Learning to Trigger: Reinforcement Learning at the 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 W. Miller**[1] **Yuxin Chen**[1*]

University of Chicago[1]
University of Michigan, Ann Arbor[2]
Fermi National Accelerator Laboratory[3]

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 before advantage normalization 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 48% (H_T) and 28% (AD), with a cumulative gain of up to 2% 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 56% (H_T) and 28% (AD) in-tolerance improvement over baselines, with further 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. Code is available at `https://github.com/Zixind/GFPO_LHC`.

1 Introduction

High-throughput discovery science relies on real-time event selection under strict bandwidth and latency constraints [1–4]. 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 [4, 5]. 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 [6] and low-latency menu adaptation [2] 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

*Corresponding to `zixin@uchicago.edu`, `chenyuxin@uchicago.edu`

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

stable background rate with improved signal efficiency, in response to changing physical conditions. For readers outside particle physics, Appendix A summarizes the trigger terminology used throughout (events, pileup, menus, and rate budgets).

Existing work [5] introduced an autonomous trigger-control framework that extends a PID baseline to a cost-function formulation jointly optimizing rate stability, signal efficiency, and computational cost. 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 [7]. Reinforcement learning (RL) [8–10] 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 [11–13] and train an RL policy [8] that observes compact streaming summaries (e.g., recent rates and distributional statistics) and updates thresholds to maximize physics utility while satisfying hard background-rate constraints [1, 14, 15]. The problem is sequential because safety limits cap how far the threshold can move per step, so the thresholds reachable later depend on the action taken now (Appendix F.1.5). 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 begin by studying two representative trigger classes trained on simulated samples with deployment on **real** collision data recorded by the CMS experiment [16]. 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 framework with metrics that quantify both feasibility and performance. Feasibility is measured through the InBand fraction, defined as the proportion of events for which the trigger rate remains within the target bandwidth tolerance, while performance is assessed through signal efficiency. We formulate online trigger threshold optimization as a reinforcement learning problem with streaming observations and demonstrate that even a simple Deep Q-Network (DQN) controller can effectively adapt trigger thresholds in real time, against control-theoretic, constrained RL, and greedy one-step baselines (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 (See psuedocode in Appendix D). Unlike PID control [5], 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 [10]. A recurrent encoder compresses the K-event window into a fixed-size state vector (Appendix F.3).

(3) We adapt Group Filtered Policy Optimization (GFPO) [17], a variant of Group Relative Policy Optimization (GRPO) [18], to the streaming trigger setting and develop two variants, GFPO-F and GFPO-FR, that filter rollouts (sampled candidate actions and their evaluated outcomes) by feasibility (accepted background rate) and signal efficiency. The adaptation is motivated by a failure mode of group-relative updates identified in Section 3.2. Both variants improve stability without degrading efficiency (and vice versa) on Monte Carlo (MC) simulated events and on **real** CMS collision data (Run 283408), transfer from simulation to CMS data with negligible performance loss (Appendix G.3), eliminate the need for gradient-based fine-tuning at deployment, and remain robust to large variations in anomaly-detection score scale (Appendix H).

(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. [5], which minimizes a global cost function, 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.

2 Related Work

Reinforcement learning for scientific decision making. RL has been applied to self-driving labs [19], accelerator tuning [20], and tokamak regulation [21]. 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 [22]. We instead adapt the *operating threshold* of an autoencoder-based score to satisfy rate constraints under drift. The closest formulation is ADT [13], which fixes a pretrained autoencoder and selects binary decision thresholds; we treat trigger thresholds as control variables in a real-time feedback loop. DSPOT [23] 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 F.1.2).

Selective updates. Rejection-style filtering stabilizes RL training, particularly for LLMs [24, 25, 17]. Most directly, GFPO [17] samples larger candidate groups and filters them based on response length and token efficiency to control verbosity. We adapt this framework to streaming trigger control, where the filter must enforce rate feasibility under distribution shift, and develop two variants: **GFPO-F** retains the top-K candidates with the smallest rate deviation; **GFPO-FR** first selects feasible candidates within an expanded tolerance band, then ranks them by signal utility. Appendix B provides an extended discussion.

3 RL formulation in Adaptive Thresholding

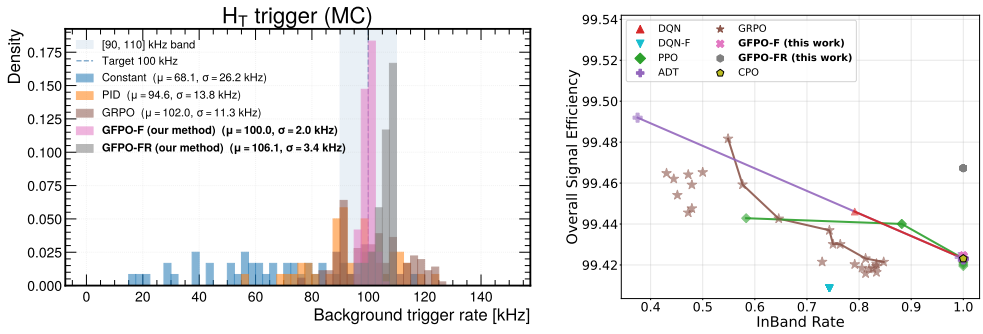
We control a single scalar threshold c_t separately for two trigger paths: an H_T trigger that selects events with high total hadronic activity, $H_T = \sum_{\text{jets}} p_T^{jet}$, and an anomaly-detection (AD) trigger that scores events by autoencoder reconstruction loss [26]. At each step, our agent chooses c_{t+1} to keep the background acceptance rate[2] 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) $(S, \mathcal{A}, T, R, \gamma)$ [27] with exogenous context [28, 29] (Appendix F.1.5), where the transition T is the environment dynamics, to which the agent has no access (the next K events arrive from a stream whose distribution drifts with beam conditions), and $\gamma \in [0, 1)$ is the discount factor. *State.* $s_t = (h_t, c_t)$, where $h_t \in \mathbb{R}^d$ is a fixed-size summary[3] of recent K events (See State representation design in Appendix F.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 $h \to 4b$ [5]; (c) discouraging action update jitter for stability [30, 4] (Eq 2). 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 (Section 4).

3.1 Warmup: DQN for single trigger

Setting Following [5], 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 $r_B^* = 0.25\%$ (equivalent to 100 kHz, since $r_{\text{kHz}} = 400 \cdot r$ at the 40 MHz bunch-crossing rate) 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 $h \to 4b$ [15]. 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 [23], which fits a Generalized Pareto Distribution to the background score tail and sets thresholds at false-positive rate $q = r_B^*$ (Appendix B), (iv) a DQN [31] controller, (v) DQN-F, an ablation in which the DQN is trained on a single calibration chunk and then frozen. We use $\gamma = 0.95$ for our valued-based DQN. Dataset description is in Appendix E.1. All learning-based controllers

[2] We use *background rate*, *rate* and *background acceptance rate* interchangeably throughout this paper.

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



(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 achieve the highest inband rate and signal efficiency among all established baselines (20% held-out MC). (a) GFPO-F sits on target ($\mu = 100.0$ kHz, $\sigma = 2.0$ kHz) and GFPO-FR runs at a slightly higher mean ($\mu = 106.1$ kHz, $\sigma = 3.4$ kHz), while the baselines spread well outside the $[90, 110]$ kHz band. (b) Each symbol represents a different λ in the reward design (Eq 2) with concave hulls connecting the upper envelope per method. Full results on background rates and signal efficiency are in Appendix G.2 and E.3.

receive the identical feature window and encoder described next; Constant, PID, and DSPOT act on the observed background rate only.

RL formulation The state has three parts (i) a length-K window of recent background event scores (H_T in GeV, or the autoencoder reconstruction loss for AD) 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 (Table 17), 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. A single-layer GRU (5,541 parameters) with a linear head encodes the window, once per control step.

We present feature construction for state representations, sequential network architecture comparison (cf. Table 16) and the sequential network learning algorithm (Algorithm 2) in Appendix F.3.

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

$$c_{t+1} = \mathrm{clip}(c_t + \Delta c_t,\, c_{\min}, c_{\max})\,, \qquad r_{t+1} = r_B(c_{t+1}), \quad \epsilon^i_{t+1} = r_i(c_{t+1}) \tag{1}$$

where $r_B(\cdot)$ is the background acceptance and r_i the acceptance of signal $i \in \{t\bar{t}\,, H \to AA \to 4b\}$. We compute one reward per trigger $k \in \{H_T, \mathrm{AD}\}$; the trigger index k (whose threshold the agent controls) is distinct from the signal index i in the efficiency term, and all threshold-dependent quantities (c_t, r_{t+1}, ϵ^i_{t+1}) are evaluated for trigger k.

$$R^k_t \;=\; \lambda_1 \underbrace{\begin{cases} 1 - \left(\dfrac{|r_{t+1} - r^\star_B|}{\tau}\right)^2 & \text{if } |r_{t+1} - r^\star_B| \le \tau \\ -\left(\dfrac{|r_{t+1} - r^\star_B|}{\tau} - 1\right) & \text{if } |r_{t+1} - r^\star_B| > \tau \end{cases}}_{\text{rate tracking}} + (1-\lambda_1) \underbrace{\left(\alpha\, \epsilon^{t\bar{t}}_{t+1} + (1-\alpha)\, \epsilon^{h\to 4b}_{t+1}\right)}_{\text{signal efficiency}} - \lambda_2 \underbrace{\left(\frac{|\Delta c_{t-1}|}{\Delta c_{\max}}\right)}_{\text{move penalty}}. \tag{2}$$

The capital R^k_t is the reward, not the background rate r. Here $\Delta c_{\max}$ is the maximum per-step update (the safety shield), τ is the tolerance width, and $\lambda_{1,2} > 0$ trade off physics utility against smooth, rate-stable control. In real LHC operations, trigger thresholds cannot be adjusted arbitrarily. They must satisfy strict operational constraints arising from detector readout bandwidth [34], computing resources [35], and detector safety limits [36]. 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 [34]. Therefore, we introduce a safety shield Δc_{max} that restricts the controller to threshold configurations that satisfy these operational constraints [37, 38].

In practice, we select them via grid search over $\{0.0, 0.25, 0.5, 0.75, 1.0\}^2$; Figure 1b reports the resulting sensitivity for the H_T trigger, and Appendix E.3 covers both triggers.

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 [5]	0.029(0)	0.070(0)	0.521(0)	99.388(0)	33.289(0)	99.294(0)	31.444(0)
	DSPOT [23]	0.031(0)	0.088(0)	0.604(0)	99.355(0)	32.612(0)	99.300(0)	31.718(0)
	DQN [31]	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 [32]	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 [13]	0.010(0)	0.023(0)	**1.000(0)**	99.423(0)	34.255(0)	99.423(0)	34.255(0)
	GRPO [18]	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 [33]	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 [5]	0.020(0)	0.051(0)	0.729(0)	95.233(0)	27.298(0)	94.990(0)	26.287(0)
	DSPOT [23]	0.022(0)	0.061(0)	0.667(0)	94.992(0)	26.767(0)	94.867(0)	26.050(0)
	DQN [31]	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 [32]	0.011(0)	0.022(0)	**1.000(0)**	95.353(0)	27.726(0)	95.353(0)	27.726(0)
	ADT [13]	0.011(0)	0.022(0)	**1.000(0)**	95.353(0)	27.726(0)	95.353(0)	27.726(0)
	GRPO [18]	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 [33]	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)**

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 $|e_t| \leq \text{P95}|e|$, capturing rare large deviations. We report a tolerance band round the target $r^- \leq \hat{r}_t \leq r^+$, where typically $r^- = r_B^* - \tau$ and $r^+ = r_B^* + \tau$ for tolerance τ. Define the in-band fraction $\text{InBand} = T^{-1} \sum_t \mathbf{1}[r^- \leq \hat{r}_t \leq r^+]$ 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 total number of signal events available at step t. We report two signal efficiencies, restricting to in-band steps or to all (i.e., overall) steps: $\epsilon_{\text{in}}^i = \frac{\sum_t \mathbf{1}[r^- \leq \hat{r}_t \leq r^+] N_{\text{pass}}^i(t)}{\sum_t \mathbf{1}[r^- \leq \hat{r}_t \leq r^+] N_{\text{tot}}^i(t)}$ 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 data 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 holdout MC samples, DQN improves InBand from 0.52 to 1.00 (H_T) and 0.73 to 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 only reaches 0.74 and AD InBand 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

Table 2: **Single-trigger control on CMS data (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 G.4 for results on online training for CMS data.

Trigger	Method	MAE↓	P95$\|e\|$ ↓	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 [5]	0.033(0)	0.085(0)	0.432(0)	97.381(0)	33.347(0)	97.497(0)	**35.242**(0)
	DSPOT [23]	0.163(0)	0.072(0)	0.432(0)	97.211(0)	33.436(0)	96.910(0)	33.278(0)
	DQN [31]	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 [32]	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 [13]	0.019(0)	0.045(0)	0.743(0)	97.412(0)	33.286(0)	97.337(0)	32.954(0)
	GRPO [18]	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 [33]	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 [5]	0.035(0)	0.080(0)	0.405(0)	75.053(0)	39.191(0)	76.210(0)	**44.573**(0)
	DSPOT [23]	0.106(0)	0.087(0)	0.419(0)	74.622(0)	38.297(0)	75.387(0)	41.843(0)
	DQN [31]	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 [32]	0.027(0)	0.061(0)	0.541(0)	75.059(0)	39.700(0)	75.601(0)	43.153(0)
	ADT [13]	0.028(0)	0.067(0)	0.541(0)	75.011(0)	39.510(0)	75.614(0)	42.930(0)
	GRPO [18]	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 [33]	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)

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 G.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 (Figure 5 in Appendix E.1), so replay data quickly becomes off-distribution. Second, as stated above, the local sensitivity $\frac{\partial r}{\partial c}$ is non-stationary and occasionally large, forcing a single Q-function to extrapolate across rapidly changing local dynamics, a known failure mode for TD bootstrapping [39]. PPO [32, 40] avoids replay but inherits the same critic-staleness problem under drift [41–43]. Appendix C 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) [18, 44] to the streaming control setting, a memory-efficient PPO variant that removes the learned critic and instead uses *within-group reward normalization*. 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 $a_t^{(g)} \sim \pi_{\theta_{\text{old}}}(\cdot|s_t)$ (with $c_t^{(g)} = c_t + a_t^{(g)}$), evaluates each on a window W_t of recent events, and forms advantages by within-group normalization:

$$A_t^{(g)} \triangleq \frac{R_t^{(g)} - \mu_t}{\sigma_t + \varepsilon}, \qquad \mu_t \triangleq \frac{1}{G}\sum_g R_t^{(g)},\ \sigma_t^2 \triangleq \frac{1}{G}\sum_g (R_t^{(g)} - \mu_t)^2. \tag{3}$$

The reward $R_t^{(g)}$ reuses Equation 2 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. We run GRPO independently for each trigger k and suppress k for the remainder of

the method. The policy is updated by maximizing

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_t\left[\frac{1}{G}\sum_g A_t^{(g)} \log \pi_\theta(a_t^{(g)} \mid s_t)\right] - \beta\, \mathbb{E}_t[\text{KL}(\pi_\theta \,\|\, \pi_{\theta_{\text{old}}})]\,. \tag{4}$$

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. Eq. 4 is REINFORCE with a sampled-group baseline [45, 46]; the KL term to the previous iterate $\pi_{\theta_{\text{old}}}$ acts as a trust region [47] on drifting data ($\beta = 0.02$; Appendix F.1.8). GRPO and GFPO use $\gamma = 0$, which is exact here because the only action-dependent state is the threshold [48] (Appendix F.1.7).

Appendix F.1.6 further compares against two greedy one-step controllers that see the same feature window as the RL agents.

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. 2 with an adaptive dual variable λ_t following the standard primal-dual optimization [49, 33, 50, 51]:

$$\lambda_{t+1} = \left[\lambda_t + \alpha_{\text{step}}(|r_t - r_B^\star| - \tau)\right]^+ \tag{5}$$

This yields *Lagrangian GRPO* (L-GRPO). On MC, L-GRPO roughly matches GRPO for AD (0.958 vs. 1.000). Under distribution shift to CMS data, 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 18). The failure mode is structural, not a hyperparameter issue (see Appendix F.1.3 for details). In $\sim 30.8\%$ (H_T) and $\sim 32\%$ (AD) of MC steps, the sampled group of G candidates contains *zero* rate-feasible actions (Figure 2 and Appendix F.2). When this happens, every candidate is rate-infeasible (out-of-band), 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.

Nor is it specific to the adaptive penalty: Constrained Policy Optimization (CPO) [33], run with the same group-relative advantages, constrains the *expected* cost of a trust-region step and leaves the feasibility of the proposed candidates unconstrained (Appendix F.1.4). Tuning λ_t rescales the penalty, and CPO reshapes the parameter step, but neither can change the proposal distribution. The agent is being asked to choose between 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) [17], 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, 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.[4]

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

$$\mathcal{F}_t(\tau) \triangleq \{g \in [G] : |\hat{e}_t(c_t^{(g)})| \le \tau\}, \tag{6}$$

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. 2:

$$A_t^{(g)} \triangleq \frac{R_t^{(g)} - \mu_t}{\sigma_t + \varepsilon}, \mu_t \triangleq \frac{1}{|\mathcal{K}_t|}\sum_{g\in\mathcal{K}_t} R_t^{(g)}, \tag{7}$$

$$\sigma_t^2 \triangleq \frac{1}{|\mathcal{K}_t|}\sum_{g\in\mathcal{K}_t}\left(R_t^{(g)} - \mu_t\right)^2. \tag{8}$$

[4] Rollouts are sampled directly from the policy's softmax distribution (temperature T=1.0) throughout the paper.

