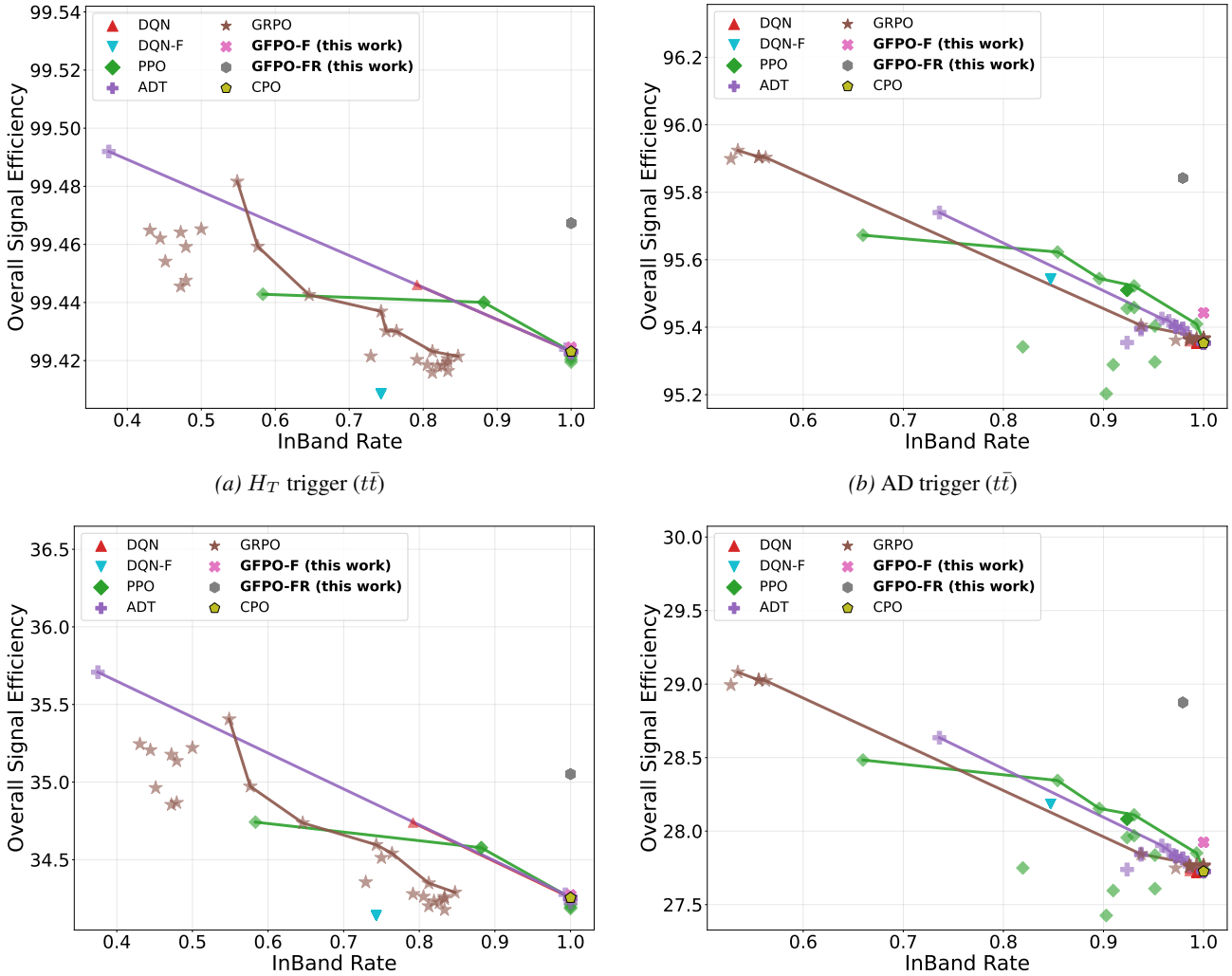


Figure 9. (a) GFPO-F for AD trigger (CMS Run 283408) (b) GFPO-FR for AD trigger (CMS Run 283408) (c) $h \to 4b$ signal efficiency for CMS Run 283408 for GFPO-F (inband) and GFPO-FR (inband) and oracle (r^+) for $h \to 4b$ (overall). The $h \to 4b$ efficiency exhibits strong temporal drift, rising from $\sim 11\%$ in early high-pileup chunks to $\sim 59\%$ in late low-pileup chunks. Because inband chunks are not *uniformly* distributed but concentrate in the high-efficiency tail of the run as (a) and (b) shows inband chunks tend to concentrate at the tail of the run, the inband average for GFPO-F (42.5%) and GFPO-FR (44.5%) both exceed the oracle overall average (41.0%), which averages uniformly over all chunks excluding first 10 chunks.



(a) H_T trigger ($t\bar{t}$)

(b) AD trigger ($t\bar{t}$)

(c) H_T trigger ($h \to 4b$)

(d) AD trigger ($h \to 4b$)

Figure 10. **Sensitivity Analysis of Reward Components (MC).** Each point represents a (λ_1, λ_2) configuration from Equation 3, with concave hulls connecting the upper envelope per method. The x-axis measures the fraction of chunks whose background rate falls within the tolerance band, and the y-axis measures overall signal efficiency. Across all four trigger, signal combinations (H_T/AD $\times$ $t\bar{t}/h \to 4b$), baseline methods (DQN, PPO, GRPO, ADT) exhibit a pronounced trade-off between inband stability and signal efficiency, with their hulls spanning a wide region of the objective space. In contrast, our methods (GFPO-F and GFPO-FR) collapse to a tight cluster in the upper-right corner, simultaneously achieving near-perfect inband rates and the highest signal efficiency, demonstrating strong robustness to the choice of $\lambda_{1,2}$.

Table 7. Anomaly detection benchmark for MC. All metrics are averaged over per chunk. Bold and underline mark the best and second-best per column. Our method GFPO-FR places top-two for 9 out of 14 metrics on H_T trigger and 7 out of 14 metrics on AD trigger, outperforming or performing on par with all other baseline methods. Our methods also achieve highest InBand rate among all methods (Table 1).

Trigger	Method	TPR(Combined)/Recall ↑	TPR($t\bar{t}$) ↑	TPR($h \to 4b$) ↑	FPR↓	FPR ($t\bar{t}$) ↓	FPR ($h \to 4b$) ↓	TNR↑	FNR↓	Prec.(Combined) ↑	Prec.($t\bar{t}$) ↑	Prec.($h \to 4b$) ↑	F1(Combined) ↑	F1($t\bar{t}$) ↑	F1($h \to 4b$) ↑
H_T	Constant	0.655	0.91	0.136	**0.001**	**0.001**	**0.001**	**0.999**	0.345	**1.000**	**1.000**	**0.996**	0.791	0.952	0.240
H_T	PID	0.740	0.98	0.252	0.002	0.002	0.002	0.998	0.260	0.999	0.999	0.992	0.850	0.989	0.402
H_T	DSPOT	0.748	0.971	0.245	0.002	0.002	0.002	0.998	**0.252**	0.999	0.999	0.991	**0.855**	**0.992**	**0.427**
H_T	ADT	0.739	0.98	0.250	0.002	0.002	0.002	0.998	0.261	0.999	0.999	0.992	0.850	0.989	0.399
H_T	DQN	0.740	0.98	0.252	0.002	0.002	0.002	0.998	0.260	0.999	0.999	0.992	0.850	0.989	0.402
H_T	PPO	0.740	0.98	0.252	0.002	0.002	0.002	0.998	0.260	0.999	0.999	0.992	0.850	0.989	0.401
H_T	GRPO	0.740	0.98	0.252	0.002	0.002	0.002	0.998	0.260	0.999	0.999	0.992	0.850	0.989	0.402
H_T	L-GRPO	0.741	0.981	0.264	0.002	0.002	0.002	0.998	0.259	0.999	0.999	0.990	0.851	0.989	0.413
H_T	CPO	0.733	0.977	0.240	0.002	0.002	0.002	0.998	0.267	0.999	0.999	0.992	0.845	0.988	0.386
H_T	GFPO-F	0.741	0.98	0.254	0.003	0.003	0.003	0.997	0.259	0.999	0.999	0.992	0.851	0.990	0.404
H_T	GFPO-FR	0.744	0.982	0.260	0.003	0.003	0.003	0.997	0.256	0.999	0.999	0.992	0.853	0.990	0.412
AD	Constant	0.576	0.795	0.131	**0.001**	**0.001**	**0.001**	**0.999**	0.424	**1.000**	**1.000**	**0.995**	0.731	0.886	0.231
AD	PID	0.700	0.936	0.219	0.002	0.002	0.002	0.998	0.300	0.999	0.999	0.991	0.823	0.966	0.359
AD	DSPOT	0.704	0.939	**0.231**	0.002	0.002	0.002	0.998	0.296	0.999	0.999	0.990	0.826	0.968	**0.374**
AD	ADT	0.700	0.936	0.220	0.003	0.003	0.003	0.997	0.300	0.999	0.999	0.991	0.824	0.967	0.360
AD	DQN	0.699	0.935	0.219	0.002	0.002	0.002	0.998	0.301	0.999	0.999	0.991	0.823	0.966	0.358
AD	PPO	0.704	0.940	0.225	0.003	0.003	0.003	0.997	0.296	0.999	0.999	0.991	0.826	0.968	0.367
AD	GRPO	0.700	0.936	0.219	0.002	0.002	0.002	0.998	0.300	0.999	0.999	0.991	0.823	0.966	0.359
AD	L-GRPO	0.699	0.936	0.221	0.002	0.002	0.002	0.998	0.301	0.999	0.999	0.990	0.823	0.966	0.361
AD	CPO	0.694	0.932	0.211	0.002	0.002	0.002	0.998	0.306	0.999	0.999	0.991	0.819	0.964	0.348
AD	GFPO-F	0.701	0.937	0.221	0.003	0.003	0.003	0.997	0.299	0.999	0.999	0.991	0.824	0.967	0.361
AD	GFPO-FR	**0.706**	**0.941**	0.226	0.003	0.003	0.003	0.997	**0.294**	0.999	0.999	0.991	**0.827**	**0.969**	0.368

Table 8. Anomaly detection benchmark for CMS Run 283408. All metrics are averaged over per chunk. Each RL policy is continually fine-tuned on streaming time chunks rather than frozen at deployment (test-time training). Bold and underline mark the best and second-best per column. Our method GFPO-FR place top-two for 7 out of 14 metrics on H_T trigger and 6 out of 14 metrics on AD triggers. While baselines like PID, DSPOT, and PPO achieve more top-two metrics for the H_T trigger, these metrics purely reflect signal efficiency without background rate constraints. As detailed in Table 12, our methods GFPO-FR and GFPO-F maintain higher InBand rates (95% and 98%, respectively) compared to PPO (97.491 %), DSPOT (17.6%) and PID (43.2%).

Trigger	Method	TPR(Combined)/Recall ↑	TPR($t\bar{t}$) ↑	TPR($h \to 4b$) ↑	FPR↓	FPR ($t\bar{t}$) ↓	FPR ($h \to 4b$) ↓	TNR↑	FNR↓	Prec.(Combined) ↑	Prec.($t\bar{t}$) ↑	Prec.($h \to 4b$) ↑	F1(Combined) ↑	F1($t\bar{t}$) ↑	F1($h \to 4b$) ↑
H_T	Constant	0.568	0.913	0.223	**0.001**	**0.001**	**0.001**	**0.999**	0.432	**0.999**	**0.999**	**0.994**	0.724	0.954	0.364
H_T	PID	0.654	0.975	0.333	0.002	0.002	0.002	0.998	0.346	0.998	0.997	0.993	0.790	0.986	0.499
H_T	DSPOT	**0.667**	0.940	0.282	0.002	0.002	0.002	0.998	**0.333**	0.998	0.998	0.992	**0.800**	**0.989**	**0.510**
H_T	ADT	0.649	**0.978**	0.333	0.003	0.003	0.003	0.997	0.351	0.998	0.997	0.992	0.791	0.988	0.499
H_T	DQN	0.653	0.974	0.332	0.002	0.002	0.002	0.998	0.347	0.998	0.997	0.993	0.789	0.985	0.498
H_T	PPO	0.653	0.977	0.333	0.002	0.002	0.002	0.998	0.347	0.998	0.997	0.993	0.791	0.987	0.498
H_T	GRPO	0.653	0.975	0.331	0.002	0.002	0.002	0.998	0.347	0.998	0.998	0.993	0.789	0.986	0.496
H_T	L-GRPO	0.653	0.976	0.332	0.002	0.002	0.002	0.998	0.347	0.998	0.997	0.992	0.789	0.986	0.487
H_T	CPO	0.651	0.974	0.329	0.002	0.002	0.002	0.998	0.349	0.998	0.997	0.992	0.787	0.985	0.487
H_T	GFPO-F	0.654	0.976	0.332	0.002	0.002	0.002	0.998	0.346	0.997	0.997	0.993	0.790	0.986	0.498
H_T	GFPO-FR	0.656	**0.978**	**0.334**	0.003	0.003	0.003	0.997	0.344	0.998	0.997	0.993	0.791	0.987	0.499
AD	Constant	0.423	0.625	0.221	**0.001**	**0.001**	**0.001**	**0.999**	0.577	**0.999**	**0.998**	0.995	0.594	0.769	0.362
AD	PID	0.573	0.753	0.392	0.002	0.002	0.002	0.998	0.427	0.998	0.997	0.994	0.728	0.858	0.563
AD	DSPOT	0.688	**0.810**	**0.567**	0.002	0.002	0.002	0.998	0.312	0.998	0.997	**0.996**	**0.815**	**0.893**	**0.722**
AD	ADT	0.574	0.754	0.394	0.002	0.002	0.002	0.998	0.426	0.998	0.997	0.994	0.729	0.858	0.565
AD	DQN	0.582	0.759	0.404	0.002	0.002	0.002	0.998	0.418	0.997	0.997	0.994	0.735	0.862	0.575
AD	PPO	0.565	0.748	0.383	0.002	0.002	0.002	0.998	0.435	0.998	0.997	0.994	0.722	0.854	0.553
AD	GRPO	0.578	0.755	0.401	0.002	0.002	0.002	0.998	0.422	0.998	0.997	0.994	0.732	0.859	0.571
AD	L-GRPO	0.596	0.757	0.435	0.002	0.002	0.002	0.998	0.404	0.998	0.997	0.993	0.746	0.860	0.591
AD	CPO	0.594	0.756	0.432	0.002	0.002	0.002	0.998	0.406	0.998	0.997	0.993	0.740	0.859	0.586
AD	GFPO-F	0.580	0.757	0.402	0.002	0.002	0.002	0.998	0.420	0.998	0.997	0.994	0.733	0.861	0.572
AD	GFPO-FR	0.578	0.755	0.401	0.002	0.002	0.002	0.998	0.422	0.998	0.997	0.994	0.732	0.859	0.571

Table 9. Anomaly detection benchmark on CMS Run 283408 (zero-shot transfer). All metrics are averaged over per chunk. MC-trained policies from Table 1 are *frozen* and *deployed* on real collision data without fine-tuning. Bold and underline mark the best and second-best per column. While baseline like DSPOT (9 out of 14 metrics) achieve more top-two metrics for the H_T trigger than GFPO-F (5 out of 14 metrics), again, these metrics purely reflect signal efficiency without background rate constraints. As detailed in Table 2, our methods GFPO-FR and GFPO-F maintain *substantially* higher InBand rates for H_T trigger (95 % and 99 % , respectively) compared to PPO (83.8 %), DSPOT (43.2%) and PID (43.2%).

