CACK: Evolving Gaussian-Process Normalcy Models for Real-Time Detection and Classification of Crowd and Video-Feed Anomalies

Mohammad Umar
Al Ain University
United Arab Emirates
202240077@aau.ac.ae

Awaiz Ahmed
Al Ain University
United Arab Emirates
202310183@aau.ac.ae

Bilal Feroz Khan
Al Ain University
United Arab Emirates
202310641@aau.ac.ae

Abstract—Camera networks must be watched for people who behave out of place, for people who may be concealing an object, and for video feeds that can no longer be trusted. This paper presents CACK, the Cognitive Anomaly Classification Kernel, which models each interpretable signal of a camera network (eight per person, three per feed) with a warped Gaussian process whose compositional kernel is discovered by an evolutionary search guided by the model's own residual diagnostics, derives four tests from each model, calibrates them with extreme-value theory, and names and explains the anomaly class. On CACK-Sim, a procedural benchmark with frame-exact labels (20 runs), CACK reaches an AUROC of 0.9990 (standard tier) and 0.9922 (subtle tier) and, on the standard tier, detects three of the four anomaly types that no component was designed for. On real footage, kernel discovery recovers the stride and step periods of human gait from unlabeled video, a frozen feed in a real shop recording is flagged at its onset, and freezes, replays, and time-warps injected into real video are detected with no false breach under a two-tier gate. On ShanghaiTech, CUHK Avenue, and UCSD Ped2, CACK reaches frame-level AUROCs of 0.707, 0.671, and 0.555: on par with classical detectors under the same front end and with early pose-based networks, below the state of the art. CACK scores a window in 14 µs on one CPU core and moves to a new camera by recalibrating on its normal footage in 14 s. Three findings are of general interest: better model evidence does not by itself improve detection, calibrated thresholds do not transfer between cameras without recalibration, and crowd-level integrity evidence must be gated separately from person-level evidence. A gait-diversity audit shows that, calibrated on typical crowds, CACK raises alarms in 96.8% to 100.0% of the windows of people who walk after a stroke, with Parkinson's disease, with a cane, or with one arm occupied, and names almost all of them concealed carriers. Calibrating on representative crowds and testing gait channels against each person's own walk lowers the worst group to 2.3% while still detecting 94% of behavioral events and 100% of glitches, at the cost of most window-level detections of the concealed-carry proxy. Behind a real tracker, which gives everyone a new identity when a feed is spliced, the published feed checks let every loop, replay, and splice pass in CACK-Sim; identity-free integrity channels catch all seven naive and informed attacks and, on re-tracked real footage, separate tampered from clean video with an AUROC of 0.93 against 0.67.

***Index Terms*—anomaly detection, Gaussian processes, kernel discovery, evolutionary computation, crowd analysis, video integrity, extreme-value theory, explainable AI**

I. Introduction

Operators of camera networks in transport hubs, plazas, and venues watch for three kinds of trouble: people behaving in ways that do not fit the scene (running against a crowd, collapsing, loitering), people who may be concealing something, and feeds that can no longer be trusted because the video has frozen, been looped, or been tampered with. Modern video anomaly detectors learn frame-level or object-level representations and flag reconstruction or prediction errors [1]–[8], or model human pose sequences [9]–[12]. They reach high accuracy on benchmarks, but their scores are not calibrated probabilities, they do not separate a misbehaving person from a misbehaving feed, and they rarely say *why* a frame is anomalous. Classical detectors on hand-crafted features are transparent, but they treat signals as unstructured vectors and miss temporal structure such as the rhythm of a gait.

CACK takes a middle path. It works on a small set of physically meaningful signals that any detector, tracker, and pose estimator can provide: walking speed, turn rate, misalignment with the learned crowd flow, a kinematic jump residual, both arm swings, their bilateral asymmetry, head bounce, and three feed-integrity signals. For each signal it learns a *normalcy model*, a Gaussian process (GP) [13] whose covariance kernel is not chosen by hand but discovered by an evolutionary search over compositional kernels [14]–[16]. Every observation window is tested against its model with four statistics derived from the GP itself, the tests are calibrated into comparable surprise values with extreme-value theory, and a light classifier names the anomaly class while the strongest evidence is verbalized.