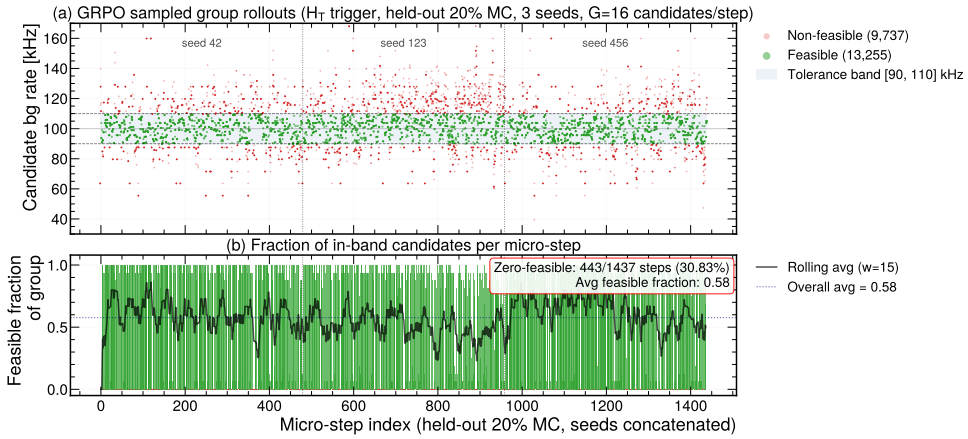


Figure 2: **GRPO's group-feasibility failure on H_T (latter 20% of MC, $G = 16$) averaged across 3 seeds.** (a) Candidate background rates; green/red denote feasible/infeasible w.r.t. the tolerance band. (b) Per-step feasible fraction $f_t = n_{\text{feas}}/G$ (run mean $\langle f_t \rangle = 0.58$). 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.

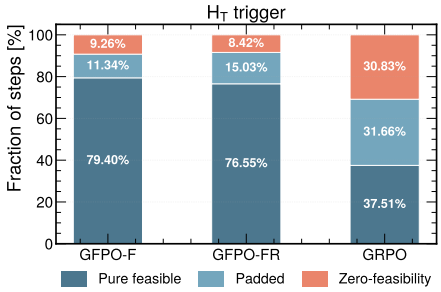


Figure 3: Average step composition for H_T (K=16, G=64 for GFPO, G=K=16 for GRPO). Each step is classified by $|F_t|$ relative to K: Pure feasible ($|F_t| \geq K$, kept set $\mathcal{K}_t \subseteq \mathcal{F}_t$), Padded ($0 < |F_t| < K$, kept set mixes feasible and out-of-band candidates), Zero feasibility ($\mathcal{F}_t = \emptyset$). AD trigger is in Figure 17.

and updates the policy with the GRPO objective restricted to $\mathcal{K}_t$ (Eq. 4 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 (i.e. TopK), regardless of feasibility, biasing learning toward the band boundary and stabilizing training when feasible candidates are scarce:

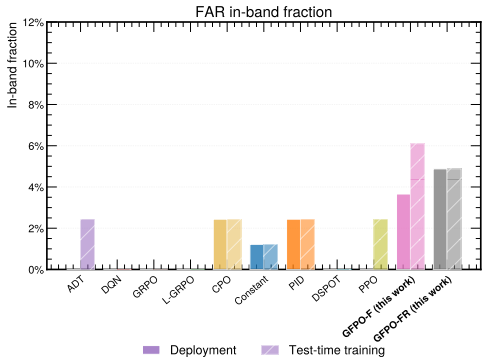
$$\mathcal{K}_t = \text{TopK}\big(\{-|\hat{e}_t(c_t^{(g)})|\}_{g\in[G]}\big). \tag{9}$$

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

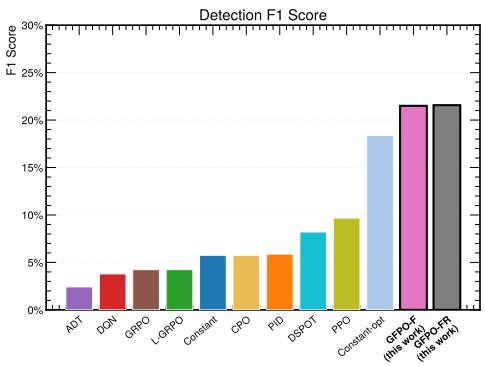
$$\mathcal{K}_t \triangleq \begin{cases} \text{TopK}\Big(\hat{\epsilon}_t^{\text{mix}}(c_t^{(g)})\Big) \subseteq \mathcal{F}_t(\tau), & \text{if } |\mathcal{F}_t(\tau)| \geq K, \\ \mathcal{F}_t(\tau) \cup \text{TopK}\Big(-|\hat{e}_t(c_t^{(g)})|\Big) \subseteq [G] \setminus \mathcal{F}_t(\tau), & \text{if } 0 < |\mathcal{F}_t(\tau)| < K, \\ \text{TopK}\Big(-|\hat{e}_t(c_t^{(g)})|\Big) \subseteq [G], & \text{if } \mathcal{F}_t(\tau) = \emptyset. \end{cases} \tag{10}$$

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. The case $\mathcal{F}_t = \emptyset$ has no analogue in the LLM setting of GFPO (Appendix F.1.9 contrasts it with DAPO-style dynamic sampling [24]).

Diagnostics. Figure 1a shows our methods concentrate background rates *well* inside the $[90, 110]$ kHz band.



(a) **GFPO dominates rate control on UNSW-NB15.** Period 1-trained policies deployed frozen on Period 2 (solid) or with online training (hatched). Fraction of chunks in the *FAR* (False Alert Rate) target band.



(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), driven by precision.

Figure 4: Anomaly detection results on UNSW-NB15 (left) and NAB (right).

Figure 3 compares both filters with vanilla GRPO on H_T (20% MC).[5] The two filters share the same 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 events drift. GFPO-FR instead ranks by signal, biasing the policy toward looser cuts that favor signal efficiency and operate nearer the edge of the band (MAE 0.016 vs. 0.004 on H_T in Table 1). 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 in Table 1).

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 domains, or a more 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 collision rate $40\,\mathrm{MHz}$ to $\sim 2-3\mathrm{kHz}$ for permanent storage, bounded by the write speed and buffer capacity of the tape [52]. Bandwidth is zero-sum, collisions are irreversible, and every false positive permanently displaces a signal event. Thus violations induce detector deadtime. None of (i), (ii), (iii) is physics-specific. The zero-feasibility failure mode arises in any setting satisfying them once the policy drifts off the feasible manifold.

We test this on two benchmarks. **UNSW-NB15** [53] preserves the full rate-constrained protocol and tests whether the failure mode and GFPO's resolution transfer to a non-physics domain. **NAB** [54] 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

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

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 defined in Section 3.1. Physics-specific features (pileup N_{PV}, distance-to-cut in GeV) are replaced by domain-specific 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. Benign network 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. Dataset description, preprocessing, and training setup are in Appendix I.1.

NAB. NAB drops the rate constraint and reports standard detection metrics (precision, recall, F1). The agent optimizes a pure detection-quality reward on 24 real telemetry streams (Appendix I.2). With no rate budget the feasibility set $\mathcal{F}_t$ in Eq. 10 is vacuous, so GFPO-F and GFPO-FR reduce to TopK selection on the detection reward and coincide by construction; NAB thus isolates the contribution of filtering the candidate group from that of feasibility enforcement: 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. Constant-opt is well-defined for NAB but not for UNSW-NB15, where rate regulation and signal TPR create a tension no static threshold shall resolve.

5.2 Experimental Results

UNSW-NB15 results. Both predictions hold (Figure 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 nearly 4% and 4.9% in-band frozen, ranging between 1.5 to $2\times$ the best classical baseline (2.5% PID). Signal TPR on Backdoors, the hard class, is dominated by GFPO-F/FR (Appendix I.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; Figure 4b), approximately 3% gain over the oracle Constant-opt (F1$= 0.184$); 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 (Appendix I.2).

6 Conclusion

We cast online trigger threshold tuning as a streaming RL problem and evaluate it on Monte Carlo simulation and CMS data for H_T and AD triggers. A simple DQN controller improves over constant menus and PID, but trigger non-stationarity (drifting backgrounds, sharp local rate sensitivity) undermines both critic-based DQN and Lagrangian GRPO. Group-Filtered Policy Optimization enforces feasibility *before* advantage normalization and yields consistent gains in rate stability and signal efficiency, with sim-to-real transfer to CMS data. The same failure mode and resolution carry over to UNSW-NB15 and NAB, suggesting feasibility-first filtering is the operative mechanism rather than a physics-specific artifact. Our framework has *limitations* we discuss explicitly here. Our framework regulates a single scalar threshold per trigger, so signal efficiencies are coupled: $t\bar{t}$ exceeds 95% in-band efficiency while $h \rightarrow 4b$ is limited to 29-36% under the same cut (Table 1); the reward mixing parameter α offers only partial control. Our HEP evaluation is restricted to CMS Open Data (Run 283408 for deployment; Run 283876 for autoencoder training); generalization to other LHC experiments such as ATLAS [55] and LHCb [56] is not tested. Further limitations on the state representation, detector conditions, and the choice of training signals are discussed in Appendix J.

A natural extension is multi-trigger menu coordination with the feasibility filter operating over a joint threshold vector [57], where the feasible region only tightens, so the Lagrangian GRPO failure mode likely persists and feasibility-first filtering matters more, rather than less.

Acknowledgements and Disclosure of Funding

We thank the anonymous reviewers and area chair for their constructive feedback. This work is supported by the University of Chicago Joint Task Force Initiative (JTFI) – Partnerships for Emerging Technologies Seed Funding and the University of Chicago AI Initiative. 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

[1] Johannes Albrecht, Leon Bozianu, Lukas Calefice, Sofia Cella, CE Cocha Toapaxi, Caterina Doglioni, VV Gligorov, James Andrew Gooding, Kaare Endrup Iversen, Patin Inkaew, 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.

[2] Chinmaya Mahesh, Kristin Dona, David W Miller, and Yuxin Chen. Towards an interpretable data-driven trigger system for high-throughput physics facilities. *arXiv preprint arXiv:2104.06622*, 2021.

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

[4] Georges Aad, Brad Abbott, Dale Charles Abbott, A Abed Abud, Kira Abeling, Deshan Kavishka Abhayasinghe, Syed Haider Abidi, OS AbouZeid, NL Abraham, Halina Abramowicz, et al. Performance of the atlas level-1 topological trigger in run 2. *The European Physical Journal C*, 82(1):7, 2022.

[5] Shaghayegh Emami, Cecilia Tosciri, Giovanna Salvi, Zixin Ding, Yuxin Chen, Abhijith Gandrakota, Christian Herwig, David W. Miller, Jennifer Ngadiuba, and Nhan Tran. 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`.

[6] Timothy Evans, Conor Fitzpatrick, and Joshua Horswill. An automated bandwidth division for the lhcb upgrade trigger. *Computing and Software for Big Science*, 9(1):7, 2025.

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

[8] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[9] Matthew J Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *AAAI fall symposia*, volume 45, page 141, 2015.

[10] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=S1lOTC4tDS`.

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

[12] Min-hwan Oh and Garud Iyengar. Sequential anomaly detection using inverse reinforcement learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & data mining*, pages 1480–1490, 2019.

[13] Xue Yang, Enda Howley, and Michael Schukat. Adt: Time series anomaly detection for cyber-physical systems via deep reinforcement learning. *Computers & Security*, 141:103825, 2024.

[14] Olmo Cerri, Thong Q Nguyen, Maurizio Pierini, Maria Spiropulu, and Jean-Roch Vlimant. Variational autoencoders for new physics mining at the large hadron collider. *Journal of High Energy Physics*, 2019(5):1–29, 2019.

[15] Sofia Cella. The atlas run-3 trigger menu. Technical report, ATL-COM-DAQ-2024-077, 2024.

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

[17] Vaishnavi Shrivastava, Ahmed Hassan Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. 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`.

[18] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[19] Amanda A Volk, Robert W Epps, Daniel T Yonemoto, Benjamin S Masters, Felix N Castellano, Kristofer G Reyes, and Milad Abolhasani. 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.

[20] Jan Kaiser, Chenran Xu, Annika Eichler, Andrea Santamaria Garcia, Oliver Stein, Erik Bründermann, Willi Kuropka, Hannes Dinter, Frank Mayet, Thomas Vinatier, et al. Reinforcement learning-trained optimisers and bayesian optimisation for online particle accelerator tuning. *Scientific reports*, 14(1):15733, 2024.

[21] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897): 414–419, 2022.

[22] Daochen Zha, Kwei-Herng Lai, Mingyang Wan, and Xia Hu. Meta-aad: Active anomaly detection with deep reinforcement learning. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 771–780. IEEE, 2020.

[23] Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. Anomaly detection in streams with extreme value theory. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1067–1075, 2017.

[24] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[25] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 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`.

[26] Ekaterina Govorkova, Ema Puljak, Thea Aarrestad, Thomas James, Vladimir Loncar, Maurizio Pierini, Adrian Alan Pol, Nicolo Ghielmetti, Maksymilian Graczyk, Sioni Summers, 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.

[27] Pushkala Jayaraman, Jacob Desman, Moein Sabounchi, Girish N Nadkarni, and Ankit Sakhuja. A primer on reinforcement learning in medicine for clinicians. *NPJ digital medicine*, 7(1):337, 2024.

[28] Thomas Dietterich, George Trimponias, and Zhitang Chen. Discovering and removing exogenous state variables and rewards for reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1262–1270. PMLR, 2018. URL `https://proceedings.mlr.press/v80/dietterich18a.html`.

[29] Yonathan Efroni, Dylan J. Foster, Dipendra Misra, Akshay Krishnamurthy, and John Langford. Sample-efficient reinforcement learning in the presence of exogenous information. In *Proceedings of the 35th Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 5062–5127. PMLR, 2022. URL `https://proceedings.mlr.press/v178/efroni22a.html`.

[30] Hoang Le, Andrew Kang, Yisong Yue, and Peter Carr. Smooth imitation learning for online sequence prediction. In *International Conference on Machine Learning*, pages 680–688. PMLR, 2016.

[31] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[32] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[33] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. Pmlr, 2017.

[34] Morad Aaboud, Georges Aad, Brad Abbott, Jalal Abdallah, Baptiste Abeloos, Rosemarie Aben, OS AbouZeid, NL Abraham, Halina Abramowicz, Henso Abreu, et al. Performance of the atlas trigger system in 2015. *The European Physical Journal C*, 77(5):1–53, 2017.

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

[36] Technical Design Report for the Phase-II Upgrade of the ATLAS Tile Calorimeter.

[37] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.

[38] Murad Dawood, Ahmed Shokry, and Maren Bennewitz. A dynamic safety shield for safe and efficient reinforcement learning of navigation tasks. In *7th Annual Learning for Dynamics & Control Conference*, pages 686–697. PMLR, 2025.

[39] Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphael Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, et al. What matters in on-policy reinforcement learning? a large-scale empirical study. *arXiv preprint arXiv:2006.05990*, 2020.

[40] Junbo Li, Peng Zhou, Rui Meng, Meet P Vadera, Lihong Li, and Yang Li. 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*, pages 6227–6243, 2026.

[41] Fabio Pardo, Arash Tavakoli, Vitaly Levdik, and Petar Kormushev. Time limits in reinforcement learning. In *International Conference on Machine Learning*, pages 4045–4054. PMLR, 2018.

[42] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

[43] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[44] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[45] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696.

[46] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.662.

[47] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[48] Nan Jiang, Alex Kulesza, Satinder Singh, and Richard Lewis. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 international conference on autonomous agents and multiagent systems*, pages 1181–1189, 2015.

[49] Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

[50] Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. In *International conference on machine learning*, pages 9133–9143. PMLR, 2020.

[51] Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SkfrvsA9FX`.

[52] Maria Arsuaga-Rios, Vladimír Bahyl, Manuel Batalha, Cédric Caffy, Eric Cano, Niccolo Capitoni, Cristian Contescu, Michael Davis, David Fernandez Alvarez, Jaroslav Guenther, et al. Lhc data storage: Preparing for the challenges of run-3. In *EPJ Web of Conferences*, volume 251, page 02023. EDP Sciences, 2021.

[53] Nour Moustafa and Jill Slay. 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)*, pages 1–6. IEEE, 2015.

[54] Alexander Lavin and Subutai Ahmad. Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, pages 38–44. IEEE, 2015.

[55] Georges Aad, Erlend Aakvaag, B Abbott, Kira Abeling, Nils Julius Abicht, SH Abidi, Asmaa Aboulhorma, Halina Abramowicz, Henso Abreu, Yiming Abulaiti, et al. The atlas trigger system for lhc run 3 and trigger performance in 2022. *Journal of Instrumentation*, 19(06): P06029, 2024.

[56] Roel Aaij, Marta Adinolfi, Salvatore Aiola, S Akar, Johannes Albrecht, M Alexander, S Amato, Yasmine Amhis, F Archilli, M Bala, et al. A comparison of cpu and gpu implementations for the lhcb experiment run 3 trigger. *Computing and Software for Big Science*, 6(1):1, 2022.

[57] Laura Boggia, Carlos Cocha, Fotis Giasemis, Joachim Hansen, Patin Inkaew, Kaare Endrup Iversen, Pratik Jawahar, Henrique Pineiro Monteagudo, Micol Olocco, Sten Astrand, 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.

[58] Isaac D Lutz, Shunzhi Wang, Christoffer Norn, Alexis Courbet, Andrew J Borst, Yan Ting Zhao, Annie Dosey, Longxing Cao, Jinwei Xu, Elizabeth M Leaf, et al. Top-down design of protein architectures with reinforcement learning. *Science*, 380(6642):266–273, 2023.

[59] Rajat Khanda, Mohammad Baqar, Sambuddha Chakrabarti, and Satyasaran Changdar. Extending group relative policy optimization to continuous control: A theoretical framework for robotic reinforcement learning. *arXiv preprint arXiv:2507.19555*, 2025.

[60] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=LzQQ89U1qm_`.

[61] Lyndon Evans and Philip Bryant. Lhc machine. *Journal of instrumentation*, 3(08):S08001–S08001, 2008.