Trigger	Method	TPR(Combined)/Recall ↑	TPR($t\bar{t}$) ↑	TPR($h \to 4b$) ↑	FPR↓	FPR ($t\bar{t}$) ↓	FPR ($h \to 4b$) ↓	TNR↑	FNR↓	Prec.(Combined) ↑	Prec.($t\bar{t}$) ↑	Prec.($h \to 4b$) ↑	F1(Combined) ↑	F1($t\bar{t}$) ↑	F1($h \to 4b$) ↑
H_T	Constant	0.568	0.913	0.223	**0.001**	**0.001**	**0.001**	**0.999**	0.432	**0.999**	**0.999**	**0.994**	0.724	0.954	0.364
H_T	PID	0.654	0.975	0.333	0.002	0.002	0.002	0.998	0.346	0.998	0.997	0.993	0.790	0.986	0.499
H_T	DSPOT	0.667	0.940	0.282	0.002	0.002	0.002	0.998	**0.333**	0.998	0.998	0.992	**0.800**	**0.989**	**0.510**
H_T	ADT	0.649	0.971	0.328	0.002	0.002	0.002	0.998	0.351	0.998	0.998	0.993	0.787	0.984	0.493
H_T	DQN	0.653	0.975	0.331	0.002	0.002	0.002	0.998	0.347	0.998	0.997	0.993	0.790	0.986	0.496
H_T	PPO	0.653	0.975	0.331	0.002	0.002	0.002	0.998	0.347	0.998	0.997	0.993	0.790	0.986	0.496
H_T	GRPO	0.653	0.975	0.331	0.002	0.002	0.002	0.998	0.347	0.998	0.997	0.993	0.789	0.986	0.498
H_T	L-GRPO	0.654	0.977	0.332	0.002	0.002	0.002	0.998	0.346	0.998	0.997	0.992	0.790	0.987	0.490
H_T	CPO	0.651	0.974	0.329	0.002	0.002	0.002	0.9975	0.349	0.998	0.997	0.992	0.787	0.985	0.487
H_T	GFPO-F	0.654	0.977	0.332	0.002	0.002	0.002	0.998	0.346	0.998	0.997	0.993	0.791	0.987	0.499
H_T	GFPO-FR	0.656	**0.978**	**0.334**	0.003	0.003	0.003	0.997	0.344	0.998	0.997	0.993	0.791	0.987	0.499
AD	Constant	0.423	0.625	0.221	**0.001**	**0.001**	**0.001**	**0.999**	0.577	**0.999**	**0.998**	0.995	0.594	0.769	0.362
AD	PID	0.573	0.753	0.392	0.002	0.002	0.002	0.998	0.427	0.998	0.997	0.994	0.728	0.858	0.563
AD	DSPOT	0.688	0.810	**0.567**	0.002	0.002	0.002	0.998	0.312	0.998	0.997	**0.996**	**0.815**	**0.893**	**0.722**
AD	ADT	0.579	0.758	0.399	0.003	0.003	0.003	0.997	0.421	0.998	0.997	0.994	0.733	0.861	0.570
AD	DQN	0.570	0.751	0.390	0.002	0.002	0.002	0.998	0.430	0.998	0.997	0.994	0.726	0.857	0.560
AD	PPO	0.584	0.762	0.406	0.003	0.003	0.003	0.997	0.416	0.998	0.997	0.994	0.737	0.864	0.577
AD	GRPO	0.574	0.756	0.391	0.002	0.002	0.002	0.998	0.426	0.998	0.997	0.994	0.728	0.860	0.561
AD	L-GRPO	0.595	0.757	0.433	0.002	0.002	0.002	0.998	0.405	0.998	0.997	0.993	0.745	0.859	0.589
AD	CPO	0.594	0.756	0.432	0.002	0.002	0.002	0.998	0.406	0.998	0.997	0.993	0.740	0.859	0.586
AD	GFPO-F	0.580	0.758	0.402	0.002	0.002	0.002	0.998	0.420	0.998	0.997	0.994	0.734	0.861	0.573
AD	GFPO-FR	0.581	0.759	0.403	0.002	0.002	0.002	0.998	0.419	0.998	0.997	0.994	0.734	0.862	0.573

E. Additional Implementation Details

E.1. Baseline Adaptation

Stream chunking. State features are computed over 5K-event windows; threshold updates Δ are applied per 50K-event chunk, following the PID setting of Emami et al. (2026). This yields 195 chunks for MC and 99 chunks for CMS Run 283408 (Table 3).

E.1.1. ADT

We adapt the ADT controller of Yang et al. (2024) to streaming trigger control; four changes are required.

Objective. Vanilla ADT classifies fixed time-series windows as normal/abnormal (10 or 12 classes, dataset-dependent). We instead regulate the per-event accept rate to a target r_B^* within tolerance τ, with decisions at event granularity rather than window granularity.

State. ADT's state vector includes the running $\{\rho_t^{TP}, \rho_t^{TN}, \rho_t^{FP}, \rho_t^{FN}\}$ over the last k windows, quantities that require ground-truth labels at decision time. In LHC trigger control no such labels exist: signal/background identity is recovered only after the full event stream is processed. We replace ADT's classification-feedback state with the rate-regulation state of BUILDSEQUENTIALSTATE (Algorithm 2).

Episode/step mapping. We identify each "episode" with one chunk and each RL step with one micro-step (a stride over events within the chunk). This preserves ADT's end-of-episode update rule while operating on a long event stream rather than fixed time-series windows.

Action space. ADT uses binary $\delta \in \{0, 1\}$. Triggers are continuous-valued (H_T in GeV, AD_{cut} in score units), so we discretize δ to match DQN and GRPO. We retain ADT's update cadence ($l = 10$ micro-steps; Yang et al. (2024) default), reusing a_{t-1} on intermediate steps. Other baselines (DQN, GRPO, GFPO-F, GFPO-FR) update every 5K-event window.

Safety shield. For all RL methods we add a safety shield that overrides the policy when its action would require *too much* threshold update. Vanilla ADT does not include such a shield; we add it because rate-budget violations trigger DAQ deadtime at the LHC (Racz, 2000; Bertelsen et al., 2016).

E.1.2. DSPOT

Algorithm. DSPOT (Siffer et al., 2017) maintains a sliding reference window of the most recent n observations and models the excess distribution above an intermediate threshold t as a Generalized Pareto Distribution (GPD). It updates the GPD parameters $(\hat{r}, \hat{\sigma})$ via maximum likelihood at each step and returns

$$z_q = t + \frac{\hat{\sigma}}{\hat{\gamma}} \left[\left(\frac{q \cdot n}{N_t} \right)^{-\hat{\gamma}} - 1 \right], \tag{15}$$

where N_t is the number of observations above t in the reference window and q is the target false-positive risk.

Adaptation. At each chunk boundary we pass background scores from the most recent window to DSPOT and use the returned z_q as the next-chunk trigger threshold c_t, with $q = r_B^* = 0.25\%$. Reference window size matches our chunk size (50K events MC, 20K CMS Run 283408); the intermediate threshold t is initialized at the 98th percentile of the calibration window (Siffer et al., 2017). No reward, policy network, or gradient update is involved.

No safety shield. DSPOT is a one-sided rate-compliance algorithm: it controls the upper tail of the score distribution to bound the false-positive rate, but has no mechanism to keep the achieved rate within a two-sided tolerance band $[r^-, r^+]$, and no objective ranking among rate-compliant thresholds. We deliberately do not impose a safety shield on its output. Out-of-band chunks therefore appear directly in the InBand fraction (Table 1: 39.8% H_T, 45.2% AD). This is the honest evaluation: DSPOT was designed for unsupervised anomaly detection in unlabelled streams (Siffer et al., 2017), where signal efficiency is undefined. Our RL methods, by contrast, enforce the operational rate constraints arising from detector readout bandwidth (Aaboud et al., 2017) and computing resources (Collaboration et al., 2020) via a safety shield, and then *rank* feasible thresholds by signal efficiency.

H_T trigger. Pileup increases H_T systematically across the full score distribution, since soft radiation from pileup vertices contributes to all jet p_T measurements. DSPOT's tail estimator is therefore less well-matched to H_T than to AD, where pileup manifests predominantly as a shift in the reconstruction-loss tail. We include DSPOT on both triggers for completeness; the asymmetry is itself informative about the qualitative difference in non-stationarity between the two trigger paths.

MC-only setting. In Table 1, calibration and deployment share the same MC stream. DSPOT achieves InBand fractions of 45.2% (AD) and 39.8% (H_T), with InBand signal efficiencies of 93.7% and 98.1% on $t\bar{t}$ close to the PID ceiling on both triggers. The InBand-fraction gap to PID (66.1% AD, 59.1% H_T) reflects DSPOT's 50-chunk GPD calibration burn-in, during which the threshold is held fixed; PID adapts from chunk one.

Sim-to-real transfer. For MC-trained, CMS-deployed runs (Table 2), the GPD is fitted on MC and applied on CMS. The InBand fraction is largely preserved (41.9% AD, 43.2% H_T): DSPOT's 5-chunk sliding window re-fits the tail online and absorbs much of the sim-to-real shift. A single catastrophic chunk, however, pushes both MAE values past their P95 (AD: 0.106 vs. 0.087; H_T: 0.163 vs. 0.072), abrupt CMS rate excursions fall outside the support of the MC-calibrated GPD. PID's integral feedback degrades gracefully under the same shift; DSPOT's tail-fit estimator does not.

Test-time training. For test-time training (Table 12) DSPOT follows the two-phase protocol of Siffer et al. (2017): a calibration window accumulates background scores to fit the GPD; the sliding window then re-fits online each chunk (Figure 11). We use 50 CMS chunks at 20K events each (Emami et al., 2026), which exhausts $\approx 68\%$ of the available CMS data and leaves 24 chunks for deployment. During calibration DSPOT holds the MC-trained cut fixed, so it observes background rates of 0.113% (AD) and 0.150% (H_T), well below the 0.25% target (Figure 11). The GPD is therefore fitted to the tail of this low-rate regime. When DSPOT activates at chunk 50, the CMS score distribution has shifted upward: deployment means are 0.233% (AD) and 0.227% (H_T). The abrupt threshold drop at chunk 50 corrects for this mismatch, but 24 chunks are too few to recover. Over all chunks DSPOT reaches 13.5% and 17.6% InBand vs. PID's 40.5% and 43.2% (Table 2), without calibration burn-in.

Why DSPOT lacks signal efficiency. DSPOT fits a GPD to the background tail to enforce a target false-positive rate, with no signal knowledge at any stage. Among all rate-compliant thresholds, it has no basis for preferring the one that maximizes $t\bar{t}$ or $h \to 4b$ retention — it returns whatever the GPD estimate dictates. This is structural, not a tuning choice: the algorithm has *no* objective beyond rate control. Our two variants (GFPO-F and GFPO-FR) fill this gap by explicitly ranking feasible thresholds by signal efficiency at each step, recovering retention DSPOT forgoes while satisfying the same rate constraint.

Note on H_T trigger. For the H_T trigger, pileup increases H_T systematically across the full score distribution rather than primarily in the tail, since additional soft radiation from pileup vertices contributes to all jet p_T measurements. DSPOT's EVT-based tail estimator is therefore less well-matched to the H_T setting than to the AD setting, where pileup drift manifests predominantly as a shift in the reconstruction loss distirbution. We include DSPOT on both triggers for completeness; the performance asymmetry between the two setttings is itself informative about the qualitatively different nature of non-stationarity in each trigger path.

E.1.3. L-GRPO

We extend GRPO by replacing the fixed rate tracking penalty weight λ_1 with a dual variable λ_t that auto-tunes online. At each chunk boundary, after observing the background rate r_t, the dual variable is updated via a projected gradient ascent step on the Lagrangian:

$$\lambda_{t+1} = [\lambda_t + \alpha_{\text{step}}(|r_t - r_B^*| - \tau)]^+ \tag{16}$$

where $[\cdot]^+$ denotes projection onto $\mathbb{R}_{\geq 0}$ to maintain dual feasibility, $\alpha_{\text{step}} > 0$ is the dual step size, and τ is the tolerance width. Intuitively, λ_t increases when the background rate violates the tolerance band and decreases when the constraint is satisfied, automatically tightening or relaxing the rate penalty over time.

The reward in Eq 3 at step t shall be written as:

$$\begin{aligned} R_t^{\cdot} = \lambda_t & \underbrace{\begin{cases} 1 - \left(\dfrac{|r_{t+1} - r_B^\star|}{\tau}\right)^2 & \text{if } |r_{t+1} - r_B^\star| \leq \tau \\ -\left(\dfrac{|r_{t+1} - r_B^\star|}{\tau} - 1\right) & \text{if } |r_{t+1} - r_B^\star| > \tau \end{cases}}_{\text{rate tracking}} \\ & + (1 - \lambda_t) \underbrace{\left(\alpha\, \epsilon_{t+1}^{t\bar{t},\cdot} + (1-\alpha)\, \epsilon_{t+1}^{h\to 4b,\cdot}\right)}_{\text{signal efficiency}} \\ & - \lambda_2 \underbrace{\left(\frac{|\Delta c_{t-1}^{\cdot}|}{\Delta c_{\max}^{\cdot}}\right)}_{\text{move penalty}} \end{aligned} \tag{17}$$

The policy update follows the standard GRPO objective in Eq. 5 with rewards rewards recomputed under the current λ_t at each stepp.

Implementation details. We initialize $\lambda_0 = \lambda_1 = 0.25$, matching the fixed baseline for GRPO, GFPO-F and GFPO-FR, so that Lagrangian-GRPO starts from the same operating point and any subsequent improvement reflects the dual adaptation rather than initialization. We set the dual step size $\alpha_{\text{step}} = 0.01$ and clip $\lambda_t \in [0.05, 0.95]$ to prevent degeneracy: a near-zero λ_t would eliminate the rate constraint entirely, while $\lambda_t \approx 1$ would suppress the signal efficiency objective. The dual variable is updated once per chunk, operating on the same timescale as the threshold update, and is not updated during shielded steps. All other hyperparameters G, K, λ_2, α in the reward mixing) follow the GRPO settings in Appendix D.3.

Relationship to PID-Lagrangian methods. The update above corresponds to the integral term of a PID-Lagrangian controller (Stooke et al., 2020), where the dual step size α_{step} plays the role of the integral gain. We omit the proportional

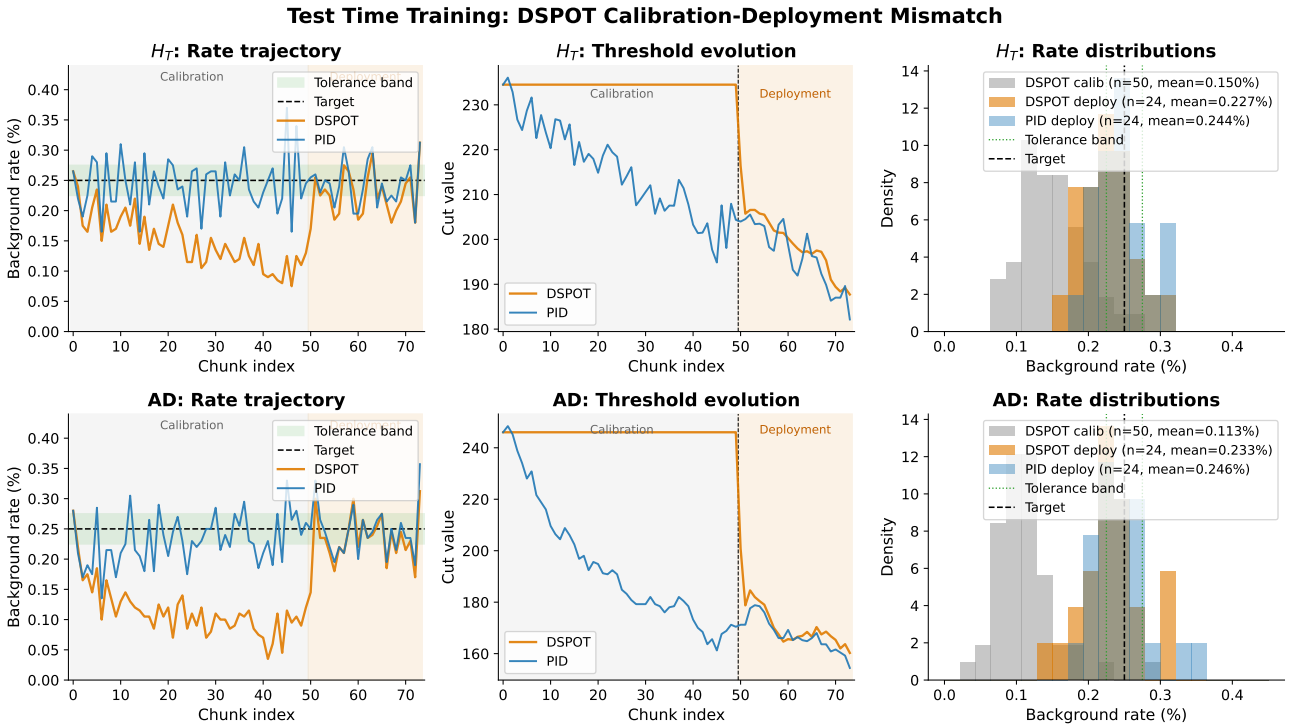


Figure 11. DSPOT calibration-deployment mismatch: the GPD is fitted on 50 calibration chunks where the background rate is 0.113% (AD) and 0.150% (H_T) under the fixed initial cut, well below the 0.25% target, causing a distributional shift of $+0.12\%$ and $+0.08\%$ at deployment that limits the inband fraction to 13.5% and 17.6% over the reamining 24 chunks, while PID tracks the target continuously without burn-in phase.

and derivative terms for simplicity, as the GRPO policy already observes the instantaneous rate error and drift through the state representation (Appendix E.3), making separate proportional and derivative correction in the dual redundant.