What is new. GPs have been used for video anomaly detection by regressing local appearance features [17] and for one-class classification [18]; compositional kernel search [14], [15], [19] and its evolutionary variants [16] fit kernels to single time series for prediction. CACK differs in three ways. First, the search operator reads the *innovations* of the current best model (a residual spectral peak, residual autocorrelation, a weak component) and proposes the structure that is still missing, and the discovered kernels are used as per-signal normalcy models rather than predictors. Second, the four tests derived from each model are tail-calibrated, so the 44 tests of a window share one probabilistic scale that supports thresholds, multiplicity arguments, and a separate gate for feed-level evidence. Third, one model family covers behavior, a

concealed-carry proxy, and video-feed integrity, and every alert is explained in terms of the signal and test that fired. Table I contrasts these properties with representative detectors.

The paper makes eight contributions. (1) The CACK framework: warped, shared-grid GP normalcy models whose kernels are discovered by an evolutionary search with an innovation-driven reasoning operator, trained on normal footage only. (2) Four complementary tests from one GP (structured energy T^2, residual energy Q, the maximum Cholesky innovation, and a generalized-least-squares level test) with extreme-value tail calibration, and a two-tier alarm that keeps feed evidence apart from person evidence. (3) CACK-Sim, a procedural crowd benchmark with frame-exact labels for ten anomaly types in three classes and four held-out types, with an evaluation protocol that keeps person-level events from being credited to concurrent feed glitches. (4) A real-world evaluation: three public benchmarks under one shared perception front end with five baselines, and a deployment study on four real surveillance clips covering kernel discovery on real gait, calibration transfer between cameras, and natural and injected feed tampering. (5) A component analysis of kernel discovery, statistical calibration, and anomaly classification, including anomaly types unseen by every component and transfer between sites. (6) A gait-diversity audit and a remedy: calibrated on typical crowds, CACK flags people whose natural gait differs from the typical one (a limp, a stroke, Parkinson's disease, a cane, or an occupied arm), and representative calibration footage together with person-referenced gait channels, which test a person's gait against their own earlier walk, removes the disparity at a cost that the study measures. (7) A real-time analysis (complexity, latency, throughput, and scalability) of the Python engine and of a browser engine that reproduces it to within 4.6×10^{-12}. (8) A study of feed integrity under realistic tracking and informed tampering: it shows that CACK's feed checks depend on track identities that a real tracker breaks at a splice, introduces identity-free integrity channels tested in frame windows with a calibration that holds for a single feed series, and evaluates them against an attacker who knows the detector, in simulation and on re-tracked real footage.

Artifact availability: The Python engine, the real-video front end, CACK-Sim, all experimental configurations and results, and the live browser console are available for evaluation and reproduction at https://github.com/4waiz/cack; the console runs at https://4waiz.github.io/cack/.