[62] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan Read, Jacob Zwart, Michael Steinbach, and Vipin Kumar. 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*, pages 558–566. SIAM, 2019.

[63] Arka Daw, Anuj Karpatne, William D Watkins, Jordan S Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. In *Knowledge guided machine learning*, pages 353–372. Chapman and Hall/CRC, 2022.

[64] Yingheng Tang, Jichao Fan, Xinwei Li, Jianzhu Ma, Minghao Qi, Cunxi Yu, and Weilu Gao. Physics-informed recurrent neural network for time dynamics in optical resonances. *Nature computational science*, 2(3):169–178, 2022.

[65] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2):209–232, 2002.

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

[67] Jincheng Mei, Wesley Chung, Valentin Thomas, Bo Dai, Csaba Szepesvari, and Dale Schuurmans. The role of baselines in policy gradient optimization. *Advances in Neural Information Processing Systems*, 35:17818–17830, 2022.

[68] Hongyi Zhou, Kai Ye, Erhan Xu, Jin Zhu, Ying Yang, Shijin Gong, and Chengchun Shi. Demystifying group relative policy optimization: Its policy gradient is a u-statistic. *arXiv preprint arXiv:2603.01162*, 2026.

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

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

[71] Daniel de Florian, Christophe Grojean, Fabio Maltoni, C Mariotti, A Nikitenko, M Pieri, P Savard, M Schumacher, R Aggleton, M Ahmad, 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.

[72] Morad Aaboud, G Aad, B Abbott, B Abeloos, DK Abhayasinghe, SH Abidi, OS AbouZeid, NL Abraham, H Abramowicz, H Abreu, 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.

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

[74] Henrik Bertelsen, G Carrillo Montoya, P-O Deviveiros, T Eifert, G Galster, J Glatzer, S Haas, A Marzin, MV Silva Oliveira, T Pauly, et al. Operation of the upgraded atlas central trigger processor during the lhc run 2. *Journal of Instrumentation*, 11(02):C02020–C02020, 2016.

[75] Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

[76] Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pages 1597–1600. IEEE, 2017.

[77] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[78] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*, 2019.

[79] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International journal of forecasting*, 37(4): 1748–1764, 2021.

[80] Arsenii Gavrikov, Julián García Pardiñas, and Alberto Garfagnini. Dinamo: Dynamic and interpretable anomaly monitoring for large-scale particle physics experiments. *Machine Learning: Science and Technology*, 6(3):035050, 2025.

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

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

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

[84] Vishal S Ngairangbam, Błażej Rozwoda, Kazuki Sakurai, and Michael Spannowsky. Enhancing anomaly detection with topology-aware autoencoders. *Machine Learning: Science and Technology*, 6(4):045051, 2025.

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

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

Appendices

A Background: The LHC Trigger Problem **18**

B Extended Related Work **18**

C Theoretical Justification **21**

D GFPO for Streaming Trigger Control: Pseudocode **26**

E Experimental Setup for LHC **27**

E.1 Dataset Summary . . . 27

E.2 Training setup . . . 29

E.3 Models and Hyperparameters . . . 29

E.4 Standard anomaly detection metrics for LHC . . . 31

F Additional Implementation Details **37**

F.1 Baseline Adaptation . . . 37

F.1.1 ADT . . . 37

F.1.2 DSPOT . . . 37

F.1.3 L-GRPO . . . 38

F.1.4 CPO . . . 40

F.1.5 MDP Structure: Factored Kernel and Markov Sufficiency . . . 43

F.1.6 Greedy One-Step Baselines with the Same Inputs: GRW and RWQ . . . 43

F.1.7 Choice of Discount Factor . . . 44

F.1.8 KL-Term Ablation . . . 44

F.1.9 DAPO-Style Dynamic Sampling . . . 45

F.1.10 Signal-Free Reward and Cross-Signal Transfer . . . 45

F.2 GRPO Failure Mode . . . 46

F.3 Sequential Network Architectures . . . 46

G Additional Experimental Results **49**

G.1 Additional DQN results . . . 49

G.2 GRPO MC training . . . 53

G.3 Sim-to-Real Transfer and Test Time Adaptation . . . 54

G.4 Testing Time Training on Real CMS Data . . . 54

G.5 CPO degeneracy on the CMS dataset . . . 56

H Ablation study on noisy anomaly scores **58**

I Anomaly Benchmarks **59**

I.1 UNSW-NB15 . . . 59

I.2 NAB . . . 63

J Extended Limitations **64**

A Background: The LHC Trigger Problem

This section collects the particle-physics terminology used in the main text.

Events and pileup. At the LHC, bunches of protons cross inside each detector at $40\,\mathrm{MHz}$. Each bunch crossing yields an *event*: a snapshot of the detector response to the outgoing particles. The overwhelming majority of events are produced by well-understood processes of no direct interest, and every crossing superimposes tens of simultaneous proton–proton interactions, called *pileup*, whose multiplicity is proxied by the number of reconstructed primary vertices N_{pv}. Pileup, the instantaneous luminosity, and detector conditions all vary during a data-taking run, which is the source of the drift in the background-score distributions shown in Appendix E.1.

Triggers and menus. Readout and storage bandwidth hard-cap the rate of events that can be kept, from $40\,\mathrm{MHz}$ of crossings to roughly $100\,\mathrm{kHz}$ after the hardware Level-1 (L1) stage and a few kHz for permanent storage [52]. A *trigger* is the real-time system that decides which events to accept. A trigger *menu* is a list of such selections, each a threshold on a scalar per-event score, for example the total hadronic activity H_T (the scalar sum of jet transverse momenta, where a jet is a collimated spray of particles) or an anomaly score from an autoencoder trained on background events. In this paper a single menu item is controlled at a time; the multi-item case is discussed as future work in Section 6.

Rates, budgets, and deadtime. The *background rate* r of a trigger is the fraction of (overwhelmingly background) events it accepts. A fractional rate r in percent corresponds to $r_{\mathrm{kHz}} = 400 \cdot r$ at the $40\,\mathrm{MHz}$ crossing rate, so the target $r_B^* = 0.25\%$ used throughout corresponds to $100\,\mathrm{kHz}$ and the tolerance $\tau = 0.025\%$ to a band of $[90, 110]\,\mathrm{kHz}$. Exceeding the readout budget induces *deadtime*, intervals during which the detector cannot record any event and all physics channels lose data simultaneously, whereas under-using the budget permanently discards events that would otherwise have been kept. Trigger control is therefore a rate-constrained selection problem under drift, which is the formulation adopted in Section 3.

Signal efficiency. The *signal efficiency* of a trigger on a physics process is the fraction of that process's events it accepts. We use two representative signals: top-quark pair production ($t\bar{t}$), whose large H_T places it well above the background bulk, and a light-Higgs decay to four b-quarks ($h \to 4b$), whose spectrum overlaps the background and is therefore sensitive to where the threshold sits within the rate band. Signal efficiencies are available only in simulation; in real operation only the background rate is measured online, which is why the deployed constraint in this paper is the rate band and signal efficiency enters only through the training reward.

B Extended Related Work

RL for scientific decision making. RL is increasingly used across scientific domains, including experiment planning in self-driving laboratories [19], online control of particle accelerators [20], and tokamak plasma regulation [21]. 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 [58] 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. [59], 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 [50, 51] replaces fixed penalty weights with adaptive dual variables. We include Lagrangian GRPO (L-GRPO) as a direct baseline (Section 3.2, Appendix F.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 [22] 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 [13], 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 F.1.

We deliberately exclude the Anomaly Transformer [60]. 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 [61], 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 [23] 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 F.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 [17] oversamples candidates per prompt and updates only on a filtered subset; DAPO [24] discards prompts whose rollouts carry trivial learning signal; analyses by Yue et al. [25] 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. We claim no novelty for the filtering principle itself; Section 4 and Appendix F.1.9 delineate how the hard-constraint setting differs from DAPO and the original GFPO, and the base optimizer is REINFORCE [45] with a sampled-group baseline [46].

Model-based and model-predictive control. A natural alternative to a learned policy is to identify the threshold→rate map and optimize over it, as in model-predictive control (MPC). MPC needs a model of the rate response to the threshold, which is non-stationary in this problem (Figures 22 and 23); under drift a fitted model is stale, the same failure that degrades the DQN/PPO critics (Appendix C). Adaptive MPC refits the model online, which requires either a parametric form for the score distribution's tail (what DSPOT assumes via extreme-value theory, and which fails under imbalance and drift, Appendix F.1.2) or an empirical quantile over the window, which is the RWQ controller of Appendix F.1.6. Our state representation already contains a measured local rate map (the occupancy features and the central-difference probe $\partial r/\partial c$), so the policy acts on the same local model a system-identification approach would fit, without requiring that model to remain valid away from the current operating point. The GRW and RWQ baselines show how far a measured

local map alone goes: RWQ matches GFPO-F on feasibility but is not deployable at L1 latency and has no mechanism for signal efficiency, and GRW performs at the level of GRPO but below the filtered variants. For these reasons we close the control loop on measured rates and do not learn a dynamics model. PID is included because it is deployed practice for rate regulation [5]; the stronger non-learning adaptive controllers in our comparison are DSPOT and RWQ.

Physics-informed sequential state representations. A related line of work improves robustness in non-stationary physical systems by embedding domain structure into sequence models [9, 62–64]. For example, Jia et al. [62] 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. [64] 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 [63]. 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 [32, 43] 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 [18] 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 C 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.

C 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 1 bounds the PPO advantage error in terms of critic mismatch; Lemma 2 establishes within-chunk unbiasedness of the GRPO gradient at rate $\mathcal{O}(G^{-1})$; Lemma 3 shows that the GRPO learning signal is invariant under per-chunk affine rescaling of returns; Corollary 1 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

$$\begin{aligned}\delta_k \;\triangleq\; & \underbrace{\sup_{s,a} \left| r_k(s,a) - r_{k-1}(s,a) \right|}_{\text{reward drift}} \\ & + \underbrace{\tfrac{\gamma R_{\max}}{1-\gamma} \sup_{s,a} \left\| P_k(\cdot|s,a) - P_{k-1}(\cdot|s,a) \right\|_{\mathrm{TV}}}_{\text{transition drift}},\end{aligned} \tag{11}$$

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 11 then sum cleanly into the simulation-lemma bound (Kearns and Singh 65, Lemma 4; Strehl and Littman 66, Lemma 1): for any policy π,

$$\left\| V_{k-1}^\pi - V_k^\pi \right\|_\infty \;\leq\; \frac{\delta_k}{1-\gamma}. \tag{12}$$

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,

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

The bound on the second term uses $\sup_{s,a} |r_{k-1} - r_k|$ on the reward channel and the standard inequality $|\mathbb{E}_P V_k^\pi - \mathbb{E}_Q V_k^\pi| \leq \mathrm{span}(V_k^\pi)\, \|P - Q\|_{\mathrm{TV}}$ together with $\mathrm{span}(V_k^\pi) \leq R_{\max}/(1-\gamma)$ on the transition channel, recovering exactly δ_k. Solving $\|V_{k-1}^\pi - V_k^\pi\|_\infty \leq \gamma \|V_{k-1}^\pi - V_k^\pi\|_\infty + \delta_k$ for the left-hand side yields Equation 12.

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 3.

Critic drift biases the PPO advantage. Let V_ϕ be the critic fit on chunks $\{1, \ldots, k-1\}$, and let $\hat{A}_t^{\mathrm{GAE}} = \sum_{l\geq 0} (\gamma\lambda)^l \delta_{t+l}$, 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. [43].

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

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

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 $l \geq 1$, 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,

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

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

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

□

A simulation-lemma argument [65] bounds $\|V_{k-1}^\pi - V_k^\pi\|_\infty \leq \delta_k/(1-\gamma)$, 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 1). 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 $R_t^{(g)}$, $g = 1, \ldots, G$.

Define the within-chunk preconditioned policy gradient

$$\widetilde{\nabla} J_k(\theta) \triangleq \mathbb{E}_{s \sim \pi_\theta, \mathcal{M}_k} \left[\frac{1}{\sqrt{\Sigma_k(s)} + \varepsilon} \cdot \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} [A_k^\pi(s,a) \nabla_\theta \log \pi_\theta(a \mid s)] \right], \tag{13}$$

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. [67] the within-chunk z-score acts as an adaptive step-size.

Lemma 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 13.

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

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

(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 $|\hat{A}_t^{(g)}| \le R_{\max}/[(1-\gamma)\varepsilon]$, 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} [68]. 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 13, 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. [68]. Every quantity above depends only on $\mathcal{M}_k$, so no information from chunks $\{1, \ldots, k-1\}$ enters the gradient. □

Affine equivariance.

Lemma 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 1 admits no analogous equivariance because V_ϕ is not re-scaled at chunk k. From the viewpoint of Mei et al. [67], the within-chunk z-score acts as an *adaptive step-size*.

Per-chunk bias bounds.

Corollary 1 (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 13. Combining Lemmas 1 and 2, for any fixed group size G,

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

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

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

Proof. The PPO bound follows by combining Lemma 1 with $\|V_\phi - V_k^\pi\|_\infty = \mathcal{O}(\delta_k/(1-\gamma))$ from Equation 12, 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 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 13. Affine invariance is immediate from Lemma 3. $\square$

Remark 1 (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 2 (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 3. The GRPO update absorbs $\mathcal{L}_k$-rescaling (per-chunk luminosity) without re-tuning λ_k; the PPO critic acquires the Lemma 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 1 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.

D GFPO for Streaming Trigger Control: Pseudocode

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

Input: Data Streams $\mathcal{D}_{\text{MC}}, \mathcal{D}_{\text{CMS}}$; 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 $\{c_t\}_{t \in \mathcal{D}_{\text{CMS}}}$.

Procedure GFPO-STEP(s_t, c_t);

Sample actions $\{a_t^{(g)}\}_{g=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid s_t)$; form candidates $c_t^{(g)} \leftarrow c_t + a_t^{(g)}$;

Evaluate background rate errors for $c_t^{(g)}$ and calculate rewards $\{R_t^{(g)}\}$; `// Eq. 2`

$\mathcal{F}_t(\tau) \leftarrow \{g \in [G] : |\hat{e}_t(c_t^{(g)})| \leq \tau\}$;

$\mathcal{K}_t \leftarrow$ kept set per variant v ; `// Eqs. 9/10`

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

  $\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{L}_{\text{GRPO}}(\theta; \mathcal{K}_t)$; `// Eqs. 8, 4`

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);

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

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`

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

E Experimental Setup for LHC

E.1 Dataset Summary

Datasets. We use the datasets of Emami et al. [5]. 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.

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

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. [5], we use latent dimension $d = 2$. For real data, we train on CMS Run 283876 [69] (the second-longest 2016 run [70]) 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 [71, 72], 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: for H_T trigger, every method in Tables 1 and 2 reaches $> 97\%$ on $t\bar{t}$ (except Constant in Table 2) but stays below $\sim 35\%$ on $h \to 4b$ for H_T (and analogously for AD), tracking the oracle bounds in Table 4. Note to keep it consistent with the main paper, we only report results for 20% MC with Table 4 as it involves RL training.

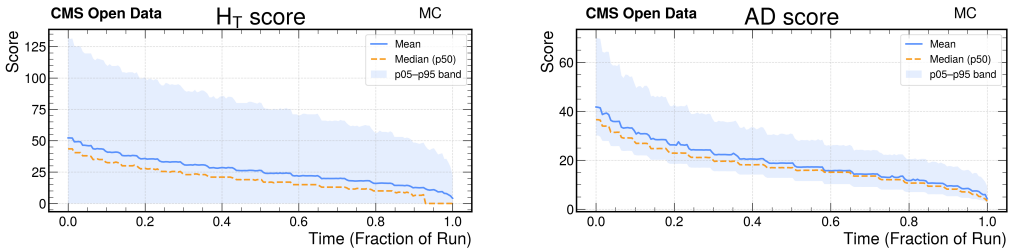
Unlike the RL evaluations, which train on the first 80% of MC chunks and report performance on the held-out final 20% to prevent train-test leakage, Figure 7 uses the full MC sample because essentially *no* learning is performed. That figure serves only as an oracle study of the underlying signal-background separation achievable with each trigger. Same setup applies to Figure 8 for real CMS data. Therefore, the maximum achievable signal efficiency for both Figures 7 and 8 are different from oracle signal efficiencies for Tables 5 and 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. [5], we exclude the first 10 chunks (initial threshold calibration) and report results for only events past the first 500K (MC) and 200K (CMS Run 283408).

Analyzing the signal efficiency gap between oracle and GFPO variants. Tables 5 and 6 report the gap between our methods and the oracle at r^+. For MC, GFPO-FR achieves the smallest gap with the oracle compared to GFPO-F. On CMS Run 283408 for $h \to 4b$ signal, both GFPO variants (42.533 and 44.477) *exceed* the oracle (41.028) in-band efficiency: in-band chunks are not uniformly distributed but concentrate in the high-efficiency tail of the run (Figure 9).

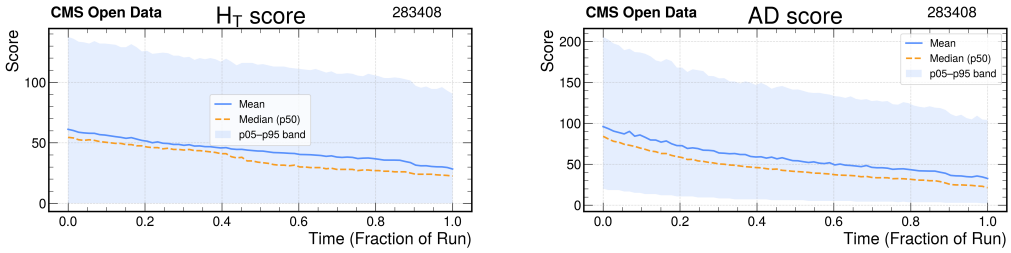
Table 4: Overall oracle signal efficiencies for 20% MC and CMS Run 283408 at $r_B^* = 0.250\%$ and the loosest feasible rate $r^+ = r_B^* + \tau = 0.275\%$. To avoid training and evaluation overlapping, we train on 80% MC and report below results on 20% MC. For oracle performance, the default way to 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 for both overall and inband signal efficiency.