Structural Failure Analysis of L-GRPO. We provide empirical support for the structural argument in Section 3.2 that L-GRPO's dual variable λ_t cannot resolve the zero-feasibility failure mode. All diagnostic runs use the AD trigger on CMS real collison data (Table 2), which represents the harder case (54.1% in-band). The H_T trigger shows qualitatively identical behavior on par with GRPO baseline and is omitted for brevity.

Zero-feasible fraction is structurally unchanged by λ_t (Figure 12). Both GRPO and L-GRPO yield $\approx 20\%$ zero-feasible steps on MC and $\approx 61\%$ on CMS, with per-chunk trajectories that are nearly indistinguishable (Figure 12b).

Since λ_t modulates reward weights only, it has no effect on the proposal distribution of G and cannot reduce the fraction of steps with zero feasible candidates.

λ_t adapts during violations but the in-band rate does not recover (Figure 13). On CMS, λ_t remains active near 0.25 and responds to violation windows, yet the in-band rate shows no systematic recovery and oscillates around 54% throughout. When the sampled group contains no feasible action, group-relative normalisation assigns positive advantage to the least-infeasible out-of-band candidate regardless of λ_t; increasing λ_t amplifies this pathological update rather than correcting it.

L-GRPO's rate trajectory is systematically below the tolerance band on CMS (Figure 14). Despite sharing the same MC-pretrained weights as GRPO at deployment, L-GRPO drifts persistently below the lower band edge while GRPO and both GFPO variants remain near the target rate. The gap isolates the reward formulation as the source of failure under distribution shift, and motivates the feasibility-first candidate selection of GFPO (Section 4).

E.1.4. CPO

Implementation details. We set $\lambda_1 = 0.25$ (in CPO this is a fixed reward weight, distinct from the trust-region dual $\nu^\star$ that the QP solves for at each update). CPO inherits the signal and move-penalty pieces of Eq. 3 but moves the rate-tracking term from the reward into the cost constraint. Concretely, for a candidate Δ-cut δ_k at time t we evaluate

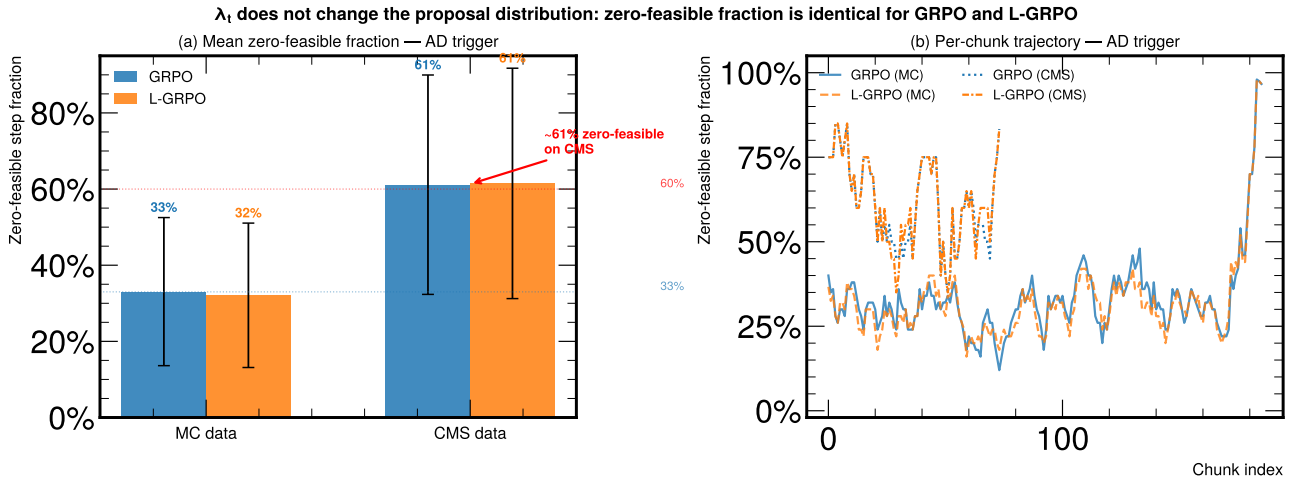


Figure 12. **Zero-feasible step fraction for GRPO and L-GRPO (AD trigger).** (a) Mean zero-feasible fraction across all chunks. On MC data, approximately 20% of micro-steps yield no rate-feasible candidate from the sampled group of G actions for both GRPO and L-GRPO. On CMS real data this rises to $\sim$61% for both methods. (b) Per-chunk trajectory over time. The two curves are nearly indistinguishable in both datasets, confirming that the dual variable λ_t modulates reward weights only and has no effect on the proposal distribution or the fraction of feasible candidates.

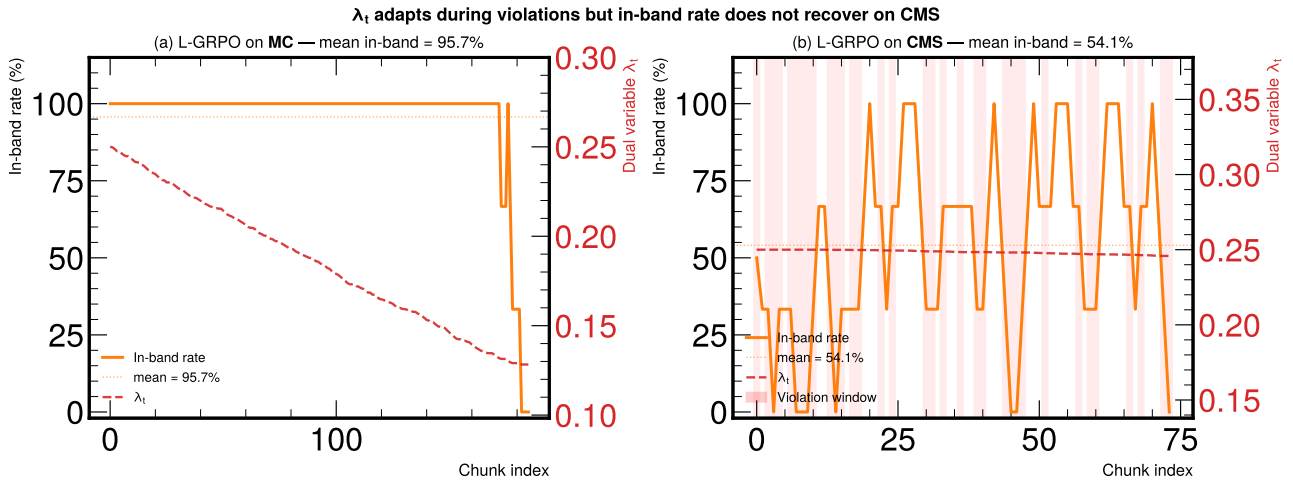


Figure 13. **Dual variable λ_t and in-band rate over chunks for L-GRPO (AD trigger).** (a) On MC data, λ_t decreases monotonically from its initial value of 0.25 as the policy stays in-band ($\sim$96% of chunks). (b) On CMS real collision data, λ_t fluctuates but remains roughly constant (≈ 0.25) while the in-band rate oscillates around 54%. Shaded regions mark chunks where the in-band rate is below 50%. Despite λ_t responding to these violation windows, the in-band rate does not systematically recover, confirming that increasing λ_t amplifies the pathological group update rather than correcting the underlying sampling failure.

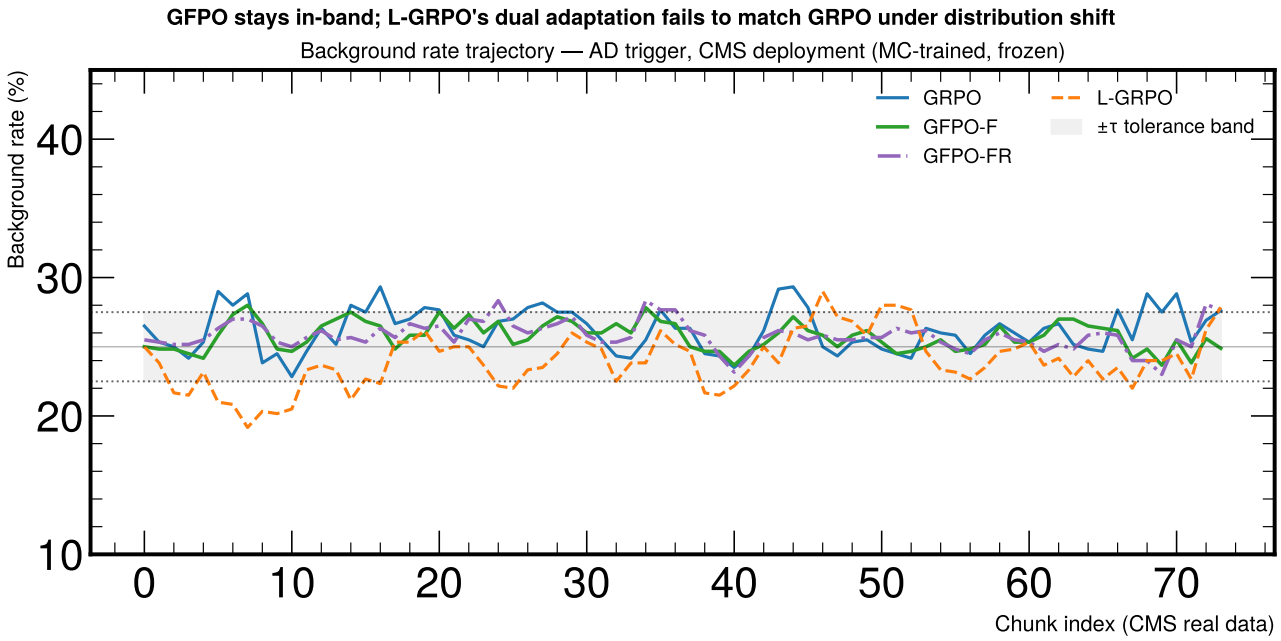


Figure 14. **Background rate trajectory on CMS real data (AD trigger, MC-trained frozen policy).** Grey band: $\pm\tau$ tolerance around the target r_B^*. GFPO-F and GFPO-FR track within the band for most of the 73-chunk deployment window. GRPO and L-GRPO sit near or below the lower band edge throughout, despite L-GRPO's dual variable adapting concurrently (Figure 13). The performance gap is therefore *not* a matter of insufficient penalty strength but of the GRPO candidate-sampling mechanism failing to produce feasible actions under distribution shift.

reward (with capital letters)

$$R_k \;=\; (1-\lambda_1)\,\underbrace{\left(\alpha\,\epsilon_{t+1}^{t\bar{t}} + (1-\alpha)\,\epsilon_{t+1}^{h\to 4b}\right)}_{\text{signal efficiency}} \;-\; \lambda_2\,\underbrace{\frac{|\Delta c_{t-1}|}{\Delta c_{\max}}}_{\text{move penalty}} \tag{18}$$

and cost (with capital letters)

$$C_k \;=\; \begin{cases} \left(|r_{t+1} - r_B^\star|/\tau\right)^2 & \text{if } |r_{t+1} - r_B^\star| \le \tau, \\ |r_{t+1} - r_B^\star|/\tau & \text{if } |r_{t+1} - r_B^\star| > \tau, \end{cases} \tag{19}$$

where r_{t+1} is the realized background rate after the candidate cut is applied. The cost is the sign-flipped, vertically shifted version of Eq. 3's rate-tracking term: it equals 0 at the target, ramps quadratically to 1 at the band edge, and grows linearly with the relative violation beyond, remaining non-negative and continuous at $e_t = 1$. The constraint enforced by CPO is on the expected per-step cost,

$$J_C(\pi_\theta) \;=\; \mathbb{E}_{(s,a)\sim\pi_\theta}\big[C(s,a)\big] \;\le\; d, \tag{20}$$

with d a hyperparameter (the per-step rate-violation budget). All other terms (α, λ_1, λ_2, τ, $r_B^\star$, $\Delta c_{\max}$) keep the values used elsewhere in the paper. We set $d = 1$, the cost evaluated at the tolerance-band edge, so that any operating point with average rate violation within τ is constraint-feasible and CPO is not driven toward the recovery branch by its budget alone.

CPO update. We adapt CPO (Achiam et al., 2017) to the bandit form of our setup. At each micro-step we sample $G = 16$ candidate Δ-cuts $a_k \sim \pi_\theta(\cdot|s)$, evaluate (r_k, c_k) under the simulator (Eqs. 18–19), and form *group-relative* advantages $\hat{A}_k^R = R_k - \bar{R}$, $\hat{A}_k^C = C_k - \bar{C}$, replacing the separate value-function critics used in the original CPO; like the GRPO/GFPO baselines, the bandit reduction removes the unreliable value-estimation step in the rare-signal regime. Every 50 micro-steps (with at least 128 samples buffered) we form the linearized trust-region program

$$\max_{\theta'}\; g^\top(\theta'-\theta) \quad \text{s.t.} \quad c + b^\top(\theta'-\theta) \le 0, \quad \tfrac{1}{2}(\theta'-\theta)^\top H(\theta'-\theta) \le \delta_{\mathrm{KL}}, \tag{21}$$

where g and b are the gradients of the importance-weighted reward and cost surrogates at θ, $c = J_C(\pi_\theta) - d$, and $H = \nabla_\theta^2 \mathrm{KL}(\pi_{\text{old}} \,\|\, \pi_\theta)$ is the Fisher matrix. We evaluate Fisher–vector products through mean-KL Hessian–vector products, solve $H^{-1}g$ and $H^{-1}b$ by conjugate gradient (10 iterations, damping 0.1), and obtain the dual variables $(\lambda^\star, \nu^\star)$ in closed form following Achiam et al. (2017, Sec. 6.2). A backtracking line search (initial step 1.0, decay 0.8, up to 10 backtracks) accepts the update if the measured KL stays within $1.5\,\delta_{\mathrm{KL}}$, the reward surrogate does not regress, and the projected cost stays under budget. When $c > 0$ and $2\delta_{\mathrm{KL}} - c^2/s < 0$ the program is infeasible and we apply the recovery step (Eq. 14 of Achiam et al., 2017), $\theta' - \theta = -\sqrt{2\delta_{\mathrm{KL}}/s}\,H^{-1}b$, which descends purely along

the cost direction; the line search is then required only to reduce the cost surrogate. We train on 3 seeds over the full MC trajectory using a single configuration ($\delta_{\mathrm{KL}} = 0.03$, $d = 1$, $G = 16$, batch threshold 128) and do not perform a hyperparameter sweep; the values chosen are conservative defaults from Achiam et al. (2017) for δ_{KL} and the CG/line-search budgets, with $d = 1$ matching the cost at the band edge.

E.2. GRPO Failure Mode

This appendix supplements the H_T analysis in the main text (Figures 2, 3) with the parallel analysis on the AD trigger.

Failure does not vanish with group size. Figure 15 plots the fraction of training steps with zero feasible rollouts as a function of $G \in \{4, 8, \ldots, 256\}$. The AD curve is flat at 34–36% across all G; H_T plateaus above 24% even at $G = 256$. If zero-feasibility were a sampling-variance artifact, scaling G would drive the fraction to zero. It does not: zero feasibility is a property of the policy distribution under the constraint, not of the estimator's sample size.

E.3. Sequential Network Architectures

Why a sequence encoder. The trigger-control problem is partially observed and non-stationary: at update step t we observe a finite recent window of background events, while the threshold-to-rate mapping drifts with pileup (N_{pv}), detector conditions, and score-distribution shifts. A scalar controller (e.g., PID (Emami et al., 2026)) sees only the instantaneous rate error. We need a state that exposes the recent evolution of the event-level distribution near the cut — near-cut occupancy, tail thickness, and the conditional dependence of pass indicators on score and kinematics. Algorithm 2 presents the full pipeline of building sequential state for all RL policy.

State. At update step t, the state is a fixed-length sequence $S_t \in \mathbb{R}^{K \times F}$ built from the most recent K background events, with F features per event (Algorithm 2). Features fall into three groups, broadcast across the sequence:

- **Event-level**: score and kinematics; pileup context.
- **Threshold geometry**: distance-to-cut; pass indicator.
- **Chunk-level control**: rate error, drift, last action, feasibility.

Architecture. All RL agents in this work share the same sequential architecture for both state encoding and policy. The same input format and network apply to the H_T and AD triggers.

Setup. With $\lambda_1 = 0.25$, $\lambda_2 = 1.00$ fixed (Section D.3), we ablate the recurrent cell in the shared sequence encoder. All five RL agents (DQN, GRPO, GFPO-F, GFPO-FR, PPO) use the same architecture: a single-layer RNN followed by a linear projection head, trained on MC and deployed on CMS Run 283408.

RNNs compared. LSTM (4 gates, 7,333 parameters) (Graves, 2012); GRU (3 gates, 5,541 parameters) (Dey & Salem, 2017); vanilla RNN with $\tanh$ (1,957 parameters); vanilla RNN with ReLU (1,957 parameters).