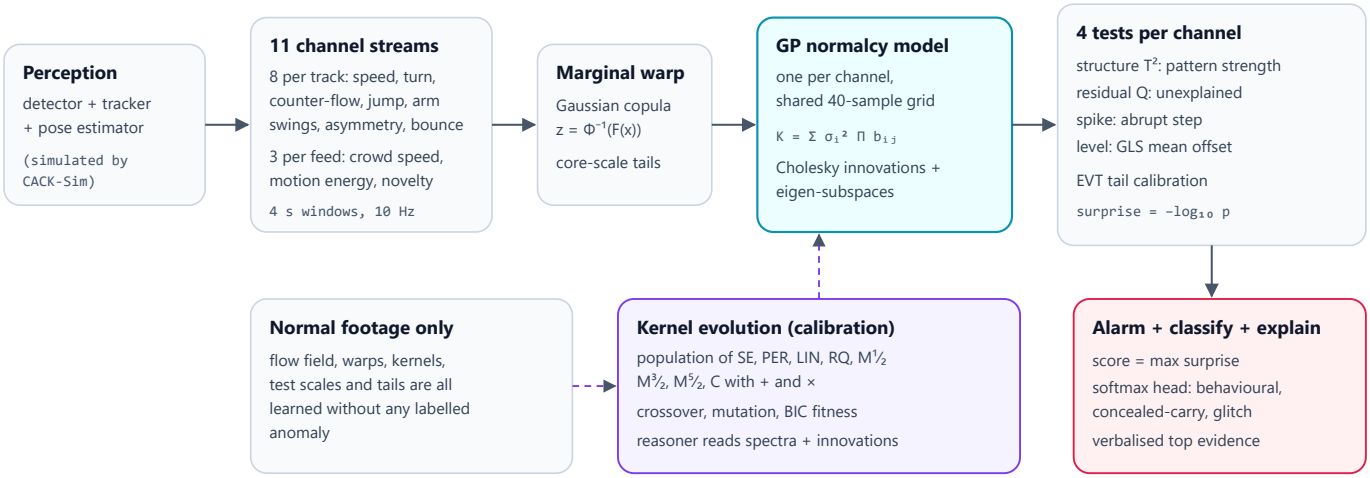


Fig. 1. Overview of CACK. Tracker output becomes eleven channel streams; each is warped to a Gaussian marginal and tested against a GP normalcy model whose kernel is discovered by reasoning-guided evolution; four tests per channel are calibrated into surprise values that drive the alarm, the classifier, and the explanation. Everything except the classification head is learned from normal footage only.

II. Related Work

Video anomaly detection. Benchmarks such as UCSD Pedestrian [1], CUHK Avenue [2], and ShanghaiTech [20] spurred reconstruction-based [3], [6], memory-augmented [6], [7], and prediction-based [4], [7], [8] detectors on frames, optical flow, or object crops, and weakly supervised learning on long surveillance videos [5]. Crowd behavior has been modeled with the social force model [21], which also underlies interaction-force features for abnormal crowd events [22]. Pose-based methods model human skeleton sequences: MPED-RNN decomposes trajectories into global and local motion [9], multi-timescale predictors forecast past and future poses [10], GEPC clusters graph embeddings of pose [11], and STG-NF fits normalizing flows to pose graphs [12]. These methods learn one representation of all people in a frame and score it by a reconstruction error, a prediction error, or a likelihood. CACK instead models a handful of interpretable signals per person with explicit temporal structure and calibrates every test.

Gaussian processes and kernel discovery. Warped GPs transform non-Gaussian marginals [23], GPs have been used for one-class classification [18] and for video anomaly detection by regressing hierarchical appearance features [17]. Compositional kernel search showed that sums and products of base kernels uncover interpretable structure [14], can be described in natural language [15], and scale to large data [19]; spectral mixture kernels learn stationary structure directly [24].

Genetic programming has evolved covariance functions from a kernel grammar [16], and large language models have been used as variation operators in evolutionary search [25]–[27], including to evolve GP kernels for Bayesian optimization [28]. CACK uses an evolutionary search of the same kind but steers it with diagnostics of the current best model's innovations, applies the result to anomaly detection, and studies how model evidence relates to detection power.

Statistical monitoring and calibration. The structured and residual energies used here mirror the Hotelling T^2 and squared-prediction-error statistics of multivariate process monitoring [29], [30]. Extreme-value theory calibrates anomaly thresholds in streams [31], conformal inference turns held-out scores into valid p-values [32], and synthetic anomaly injection has been used for unsupervised model selection [33]. Isolation forests [34] and one-class support vector machines (SVMs) [35] are strong classical detectors [36].

Carried objects, tampering, and tracking. Carried objects have been detected from silhouettes in short video sequences [37], and duplicated frames expose video forgeries [38]. The concealed-carry signature and the reality-glitch channels of CACK are kinematic analogues of these ideas. The real-video front end uses a Keypoint R-CNN [39] and a SORT-style tracker [40]; the ShanghaiTech poses come from AlphaPose [41].

TABLE I. Positioning of CACK Against Representative Video Anomaly Detectors

Method family	Input	Temporal model	Calibrated scores	Explains per signal	Feed integrity	Names class	CPU real time
Frame reconstruction / prediction [3], [4], [6]–[8]	pixels, flow	CNN, a few frames	–	–	–	–	–
Pose-based [9]–[12]	skeletons	RNN, GCN, flow	–	–	–	–	partly
GP regression on features [17]	local features	GP	–	–	–	–	–
Classical one-class [34], [35]	any features	none	–	–	–	–	yes
CACK (this work)	tracks, pose	discovered GP kernels	yes (EVT)	yes	yes	yes	yes

"Calibrated" means that a score maps to a tail probability under normal data, estimated per test. "CPU real time" refers to the scoring stage; every family needs a detector or feature extractor in front of it.