		20 % MC				CMS Run 283408			
		ϵ_{ov}		ϵ_{in}		ϵ_{ov}		ϵ_{in}	
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 ($r^+ = r_B^* + \tau$)	**99.560**	**36.844**	—	—	**98.049**	**33.693**	—	—
	PID	99.388	33.289	99.294	31.444	97.381	33.347	97.497	**35.242**
	GFPO-F (Ours)	99.425	34.273	99.425	34.273	97.587	33.347	97.508	33.262
	GFPO-FR (Ours)	99.468	35.057	99.468	35.057	97.734	33.479	97.821	33.451
AD	Oracle ($r^+ = r_B^* + \tau$)	**96.092**	**29.910**	—	—	**76.664**	**41.028**	—	—
	PID	95.233	27.298	94.990	26.287	75.053	39.191	76.210	**44.573**
	GFPO-F (Ours)	95.442	27.926	95.442	27.926	75.562	40.083	76.209	42.533
	GFPO-FR (Ours)	95.842	28.875	95.842	28.868	75.582	40.309	76.515	44.477



(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).



(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).

Table 5: Oracle signal efficiencies (20 % MC) at $r_B^* = 0.250\%$ and $r^+ = r_B^* + \tau = 0.275\%$. To avoid training and evaluation overlapping, we train on 80% MC and report below results on 20% MC. 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^+)	**99.560**	**36.844**			**99.560**	**36.844**		
	GFPO-F	99.425	34.273	+0.135	+2.571	99.425	34.273	+0.135	+2.571
	GFPO-FR	99.468	35.057	+0.093	+1.787	99.468	35.057	+0.092	+1.787
AD	Oracle (r^+)	**96.092**	**29.910**			**96.092**	**29.910**		
	GFPO-F	95.442	27.926	+0.650	+1.984	95.442	27.926	+0.650	+1.984
	GFPO-FR	95.842	28.875	+0.250	+1.035	95.842	28.868	+0.251	+1.042

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

E.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 (cf. 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 $\tau = \pm 0.025\%$ ($\approx \pm 10$ kHz).

Reporting. Mean over 3 random seeds per method.

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

E.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 2 (25 configurations per trigger). Figure 10 plots in-band rate against signal efficiency across all configurations and all four trigger×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 $\lambda_1 = 0.25$, $\lambda_2 = 1.00$. Sweeping (λ_1, λ_2) moves the convex weight between the three reward terms jointly, and DQN follows this trade-off: its upper hull spans in-band rates from ≈ 0.8 to 1.0, so its operating point is dictated by the weighting, whereas GFPO-F/FR collapse the sweep to a single high-feasibility, high-efficiency cluster.

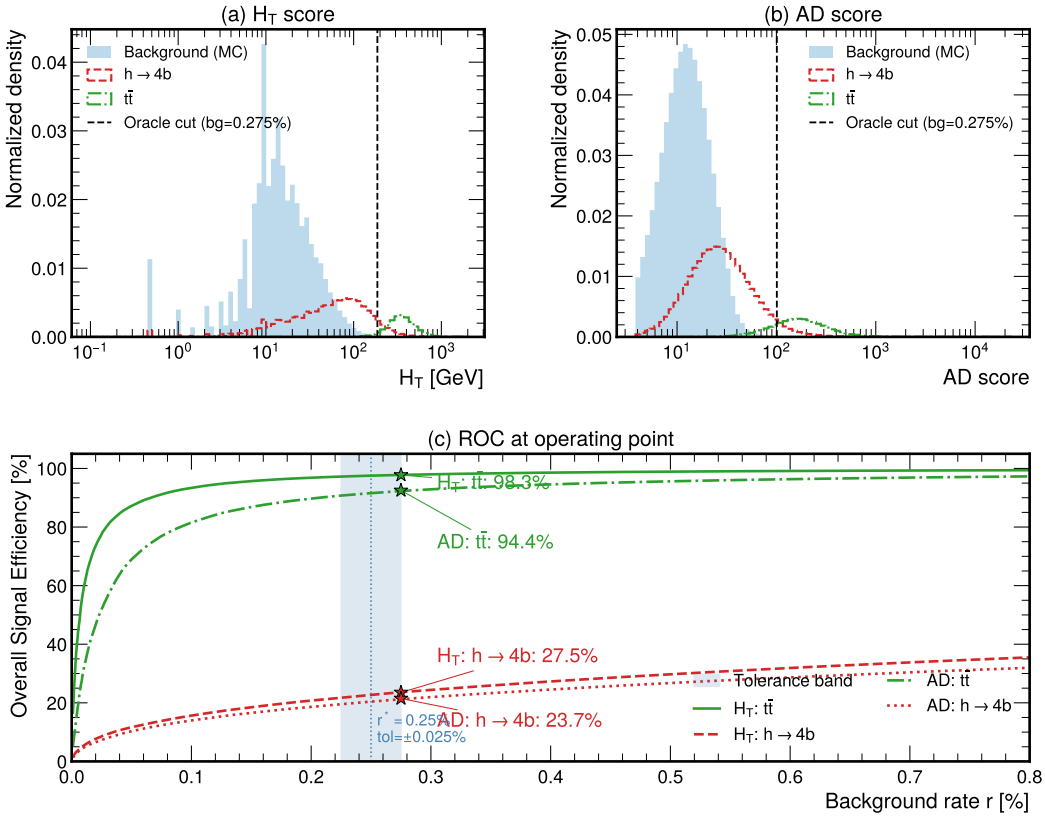


Figure 7: Signal-background discrimination for the H_T and AD triggers (**Full MC**, no train/test split). (a) H_T score and (b) AD score distributions for background (MinBias, shaded blue), $h \to 4b$ signal (red solid), and $t\bar{t}$ signal (green dashed). The dashed black line indicates the oracle 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.

Window size K. K enters in two places with opposing effects: it sets the resolution of the near-cut occupancy and distance-to-cut features, and it sets the sample size of the window-aggregated rate estimate $\hat{e}_t = r_{t,W_t} - r_B^*$ that defines the feasibility set. Small K makes $\hat{e}_t$ noisy, so feasibility decisions become unreliable; large K makes the window stale relative to the drift rate (Figures 5 and 6). Table 7 sweeps $K \in \{32, 64, 128, 256\}$ for GRPO on the AD trigger; InBand and overall efficiency are insensitive to K over this range (CMS InBand 0.536–0.544), so the default $K = 128$ is not a tuned choice. The 22 per-event and chunk-level features (Table 17) were selected with domain experts to expose the quantities that govern local rate sensitivity, and we do not ablate their composition.

Table 7: **Window-size ablation for GRPO (AD trigger).** CMS Run 283408, zero-shot. MC columns (held-out-20% protocol) to be filled.

K	MC InBand	MC $\epsilon_{\text{ov}}^{t\bar{t}}$	CMS InBand	CMS $\epsilon_{\text{ov}}^{t\bar{t}}$
32	TODO	TODO	0.536	75.058
64	TODO	TODO	0.538	75.057
128 (default)	TODO	TODO	0.541	75.059
256	TODO	TODO	0.544	75.061

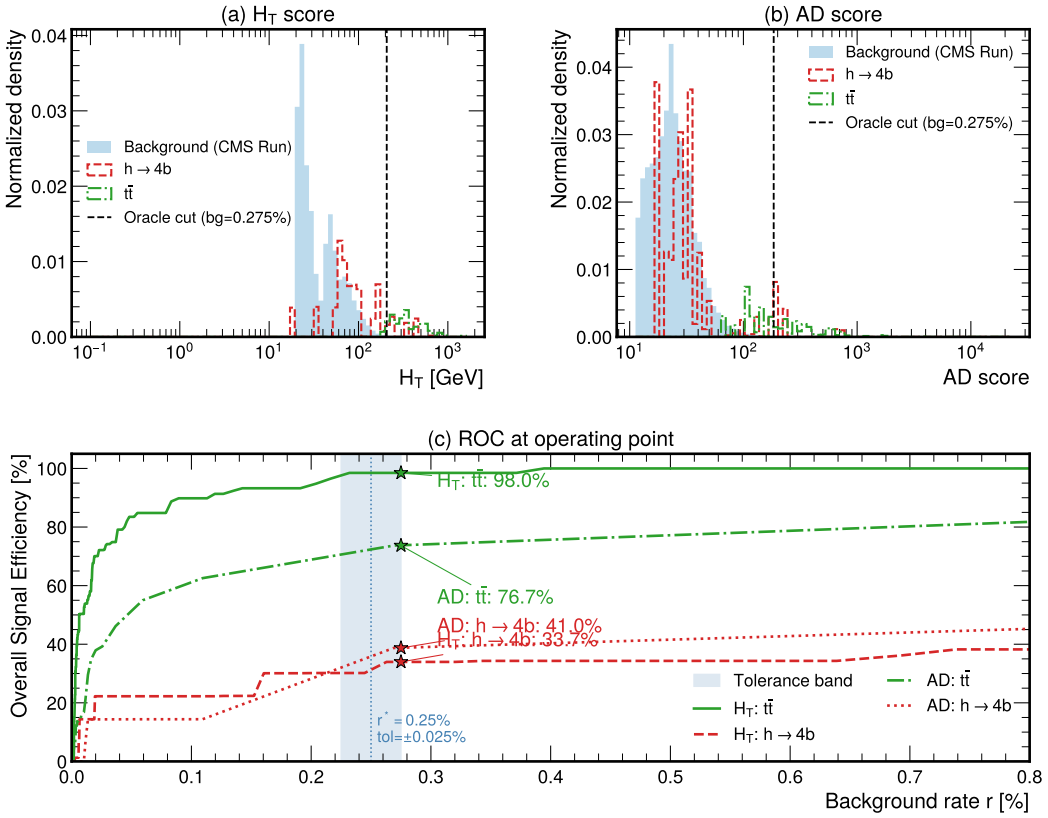


Figure 8: Signal-background discrimination for the H_T and AD triggers (CMS Run 283408). (a) H_T score and (b) AD score distributions for background (MinBias, shaded blue), $h \to 4b$ signal (red solid), and $t\bar{t}$ signal (green dashed). The dashed black line indicates the oracle cut corresponding 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 $h \to 4b$.

E.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 8, 9, and 10 summarize the anomaly detection results for the three experimental settings corresponding to Tables 1, 18, and 2, respectively, which report the LHC-specific operational metrics discussed in the main text.

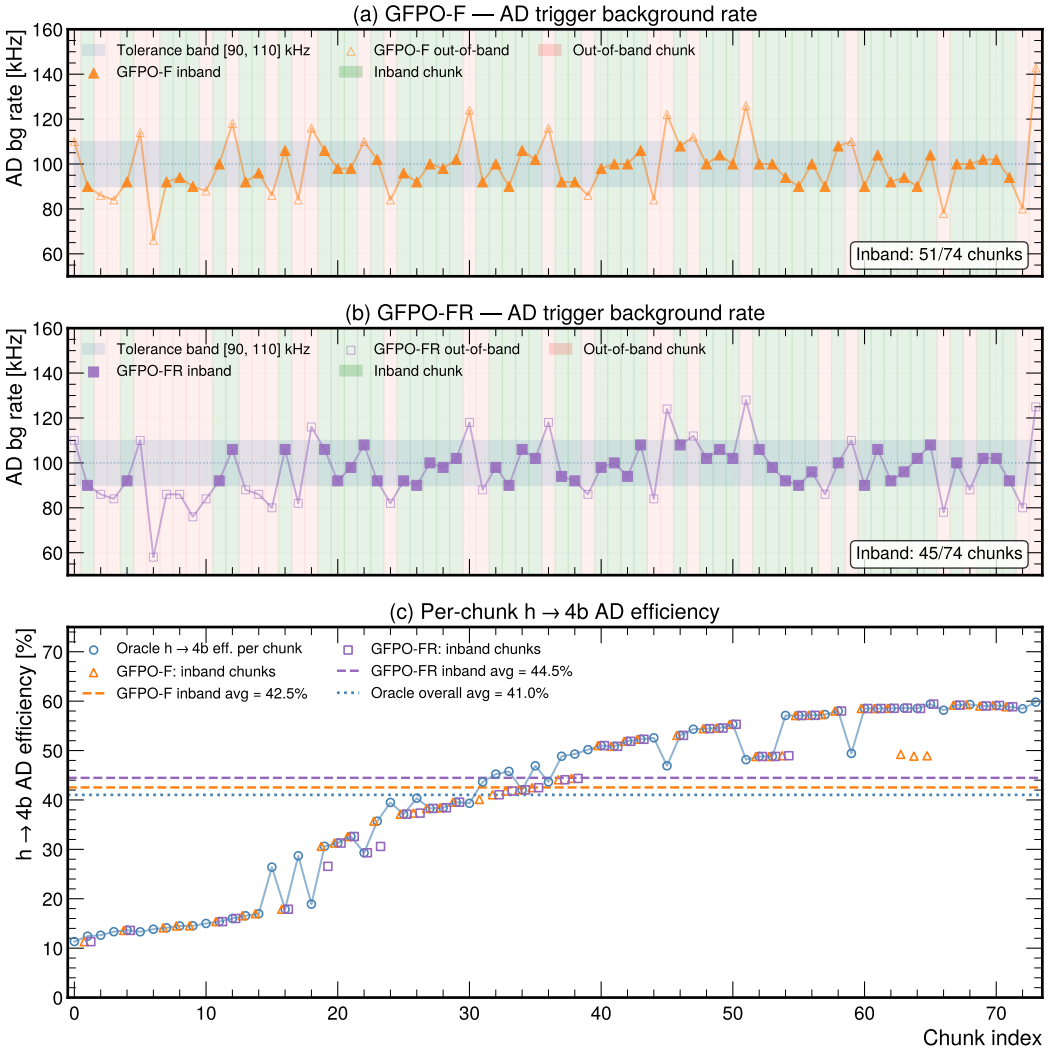
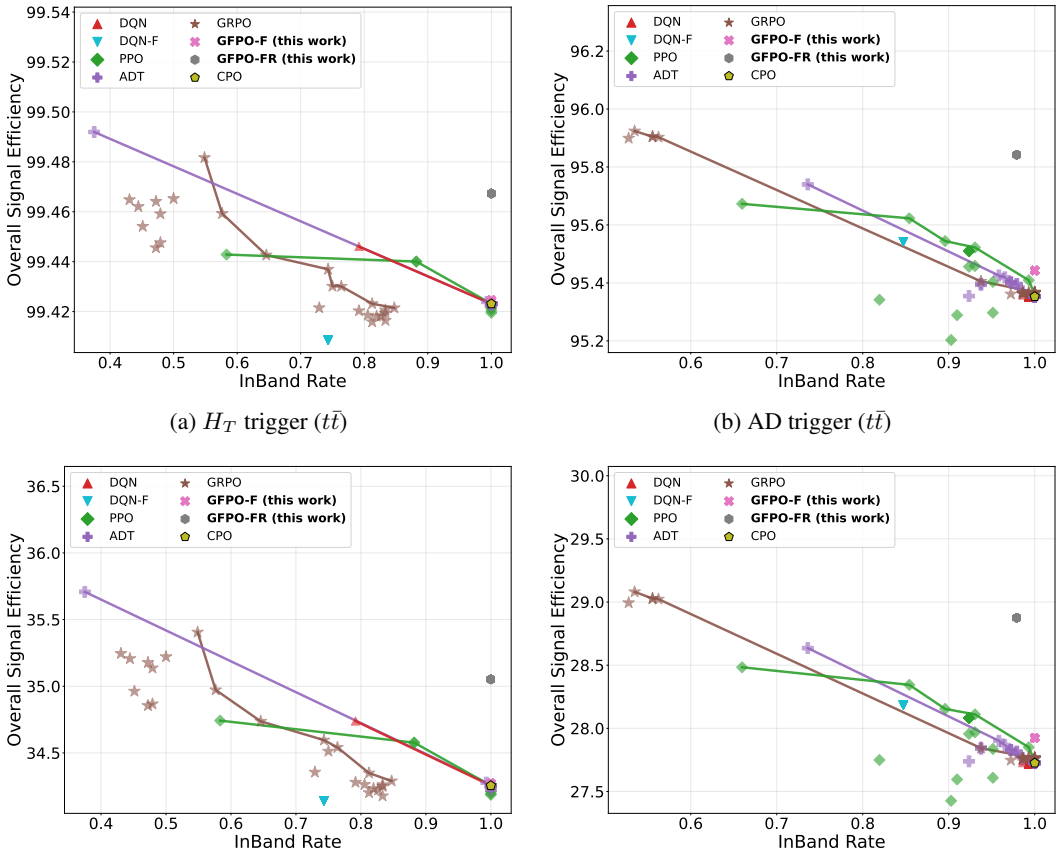


Figure 9: (a) GFPO-F for AD trigger (CMS Run 283408) (b) GFPO-FR for AD trigger (CMS Run 283408) (c) $h \to 4b$ signal efficiency for CMS Run 283408 for GFPO-F (inband) and GFPO-FR (inband) and oracle (r^+) for $h \to 4b$ (overall). The $h \to 4b$ efficiency exhibits strong temporal drift, rising from $\sim 11\%$ 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 2, 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 8: 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 9: 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 18, 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 10: 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

F Additional Implementation Details

F.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. [5]. This yields 195 chunks for MC and 99 chunks for CMS Run 283408 (Table 3).

F.1.1 ADT

We adapt the ADT controller of Yang et al. [13] 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. [13] 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 [73, 74].