Results. Table 10 reports means over the five agents. On the H_T trigger, signal efficiencies are nearly architecture-invariant: $t\bar{t}$ overall varies by under 0.1% (97.57–97.64%) and $h \to 4b$ overall by under 0.1% (33.29–33.38%) across all four cells. InBand rate is the discriminating axis: GRU reaches 0.849, vanilla RNN 0.789.

Choice. Trigger rate scales with instantaneous luminosity (Aad et al., 2022; Collaboration et al., 2020; Govorkova et al., 2022), so rate stability is operationally critical. We therefore prioritize InBand rate and adopt GRU as stated in Section 4.

State representation. We summarize the K-event window with F features per event (Algorithm 2). Features fall into three groups, listed in Table 11: per-event physics features, chunk-level control state (broadcast across all K events), and multi-resolution near-cut occupancy. We justify the non-obvious choices below; the same input format is used for both the H_T and AD triggers, though constructed separately. Total dimensionality is $F = 22$ per timestep $(5 + 14 + 3)$.

Widths are $(5, 10, 20)$ GeV for H_T and $(0.25, 0.5, 1.0)$ for the AD score. mid and span are computed once at the start of the first chunk: $\mathrm{mid} = 0.5(P_{95} + P_{99.99})$ and $\mathrm{span} = P_{99.99} - P_{95}$ over the calibration window, analogous to how a static LHC trigger menu is set.

Normalized threshold (Figure 19). Pileup drift makes the percentile rank of a fixed raw threshold non-stationary (top row): the same c produces different rates over time. Normalization by the calibration mid and span stabilizes the threshold representation; even after normalization, the ideal operating point drifts (bottom row), so the agent must observe c_{norm} to know whether its threshold sits high or low relative to the score distribution. Without it, the agent would observe only the rate error, purely reactive, not anticipatory.

Window position. The per-event index $\tau_i = (i-1)/(K-1) \in \{0, \frac{1}{K-1}, \ldots, 1\}$ provides an explicit recency signal, freeing the GRU from having to recover position informa-

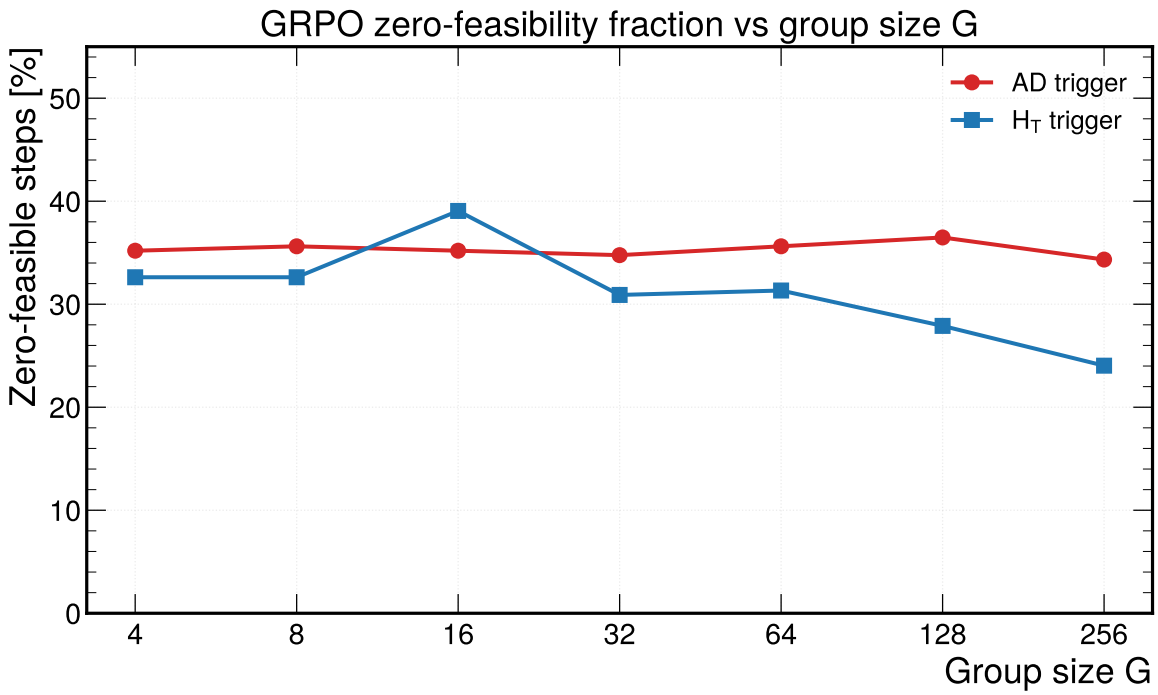


Figure 15. **Zero-feasibility fraction of GRPO vs. group size G.** Fraction of training steps on which every rollout violates the rate budget, for AD and H_T triggers. Even at $G = 256$, 24–34% of steps yield no feasible sample, so the group-relative baseline collapses to an uninformative signal on a constant fraction of updates. Scaling G alone cannot recover constraint satisfaction, motivating the constraint-aware variants of GRPO in Section 4.

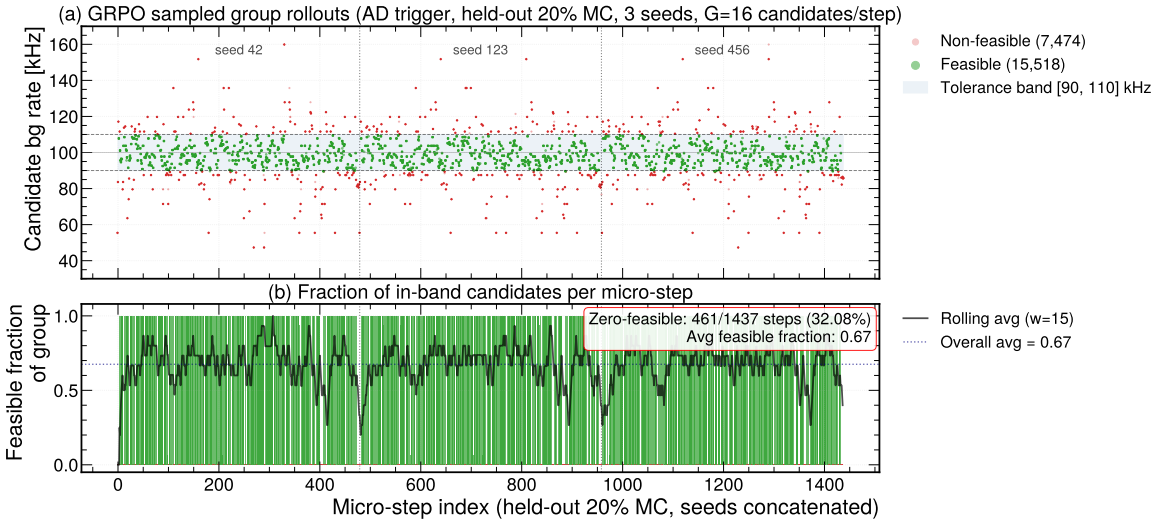


Figure 16. **GRPO group-feasibility failure (AD trigger, MC, G=16); appendix complement to Fig. 2.** (a) Background rates of all G=16 candidate rollouts per micro-step; green/red denotes feasible/infeasible w.r.t. the $[90, 110]$ kHz band. (b) Per-step feasible fraction $f_t = n_{\text{feas}}/G$ (run mean $\langle f_t \rangle = 0.67$). On 32.1% of steps $f_t = 0$: every candidate lies outside the band, the within-group advantage has zero variance, and the policy receives no feasibility signal. A 67% in-band mean coexisting with a $\sim 32.08\%$ zero-feasible tail is the regime where group-relative updates fail intermittently, motivating the feasibility-filtered GFPO update.

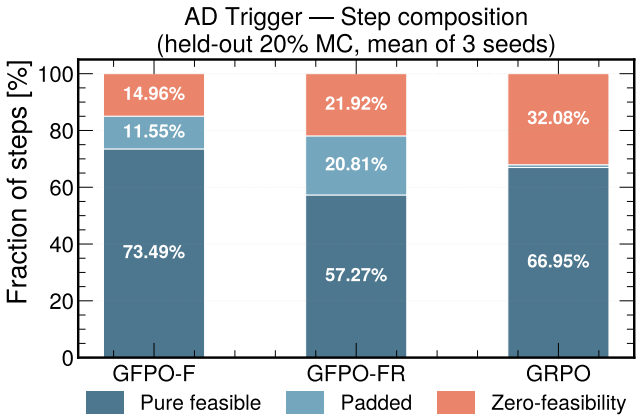


Figure 17. **Step composition for the AD trigger; appendix complement to Fig. 3.** Categories follow Fig. 3: *Pure feasible* ($|\mathcal{F}_t| \geq K$), *Padded* $0 < |\mathcal{F}_t| < K$), *Zero-feasibility* ($\mathcal{F}_t = \emptyset$). GFPO-F/FR achieve $\sim$57 73% pure-feasible kept sets and pad on 11–21% of steps. GRPO is dominated by partial-feasibility (66.95%) and zero-feasibility (32.08%) steps; the latter matches the value in Fig. 16 and is the failure mode shielded by GFPO.

Table 10. State representation ablation on CMS real data (Run 283408). All models trained on 80% MC with λ_1=0.25, λ_2=1.00. Results averaged across five RL methods (DQN, GRPO, GFPO-F, GFPO-FR, PPO).

Trigger	RNN	Params	Gates	InBand	$\epsilon^{t\bar{t}}_{\text{inband}} \uparrow$	$\epsilon^{h\to 4b}_{\text{inband}} \uparrow$	$\epsilon^{t\bar{t}}_{\text{overall}} \uparrow$	$\epsilon^{h\to 4b}_{\text{overall}} \uparrow$
H_T	LSTM	7,333	4	0.830	97.50	33.07	97.57	33.29
H_T	RNN	1,957	1	0.789	**97.68**	**33.60**	**97.64**	**33.38**
H_T	RNN–ReLU	1,957	1	0.800	97.64	33.54	97.60	33.36
H_T	GRU	5,541	3	**0.849**	97.48	33.06	97.57	33.30
AD	LSTM	7,333	4	0.570	75.89	43.01	75.30	39.80
AD	RNN	1,957	1	0.570	75.94	43.22	75.29	39.84
AD	RNN–ReLU	1,957	1	0.568	75.89	43.24	**75.34**	**39.86**
AD	GRU	5,541	3	**0.573**	**76.02**	**43.32**	75.31	39.81

tion from its hidden-state recurrence alone. This follows the positional-encoding inductive bias of Vaswani et al. (2017) (and its time-series extensions, e.g., Kazemi et al. (2019); Lim et al. (2021)); we broadcast the same ramp identically every chunk.

EMA rate error (Figure 21). The instantaneous error is window-to-window noisy; the EMA at $\lambda = 0.95$ smoothly tracks the sustained drift out of the tolerance band, analogous to the integral term in PID. This gives the agent memory of accumulated deviations and distinguishes transient fluctuations from regime shifts.

Sensitivity probe (Figure 22). A central-difference estimate of $\partial r/\partial c$ gives the agent local gradient information about the trigger curve: a steep slope means small threshold changes produce large rate effects (fine adjustments needed); a shallow slope means the agent can move further safely.

Pileup features (Figure 20). N_{pv} distributions drift over the run; including window mean and std lets the agent condition on the current pileup regime.

Near-cut indicators (Figure 23). Multi-resolution counts of how many events sit close to the cut predict how sensitive the rate is to small threshold adjustments.

Look-ahead rate and survival ratios (Figure 18). The look-ahead rates $p_1 = r(c{+}\Delta)$ and $p_2 = r(c{+}2\Delta)$ evaluate the background rate at the next two grid points along the discrete action axis ($\Delta = 1$ GeV for H_T, $\Delta = 0.5$ for AD); the survival ratios $r_1 = p_1/p_0$ and $r_2 = p_2/p_1$ report the rate decay between consecutive shifts, where $p_0 = r(c) \approx r^*_B$ at the natural operating point $c = P_{99.75}$. Where the central-difference sensitivity probe captures the local slope at c, (p_1, p_2, r_1, r_2) describes the *shape* of the cut neighborhood reachable in one or two actions: a thin tail produces small r_1, r_2 (one upward step removes most of the rate); a fat tail produces large r_1, r_2 (several steps needed). The four scalars drift across the run on both triggers, motivating their inclusion as state inputs.

Table 11. State features. μ, σ: window mean and std of the trigger observable. r_t: chunk-t background rate. r_B^*: target rate. c: current threshold.

Group	Feature	Definition
Per-event physics	Normalized observable	$(x_i - \mu)/\sigma$
	Pass flag	$\mathbf{1}[x_i \geq c]$
	Signed distance to cut	$(x_i - c)/\sigma$
	Normalized $N_{pv,i}$	$(N_{pv,i} - \overline{N_{pv}})/\mathrm{std}(N_{pv})$
	Window position	$\mathrm{linspace}(0, 1, K)$
Chunk-level (broadcast)	Relative rate error	$(r_t - r_B^*)/r_B^*$
	Absolute rate error	$\lvert r_t - r_B^* \rvert / r_B^*$
	Rate drift	$(r_t - r_{t-1})/r_B^*$
	InBand flag	$\mathbf{1}[\lvert r_t - r_B^* \rvert \leq \tau]$
	Normalized threshold	$(c_t - \mathrm{mid})/\mathrm{span}$
	Last action	$\Delta c_{t-1}/\Delta c_{\max}$
	Pileup mean	$\overline{N_{pv}}$ over window
	Pileup std	$\mathrm{std}(N_{pv})$ over window
	EMA rate error (λ=0.95)	$\mathrm{EMA}_\lambda[(r_t - r_B^*)/r_B^*]$
	Sensitivity probe	$\partial r/\partial c$, central-difference, normalized by r_B^*
	Look-ahead rate p_1	$r(c + \Delta)$
	Look-ahead rate p_2	$r(c + 2\Delta)$
	Survival ratio r_1	$r(c + \Delta)/r(c)$
	Survival ratio r_2	$r(c + 2\Delta)/r(c + \Delta)$
Near-cut indicators	Width w_1	$\mathbf{1}[\lvert x_i - c \rvert \leq w_1]$
	Width w_2	$\mathbf{1}[\lvert x_i - c \rvert \leq w_2]$
	Width w_3	$\mathbf{1}[\lvert x_i - c \rvert \leq w_3]$

F. Additional Experimental Results

F.1. Additional DQN results

Here, in this section, we want to demonstrate even with off-policy RL that cannot handle distribution shifts well, simple RL policy DQN shall boost signal efficiency over the trajectory compared to fixed predefined PID loop.

DQN Background-rate trajectories. For clarity, we plot out DQN and DQN-F qualitative results for per-timestep trajectories to demonstrate simple DQN can boost in band rate visually. Figure 24 traces the per-chunk background trigger rate over a simulated MC run for the H_T trigger and the anomaly-detection (AD) trigger. The Constant controller is set at the start of the run and never updates; pileup drift pulls its rate steadily downward, the canonical failure mode of static thresholds under non-stationary input rates. PID reacts to the rate signal but oscillates around the tolerance band, repeatedly overshooting and undershooting. DQN tracks the band more consistently, and DQN-F (the feasibility-filtered variant) further suppresses out-of-band swings and recovers faster from pileup transitions. The two panels show qualitatively the same pattern, indicating that the controller-level differences are not specific to the choice of physics observable.

F.2. GRPO MC training

Beyond vanilla DQN, we evaluate the GRPO family (GRPO, GFPO-F, GFPO-FR). Figures 25 and 26 show the background-rate distribution and per-timestep trajectory for each controller on MC. GRPO, GFPO-F, and GFPO-FR concentrate near the 100 kHz target in both views; Constant decays as pileup drifts, and PID oscillates with frequent excursions outside the band.

F.3. Sim-to-Real Transfer and Test Time Adaptation

Figure 27 shows background rates for all methods under test-time adaptation on CMS Run 283408, where models are trained on MC samples and deployed on real CMS data. Notably, the results are nearly *identical* to those in Figure 28, suggesting that test-time adaptation achieves performance on par with test-time training. Therefore, we answer the question raised in the main paper: test-time adaptation, training on MC simulations and deploying on real CMS collision data, is a preferable and practical alternative to test-time training.