III. Problem Setting and Anomaly Taxonomy

A perception front end reports, at 10 Hz, the position of every tracked person and three pose signals (left and right upper-arm angle and vertical head displacement). CACK must, for every person and every instant, decide whether the last few seconds are anomalous, name the anomaly class, and justify the decision. Only *normal* footage is available to learn what normal is; a small labeled set may be used to name classes, never to detect. Table II lists the taxonomy of CACK-Sim, with severity ranges for the standard and subtle tiers, and the four types held out of every design decision to test generalization.

TABLE II. Anomaly Taxonomy of CACK-Sim (Standard / Subtle Tier)

Class	Type	Mechanism in CACK-Sim
Behavior	Sprint	Preferred speed 2.6–4.5 / 2.0–2.8 m/s for 3–6 s
	Loiter	Stops for 8–15 / 6–10 s, shuffling
	Erratic	Heading jumps of 60–150° / 30–70° every 0.4–1 s
	Counter-flow	Walks against a lane's direction
	Collapse	Abrupt stop, head drops by 110 cm, gait stops
Concealed carry	Carrier	One arm's swing restricted to 35–65% / 65–85%, arm held forward 4–10° / 2–6°, stiffer gait, periodic "security touches"
Reality glitch	Teleport	A track jumps 3–8 / 0.8–2 m and continues
	Freeze	The feed repeats one frame for 2.5–4 s
	Loop	The last 3–5 s are replayed
	Time-warp	Playback at 2× / 1.5× for 3–5 s
Held out: behavior	Sudden limp	Head bob at the stride (not step) frequency of 3–5 / 1.5–3 cm, speed pulsing by 25–40% / 15–25%, 25% slower, for 8–14 s
	Stagger	Heading sways by ±25–40° / ±12–22° at 0.35–0.6 Hz, 15% slower, for 6–12 s
Held out: glitch	Camera shake	Every observed position moves with a common jitter of 15 / 8 cm for 2–4 s
	Stutter	Every other frame is dropped and its predecessor shown twice for 3–5 s

The concealed-carry signature is a *proxy* modeled on behavioral indicators described in security-screening practice: restricted arm swing and gait asymmetry on the carrying side, and repeated touching of the carried object. It is a controlled test of whether CACK can pick up subtle, one-sided kinematic changes; it is not a validated weapon detector (Section VIII).

IV. Method

Fig. 1 gives an overview of CACK. The subsections below follow a window of tracker output through the system, from channels to an explained alert, then describe the real-video front end and the computational cost.

A. Channels

For each track CACK computes eight channels per frame: speed over a 0.2 s baseline; turn rate; a counter-flow index $c\,(1-\cos\theta)/2$ that compares the heading with the mean walking direction of the person's 2 m cell (learned from normal footage, with coherence c); a kinematic jump residual $\log\|p_t - 2p_{t-1} + p_{t-2}\|$; left and right arm swing; the bilateral asymmetry $a_{\mathrm{L}} + a_{\mathrm{R}}$, which cancels for a symmetric gait because the arms swing in anti-phase; and head bounce. Three feed channels summarize the whole frame with medians over tracks, so a single misbehaving track cannot move them: crowd speed; frame motion energy, each person's displacement since the previous frame plus a small term for the change of both arm angles where these are known (an arm hidden by an umbrella or a bag leaves the displacement); and frame novelty, the distance from the current frame to the most similar frame of the last eight seconds, which collapses during a replay. A camera provides a *single* crowd-level series whose level and noise depend on how many people are in view, so each feed channel is self-normalized as a causal robust z-score (a one-minute rolling median and a five-minute rolling median absolute deviation). Channels are cut into windows of W samples: $W = 40$ (4 s) on CACK-Sim and $W = 20$ (2 s) on the short clips of the public benchmarks.