F.1.2 DSPOT

Algorithm. DSPOT [23] 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

$$z_q = t + \frac{\hat{\sigma}}{\hat{\gamma}} \left[\left(\frac{q \cdot n}{N_t} \right)^{-\hat{\gamma}} - 1 \right], \tag{14}$$

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 [23]. 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 [23], where signal efficiency is undefined. Our RL methods, by contrast, enforce the operational rate constraints arising from detector readout bandwidth [34] and computing resources [35] 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 18) DSPOT follows the two-phase protocol of Siffer et al. [23]: 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 [5], 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.

F.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:

$$\lambda_{t+1} = [\lambda_t + \alpha_{\text{step}}(|r_t - r_B^*| - \tau)]^+ \tag{15}$$

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

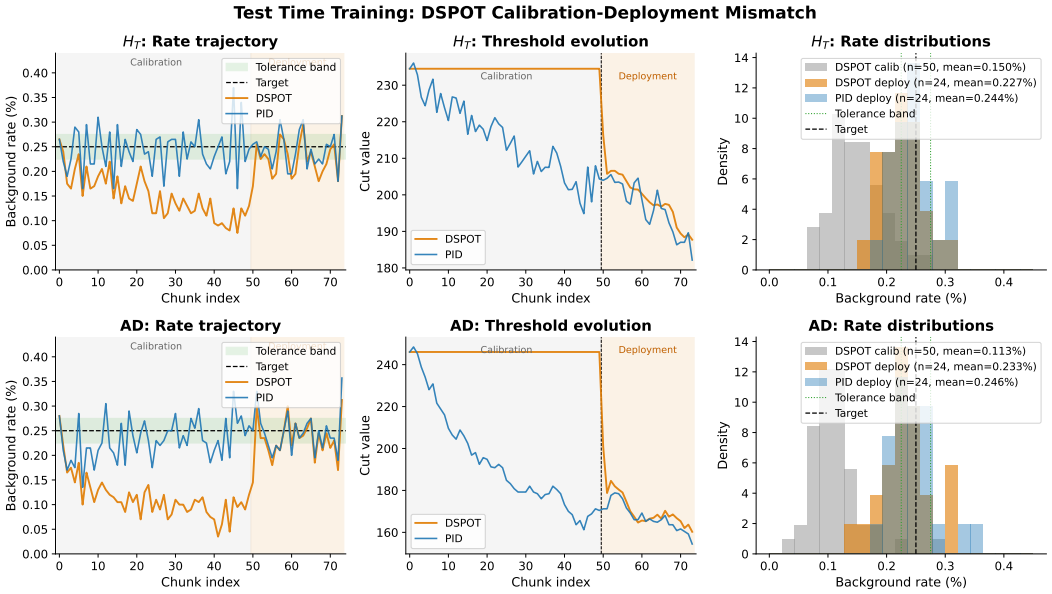


Figure 11: DSPOT calibration-deployment mismatch: the GPD is fitted on 50 calibration chunks where the background rate is 0.113% (AD) and 0.150% (H_T) under the fixed initial cut, well below the 0.25% target, causing a distributional shift of $+0.12\%$ and $+0.08\%$ at deployment that limits the inband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

band and decreases when the constraint is satisfied, automatically tightening or relaxing the rate penalty over time.

The reward in Eq 2 at step t shall be written as:

$$
\begin{aligned}
R_t^{\cdot} = \lambda_t \underbrace{\begin{cases} 1 - \left(\dfrac{|r_{t+1} - r_B^\star|}{\tau} \right)^2 & \text{if } |r_{t+1} - r_B^\star| \le \tau \\ - \left(\dfrac{|r_{t+1} - r_B^\star|}{\tau} - 1 \right) & \text{if } |r_{t+1} - r_B^\star| > \tau \end{cases}}_{\text{rate tracking}} \\
+ (1 - \lambda_t) \underbrace{\left(\alpha \, \epsilon_{t+1}^{t\bar{t},\cdot} + (1 - \alpha) \, \epsilon_{t+1}^{h \to 4b,\cdot} \right)}_{\text{signal efficiency}} \\
- \lambda_2 \underbrace{\left(\frac{|\Delta c_{t-1}^{\cdot}|}{\Delta c_{\max}^{\cdot}} \right)}_{\text{move penalty}}
\end{aligned}
\tag{16}
$$

The policy update follows the standard GRPO objective in Eq. 4 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 E.3.

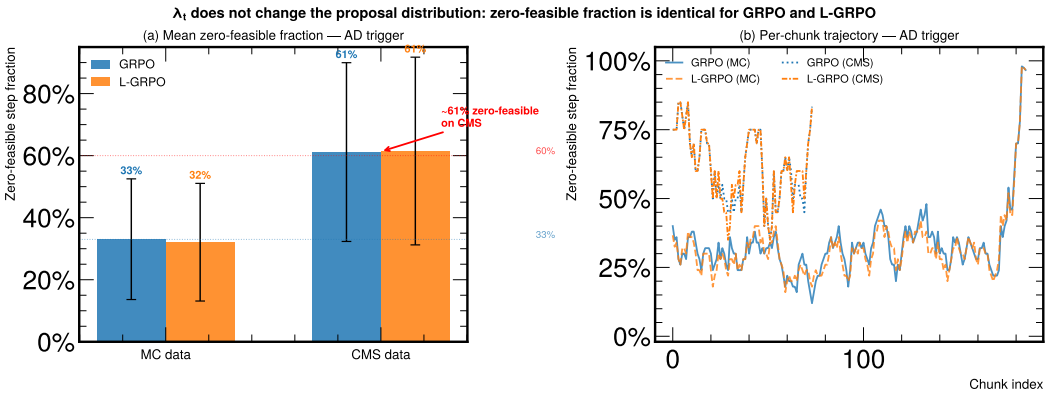


Figure 12: **Zero-feasible step fraction for GRPO and L-GRPO (AD trigger).** (a) Mean zero-feasible fraction across all chunks. On MC data, approximately 20% of micro-steps yield no rate-feasible candidate from the sampled group of G actions for both GRPO and L-GRPO. On CMS real data this rises to $\sim$61% 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.

Relationship to PID-Lagrangian methods. The update above corresponds to the integral term of a PID-Lagrangian controller [50], where the dual step size α_{step} plays the role of the integral gain. We omit the proportional and derivative terms for simplicity, as the GRPO policy already observes the instantaneous rate error and drift through the state representation (Appendix F.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).

F.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. 2 but moves the rate-tracking term from the reward into the cost constraint. Concretely,

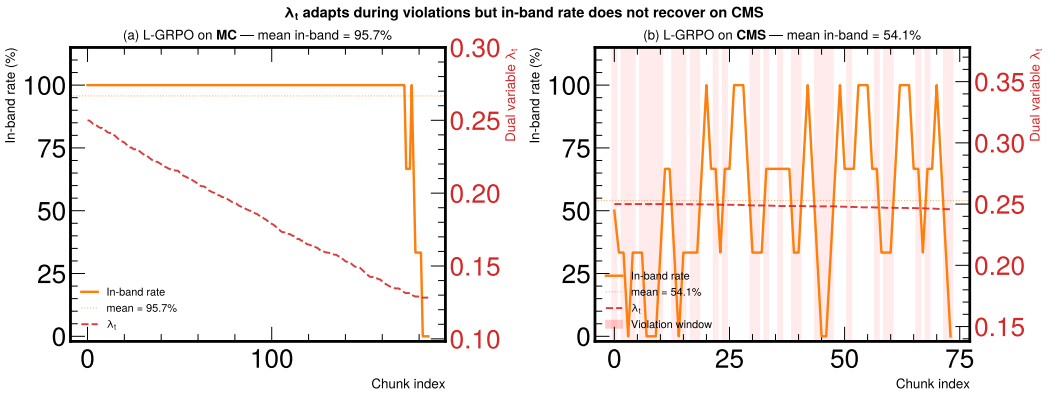


Figure 13: **Dual variable λ_t and in-band rate over chunks for L-GRPO (AD trigger).** (a) On MC data, λ_t decreases monotonically from its initial value of 0.25 as the policy stays in-band ($\sim$96% of chunks). (b) On CMS real collision data, λ_t fluctuates but remains roughly constant (≈ 0.25) while 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.

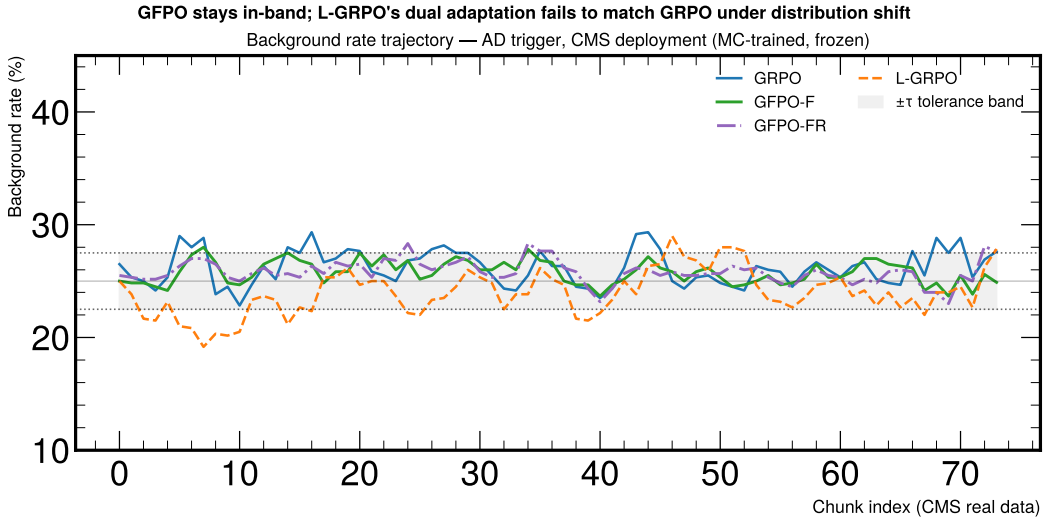


Figure 14: **Background rate trajectory on CMS real data (AD trigger, MC-trained frozen policy).** Grey band: $\pm\tau$ tolerance around the target r_B^*. GFPO-F and GFPO-FR track within the band for most of the 73-chunk deployment window. GRPO and L-GRPO sit near or below the lower band edge throughout, 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.

for a candidate Δ-cut δ_k at time t we evaluate reward (with capital letters)

$$R_k \;=\; (1-\lambda_1)\;\underbrace{\left(\alpha\,\epsilon_{t+1}^{t\bar{t}} + (1-\alpha)\,\epsilon_{t+1}^{h\to 4b}\right)}_{\text{signal efficiency}} \;-\; \lambda_2\;\underbrace{\frac{|\Delta c_{t-1}|}{\Delta c_{\max}}}_{\text{move penalty}} \tag{17}$$

and cost (with capital letters)

$$C_k \;=\; \begin{cases} \left(|r_{t+1} - r_B^\star|/\tau\right)^2 & \text{if } |r_{t+1} - r_B^\star| \le \tau, \\ |r_{t+1} - r_B^\star|/\tau & \text{if } |r_{t+1} - r_B^\star| > \tau, \end{cases} \tag{18}$$

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. 2'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,

$$J_C(\pi_\theta) \;=\; \mathbb{E}_{(s,a)\sim\pi_\theta}\big[C(s,a)\big] \;\le\; d, \tag{19}$$

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 [33] 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. 17–18), 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

$$\max_{\theta'}\; g^\top(\theta'-\theta) \quad \text{s.t.} \quad c + b^\top(\theta'-\theta) \le 0, \quad \tfrac{1}{2}(\theta'-\theta)^\top H(\theta'-\theta) \le \delta_{\mathrm{KL}}, \tag{20}$$

where g and b are the gradients of the importance-weighted reward and cost surrogates at θ, $c = J_C(\pi_\theta) - d$, and $H = \nabla_\theta^2 \mathrm{KL}(\pi_{\mathrm{old}} \,\|\, \pi_\theta)$ 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. [33, 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 33), $\theta' - \theta = -\sqrt{2\delta_{\mathrm{KL}}/s}\, H^{-1}b$, 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. [33] for δ_{KL} and the CG/line-search budgets, with $d = 1$ matching the cost at the band edge.

CPO on zero-feasible groups (expanded from Section 3.2). The natural response to the L-GRPO failure is to reach for a more principled constrained method, and CPO is the canonical one: rather than fold the rate constraint into the reward, it enforces the constraint directly, solving a linearized program at each update subject to a trust-region constraint on the Kullback–Leibler (KL) divergence between consecutive policies [47], which approximately satisfies the expected-cost constraint with a bounded per-iteration violation. Does moving the constraint out of the reward and into the update fix the failure? 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 (Eqs. 17 and 18). 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 in LHC settings it is the latter that matters. Consider what happens when the current policy concentrates on the (background) rate-infeasible region, the same condition that produces all-infeasible groups. CPO's linearized

constraint becomes infeasible, and the algorithm falls back to a recovery step [33, 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 (AD trigger in Table 2). Intuitively, CPO spends half its updates merely climbing back towards feasibility, never improving the in-band 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 18 are nearly identical; see Appendix G.5). Like the adaptive penalty, CPO acts on the objective and the parameter update, but *never* on the proposal distribution.

F.1.5 MDP Structure: Factored Kernel and Markov Sufficiency

This section expands the remark in Section 3 on the structure of the transition kernel. Writing the state as $s_t = (h_t, c_t)$, where h_t summarizes the recent event window and c_t is the current threshold, the kernel factors as

$$T(h_{t+1}, c_{t+1} \mid h_t, c_t, a_t) \;=\; P_{\text{exo}}(h_{t+1} \mid h_t) \cdot \mathbb{1}\big[c_{t+1} = \text{shield}(c_t + a_t)\big], \tag{21}$$

so the problem is an MDP with exogenous context in the sense of Dietterich et al. [28] and Efroni et al. [29]: pileup, instantaneous luminosity, and the background-score distribution evolve independently of the action, and the only sequential dependence runs through the threshold. The reward (Equation 2) depends on the action only through the next-window rate and signal efficiency at the executed threshold.

Relation to a contextual bandit. If the action set were the full threshold range at every step, the factored kernel and the immediate reward would reduce the problem to a contextual bandit with context h_t. The safety shield breaks this reduction: the reachable action set $\mathcal{A}(c_t) = \{c : |c - c_t| \leq \Delta c_{\max}\}$ is a function of past actions. When drift moves the rate-feasible threshold by more than $\Delta c_{\max}$ within one chunk, no per-round rule can recover in a single step, and the policy must trade present rate error against future reachability. This is exactly the regime probed by the zero-feasibility analysis of Appendix F.2 and by the greedy one-step controllers of Appendix F.1.6.

Markov sufficiency of the window state. Treating $s_t = (h_t, c_t)$ as a Markov state is a modeling assumption; we have not shown it to hold for the trigger stream. The assumption is weak at $\gamma = 0$ (Appendix F.1.7), where the state only needs to be informative for the next-window reward, and we support it empirically in three ways: the oracle upper bound of Table 4 leaves little headroom between the learned policies and the best achievable threshold; the greedy one-step controllers with the same inputs of Appendix F.1.6 bound what the engineered state alone can deliver; and the window-size ablation of Table 7 (Appendix E.3) shows the results are insensitive to the length K of the window. We do not establish sufficiency formally.

F.1.6 Greedy One-Step Baselines with the Same Inputs: GRW and RWQ

To isolate whether the gains of the learned controllers come from sequential RL or from the engineered state, we add two controllers that receive the identical feature window used by all RL agents (Table 17) but act greedily at each step.

GRW (greedy reward, one step). GRW enumerates the discrete action grid $\mathcal{A}$, evaluates the exact one-step reward of Equation 2 for every candidate threshold on the current window, and executes the argmax. It is a model-based myopic controller: it uses the measured local rate map (occupancy features and the $\partial r/\partial c$ probe) directly, with no learned policy and no sampling.

RWQ (rolling-window quantile). RWQ sets the next threshold at the empirical $(1 - r_B^*)$-quantile of the background scores in the rolling window, i.e., the threshold that would have produced exactly the target rate on the most recent data. It is the greedy rate controller; it differs from DSPOT (Appendix F.1.2) only in that it takes the exact empirical quantile where DSPOT fits a GPD tail after a 50-chunk calibration burn-in. RWQ has no mechanism for signal efficiency, and it requires an exact online quantile over the full $\sim 5 \times 10^4$-score buffer per micro-step, so we use it as an offline reference; it is not a drop-in controller at L1 latency.

Table 11 reports the results on CMS Run 283408 (zero-shot). The two axes are not independent: a controller can drive InBand to 1 by targeting rate alone, and RWQ does exactly this, but among

rate-compliant thresholds it has no basis for preferring the one that retains signal. GRW maximizes the composite reward greedily and spends rate budget on signal, failing feasibility without gaining efficiency over GRPO. GFPO-F and GFPO-FR span the interior of this frontier by construction, since the filter enforces feasibility first and then ranks the feasible set by signal utility. On CMS-H_T, GFPO-F attains both the highest InBand (0.99) and the highest $t\bar{t}$ efficiency (97.587). On CMS-AD the trade is real and we report it directly: RWQ reaches InBand 1.000 against GFPO-F's 0.689, for 0.55 pp less $t\bar{t}$ and 0.36 pp less $h \to 4b$ efficiency.

Table 11: **Greedy one-step baselines on CMS Run 283408 (zero-shot).** GRW and RWQ see the identical feature window as the RL agents but act greedily. InBand↑ and overall signal efficiency $\epsilon_{\text{ov}} \uparrow$ (%). MC columns are to be filled with the held-out-20% protocol of Table 1.