F.4. Testing Time Training on Real CMS Data

We include a study on test-time training over real CMS collision data to evaluate whether adaptive thresholding remains effective beyond the controlled assumptions of simulation. While Monte Carlo studies are useful for development, real detector streams can exhibit mismatches in feature distri-

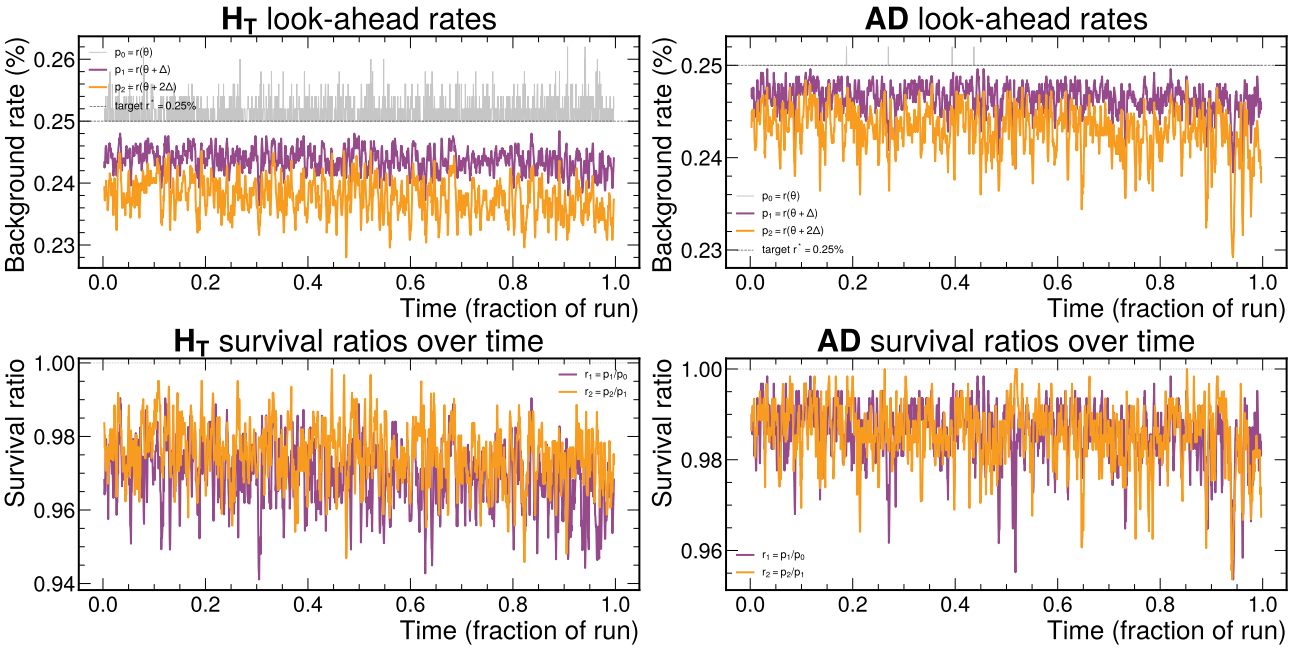


Figure 18. Look-ahead rates (p_1, p_2) and survival ratios (r_1, r_2) over time, evaluated at each window's natural operating point $\theta = P_{99.75}$ so that $p_0 \approx r_B^*$ by construction. The four scalars give the agent a discrete forward-rate map of the tail at $\theta + \Delta$ and $\theta + 2\Delta$ (with $\Delta = 1$ GeV on H_T and $\Delta = 0.5$ on AD), complementing the local sensitivity probe $\partial r/\partial\theta$. Their drift over the run motivates including (p_1, p_2, r_1, r_2) as state inputs.

butions, background composition, detector response, and running conditions that are difficult to model perfectly (cf. Figure 5 and Figure 6) (Gavrikov et al., 2025). This concern is especially pronounced for anomaly-detection triggers, whose decisions are made through a threshold on a learned score distribution. Since CMS has already demonstrated anomaly-detection algorithms in realistic Level-1 trigger settings on live data (Gandrakota, 2024; CMS Collaboration, 2024a; Zipper & collaboration, 2024), it is important to test whether an RL agent can adapt thresholds online when exposed to the variability of real data rather than only simulated samples.

GFPO-F and GFPO-FR remain strongly competitive on real CMS collision data, achieving the best overall stability–efficiency tradeoff for both trigger settings. We present online training per time chunk in Table 12. Compared with PID, DQN, PPO, ADT, and GRPO, our proposed GFPO variants are more consistently strong across both control and physics metrics on real CMS data. Classical PID can still be competitive on isolated metrics, but its performance is less balanced overall. In contrast, GFPO-F and GFPO-FR repeatedly appear at or near the top across in-band occupancy and signal efficiency, indicating better robustness and adaptivity to real-time collision data variability. Figure 28 presents the trajectory of background rates for all methods under test time training on CMS Run 283408 (i.e., directly learning an RL policy on real collision data). Remarkably, the frozen MC-trained rollout (Table 1) and the test-time training (Table 12 and Figure 28) achieve nearly *identical* performance on CMS data, InBand rate, and signal efficiency differ by less than 2 percentage points across all methods and triggers. This demonstrates that the online, per-chunk threshold adaptation mechanism (adjusting the cut c_t at each micro-step) is the primary driver of domain transfer, not continued weight updates. The policy network learned on MC generalizes well enough such that the action-level adaptation (selecting Δc to stay within the background rate tolerance band) compensates for the MC to CMS distribution shift without requiring gradient-based fine-tuning at deployment time.

F.5. CPO degeneracy on the CMS dataset

The CPO entries in Tables 2 and 12 agree to floating-point precision, with *zero* per-seed standard deviation. This is structural rather than a seeding artifact: the three trained checkpoints differ in π_θ, and every other RL baseline on the same protocol shows nonzero seed spread. The cause is that CPO's executed trajectory is screened from the policy.

Setup. At each micro-step CPO samples G=16 candidate actions $a_k \sim \pi_\theta(\cdot \mid s)$ from a discrete set $\mathcal{A} = \{-3, -1.5, 0, 1.5, 3\}$ (AD) or $\{-2, -1, 0, 1, 2\}$ (H_T) of action space size $|\mathcal{A}| = 5$, evaluates each candidate's reward

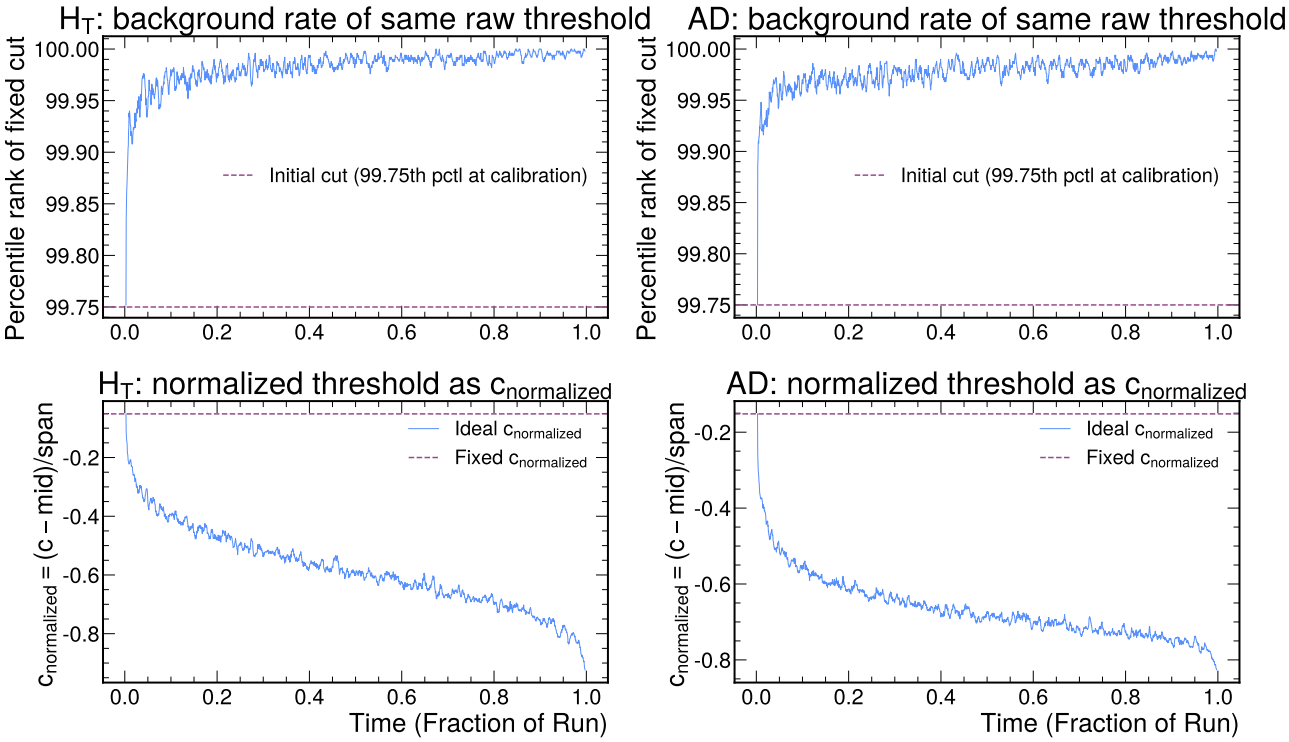


Figure 19. Non-stationarity of trigger thresholds under pileup drift (MC). **Top row**: The percentile rank of a fixed raw threshold (calibrated to the 99.75th percentile of the first 50k events) drifts substantially over time for both H_T (left) and AD (right), meaning the same raw threhsold yields a changing background rate. **Bottom row**: The normalized threshold $c_{\text{normalized}}$ where mid and span are derived from the P95-P99.99 calibration range of the first 50k events, illustrates that the ideal operating point (blue) deviates from the fixed calibration value (red dashed line). The ideal $c_{\text{normalized}}$ (blue) is calculated as the 99.75th percentile of each 50k events' score distribution. This is the threshold that would give exactly $r^* = 0.25\%$. This suggests including $c_{\text{normalized}}$ as a state feature so the agent can determine where c_t sits relative to the score distribution.

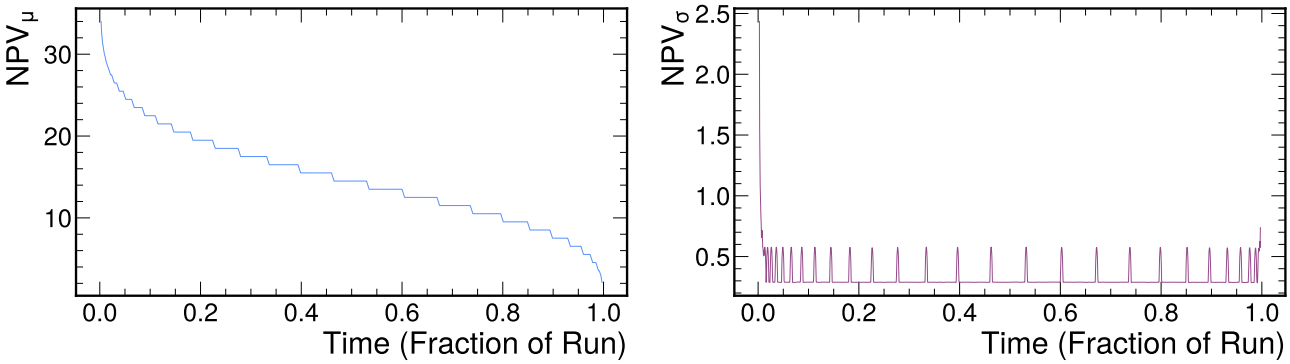


Figure 20. N_{pv} (Number of Primary Vertices) distribution over time (MC).

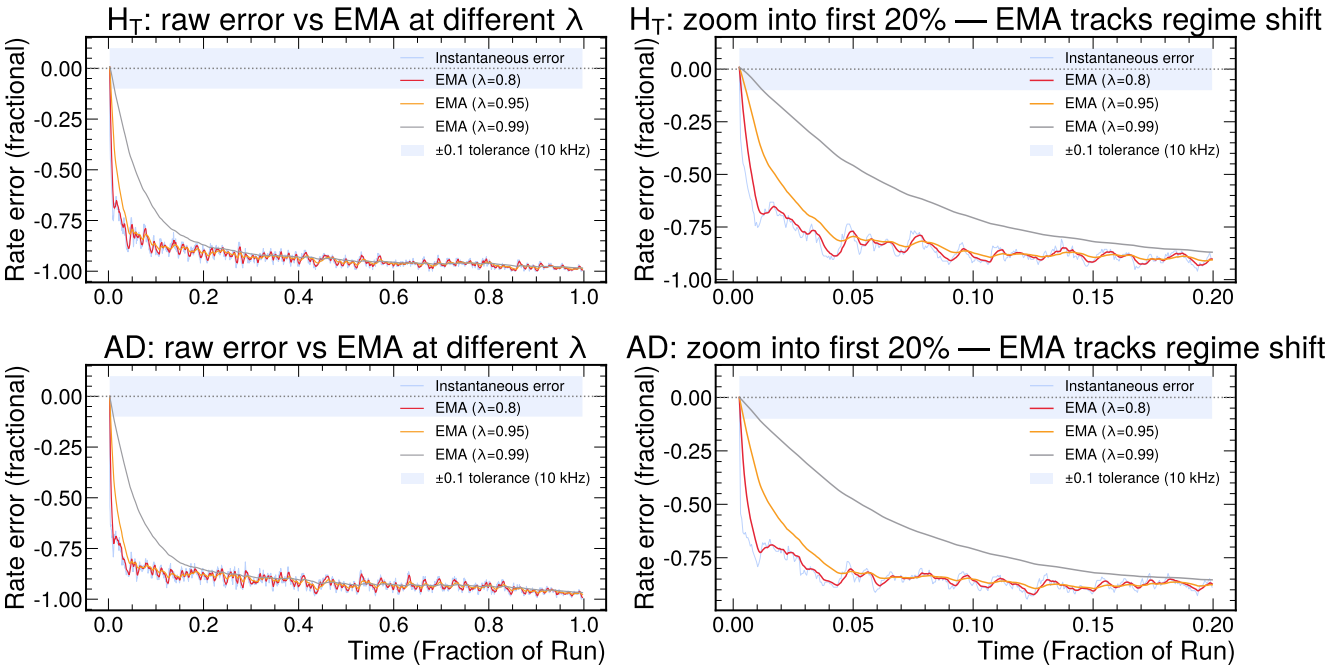


Figure 21. Exponential Moving Average (EMA) of the fractional rate error at different smoothing factors λ, computed over 50k-event sliding windows with a fixed threshold calibrated at the start of the run (MC). The light blue band indicates the ± 10 kHz tolerance around the 100 kHz target rate.

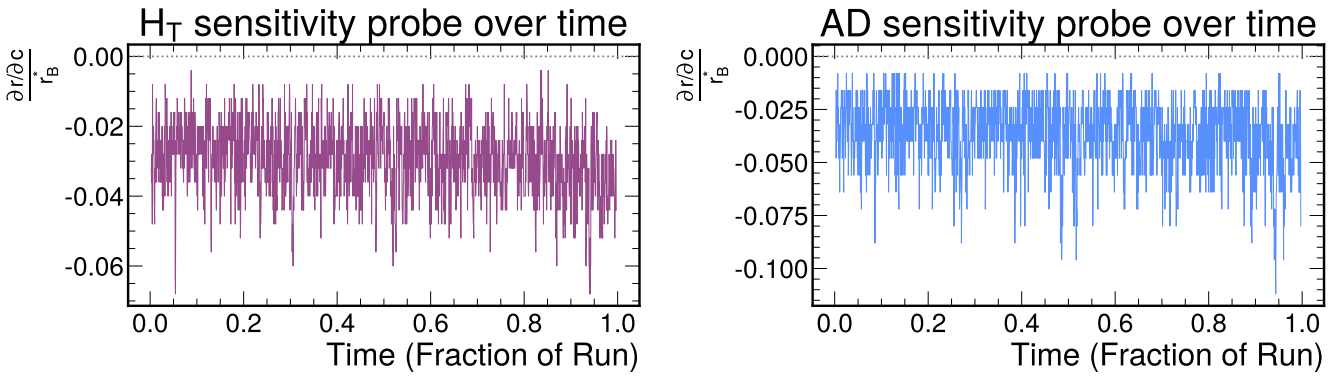


Figure 22. Sensitivity probe over time (MC).