B. Marginal Warping

Several channels are far from Gaussian: the counter-flow index is zero-inflated and turn rates are heavy-tailed. Fitting a GP to such values inflates its variance and makes outliers look plausible. Following warped GPs [23], each channel is mapped through the Gaussian copula $z = \Phi^{-1}(\hat{F}(x))$ estimated on normal footage at 121 quantiles. Outside the calibrated range the map continues linearly with the *smaller* of the local tail slope and the core scale, so a heavy normal tail cannot hide values far beyond anything seen during calibration.

C. Shared-Grid Gaussian-Process Normalcy Models

All windows of a channel share the grid $t_1, \dots, t_W$ and therefore one covariance matrix $K = K_\theta + \sigma_n^2 I$. For M independent normal windows stacked in Y, the pooled negative log marginal likelihood depends on the data only through the scatter $S = Y^\top Y$:

$$\mathcal{L}(\theta) = \tfrac{1}{2}\left[\operatorname{tr}\!\left(K^{-1}S\right) + M\log|K| + MW\log 2\pi\right], \quad (1)$$

$$\frac{\partial \mathcal{L}}{\partial \theta_j} = \tfrac{1}{2}\operatorname{tr}\!\left[\left(MK^{-1} - K^{-1}SK^{-1}\right)\frac{\partial K}{\partial \theta_j}\right], \quad (2)$$

so each optimizer step costs $\mathcal{O}(W^3)$ regardless of how many windows are pooled. Hyperparameters are fitted with L-BFGS-B on the exact gradients (2), bounded to what 10 Hz sampling can resolve (lengthscales of at least one sample, periods of at least three), with a deterministic short-lengthscale start that avoids the "treat rhythm as noise" optimum.

D. Kernel Discovery by Reasoning-Guided Evolution

Kernels are expressions over eight base kernels (constant, squared exponential (SE), periodic (PER), linear, rational quadratic (RQ), and Matérn 1/2, 3/2, and 5/2) combined with $+$ and $\times$. Every expression is expanded to a canonical sum-of-products form $k = \sum_i \sigma_i^2 \prod_j b_{ij}$ in which each additive term owns its variance, so equivalent expressions share one key and every fitted model decomposes into interpretable components. A population of eight kernels evolves for six generations under the Bayesian information criterion (BIC) [42]. Offspring come from *crossover* (join two fitness-selected parents with $+$ or $\times$, or swap terms), *mutation* (replace a base kernel, or add, multiply, or drop a term), and *reasoner proposals*. The symbolic reasoner reads the averaged, tapered periodogram, the variogram exponent, and the between-window variance share of the raw windows to seed hypotheses (for arm swing on the plaza camera: "a 1.07 s rhythm dominates the spectrum (346x the median power) and drifts between cycles, so model it as quasi-periodic PER*SE"), and it diagnoses the innovations of the current best model: a residual spectral peak calls for a periodic term, positive lag-one correlation for a short-range term, alternating signs for a rougher Matérn term, and a weak term for pruning. Offspring inherit the fitted hyperparameters of every term they share with evaluated kernels (Lamarckian warm starts), and survivors are chosen by $(\mu + \lambda)$ truncation. An optional language-model reasoner receives the same diagnostics and proposes validated kernel expressions; it is disabled in all reported experiments, which run offline.

E. Four Tests per Channel

Let $K = LL^\top$ and $e = L^{-1}y$ for a warped window y. Because L is lower triangular, e_i is exactly the standardized one-step-ahead prediction error of y_i given $y_1, \dots, y_{i-1}$ under the GP, so $e \sim \mathcal{N}(0, I)$ for normal windows. Two statistics decompose the squared Mahalanobis distance through the eigendecomposition $K_\theta = U\Lambda U^\top$. With whitened coordinates $w = (\Lambda + \sigma_n^2 I)^{-1/2} U^\top y$ and a structured subspace $\mathcal{S}$ made of the directions whose signal variance stands four times above both the noise and the spectral median, the four tests are

$$T^2 = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} w_i^2, \qquad Q = \frac{1}{W - |\mathcal{S}|}\sum_{i\notin\mathcal{S}} w_i^2, \quad (3)$$

$$\text{spike} = \max_i |e_i|, \qquad \ell = \frac{\mathbf{1}^\top K^{-1} y}{\sqrt{\mathbf{1}^\top K^{-1}\mathbf{1}}}. \quad (4)$$