	MC InBand		CMS InBand		CMS H_T ϵ_{ov}		CMS AD ϵ_{ov}	
Method	H_T	AD	H_T	AD	$t\bar{t}$	$h \to 4b$	$t\bar{t}$	$h \to 4b$
PID [5]	TODO	TODO	0.432	0.405	97.381	33.347	75.053	39.191
GRW (greedy 1-step)	TODO	TODO	0.554	0.432	97.383	33.351	74.892	39.323
RWQ (greedy quantile)	TODO	TODO	0.986	1.000	97.487	33.402	75.012	39.723
GRPO (sequential)	TODO	TODO	0.838	0.541	97.491	33.397	75.059	39.700
GFPO-F (Ours)	TODO	TODO	0.990	0.689	97.587	33.347	75.562	40.083

F.1.7 Choice of Discount Factor

The default in RL is to discount future reward, and our value-based baselines DQN and PPO use the conventional discount $\gamma = 0.95$. In this control problem the discount is a free choice, because three structural features make the problem myopic [48]. First, reward is immediate: each action's effect lands on the next batch, with no delayed rewards. Second, the only action-dependent state is the threshold, which the policy resets at every step, so look-ahead buys nothing that immediate correction does not. Third, the uncontrolled part of the state drifts *exogenously* with beam conditions (changing pileup) and is unaffected by the policy. Together these make the greedy policy optimal, so every γ induces the same optimum and $\gamma = 0$ loses nothing. We verify this empirically: DQN at $\gamma = 0$ matches its $\gamma = 0.95$ counterpart (Table 12), which is why the conventional $\gamma = 0.95$ is harmless. For GRPO and GFPO, $\gamma = 0$ also simplifies the estimator: the return collapses to the immediate reward, so the group-relative advantage estimates the per-step advantage *exactly*, with no value function. The one temporal concern is cross-step smoothness: a myopic policy could chase the per-step optimum with large, oscillating threshold jumps. We suppress this directly through the move-penalty term $\lambda_2 \left(|\Delta c_{t-1}|/\Delta c_{\max}\right)$ in R_t (Equation 2), which charges each step for the magnitude of its threshold change. Because this penalty enters the immediate reward, smoothness is enforced at the step where the jump occurs, and a discounted horizon is not needed for it.

Table 12: **DQN discount ablation on CMS Run 283408 (zero-shot),** $\gamma \in \{0, 0.95\}$**.** MAE, P95, and overall efficiency are unchanged; the only visible movement is in the in-band efficiencies, which change only because the set of in-band chunks changes. Standard deviation over 3 seeds in parentheses. MC rows (held-out-20% protocol) to be filled.

Trigger	γ	MAE↓	P95$\lvert e\rvert$ ↓	InBand↑	$\epsilon_{\text{ov}}^{t\bar{t}} \uparrow$	$\epsilon_{\text{ov}}^{h\to 4b} \uparrow$	$\epsilon_{\text{in}}^{t\bar{t}} \uparrow$	$\epsilon_{\text{in}}^{h\to 4b} \uparrow$
H_T	0	0.0183(0)	0.038(0)	0.878(0)	97.517(0)	33.243(0)	97.465(0)	32.735(0)
H_T	0.95	0.0173(0)	0.037(0)	0.865(0)	97.510(0)	33.213(0)	97.447(0)	33.644(0)
AD	0	0.027(0)	0.061(0.002)	0.531(0.002)	75.089(0.067)	39.885(0.140)	75.892(0.510)	43.795(1.511)
AD	0.95	0.027(0)	0.060(0.001)	0.527(0.002)	75.093(0.060)	39.930(0.130)	76.013(0.510)	44.178(1.509)
MC (H_T, AD)	0 / 0.95	TODO: re-run under the held-out-20% protocol						

F.1.8 KL-Term Ablation

The KL term in Eq. 4 is taken to the previous iterate $\pi_{\theta_{\text{old}}}$ (there is no pretrained reference policy), so it acts as a trust region bounding the per-update policy change. Table 13 sets $\beta = 0$ (we use $\beta = 0.02$ elsewhere). InBand is insensitive to it, consistent with the mechanism we claim: the feasibility filter already restricts which candidates enter the update, leaving little for a constraint on update magnitude to do.

Table 13: **Ablation of the KL term, InBand↑ on CMS Run 283408 (zero-shot).** MC columns (held-out-20% protocol) to be filled; in the rebuttal-time evaluation $\beta = 0$ left MC InBand unchanged for both methods.

Method	β	MC-H_T	MC-AD	CMS-H_T	CMS-AD
GRPO	0.02	TODO	TODO	0.838	0.541
GRPO	0	TODO	TODO	0.838	0.533
GFPO-F	0.02	TODO	TODO	0.990	0.689
GFPO-F	0	TODO	TODO	0.990	0.694

F.1.9 DAPO-Style Dynamic Sampling

DAPO [24] discards groups whose rewards have zero variance ($\sigma_t = 0$) and resamples until the batch is filled. The zero-feasibility failure of Section 4 is the *opposite* regime: on zero-feasible steps our groups are non-degenerate ($\sigma_t > 0$, well-defined advantages, non-zero gradients), and because mean subtraction makes the group advantage shift-invariant, the update confidently reinforces the least-infeasible of G out-of-band candidates. DAPO retains any group with reward spread, including exactly those groups. Table 14 confirms that DAPO-style dynamic sampling is indistinguishable from GRPO on our problem.

Relation to the original GFPO. Filtering before advantage normalization is inherited from GFPO [17]; our contribution is its adaptation to a hard *external* constraint, which differs from the LLM setting in three ways. (i) The filter criterion $|\hat{e}_t(c_t^{(g)})| \leq \tau$ is an absolute threshold set by DAQ deadtime and readout bandwidth, whereas GFPO filters by a preference over responses; the feasible set here moves with the environment, independently of the policy. (ii) The case $\mathcal{F}_t = \emptyset$ has no LLM analogue; handling it requires both the padding rule of Eq. 10 and the shielded update, and, as argued above, it is the opposite of the degenerate-group case that DAPO addresses. (iii) Reward re-weighting does not repair the failure: L-GRPO leaves the zero-feasible fraction unchanged (Appendix F.1.3), and CPO's trust-region update falls back to cost descent on exactly these steps (Appendix F.1.4). This negative result points to the candidate set itself as the place to intervene.

Table 14: **DAPO-style dynamic sampling vs. GRPO, InBand↑ on CMS Run 283408 (zero-shot).** Standard deviation over 3 seeds in parentheses. MC columns (held-out-20% protocol) to be filled.

Trigger	MC (DAPO)	MC (GRPO)	CMS (DAPO)	CMS (GRPO)
H_T	TODO	TODO	0.824(0.01)	0.838(0.02)
AD	TODO	TODO	0.541(0.01)	0.541(0.01)

F.1.10 Signal-Free Reward and Cross-Signal Transfer

Signal labels are unavailable in real trigger operation; they enter our method only through the training reward, and the deployed constraint (hold the background rate in band) is label-free. GFPO-F's keep rule (Eq. 9) ranks candidates by $|\hat{e}_t|$ alone and uses no signal information at any stage; only GFPO-FR's rule involves $\hat{\epsilon}_t^{\text{mix}}$. Two ablations quantify the training-side exposure. (a) *Signal-free reward* ($\lambda_1 = 1$ in Eq. 2): GFPO-F and GFPO-FR are unchanged (InBand $|\Delta| \leq 0.001$ on both triggers in the rebuttal-time MC evaluation), while filter-less GRPO collapses on H_T, so it is the filter that maintains the operating point. (b) *Cross-signal transfer*: training on one signal only ($\alpha \in \{0, 1\}$) and evaluating on the other leaves InBand invariant (range ≤ 0.007) with an efficiency spread ≤ 0.14 pp. Table 15 is to be re-populated under the held-out-20% MC protocol used in Table 1.

Table 15: **Signal-free reward ablation ($\lambda_1 = 1$), InBand↑ on MC (H_T / AD).** To be filled under the held-out-20% protocol.

Method	InBand (default reward)	InBand (signal-free)
GFPO-F	TODO / TODO	TODO / TODO
GFPO-FR	TODO / TODO	TODO / TODO
GRPO (no filter)	TODO / TODO	TODO / TODO

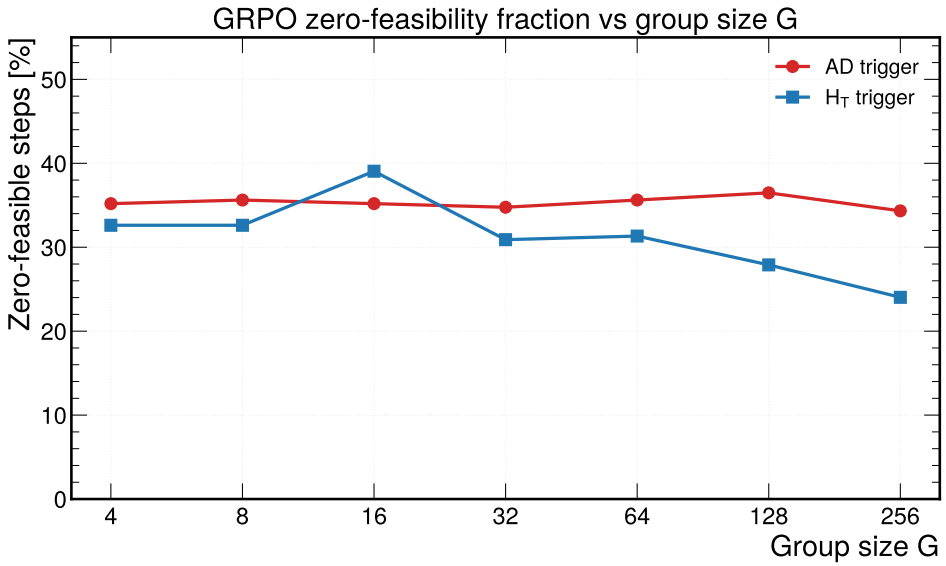


Figure 15: **Zero-feasibility fraction of GRPO vs. group size G.** Fraction of training steps on which every rollout violates the rate budget, for AD and H_T triggers. Even at $G = 256$, 24–34% of steps yield no feasible sample, so the group-relative baseline collapses to an uninformative signal on a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

F.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 (Figures 16 and 17).

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.

F.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 [5]) 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 and 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.

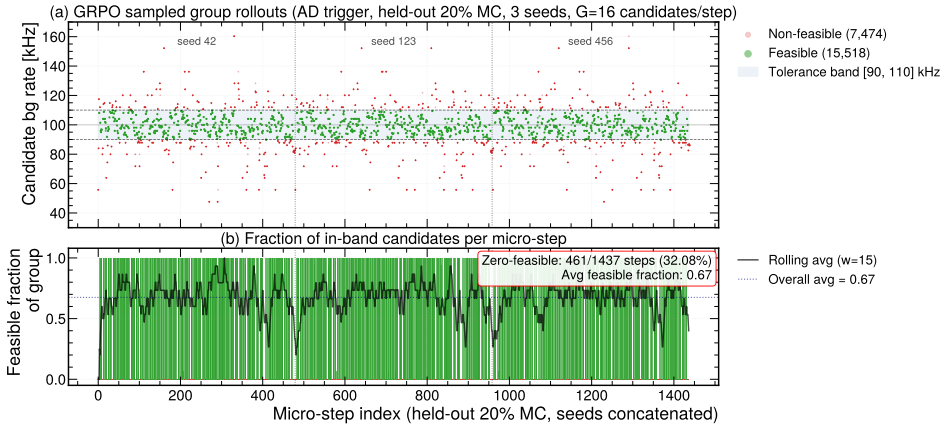


Figure 16: **GRPO group-feasibility failure (AD trigger, MC, G=16); appendix complement to Fig. 2.** (a) Background rates of all G=16 candidate rollouts per micro-step; green/red denotes feasible/infeasible w.r.t. the $[90, 110]$ kHz band. (b) Per-step feasible fraction $f_t = n_{\text{feas}}/G$ (run mean $\langle f_t \rangle = 0.67$). On 32.08% 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.

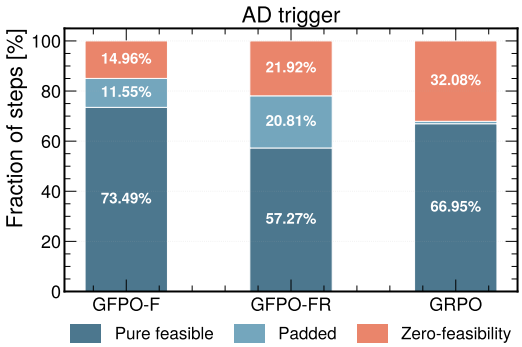


Figure 17: **Step composition for the AD trigger; Complement to Figure 3.** Categories follow Figure 3: *Pure feasible* ($|\mathcal{F}_t| \geq K$), *Padded* $0 < |\mathcal{F}_t| < K$), *Zero-feasibility* ($\mathcal{F}_t = \emptyset$). GFPO-F/FR achieve $57 \sim 73\%$ pure-feasible kept sets and pad on $14 \sim 21\%$ of steps. GRPO is dominated by partial-feasibility (66.95%) and zero-feasibility (32.08%) steps. The latter matches the value in Figure 16 and is the failure mode shielded by GFPO.

Setup. With $\lambda_1 = 0.25$, $\lambda_2 = 1.00$ fixed (Section E.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) [75]; GRU (3 gates, 5,541 parameters) [76]; vanilla RNN with $\tanh$ (1,957 parameters); vanilla RNN with ReLU (1,957 parameters).

Results. Table 16 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 [4, 35, 26], so rate stability is operationally critical. We therefore prioritize InBand rate and adopt GRU as stated in Section 4.

Table 16: 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

Table 17: 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 $N_{pv,i}$	$(N_{pv,i} - \overline{N_{pv}})/\text{std}(N_{pv})$
	Window position	$\text{linspace}(0, 1, K)$
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	$(c_t - \text{mid})/\text{span}$
	Last action	$\Delta c_{t-1}/\Delta c_{\max}$
	Pileup mean	$\overline{N_{pv}}$ over window
	Pileup std	$\text{std}(N_{pv})$ over window
	EMA rate error (λ=0.95)	$\text{EMA}_\lambda[(r_t - r^*_B)/r^*_B]$
	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	$\mathbf{1}[\lvert x_i - c\rvert \leq w_3]$

Recurrent vs. pooled encoder. We compare the GRU against a parameter-matched non-recurrent encoder that mean-pools the K-event window and feeds a 2-layer MLP. For GRPO on CMS-AD the pooled encoder reaches InBand 0.489 vs. the GRU's 0.541, and $\epsilon^{t\bar{t}}_{\text{ov}}$ 75.012 vs. 75.059. We attribute the gap to representation: mean-pooling discards the near-cut occupancy structure that governs local rate sensitivity ($\partial r/\partial c$, Figure 23), which the sequence encoder retains. The mechanism studied in the paper, zero-feasibility of the sampled group and its resolution by feasibility filtering, is measured on the candidate rollouts and does not depend on the choice of encoder.

State representation. We summarize the K-event window with F features per event (Algorithm 2). Features fall into three groups, listed in Table 17: 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 information from its hidden-state recurrence alone. This follows the positional-encoding inductive bias of Vaswani et al. [77] (and its time-series extensions, e.g., Kazemi et al. [78], Lim et al. [79]); 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.

G Additional Experimental Results

G.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

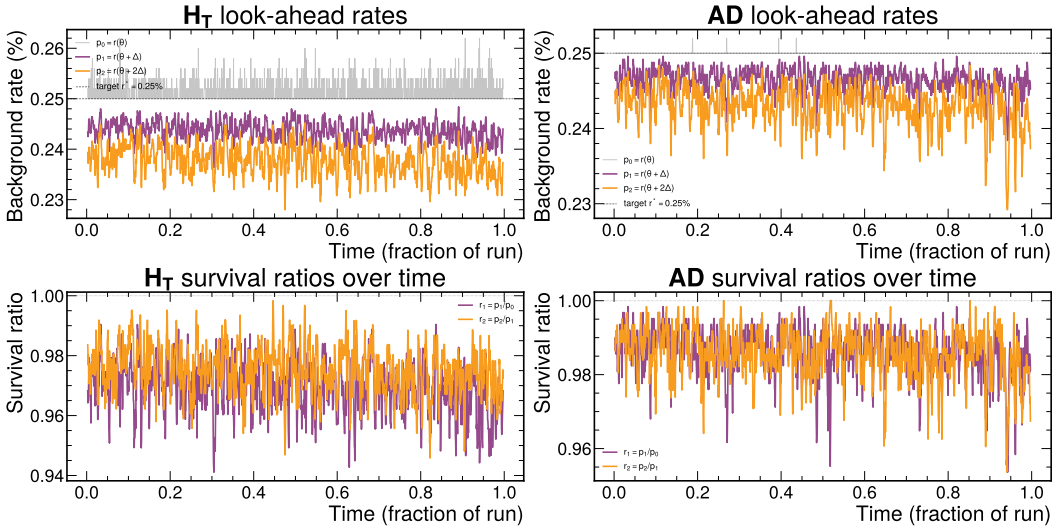


Figure 18: Look-ahead rates (p_1, p_2) and survival ratios (r_1, r_2) over time, evaluated at each window's natural operating point $\theta = P_{99.75}$ so that $p_0 \approx r_B^*$ by construction. The four scalars give the agent a discrete forward-rate map of the tail at $\theta + \Delta$ and $\theta + 2\Delta$ (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.