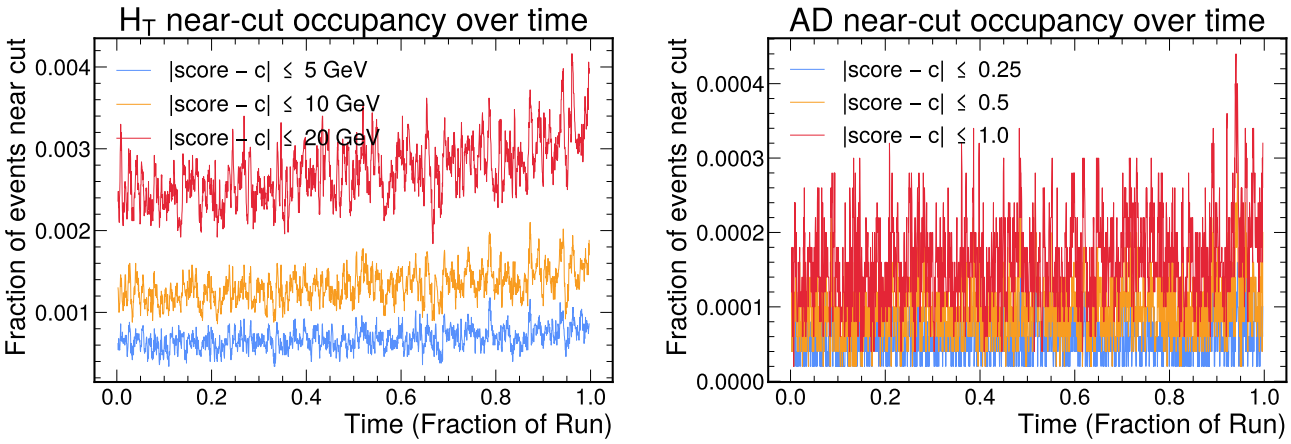
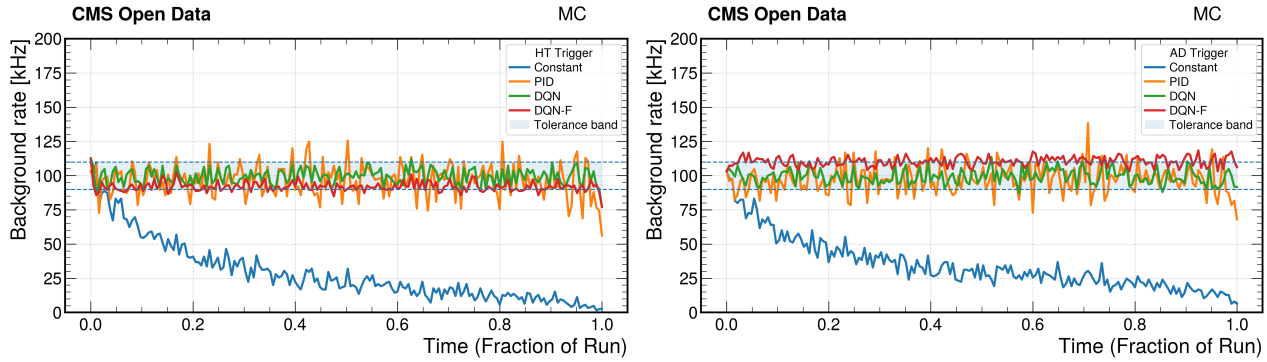


Figure 23. Near-threshold occupancy over time (MC).



(a) H_T trigger *(b)* AD trigger

Figure 24. Background trigger rates under DQN for (a) H_{T} and (b) AD trigger. DQN stays in-band more frequently, reduces overshoot or undershoot and adapts faster in response to pileup changes.

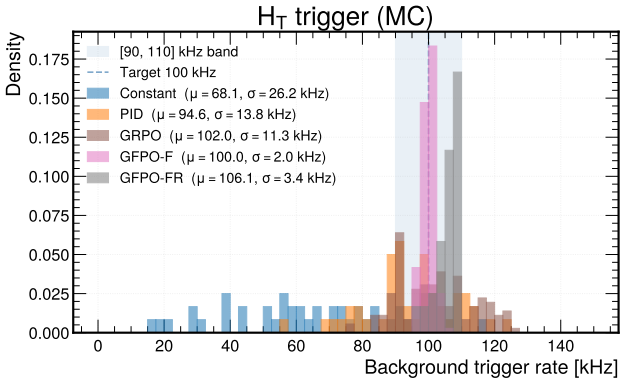
R_k and cost c_k in the simulator (Eqs. 18–19), and executes

$$k^\star = \begin{cases} \arg\max_{k\,:\,c_k \le 10^{-9}} R_k & \text{if some } c_k \le 10^{-9}, \\ \arg\min_k c_k & \text{otherwise,} \end{cases} \tag{22}$$

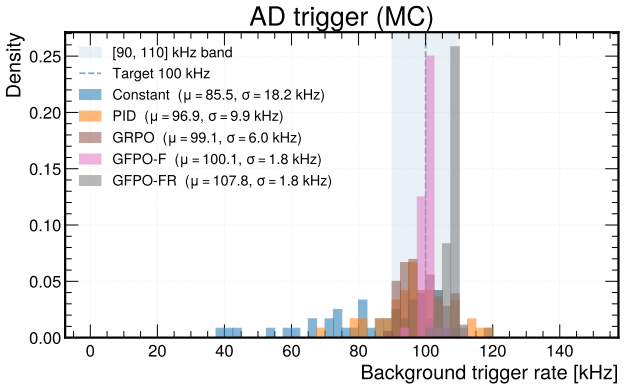
followed by a safety-shield override $\delta = \pm\Delta_{\max}$ whenever $|\bar{r}_{\mathrm{bg}} - r_B^\star| > \tau$. We partition micro-steps by rate state.

The executed action is independent of π_θ. *Out of band* ($|\bar{r}_{\mathrm{bg}} - r_B^\star| > \tau$), the shield overrides the selection with $\delta = \pm\Delta_{\max}$, a function of $(\bar{r}_{\mathrm{bg}}, r_B^\star, \tau, \Delta_{\max})$ alone, independent of the sampled group and of π_θ. *In band*, both selection branches of Eq. 22 rank candidates by their values (R_k, c_k), and R_k and c_k are functions of the candidate cut δ_k and the data through the simulator, not of π_θ. (Because the in-band cost is $c = e^2$ with $e = |\bar{r}_{\mathrm{bg}} - r_B^\star|/\tau$, the tolerance $c_k \le 10^{-9}$ demands $e_k \approx 0$, which discrete cuts essentially never meet, so the otherwise-branch $k^\star = \arg\min_k c_k$ is the operative one.) The policy enters only by determining which δ appear in the sample. The operative δ is absent with probability at most $(4/5)^{16} \approx 0.03$ under a uniform policy, and far less once the trained policy concentrates its mass, so the in-sample optimizer coincides with the full-action-set optimizer. The executed δ therefore depends only on the data and the current cut, and the current cut is itself a function of past executed actions, so the rollout is a deterministic function of the data. We confirm this directly: the per-micro-step executed-δ sequence is bit-identical across the three CMS seeds despite the differing θ.

Test-time training does not help. Under test-time training the recovery step (Achiam et al., 2017, Eq. 14) dominates, because the batch estimate exceeds the budget, $\hat{J}_C > d$ (equivalently $\hat{c} > 0$, Eq. 20), on roughly half of the CMS update chunks (consistent with the ~ 0.54 in-band fraction for AD trigger in Table 2). The recovery branch fires when $C > 0$ and the trust region cannot absorb the violation, $2\delta_{\mathrm{KL}} - c^2/s < 0$, and its direction $\theta' - \theta \propto -H^{-1}b$ moves θ purely along the cost gradient. But neither the recovery step nor an ordinary update alters the (δ_k, R_k, c_k) values that Eq. equation 22 consults, and the selection screens θ as above. The executed trajectory, and every metric derived

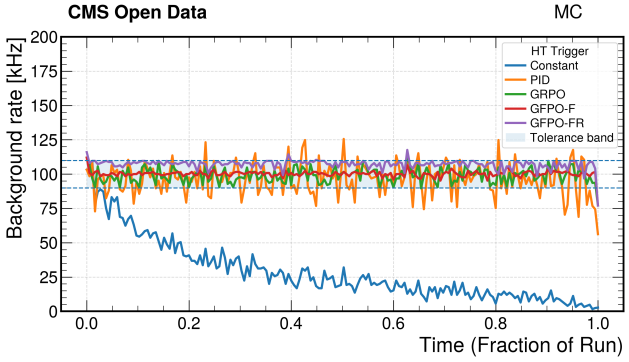


(a) H_T trigger

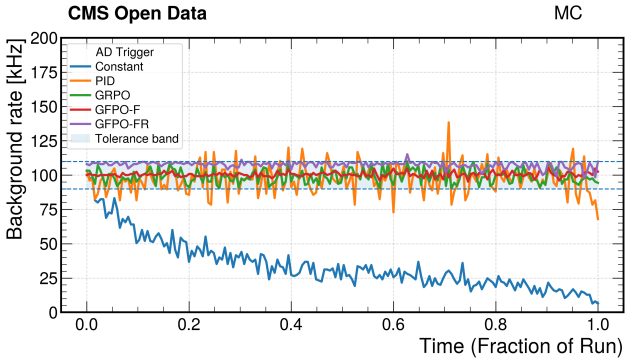


(b) AD trigger

Figure 25. Background rates distribution for (a) H_{T} trigger, (b) AD trigger. Our methods (GFPO-F and GFPO-FR) have the highest in-band rate of all established baselines.



(a) H_T trigger



(b) AD trigger

Figure 26. Background rates under Constant Menu, PID loop, GRPO, GFPO-F and GFPO-FR (MC) for (a) H_{T} trigger, (b) AD trigger. RL-based agents stay within the band for most times compared with non-learning based methods.

from it, is therefore identical to the frozen rollout.

G. Ablation study on noisy anomaly scores

In HEP anomaly detection, a long line of work has explicitly highlighted the sensitivity of reconstruction-based methods to hyperparameters and score design, while more recent studies show that the geometry and topology of the latent space can materially change anomaly-separation performance (Ngairangbam et al., 2025; Gandrakota, 2024; Zipper & collaboration, 2024). Since LHC anomaly triggers operate by thresholding such scores in a real-time setting, understanding how the latent dimension affects score quality is necessary before treating any threshold as reliable.

Latent dimension of the autoencoder training for AD trigger. We ablate the autoencoder bottleneck size to quantify how representation capacity and the resulting variability in reconstruction scores affect downstream threshold control. While the main paper uses $d = 2$, we retrain the autoencoder on MC with larger latents $d \in \{4, 6, 8, 10, 12, 14, 16\}$. To reflect realistic deployment, we train on one MinBias sample and apply the model to a disjoint MinBias dataset, introducing the distribution shift expected between offline training and online operation.

Table 13 shows that enlarging d changes the score distribution, yet the qualitative controller behavior is stable. The PID baseline is consistently less reliable due to *fixed* and *predefined* coefficients, with substantially lower InBand fractions (about 0.30–0.46 with $d \in \{4, 6, 8\}$), suggesting sensitivity to score calibration under latent dimension change. These observations are expected, as the parameter for the AD trigger in (Emami et al., 2026) is calibrated on a single MinBias sample with latent dimension $d = 2$ and lacks *generalizability* to other latent dimensions. In contrast, RL-based controllers (DQN/GRPO/GFPO-F/GFPO-FR) achieve high feasibility across all d (InBand ≈ 0.76–1.00) with low MAE and P95$|e|$, suggesting that closed-loop adaptation is weakly dependent on precise anomaly-score calibration and is *generalizable* across different autoencoder latent dimensions (Gandrakota, 2024).

Among RL variants, the filtered objectives expose the expected feasibility–efficiency trade-off: feasibility-first filtering (GFPO-F) achieves near-perfect background constraint satisfaction, while relaxed filtering (GFPO-FR) shall improve conditional signal efficiency in some settings at the cost of more violations. Overall, these results resonate our main paper's findings with $d = 2$. That is, our RL controllers primarily leverage rate feedback over windows rather than absolute score calibration, and therefore remain robust to moderate changes in autoencoder capacity and score quality, consistent with prior dynamic-thresholding observations (Yang, 2026).

Table 12. **Single-trigger control on CMS Run 283408 (test-time training).** The setup is the same as Table 1 with same CMS dataset, except each RL policy is continually fine-tuned on streaming time chunks rather than frozen at deployment (test-time policy updates). Bold and underline mark the best and second-best per column; standard deviations are shown in parentheses.

Trigger	Method	MAE↓	P95|e| ↓	InBand ↑	$\epsilon^{t\bar{t}}_{\text{overall}}$ ↑	$\epsilon^{h\to 4b}_{\text{overall}}$ ↑	$\epsilon^{t\bar{t}}_{\text{inband}}$ ↑	$\epsilon^{h\to 4b}_{\text{inband}}$ ↑
H_T **trigger**								
H_T	Constant	0.118(0)	0.175(0)	0.041(0)	91.325(0)	22.365(0)	95.842(0)	15.198(0)
H_T	PID (Emami et al., 2026)	**0.003**(0)	0.085(0)	0.432(0)	97.381(0)	33.347(0)	97.50(0)	**35.242**(0)
H_T	DSPOT (Siffer et al., 2017)	0.080(0)	0.157(0)	0.176(0)	93.998(0)	28.200(0)	96.910(0)	33.278(0)
H_T	DQN (Mnih et al., 2015)	0.018(0)	0.037(0.003)	0.824(0.01)	97.510(0.09)	33.212(0.10)	97.420(0.09)	33.40(0.19)
H_T	DQN-F	0.118(0)	0.175(0)	0.04(0)	91.33(0)	22.36(0)	95.84(0)	15.20(0)
H_T	PPO (Schulman et al., 2017)	0.016(0.004)	0.035(0.012)	0.824(0.14)	97.491(0.19)	33.26(0.21)	97.413(0)	32.32(1.01)
H_T	ADT (Yang et al., 2024)	0.018(0.002)	0.042(0.011)	0.811(0.04)	97.460(0.10)	33.26(0.11)	97.334(0.02)	33.250(0.42)
H_T	GRPO (Shao et al., 2024)	0.016(0.002)	0.037(0.008)	0.824(0.05)	97.491(0.09)	33.222(0.10)	97.358(0.14)	32.940(0.83)
H_T	L-GRPO	0.016(0.001)	0.033(0)	0.94(0.01)	97.486(0.04)	33.216(0.08)	97.341(0.04)	32.533(0.15)
H_T	CPO (Achiam et al., 2017)	0.014(0)	0.025(0)	0.905(0)	97.498(0)	33.139(0)	97.392(0)	32.901(0)
H_T	GFPO-F (Our method)	0.007(0)	**0.021(0.001)**	**0.98(0.01)**	97.61(0)	33.32(0)	97.57(0.01)	33.16(0.07)
H_T	GFPO-FR (Our method)	0.015(0)	0.022(0)	0.95(0)	**97.97(0)**	**33.49(0)**	**97.98(0)**	33.59(0)
AD trigger								
AD	Constant	0.142(0)	0.195(0)	0.00(0)	62.548(0)	14.399(0)	NA	NA
AD	PID (Emami et al., 2026)	0.035(0)	0.080(0)	0.405(0)	75.053(0)	39.191(0)	76.210(0)	44.573(0)
AD	DSPOT (Siffer et al., 2017)	0.104(0)	0.180(0)	0.135(0)	67.843(0)	27.225(0)	**80.979(0)**	**56.686(0)**
AD	DQN (Mnih et al., 2015)	0.027(0)	0.060(0)	0.527(0.01)	74.971(0.07)	39.66(0.23)	75.743(0.21)	43.621(0.71)
AD	DQN-F	0.142(0)	0.195(0)	0(0)	62.548(0)	14.399(0)	NA	NA
AD	PPO (Schulman et al., 2017)	0.027(0)	0.061(0)	0.54(0)	75.06(0)	39.70(0)	75.600(0)	43.150(0)
AD	ADT (Yang et al., 2024)	0.027(0.001)	0.063(0.002)	0.52(0.03)	75.011(0.31)	39.510(0.20)	75.614(0.67)	42.930(0.92)
AD	GRPO (Shao et al., 2024)	0.027(0.001)	0.061(0.004)	0.541(0.01)	75.059(0.11)	39.700(0.06)	75.601(0.10)	43.153(0.41)
AD	L-GRPO	0.027(0.001)	0.061(0.002)	0.541(0.01)	75.059(0.06)	39.68(0.04)	75.597(0.09)	43.520(0.46)
AD	CPO (Achiam et al., 2017)	0.027(0.000)	0.061(0)	0.541(0)	75.059(0)	39.703(0)	75.60(0)	43.15(0)
AD	GFPO-F (Our method)	**0.021(0.001)**	**0.053(0.003)**	**0.676(0.02)**	**75.640(0.19)**	**40.370(0.10)**	76.183(0.02)	42.910(0.68)
AD	GFPO-FR (Our method)	0.027(0)	0.060(0)	0.595(0)	75.580(0)	40.240(0)	76.397(0)	44.163(0)

H. Anomaly Benchmarks

H.1. UNSW-NB15

Setup. UNSW-NB15 (Moustafa & Slay, 2015) comprises network traffic from two collection campaigns with documented distributional shift, giving a natural sim-to-real test of the MC→CMS protocol: we train the autoencoder and RL policy on Period 1 and deploy the frozen policy on Period 2. The background class is normal traffic. We designate *Exploits* as the easy signal (anomalous payloads produce reconstruction losses well-separated from normal, analogous to $t\bar{t}$ (CMS Collaboration, 2017)) and *Backdoors* as the hard signal (low-footprint behaviour overlaps substantially with normal scores, analogous to $h \to 4b$ (Aaboud et al., 2018)). The per-window connection rate replaces N_{PV} as the context variable.