T^2 asks whether the learned pattern (a gait rhythm) is too strong or too weak, Q whether there is variation the model cannot explain or whether natural jitter has vanished, the spike catches a single implausible step, and ℓ, the standardized generalized-least-squares mean, is the matched filter for an offset and is exactly $\mathcal{N}(0,1)$ under the model. The level test matters because a linear GP predictor is scale-equivariant: a model that captures a rhythm predicts a scaled or shifted

rhythm almost perfectly after one cycle, so amplitude and offset anomalies survive in only a few whitened directions.

F. Extreme-Value Calibration and Aggregation

Each statistic is turned into a robust z-score against calibration windows (median and median absolute deviation). Its magnitude is then mapped to a *surprise* $s = -\log_{10} \hat{P}(A \geq a)$ using the empirical distribution below the 95th percentile and an exponential peaks-over-threshold tail above it [31]. The tests are two-sided, except for the spike and for the directions in which only an increase is anomalous (the structured and residual tests of speed, turn rate, and flow misalignment, and the flow level test), which are upper-tailed. Under normal conditions every surprise is approximately $-\log_{10}$ of a uniform variable, so the 44 tests of a window are equally loud, and the window score $\max s$ has an interpretable scale: a score of s means that the strongest test sits at an estimated tail probability of 10^{-s} on the calibration footage. The alarm threshold is the score quantile on normal calibration windows that meets a chosen false-alarm budget; Section VI measures how well such thresholds hold on new footage.

G. Classification and Explanation

Detection needs no labels. To name a detected anomaly, a class-balanced, L2-regularized softmax regression maps the evidence vector (signed log-compressed z-scores of all 44 tests plus their magnitudes) to behavior, concealed carry, or glitch. The explanation lists the channels with the largest surprise and phrases each by the test that fired, for example "arm-swing asymmetry: its level is shifted up" or "frame novelty: its natural jitter has vanished".

H. Real-Video Front End

For real footage, frames are sampled at 10 Hz and passed to a Keypoint R-CNN with a ResNet-50 feature pyramid [39], which returns person boxes and 17 COCO keypoints. A SORT-style tracker [40] (a constant-velocity Kalman filter on box center, area, and aspect ratio, and Hungarian assignment on intersection-over-union) links detections into tracks; tracks shorter than five samples are dropped. When the background moves (a hand-held camera), a similarity transform estimated from optical flow on the background, with people masked out, maps every frame into the coordinates of the first. Without camera calibration, image coordinates are rectified to approximate meters with a perspective map learned from the people themselves: the pixel height of pedestrians is regressed on the image row of their feet, $h(y) = a + by$, a person 1.7 m tall defines the local scale $1.7/h(y)$, and positions integrate that scale. The arm-swing channels are the signed angles between the torso axis (shoulder center to hip center) and each upper arm, positive in the walking direction, and head bounce is the height of the head above the feet relative to the track's median, in centimeters, so a collapse lowers it by about a meter, as in CACK-Sim. Keypoint gaps of up to 0.5 s are interpolated; windows with missing values are not scored. Because the detector sees each frame independently, feed tampering can be applied to the cached per-frame detections before tracking, which is exactly equivalent to running the whole front end on the tampered video.

I. Person-Referenced Gait Channels

Arm swing, pace, and head bounce are personal. A limp, a stroke, Parkinson's disease, a cane, or a bag carried in one hand changes them for as long as a person is in view, so against a model of the whole population such a gait is anomalous in every window (Section VI-M). CACK therefore offers a person-referenced mode for its five gait channels: speed, both arm swings, their asymmetry, and head bounce. Each statistic u of a window (Section IV-E) is compared with the median $\tilde{u}$ of the same statistic over the person's earlier windows that do not overlap it, and standardized as $z = (u - \tilde{u} - m)/s$, where m and s are the median and the scaled median absolute deviation of $u - \tilde{u}$ on normal footage; its tail is calibrated like every other test (Section IV-F). A stable way of walking thus produces no evidence, while a change in it, such as a sprint, a collapse, or a sudden reach toward the body, does. Only *sound* windows, whose path and feed tests all have a surprise below 4, enter a reference: a frozen or replayed feed shows no real gait, and without this gate the windows that follow a feed glitch caused most of the false alarms of the person-referenced tests. Until a person has five sound earlier windows, about ten seconds after they come into view, their gait channels are tested against the population; path and feed channels always are. The evidence vector keeps its layout, so the class head and the explanations are unchanged, except that explanations mark evidence measured against the person's own gait.