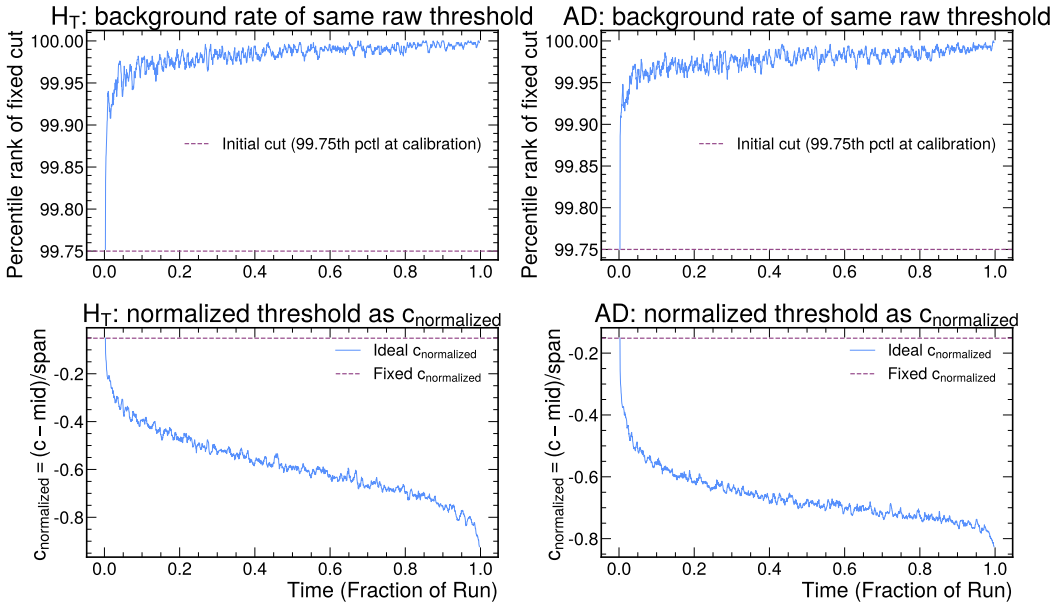


Figure 19: Non-stationarity of trigger thresholds under pileup drift (MC). **Top row**: The percentile rank of a fixed raw threshold (calibrated to the 99.75th percentile of the first 50k events) drifts substantially over time for both H_T (left) and AD (right), meaning the same raw threhsold yields 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.

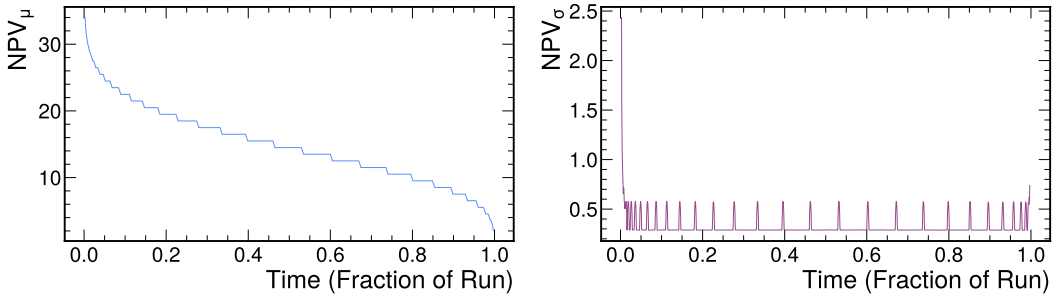


Figure 20: N_{pv} (Number of Primary Vertices) distribution over time (MC).

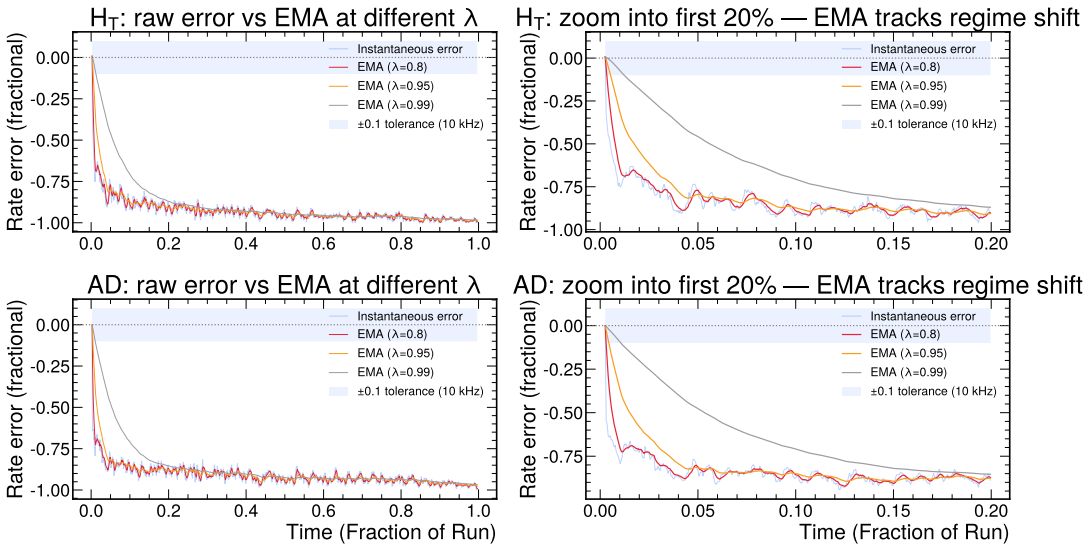


Figure 21: Exponential Moving Average (EMA) of the fractional rate error at different smoothing factors λ, computed over 50k-event sliding windows with a fixed threshold calibrated at the start of the run (MC). The light blue band indicates the ± 10 kHz tolerance around the 100 kHz target rate.

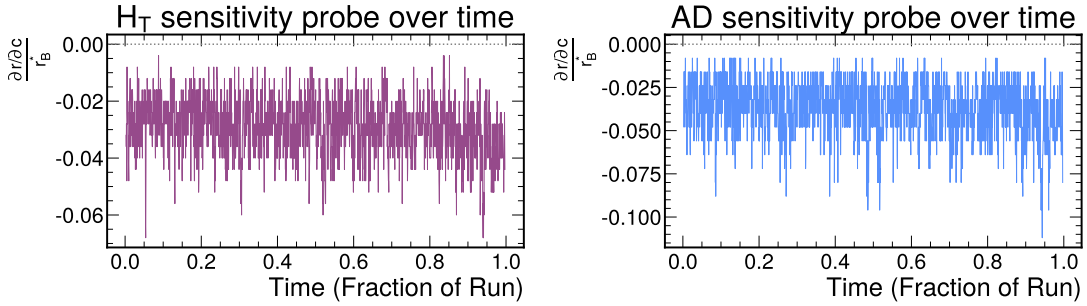


Figure 22: Sensitivity probe over time (MC).

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**

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

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

$e_t \leftarrow (r_t - r_B^*)/r_B^*$;

$e_t^{\mathrm{abs}} \leftarrow |r_t - r_B^*|/r_B^*$;

$\Delta r_t \leftarrow (r_t - r_{t-1})/r_B^*$;

$b_t \leftarrow \mathbb{1}[|r_t - r_B^*| \leq \tau]$;

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**

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

`// 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;

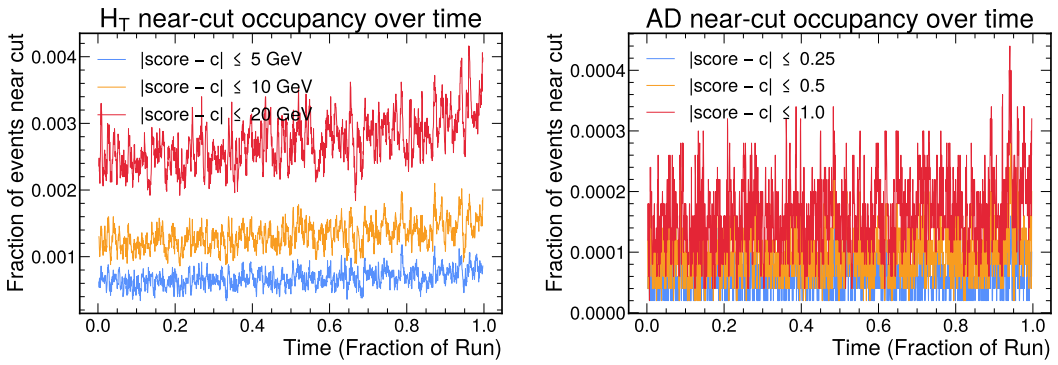
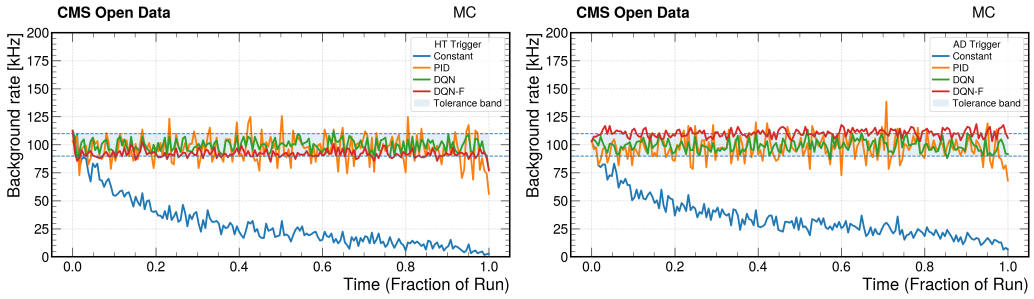


Figure 23: Near-threshold occupancy over time (MC).



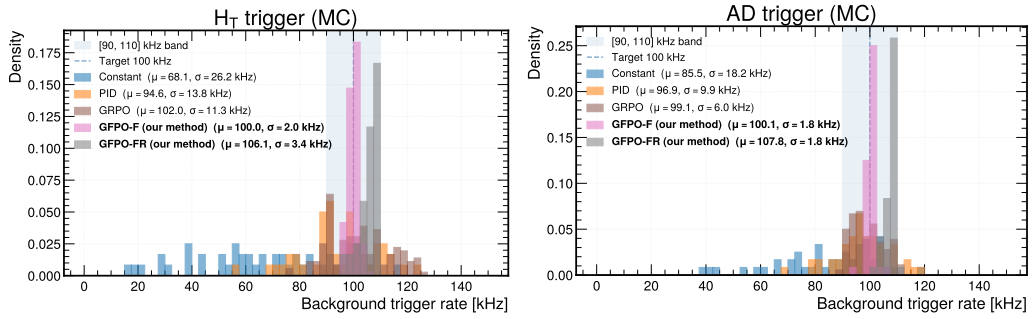
(a) H_T trigger (b) AD trigger

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.

transitions. The two panels show qualitatively the same pattern, indicating that the controller-level differences are not specific to the choice of physics observable.

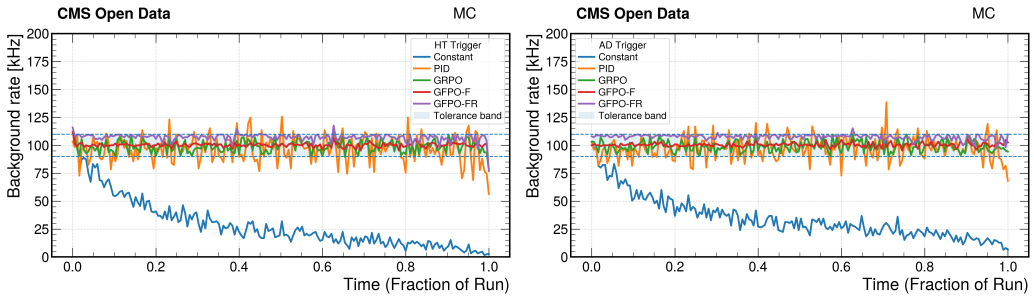
G.2 GRPO MC training

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.



(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. **GFPO-FR (106.1 kHz) operates nearer the band edge for signal compared to GFPO (100.0 kHz).**



(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.

Signal efficiency over time on MC. Emami et al. [5] showcases that adaptive thresholding recovers signal efficiency as a run progresses, while a static menu steadily loses it. We move one step further and demonstrate RL methods have better signal efficiency acceptance over PID baselines. Figure 27 plots per-chunk signal efficiency over the held-out 20% MC run for the H_T and AD triggers, now including the two non-learning baselines, Constant and PID, alongside the adaptive policies (3-seed average, centered window $W{=}5$; at $W{=}$full the curves reproduce the ϵ_{ov} entries of Table 1). The Constant threshold decays on both triggers and both signals, since it cannot track the drifting held-out distribution. The adaptive policies instead climb, re-centering the operating point to hold the background rate on target while recovering signal. The effect is largest on the rare, harder $h \to 4b$ signal: $t\bar{t}$ efficiency is already near-saturated on H_T, whereas the $h \to 4b$ panels open a wide margin, with GFPO-FR reaching roughly 48% (H_T) and 40% (AD) in the late chunks against roughly 24% for Constant. The ordering GFPO-FR $\gtrsim$ GFPO-F $\approx$ GRPO $>$ PID $>$ Constant holds throughout, and efficiency rises gradually rather than in a single jump, indicating *sustained* and *smooth* adaptation to the drift rather than a one-time correction.

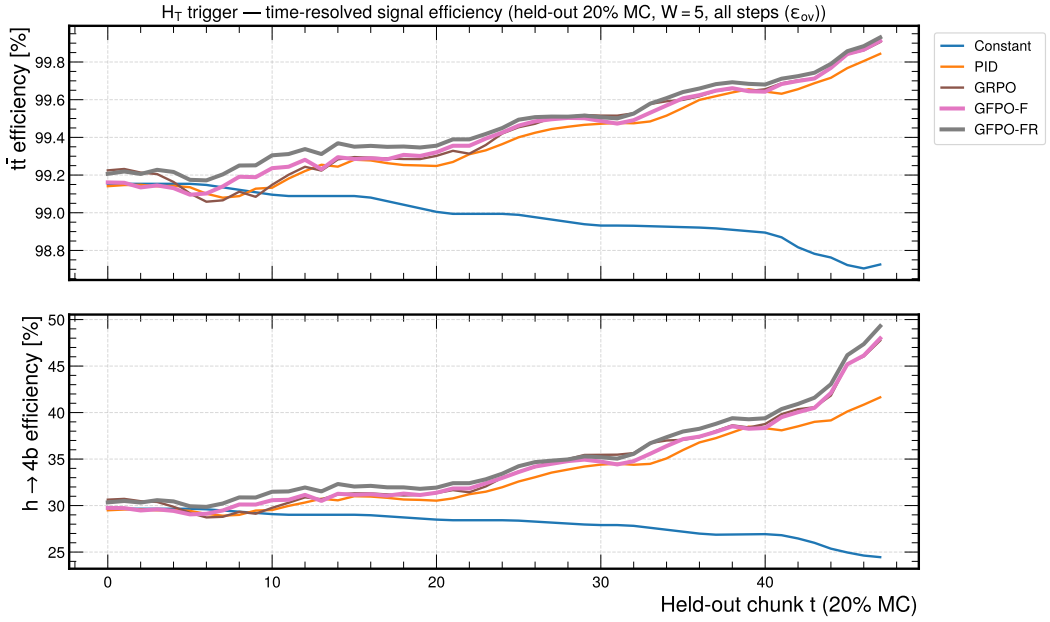
G.3 Sim-to-Real Transfer and Test Time Adaptation

Figure 28 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 29, 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.

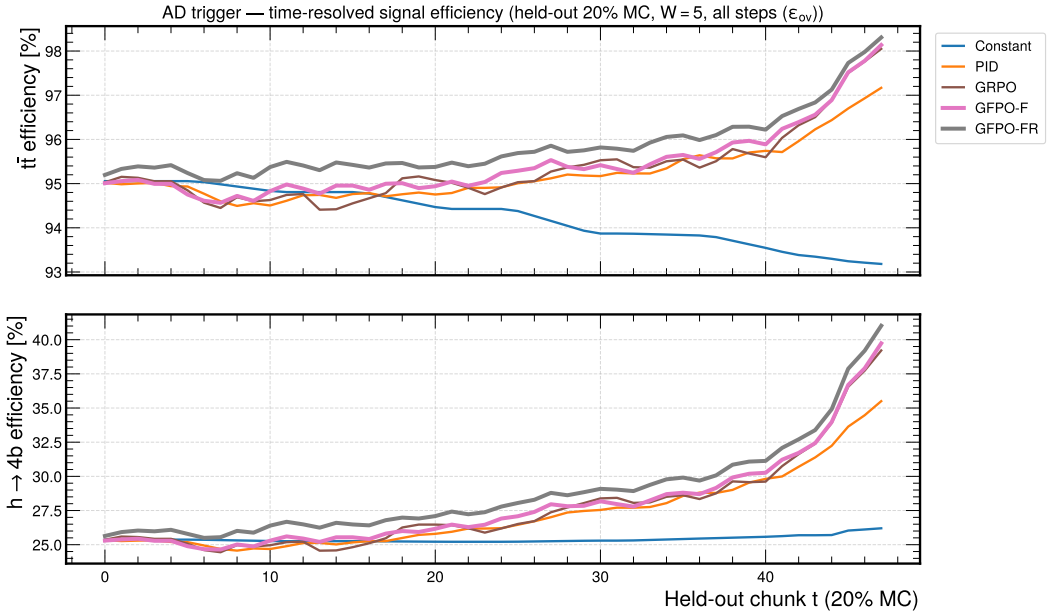
G.4 Testing Time Training on Real CMS Data

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 distributions, background composition, detector response, and running conditions that are difficult to model perfectly (cf. Figure 5 and Figure 6) [80]. 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 [81–83], 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 18. 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

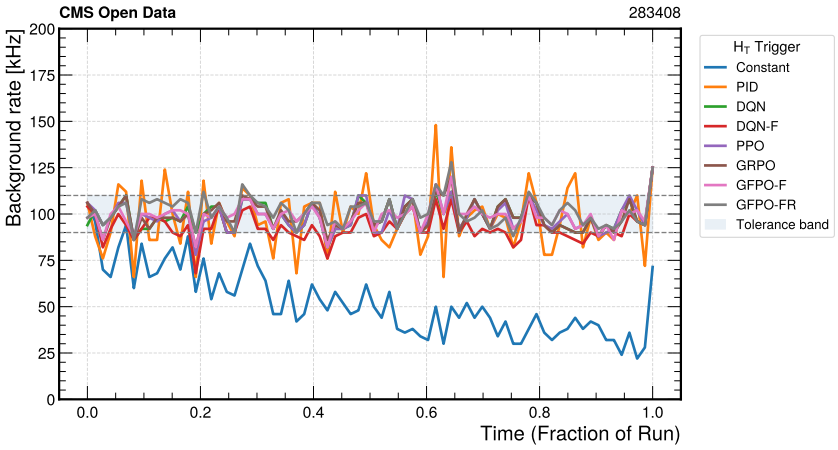


(a) H_T trigger

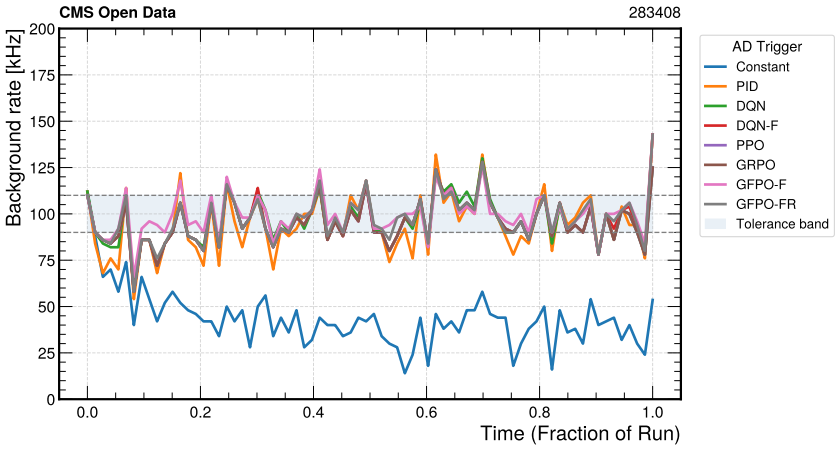


(b) AD trigger

Figure 27: Signal efficiency over time under Constant Menu, PID loop, GRPO, GFPO-F and GFPO-FR (MC) for (a) H_{T} trigger, (b) AD trigger. RL-based agents have higher signal efficiency for both $t\bar{t}$and $h \to 4b$over time than PID and constant menus.