Choice of FAR target and tolerance on UNSW-NB15. Recall FAR denotes False Alert Rate, the per-chunk fraction of benign records flagged as anomalies. We set $r^* = 0.5\%$ with tolerance $\tau = 0.05\%$, mirroring the $\pm 10\%$ relative tolerance of the LHC trigger setting ($r_{\text{B}^*} = 0.25\%$ for LHC, $\tau = 0.025\%$). Holding relative stringency fixed lets the in-band quadratic reward and the weight λ_1 transfer across domains without retuning. The absolute level is doubled (0.5% vs. 0.25%) for a quantization reason. UNSW-NB15 streams in chunks of $C = 1000$ records; Period 2 has $\sim 55\%$ attack prevalence (Moustafa & Slay, 2015), leaving $N^- \approx 447$ negatives per chunk and quantizing per-chunk FAR on a grid of step $1/N^- \approx 0.224\%$, where FP is the integer count of benign records flagged in a chunk. The LHC band $[0.225\%, 0.275\%]$ contains no realizable FP count: $\text{FP} = 1$ undershoots ($\text{FAR} = 0.224\%$) and $\text{FP} = 2$ overshoots ($\text{FAR} = 0.447\%$), so "in-band" is unreachable. The chosen band $[0.45\%, 0.55\%]$ contains exactly one realizable count, $\text{FP} = 2$ at $\text{FAR} \approx 0.45\%$ (within chunk-level variation in N^-), making it the tightest tolerance compatible with the integer FP grid. The LHC analogue avoids this floor: its per-chunk negative count is several orders of magnitude larger, so $0.25\% \pm 0.025\%$ sits well above quantization there.

RL setting. The agent observes a length-8 history of $(p_{10}, p_{25}, p_{50}, p_{75}, p_{90}, p_{95}, p_{99}, \text{threshold}, \text{FAR}, \text{TPR}, |\text{FAR} - r^*|, \text{attack prevalence})$ features (12-dim per chunk). Actions are discrete threshold deltas on a 21-bin grid covering ± 0.5 in normalised score space. We reuse the LHC rate-control reward of Section 3.1 (Eq. 3): an in-band term on FAR (target $r^* = 0.5\%$, tolerance $\tau = 0.05\%$), a TPR bonus gated on feasibility, and a movement penalty, with $\lambda_1 = 0.25$, $\alpha = 0$, $\beta = 0.005$.

Training setup for deployment. We train each method for $E = 50$ passes over Period 1 (the 175-chunk UNSW-

NB15 training-set stream, 68% attack prevalence) with Adam (learning rate 3×10^{-4}, mini-batch size 256, 2 inner epochs per update). GRPO and L-GRPO use group size $G = 16$; GFPO-F and GFPO-FR use $G = 64$ with $K = 16$, matching the LHC setups. We then freeze the policy and deploy on Period 2 (the 82-chunk test-set stream, 55.3% attack prevalence); no gradient updates occur during deployment. We report means $\pm$ standard deviations over 5 random seeds per method (seeds 0–4). All UNSW-NB15 experiments run on a single Apple M4 CPU (10 cores); the full 5-seed train+frozen-rollout pipeline completes in approximately 241 seconds (~12 s per method per seed).

Training setup for Test-time training. Here, we report the setting for test-time training results in Figure 29. We additionally evaluate a TTT variant in which each agent continues updating its policy during Period 2 rollout. At every chunk the agent (i) selects a threshold delta greedily, (ii) observes $(\mathrm{FAR}, \mathrm{TPR})$ and computes the in-band reward, and (iii) performs a single gradient step: DQN draws one minibatch from a replay buffer; PPO performs one PPO update; GRPO and L-GRPO take one group update with $G = 16$; GFPO-F and GFPO-FR take one feasibility-filtered group update with $G = 64$, $K = 16$. L-GRPO and GFPO-FR additionally update their dual variable λ once per chunk on the observed FAR. This yields a single in-order pass of 82 updates per agent, versus the $175 \times 50 = 8{,}750$ updates of the Period 1 initialization; all other hyperparameters (reward, target/tolerance, group sizes, learning rate) are unchanged.

Structural ceiling on in-band rate. Of Period 2's 82 chunks, 43 contain only attack traffic and have $\mathrm{FAR} \equiv 0$ regardless of threshold (the denominator $\mathrm{FP} + \mathrm{TN}$ is zero, thus reported as 0); they can never lie within $[r^* - \tau, r^* + \tau] = [0.45\%, 0.55\%]$. The remaining 39 benign-containing chunks (35 all-benign, 4 mixed) set a hard upper bound of $39/82 \approx 47.6\%$ on the achievable in-band rate. Reported in-band fractions should be read against this ceiling: in Figure 4a, GFPO-FR's 6.1% ($5/82$ chunks) corresponds to $\approx 12.8\%$ of the 47.6% structural maximum.

D-SPOT failure. D-SPOT achieves 0% in-band rate despite being designed for streaming anomaly detection (Figure 4a). The cause is class imbalance: with $\approx 55\%$ attack prevalence, the sliding window used to fit the Generalized Pareto distribution is dominated by attack-score values, pushing the estimated tail threshold into attack-score territory. The threshold is then too low to constrain FAR near r^*, and the method flags effectively all traffic. This is a known failure mode of EVT-based detectors on imbalanced streams (Siffer et al., 2017). GFPO sidesteps this failure by closing the loop on FAR rather than estimating the score tail: the policy adjusts the threshold via per-chunk FAR feedback, feasibility filtering retains only training samples already in-band, and all-attack chunks are excluded from rate-control updates. The class imbalance that corrupts DSPOT's GPD fit never enters GFPO's training signal, which is why GFPO-F and GFPO-FR maintain the highest Backdoors TPR (Figure 4a) while satisfying the FAR budget, exactly the regime where DSPOT collapses.

Per-class TPR. Figure 29 reports per-method TPR by attack class (*Exploits*, easy; *Backdoors*, hard) and adaptation regime (frozen Deployment, Test-time training), underlying the in-band/MAE summary in Section 5.2. Constant and PID hit non-trivial Exploits TPR ($\sim$22%) because their fixed thresholds happen to land near the easy-signal mode, but they collapse on Backdoors (6–8%) and stay outside the FAR band entirely. DQN, DSPOT, and ADT (under deployment) reach near-zero TPR on both classes, the over-conservative collapse diagnosed above. GRPO and L-GRPO collapse to near-zero TPR via the zero-feasibility failure mode of Section 4: when no sampled action satisfies the FAR constraint, the group-relative advantage degenerates and the policy converges to a single over-conservative threshold. PPO recovers some TPR but only by abandoning the FAR budget, with mean absolute FAR error two orders of magnitude worse than ours.

GFPO-F and GFPO-FR (*ours*) match the best Exploits TPR ($\approx 22\%$) and dominate on Backdoors ($\approx 12\%$), roughly $2\times$ the next-best learned baseline, while satisfying the FAR budget. Test-time training widens the Backdoors gap without trading off Exploits, indicating that per-chunk FAR feedback provides a usable adaptation signal in exactly the rare-class regime where static thresholds fail.

H.2. NAB

We evaluate on the Numenta Anomaly Benchmark (NAB) (Lavin & Ahmad, 2015), restricted to its two real-world streaming subsets, *realKnownCause* and *realAWSCloudwatch*. Together these comprise 24 univariate time series of server, machine, and cloud-monitoring telemetry with hand-labelled anomaly windows, totalling 135,900 timesteps after we discard series shorter than 200 steps (insufficient for a meaningful train/test split). The goal of this agent is to optimize a pure detection-quality reward $r_t = \mathrm{TPR} - \alpha \cdot \mathrm{FPR} - \beta\,|c_t|/\Delta_{\max}$.

Preprocessing. Each series is processed chronologically. Raw values are converted to anomaly scores via a sliding-window robust z-score, $s_t = |x_t - \tilde{\mu}_{W_t}|/(\widetilde{\mathrm{MAD}}_{W_t} + \epsilon)$, with a causal window of 100 timesteps, then normalized to $[0, 1]$ via empirical CDF rank on the training portion. We split each series 70/30 chronologically into training and test segments, then partition into non-overlapping chunks of 100 timesteps, yielding 954 training and 405 test chunks. MAD denotes the median absolute deviation, a robust scale estimator $\mathrm{MAD}(W) = \mathrm{median}_x|x - \mathrm{median}(W)|$; we substitute it for the standard deviation in the z-score so that anomalies in the recent window do not inflate the normality envelope used to score them.

RL setting. The agent observes a length-8 history of $(p_{25}, p_{50}, p_{75}, p_{90}, p_{95}, p_{99}, \text{threshold}, \text{flagging rate}, \text{TPR}, \text{FPR})$ features. Actions are discrete threshold deltas on a 21-bin grid covering ± 0.3 in score space. Unlike the LHC setting, NAB has no rate constraint: the reward is $\mathrm{TPR} - \alpha\mathrm{FPR} - \beta|\Delta|/\Delta_{\max}$, which is a pure detection quality with a movement penalty.

Training setup We train each method for $E = 50$ passes over the training partition with Adam (learning rate 3×10^{-4}, batch size 64). GRPO and L-GRPO use group size $G = 16$; GFPO-F and GFPO-FR use $G = 64$ with $K = 16$, matching the LHC setup in Figure 3. We report means over 5 random seeds per method. All NAB experiments run on a single Apple M4 CPU (10 cores), with average wall-clock time of approximately 10 minutes per method per series.

Environment. The RL environment we designed uses reward $r_t = \mathrm{TPR} - \alpha \cdot \mathrm{FPR} - \beta\,|\delta_t|/\Delta$, where $\alpha = 0.10$ penalizes false positives, $\beta = 0.005$ discourages unnecessary threshold movement, and $\Delta = 0.3$ is the action range. The action space consists of 21 discrete threshold deltas uniformly spaced in $[-0.3, +0.3]$. The 10-dimensional state encodes score-distribution percentiles $(p_{25}, p_{50}, p_{75}, p_{90}, p_{95}, p_{99})$, the current threshold, the instantaneous flagging rate, and exponentially averaged recent TPR and FPR over the last 8 chunks. All agents are initialized with the threshold at the 97th training-score percentile ($\approx$3% flagging rate).

Precision & Recall Figure 30 reports the per-method precision and TPR underlying the F1 summary in Section 5.2 (Figure 4b). Classical controllers (Constant, PID, DSPOT) reach 11–20% precision but TPR below 10%, defaulting to conservative thresholds that miss most true anomalies. Generic RL methods (PPO, DQN, and the oracle-tuned Constant-opt) push TPR above 20% but drop to 11–14% precision, lowering thresholds to fire on more events at the cost of false positives. GRPO and L-GRPO collapse to the high-precision/near-zero-TPR corner ($\sim$20%/2.4%): under the sparse anomaly reward, the group-relative advantage carries little signal between high-threshold modes and the policy settles on a near-silent one.

GFPO-F and GFPO-FR (*ours*) reach precision $\approx 28\%$ and TPR $\approx 42\%$, surpassing every baseline on both axes and yielding the F1 lead in Figure 4b. Because NAB drops the rate constraint, this gain isolates our sequence-based state and adaptive thresholding from feasibility-filtered policy optimization, confirming that GFPO improves detection quality even when the FAR-handling mechanism is inactive.

Algorithm 2 Sequential state construction and recurrent controller for adaptive threshold at LHC

Input: Background events $\{(x_n, N_{pv,n}\}_{n=1}^{N_t}$; current cut c_t; measured background rates r_t, r_{t-1}; target rate r_B^*; tolerance τ; window length K; calibration constants (m, s) (m=mid, s=span); near-cut widths $\{w_j\}_{j=1}^{J}$ with J=3; last action a_{t-1}, max step $a_{\max}$; tail step δ; integrator $e_t^{(I)}$; sensitivity probe $\widehat{\partial r/\partial c}$.

Output: Sequence observation $S_t \in \mathbb{R}^{K\times F}$ (F=22) and action $\Delta c_t \in \mathcal{A}$.

Procedure BUILDSEQUENTIALSTATE($\{x\}, \{\mathrm{N_{pv}}\}, c_t, r_t, r_{t-1}, r^*, \tau, K$) `// 1) Recent window extraction (most-recent-biased)`

$(x_{1:K}, \mathrm{N_{pv,1:K}}) \leftarrow$ DOWNSAMPLEORPADRECENT($\{x\}, \{\mathrm{N_{pv}}\}, K$)

`// 2) Per-event normalization`

for $k \leftarrow 1$ **to** K **do**

  $x_k^{\mathrm{norm}} \leftarrow (x_k - m)/s$

$\mu_{\mathrm{N_{pv}}} \leftarrow \mathrm{Mean}(\mathrm{N_{pv,1:K}})$; $\sigma_{\mathrm{N_{pv}}} \leftarrow \mathrm{Std}(\mathrm{N}_{PV,1:K})$ **for** $k \leftarrow 1$ **to** K **do**

  $\mathrm{N}_{PV,k}^{\mathrm{norm}} \leftarrow (\mathrm{N}_{pv,k} - \mu_{\mathrm{N_{pv}}})/(\sigma_{\mathrm{N_{PV}}} + \epsilon)$

`// 3) Threshold-relative geometry (per event)`

$c^{\mathrm{norm}} \leftarrow (c_t - m)/s$ **for** $k \leftarrow 1$ **to** K **do**

  $d_k^{\mathrm{norm}} \leftarrow (x_k - c_t)/s$; $p_k \leftarrow 1[x_k \geq c_t]$

`// 4) Chunk-level rate and control features (broadcast)`

$e_t \leftarrow (r_t - r_B^*)/r_B^*$ $e_t^{\mathrm{abs}} \leftarrow |r_t - r_B^*|/r_B^*$ $\Delta r_t \leftarrow (r_t - r_{t-1})/r_B^*$ $b_t \leftarrow 1[|r_t - r_B^*| \leq \tau]$ **if** *last action is available* **then**

  $\tilde{a}_{t-1} \leftarrow a_{t-1}/a_{\max}$

else

  $\tilde{a}_{t-1} \leftarrow 0$

for $k \leftarrow 1$ **to** K **do**

  $\mathrm{tpos}_k \leftarrow (k-1)/(K-1)$

`// 5) Near-cut occupancy channels (per event)`

for $k \leftarrow 1$ **to** K **do**

  for $j \leftarrow 1$ **to** J **do**

    $n_{k,j} \leftarrow 1[|x_k - c_t| \leq w_j]$

`// 6) Tail-shape features (broadcast)`

$p_0 \leftarrow r(c_t)$; $p_1 \leftarrow r(c_t + \delta)$; $p_2 \leftarrow r(c_t + 2\delta)$ $\rho_1 \leftarrow p_1/p_0$; $\rho_2 \leftarrow p_2/p_1$

`// 7) Concatenate into sequence tensor (F=22 features per event)`

for $k \leftarrow 1$ **to** K **do**

  $S_t[k] \leftarrow$ CONCAT$\big(x_k^{\mathrm{norm}}, \mathrm{N}_{PV,k}^{\mathrm{norm}}, p_k, d_k^{\mathrm{norm}}, \mathrm{tpos}_k, n_{k,1}, n_{k,2}, n_{k,3}, e_t, e_t^{\mathrm{abs}}, \Delta r_t, b_t, c^{\mathrm{norm}}, \tilde{a}_{t-1}, \mu_{\mathrm{N_{PV}}}, \sigma_{\mathrm{N_{PV}}}, e_t^{(I)}, \widehat{\partial r/\partial c}, p_1, p_2, \rho_1, \rho_2\big)$

return S_t

Controller (sequence network). $h_t \leftarrow \mathrm{GRU}(S_t)$; `// encode the event sequence`