J. Computational Complexity

Let C be the number of channels, W the window length, and N the number of people in view. Scoring one window warps CW samples by interpolation, projects each channel onto its precomputed whitening matrix (W^2 multiply-adds), solves one triangular system ($W^2/2$), and evaluates the level weights (W), so its cost is $\mathcal{O}(CW^2)$ regardless of kernel complexity, because the Cholesky factor and the whitening projector are precomputed. Scoring a frame costs $\mathcal{O}(NCW^2)$; the streaming person channels cost $\mathcal{O}(N)$ per frame and the feed channels $\mathcal{O}(N \log N)$ for the medians plus $\mathcal{O}(NL)$ for the novelty lag search over L lags; tracking costs $\mathcal{O}(N^3)$ in the worst case of the Hungarian assignment. Calibration fits each candidate kernel in $\mathcal{O}(IRW^3)$ for I optimizer iterations and R restarts, independent of the number of pooled windows once their scatter ($\mathcal{O}(MW^2)$) is computed; kernel evolution evaluates about $G(\lambda + \rho)$ candidates per channel for G generations with λ offspring and ρ reasoner proposals. A camera model stores two $W \times W$ matrices per channel (299 kB for all eleven channels at $W = 40$). Section VI-K measures these costs.

K. Self-Calibration on Unlabeled Footage

A calibrated threshold belongs to a camera (Section VI-L), and moving CACK to a new camera has so far needed normal footage of it: recalibration re-learns the flow field, the marginal warps, and the test calibration on curated clips while every channel keeps its discovered kernel. A newly installed

camera offers only its live stream, which carries no labels and contains whatever happened in front of it. Recalibrating on that stream as if it were normal lets its anomalies into the extreme-value tails, which then stop being surprised by them. Self-calibration separates what a minority of anomalous windows barely moves from what it dominates. First, the flow field, the warps, and the location and scale of every test (medians, quantiles, and median absolute deviations) are re-learned on the whole stream, while every tail is kept from the original calibration; the result is poorly calibrated for the new camera but ranks its windows. Second, every window that this adapted detector flags at a 0.1% budget, by the original calibration's threshold for that budget, is set aside together with its frames, and the camera is recalibrated on the rest. The first step is then repeated once, on the footage the second kept, whose statistics are cleaner, before the final recalibration. No label is used at any step, the discovered kernels are kept throughout, and the alarm threshold of a budget is the corresponding quantile of the scores on the footage that was kept.

L. Identity-Free Integrity Channels

Frame motion and frame novelty compare every track with its own past, and all three scene channels are tested inside the windows of people. Behind CACK-Sim's perfect tracker both are harmless. A real tracker, however, associates a detection with a track only near the track's predicted position [40], so when a spliced feed makes every box jump it loses everyone and starts new tracks, the identity switches that tracking benchmarks count [43]. The new tracks have no past to compare with, and until they are one window old no person window exists in which the feed could be tested at all. CACK can instead test a feed with three channels that never use identities. Let D_t be the set of detections of frame t, $d_i(t,k)$ the distance from detection i of D_t to its nearest neighbor in D_{t-k}, $a_i(t,k)$ the change of both arm angles against that neighbor, weighted as in frame motion, and q the upper quartile over the detections of D_t:

$$\begin{aligned} \mathrm{pace}_t &= \log q\big[d(t,2)\big], \\ \mathrm{still}_t &= \log q\big[d(t,1)+a(t,1)\big], \\ \mathrm{echo}_t &= \log \min_{2\le k\le M} q\big[d(t,k)\big], \end{aligned} \tag{5}$$