(a) H_T trigger



(b) AD trigger

Figure 28: **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.

occupancy and signal efficiency, indicating better robustness and adaptivity to real-time collision data variability. Figure 29 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 18 and Figure 29) 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.

G.5 CPO degeneracy on the CMS dataset

The CPO entries in Tables 2 and 18 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.

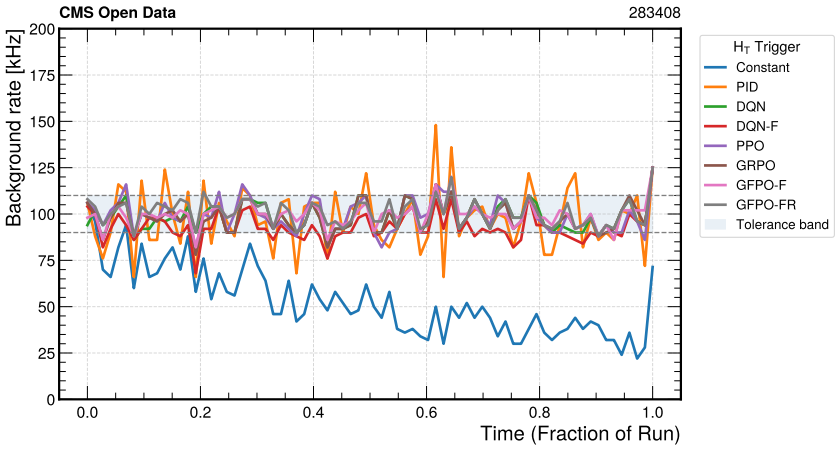
Setup. At each micro-step CPO samples $G{=}16$ candidate actions $a_k \sim \pi_\theta(\cdot \mid s)$ from a discrete set $\mathcal{A} = \{-3, -1.5, 0, 1.5, 3\}$ (AD) or $\{-2, -1, 0, 1, 2\}$ (H_T) of action space size $|\mathcal{A}| = 5$, evaluates each candidate's reward R_k and cost c_k in the simulator (Eqs. 17–18), and executes

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

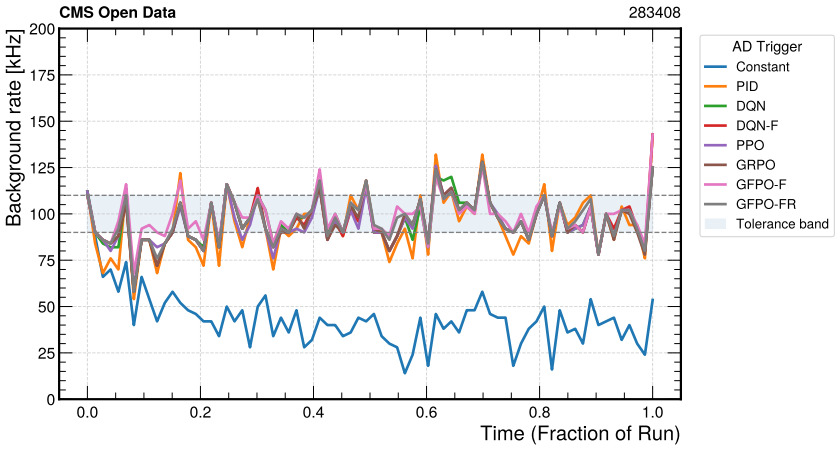
followed by a safety-shield override $\delta = \pm\Delta_{\max}$ whenever $|\bar{r}_{\text{bg}} - r_B^\star| > \tau$. We partition micro-steps by rate state.

The executed action is independent of π_θ. *Out of band* ($|\bar{r}_{\text{bg}} - r_B^\star| > \tau$), the shield overrides the selection with $\delta = \pm\Delta_{\max}$, a function of $(\bar{r}_{\text{bg}}, r_B^\star, \tau, \Delta_{\max})$ 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 $e = |\bar{r}_{\text{bg}} - r_B^\star|/\tau$, 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 [33, Eq. 14] dominates, because the batch estimate exceeds the budget, $\hat{J}_C > d$ (equivalently $\hat{c} > 0$, Eq. 19), on roughly half of the CMS update chunks (consistent with the $\sim$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, $2\delta_{\text{KL}} - c^2/s < 0$, 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 from it, is therefore identical to the frozen rollout.



(a) H_T trigger



(b) AD trigger

Figure 29: **Background trigger rates under MC→CMS transfer with test-time training (Table 18).** 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.

H 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 [84, 81, 83]. 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 19 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

Table 18: **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 [5]	**0.003**(0)	0.085(0)	0.432(0)	97.381(0)	33.347(0)	97.50(0)	**35.242**(0)
H_T	DSPOT [23]	0.080(0)	0.157(0)	0.176(0)	93.998(0)	28.200(0)	96.910(0)	33.278(0)
H_T	DQN [31]	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 [32]	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 [13]	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 [18]	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 [33]	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 [5]	0.035(0)	0.080(0)	0.405(0)	75.053(0)	39.191(0)	76.210(0)	44.573(0)
AD	DSPOT [23]	0.104(0)	0.180(0)	0.135(0)	67.843(0)	27.225(0)	**80.979(0)**	**56.686(0)**
AD	DQN [31]	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 [32]	0.027(0)	0.061(0)	0.54(0)	75.06(0)	39.70(0)	75.600(0)	43.150(0)
AD	ADT [13]	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 [18]	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 [33]	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)

the AD trigger in [5] 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 [81].

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 [85].

I Anomaly Benchmarks

I.1 UNSW-NB15

Setup. UNSW-NB15 is a network intrusion detection benchmark: each record is a network connection (a flow of packets between two hosts) described by flow-level features (duration, byte and packet counts, protocol), labeled as benign traffic or as one of nine attack categories. UNSW-NB15 [53] 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 $t\bar{t}$ [86]) and *Backdoors* as the hard signal (low-footprint behaviour overlaps substantially with normal scores, analogous to $h \to 4b$ [72]). The per-window connection rate replaces N_{PV} as the context variable.

Table 19: 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$\lvert e\rvert$↓	InBand↑	$\epsilon^{tt}_{\text{overall}}$ ↑	$\epsilon^{h\to 4b}_{\text{overall}}$ ↑	$\epsilon^{tt}_{\text{inband}}$ ↑	$\epsilon^{h\to 4b}_{\text{inband}}$ ↑
$d = 4$	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**

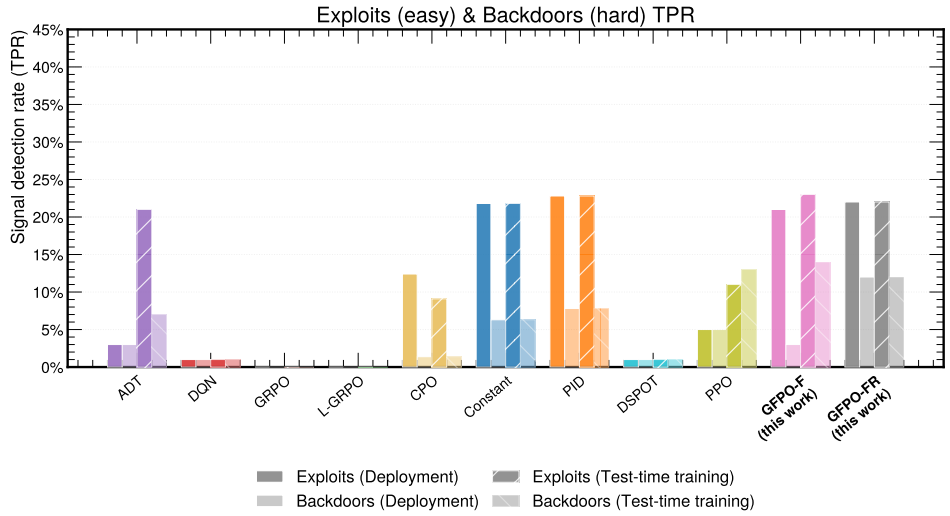


Figure 30: **UNSW-NB15 TPR by attack difficulty and adaptation regime.** For each controller we report TPR on two attack classes: *Exploits* (frequent, high-signal; easy) and *Backdoors* (rare, low-signal; hard). Bars span $\{\text{Deployment}, \text{Test-time training}\} \times \{\text{Exploits}, \text{Backdoors}\}$: 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.

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 ($r_B^* = 0.25\%$ 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 [53], 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. 2): 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 30. 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 [23]. 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.

Analysis on Per-class TPR. Figure 30 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.

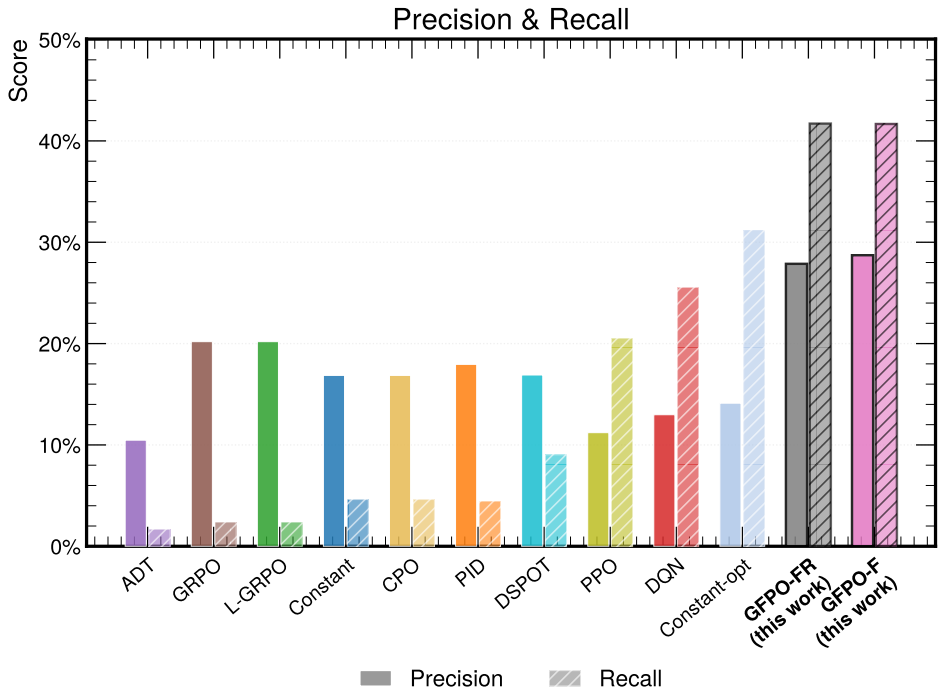


Figure 31: **NAB precision and recall per method (mean over 24 test files).** Solid bars: precision; hatched bars: recall. Methods sorted left-to-right by ascending F1. All controllers operate on the same per-chunk anomaly scores under the same FAR budget. GFPO-F and GFPO-FR (*ours*) dominate the Pareto 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.

I.2 NAB

We evaluate on the Numenta Anomaly Benchmark (NAB) [54], 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). These streams (AWS CloudWatch CPU/network metrics, machine and ambient temperatures, server CPU under known faults) are sampled at regular intervals, processed strictly online with no lookahead, and best match the trigger setting of flagging anomalies on a single live metric under a controlled alert budget. 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 $(p_{25}, p_{50}, p_{75}, p_{90}, p_{95}, p_{99}, \text{threshold}, \text{flagging rate}, \mathrm{TPR}, \mathrm{FPR})$ features. Actions are discrete threshold deltas on a 21-bin grid cover-

Table 20: 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	**0.312**	0.184
PID	0.180	0.045	0.059
D-SPOT	0.169	0.090	0.082
DQN	0.205	0.024	0.038
ADT	0.103	0.014	0.024
GRPO	0.198	0.031	0.042
L-GRPO	0.198	0.031	0.042
CPO	0.169	0.046	0.058
PPO	0.145	0.117	0.097
GFPO-F	**0.320**	0.198	0.215
GFPO-FR	0.316	0.198	**0.216**

ing ± 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 (≈3% flagging rate).

Precision & Recall Figure 31 reports the per-method precision and recall/TPR underlying the F1 summary in Section 5.2 (Figure 4b). Classical controllers (Constant, PID, DSPOT) reach 11–20% precision but TPR/recall below 10%, defaulting to conservative thresholds that miss most true anomalies. PPO and the oracle-tuned Constant-opt push TPR/recall above 10% but drop to 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 (∼20%/3.1%): 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.

Table 20 shows GFPO-F and GFPO-FR (*ours*) reach precision $\approx 32\%$ and TPR $\approx 19\%$, surpassing every baseline on both axes and yielding the highest F1 in Figure 4b. As NAB drops the rate constraint, the feasibility set is vacuous and GFPO-F/FR reduce to TopK quality filtering on the detection reward (Section 5.1); since all RL baselines share the same encoder and state, this gain isolates quality filtering from feasibility filtering, confirming that the filter-before-normalization step improves detection quality even when the FAR-handling mechanism is dropped.

J Extended Limitations

This section expands the limitations summarized in Section 6.

Markov sufficiency of the window state. Treating $s_t = (h_t, c_t)$ as a Markov state is a modeling assumption (Appendix F.1.5). At $\gamma = 0$ the requirement is weak: the state need only be informative for the next-window reward, and the oracle upper bound (Table 4) leaves little headroom between

the learned policies and the best achievable threshold. We also compare against greedy one-step controllers with the same inputs (Appendix F.1.6) and a pooled, non-recurrent encoder (Appendix F.3), which bound the value of the sequential state empirically, but we do not establish sufficiency formally.

Detector faults. We have not tested detector faults such as dead or noisy modules. The controller closes the loop on the measured background rate, with no model of detector state, so any fault that moves $\hat{r}_t$ is read directly by the feasibility test; the regime where this breaks is a shift requiring more than $\Delta c_{\max}$ of correction within a single chunk, in which case no per-step rule recovers immediately and the shield bounds the response. CMS Open Data releases no labeled fault stream, so the only partial evidence we can offer is indirect: the autoencoder latent-dimension and anomaly-score-noise ablations (Appendix H), in which the feasibility-filtered policies retain their advantage under large changes of the score scale and under injected noise.

Aggregated statistics and zero-shot transfer. The controller learns threshold adjustment from aggregated window statistics (rates, near-threshold occupancy, pileup context) and never sees detector-level physics. This abstraction makes zero-shot MC $\rightarrow$ CMS transfer possible but does not guarantee it: GRPO, L-GRPO, and DQN-F receive the same statistics and degrade substantially on CMS (Table 2), so feasibility-filtered training is also required.

Signal mismatch. The two training signals bracket the relevant regime ($t\bar{t}$ is well separated from the background, $h \rightarrow 4b$ overlaps its bulk) and the deployed constraint, the background rate, is signal-independent. A deployed signal spectrum unlike both could still shift the efficiency-optimal operating point within the band. GFPO-FR is the more exposed variant, since its keep rule uses a signal-efficiency estimate; GFPO-F's keep rule (Eq. 9) uses no signal information at any stage, and a signal-free training reward leaves both filtered variants unchanged (Appendix F.1.10).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We are first implementing RL for adaptive trigger control at LHC (real CMS collision data) and it is a valid problem to study under real world context. We provide sufficient empirical evidence for real collision data and anomaly detection benchmarks.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations explicitly in Section 6 and, in expanded form, in Appendix J. We treat the false-positive budget as a single scalar constraint per trigger; production trigger menus involve dozens of interacting streams whose joint bandwidth allocation we leave to future work. We further state that (i) the Markov-sufficiency of our window state is a modeling assumption, (ii) the evaluation is restricted to CMS Open Data with no detector-fault stream, (iii) the controller acts on aggregated window statistics and never sees detector-level physics, and (iv) a deployed signal spectrum unlike both training signals could shift the efficiency-optimal operating point.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This is a specific real-world application at Large Hadron Collider and we include a theory justification in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all hyperparameters choices in main paper and appendix and code in supplement.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our code provides how to access data and how to run all baselines.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes. We provide all hyperparameter and how to access CMS data in CERN data portal.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For our LHC experiments we report statistical significance as standard deviation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See training setup in Appendix E.2 for LHC experiments. See training setup in Appendix I.1 for UNSW-NB15 and Appendix I.2 for NAB.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes. The authors have read the NeurIPS Code of Ethics and confirm that this work conforms to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: There is no societal impact of the work performed.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper has no such risk.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have cited all papers that code we have used. All papers list their proposed datasets.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not introduce new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core method development in this research does not involve LLMs as any important, original or non-standard components. The LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.