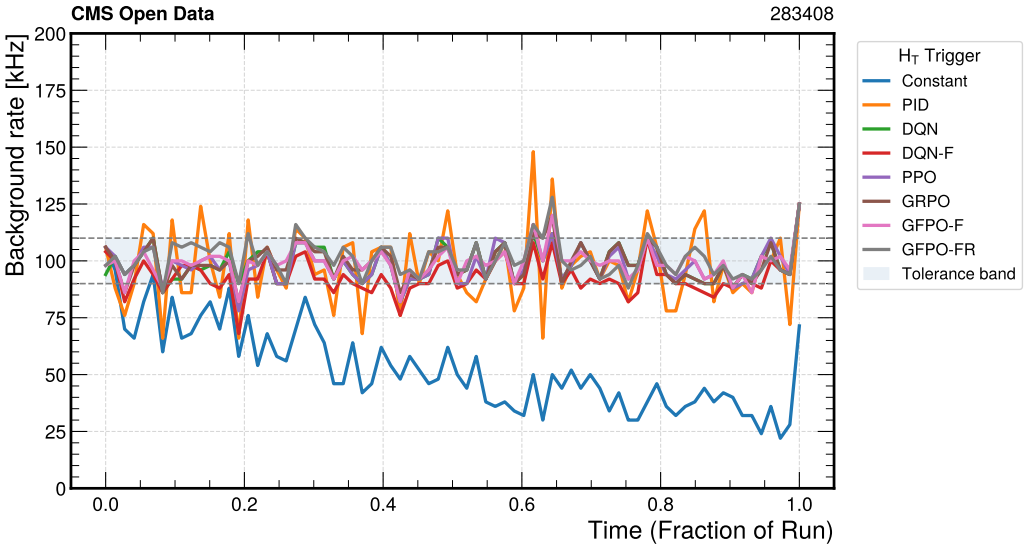
if *policy-based* **then** `// policy-based`

  $\mathrm{logits}_t \leftarrow \mathrm{MLP}(h_t)$ $\Delta c_t \sim \pi(\cdot \mid \mathrm{logits}_t)$ `// sample or argmax`

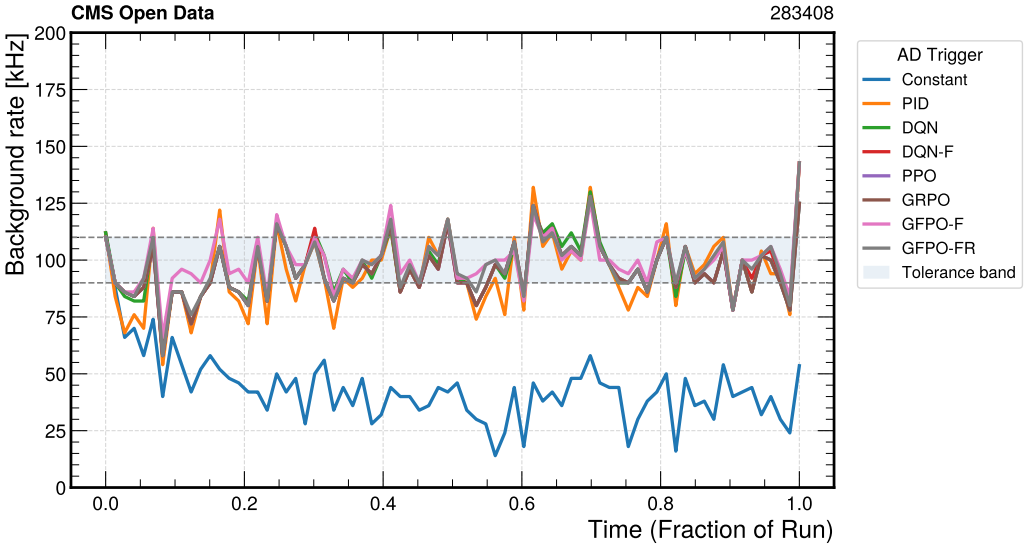
else `// value-based`

  $Q_t \leftarrow \mathrm{MLP}(h_t)$ $\Delta c_t \leftarrow \arg\max_{a\in\mathcal{A}} Q_t[a]$

return Δc_t

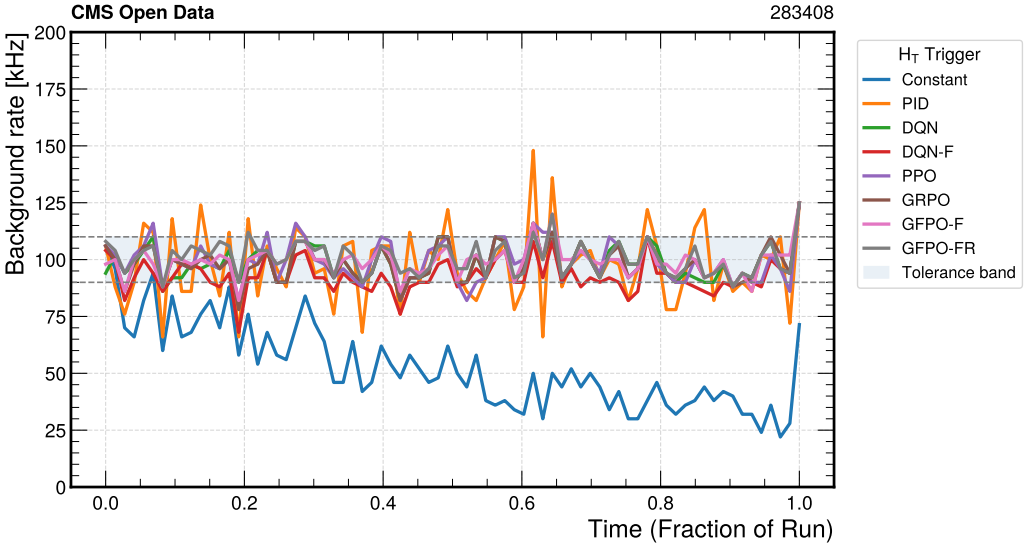


(a) H_T trigger

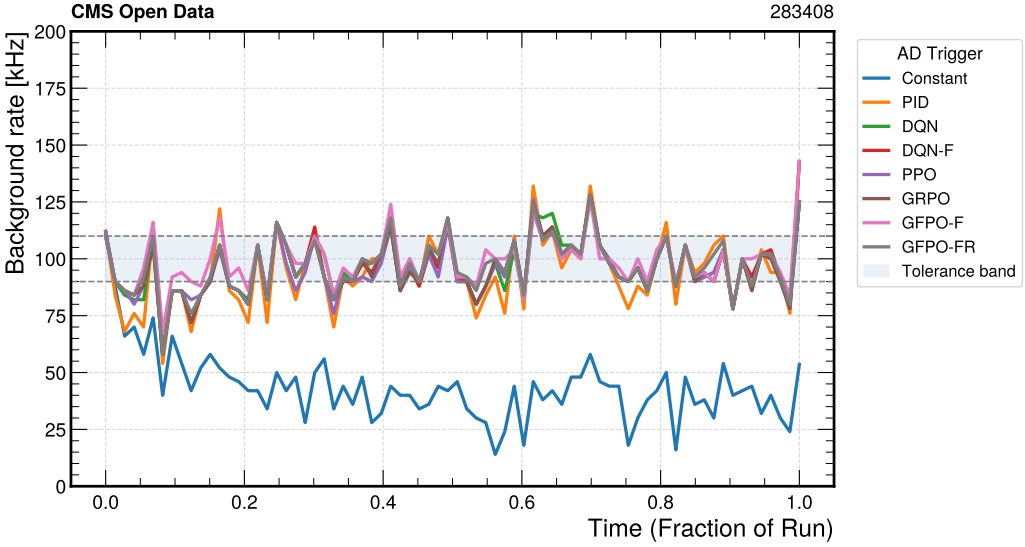


(b) AD trigger

Figure 27. **Background trigger rates under MC→CMS transfer (Table 2).** Per-step rates on CMS Run 283408 with policies trained on MC and frozen at deployment (no online updates), for the (a) H_T trigger and (b) AD trigger.



(a) H_T trigger



(b) AD trigger

Figure 28. **Background trigger rates under MC→CMS transfer with test-time training (Table 12).** Per-step rates on CMS Run 283408 with policies trained on MC and updated online during the run on real collision data (one gradient step per chunk), for the (a) H_T trigger and (b) AD trigger.

Table 13. Single-trigger control (MC) across latent dimensions $d \in \{4, 6, 8, 10, 12, 14, 16\}$ for AD trigger (MC). Rates are in percent units with target $r^* = 0.25\%$ and tolerance $\pm 0.025\%$. Standard deviations are omitted because they are negligible (near zero) across all entries.

Latent dim	Method	MAE↓	P95|e|↓	InBand↑	$\epsilon^{t\bar{t}}_{\text{overall}}$ ↑	$\epsilon^{h\to 4b}_{\text{overall}}$ ↑	$\epsilon^{t\bar{t}}_{\text{inband}}$ ↑	$\epsilon^{h\to 4b}_{\text{inband}}$ ↑
$d = 4$	Constant	0.164	0.216	0.016	88.084	13.415	94.922	19.155
	PID	0.035	0.072	0.301	97.918	25.993	98.086	26.277
	DQN	0.011	0.024	0.984	98.002	26.411	97.999	26.323
	ADT	0.021	0.054	0.554	98.040	26.475	98.010	26.164
	PPO	0.016	0.030	0.860	98.105	27.057	98.085	26.837
	GRPO	0.011	0.022	**1.000**	98.012	26.435	98.039	26.384
	GFPO-F	**0.003**	**0.008**	**1.000**	98.095	26.718	98.111	26.768
	GFPO-FR	0.020	0.033	0.849	**98.260**	**27.486**	**98.343**	**27.721**
$d = 6$	Constant	0.130	0.202	0.032	80.586	14.815	86.915	16.584
	PID	0.024	0.057	0.456	91.858	22.049	91.623	21.602
	DQN	0.015	0.033	0.876	92.202	22.509	92.139	22.266
	ADT	0.019	0.048	0.608	92.195	22.445	92.083	22.051
	PPO	0.015	0.034	0.833	92.209	22.519	92.205	22.361
	GRPO	0.015	**0.032**	0.871	92.231	22.526	92.114	22.194
	GFPO-F	**0.010**	0.033	**0.882**	92.718	23.100	92.735	23.082
	GFPO-FR	0.022	0.044	0.737	**93.237**	**23.709**	**93.407**	**23.992**
$d = 8$	Constant	0.159	0.229	0.032	92.084	15.554	95.235	19.186
	PID	0.032	0.067	0.387	97.861	25.439	97.893	25.050
	DQN	0.018	0.040	0.763	98.097	26.472	98.086	26.210
	ADT	0.018	0.042	0.634	98.101	26.348	98.039	25.846
	PPO	0.021	0.048	0.694	98.061	26.145	97.967	25.571
	GRPO	0.018	**0.038**	0.780	98.108	26.509	98.078	26.191
	GFPO-F	**0.014**	0.039	**0.839**	98.278	27.287	98.221	26.908
	GFPO-FR	0.022	0.044	0.667	**98.380**	**27.825**	**98.307**	**27.329**
$d = 10$	Constant	0.205	0.246	0.016	79.062	9.362	91.594	16.695
	PID	0.022	0.053	0.651	96.369	23.794	96.900	25.151
	DQN	0.015	0.030	0.909	96.542	24.167	96.679	24.478
	ADT	0.016	0.042	0.677	96.566	24.301	96.649	24.453
	PPO	0.020	0.038	0.737	96.398	23.619	96.404	23.574
	GRPO	0.014	0.028	0.903	96.547	24.153	96.701	24.531
	GFPO-F	**0.007**	**0.022**	**0.952**	96.632	24.394	96.705	24.568
	GFPO-FR	0.019	0.030	0.887	**96.854**	**25.036**	**97.031**	**25.534**
$d = 12$	Constant	0.182	0.236	0.016	84.400	12.263	93.426	18.335
	PID	0.016	0.040	0.758	96.848	24.629	96.826	24.311
	DQN	0.013	0.024	0.952	96.931	24.891	96.991	25.048
	ADT	0.013	0.032	0.812	96.906	24.737	96.947	24.909
	PPO	0.020	0.031	0.715	96.757	24.287	96.671	23.829
	GRPO	0.013	0.026	0.946	96.946	24.920	96.952	24.882
	GFPO-F	**0.005**	**0.018**	**0.984**	97.056	25.264	97.069	25.313
	GFPO-FR	0.020	0.030	0.876	**97.299**	**26.055**	**97.321**	**26.157**
$d = 14$	Constant	0.159	0.230	0.043	90.830	15.606	94.393	19.180
	PID	0.014	0.036	0.839	97.310	26.116	97.299	25.943
	DQN	0.012	0.024	0.973	97.357	26.374	97.378	26.361
	ADT	0.014	0.034	0.774	97.365	26.473	97.299	26.237
	PPO	0.012	0.024	0.952	97.404	26.479	97.415	26.464
	GRPO	0.012	0.024	0.973	97.365	26.384	97.350	26.282
	GFPO-F	**0.004**	**0.015**	**0.989**	97.401	26.487	97.407	26.510
	GFPO-FR	0.019	0.026	0.919	**97.596**	**27.174**	**97.646**	**27.373**
$d = 16$	Constant	0.165	0.233	0.032	84.263	14.358	88.231	17.853
	PID	0.012	0.030	0.898	94.297	25.031	94.271	24.916
	DQN	0.012	0.024	0.989	94.331	25.132	94.259	24.946
	ADT	0.013	0.032	0.753	94.391	25.318	94.410	25.273
	PPO	0.012	0.026	0.946	94.420	25.271	94.467	25.271
	GRPO	0.012	0.024	0.978	94.341	25.152	94.289	24.984
	GFPO-F	**0.003**	**0.010**	**1.000**	94.409	25.319	94.409	25.319
	GFPO-FR	0.018	0.024	0.984	**94.697**	**25.875**	**94.724**	**25.928**

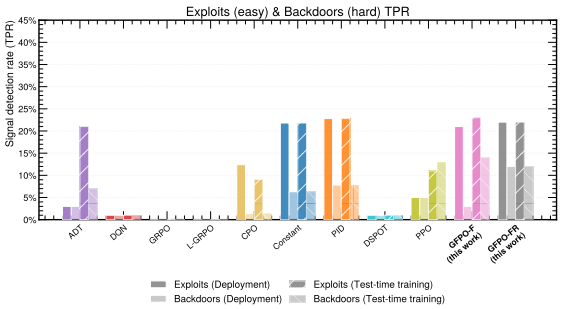


Figure 29. **UNSW-NB15 TPR by attack difficulty and adaptation regime.** For each controller we report TPR on two attack classes: *Exploits* (frequent, high-signal; easy) and *Backdoors* (rare, low-signal; hard). Bars span $\{\text{Deployment}, \text{Test-time training}\} \times \{\text{Exploits}, \text{Backdoors}\}$: solid = frozen policy, hatched = online adaptation; high opacity = Exploits, low opacity = Backdoors. GFPO-F and GFPO-FR (*ours*) maintain the highest Backdoors TPR while satisfying the FAR budget (Fig. 4a); GRPO and L-GRPO collapse to near-zero TPR via the zero-feasibility failure mode. Test-time or online training widens the GFPO advantage on Backdoors without trading off Exploits.

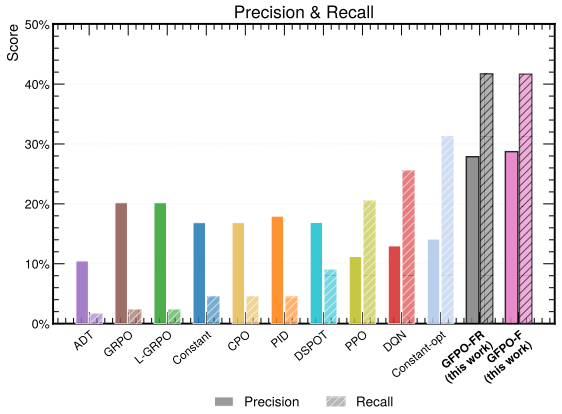


Figure 30. **NAB precision and recall per method (mean over 24 test files).** Solid bars: precision; hatched bars: recall. Methods sorted left-to-right by ascending F1. All controllers operate on the same per-chunk anomaly scores under the same FAR budget. GFPO-F and GFPO-FR (*ours*) dominate the Pareto frontier: precision $\approx 28\%$ and recall $\approx 42\%$. Classical controllers (Constant, PID, DSPOT) reach 11–20% precision but recall below 10%, too conservative to fire on more than a small fraction of true anomalies. PPO, DQN, and the oracle Constant-opt push recall above 20% at a precision cost. GRPO and L-GRPO collapse to the high-precision/near-zero-recall corner ($\sim 20\%$ / 2.4%): under the sparse anomaly reward, the group-relative advantage carries little signal between high-threshold modes, and the policy converges to a near-silent threshold.

Table 14. NAB benchmark results (405 test chunks, 24 real-world series). Constant-opt is an oracle static threshold.

Method	Precision ↑	Recall ↑	F1 ↑
Constant	0.169	0.046	0.058
Constant-opt	0.141	<u>0.312</u>	0.184
PID	0.180	0.045	0.059
D-SPOT	0.169	0.090	0.082
DQN	0.130	0.255	0.144
ADT	0.105	0.017	0.026
GRPO	0.202	0.024	0.037
L-GRPO	0.202	0.024	0.037
CPO	0.169	0.046	0.058
PPO	0.112	0.205	0.119
GFPO-F	**0.287**	**0.417**	**0.330**
GFPO-FR	<u>0.279</u>	**0.417**	<u>0.323</u>