with a memory of $M = 600$ samples (one minute), each self-normalized like the scene channels. A frozen feed repeats its detections exactly and drives still and echo to their floors; a replay of anything the camera showed within the last minute drives echo to its floor at the replay's lag; fast-forward and slow motion move pace; a splice to other footage moves every detection at once, which makes still and pace jump. The upper quartile keeps the channels informative while most people stand still, and because detections are matched to their nearest neighbors the channels are functions of the per-frame sets of detections alone, which no tracker can change. They are tested in *frame windows*, one window per frame whoever is in view, by the GP models, four tests, and extreme-value tails of every other channel. A feed is a single series, so a few minutes of footage give each channel only a few hundred values and overlapping windows: the marginal warps keep their tails within the quantiles that at least ten calibration values lie beyond, and the exceedances of overlapping windows are declustered by runs one window long [44] before a tail is fitted. The search for an echo compares every detection for lags up to eight seconds and an evenly spread subsample of 16 detections per frame beyond, which costs 2.7 ms per frame of a crowd of 48 people on one core.

Threat model. We consider an attacker who controls the video stream but not the scene in front of the camera: they can repeat, reorder, drop, or resample frames, add noise to them, and substitute footage the camera recorded at another time, and they know how CACK checks a feed, including the eight-second horizon of frame novelty and the integrity gate, as the evaluation of a defense requires [45]. Besides the naive glitches of CACK-Sim (freeze, loop, time-warp), we evaluate four informed attacks: a *replay* of footage 10 to 50 s old, beyond the novelty horizon; a *hold*, a frozen frame kept moving by independent jitter of every detection (5 cm, or 3% of a person's height in real video); smooth *slow motion* at half speed, interpolated between frames, which lets the attacker show the past while the present happens; and a *splice* to footage of another time, beyond the one-minute memory, whose only trace is the seam. Synthesizing a convincing crowd is beyond this model; pixel-level forensics address other manipulations [38], [46].

V. Experimental Setup

A. CACK-Sim

Real footage rarely comes with frame-exact labels for concealed carry or tampering, so CACK-Sim was built as a procedural simulator of the perception front end. Pedestrians follow a social-force model [21] with goal attraction, anisotropic repulsion, walls, and Ornstein–Uhlenbeck wander, along the lanes of four scene layouts. Each person has a gait whose cadence follows speed, anti-phase arm swings, a head bounce at twice the stride frequency, person-specific amplitudes, and slow amplitude drift; tracking and pose noise are added at the observation layer. Feed glitches also act on the observation layer (observed frame t shows world frame $\mathrm{src}[t]$), so a frozen or replayed feed reproduces its frames exactly, noise included. Labels cover the frames each event touches, including the two frames its derivative features are affected by.

Protocol. A run is a (tier, site, seed) triple with 4 anomaly-free calibration clips, two labeled clips used only to fit classification heads, 4 held-out test clips, and 2 further test clips that contain only the four held-out types, each 120 s long with about 48 people in view. Two sites (plaza and concourse), two tiers, and 5 seeds give 20 runs in total. Windows end every 0.5 s; a window is anomalous if at least five of its frames are (or it contains a teleport), normal if none are, and windows that merely graze an event are excluded. A feed glitch is visible to everyone in view, so a window of a person-level event that overlaps a feed glitch could be flagged because of the glitch; per-class metrics, per-type metrics, and event recall of person-level events therefore use only windows that do not overlap a feed glitch. The metrics are the window-level area under the ROC curve (AUROC), average precision (AP), and the detection rate (TPR) at 1% and 0.1% false alarms on normal test windows, event recall and median detection latency at the 1% operating point, and the *realized* false-alarm rate on normal

test windows at thresholds set on the calibration footage; all are reported as mean ± standard deviation over runs. Event recall counts the events that at least one window covers: about 4% of injected events fall before the first complete window of a clip (the feed channels need about 22 s of history).

Gait diversity. Real crowds include many people whose natural gait is not the typical one: mobility is the most common type of disability among U.S. adults, reported by 13.7% [47], and many more people carry a bag, a phone, or a drink in one hand. CACK-Sim can therefore give a share of pedestrians one of six natural gait profiles, each drawn per person: a *limp* (65–90% of the preferred speed, one step 12–26% shorter than the other, and a head nod at the stride frequency), a *post-stroke* (hemiparetic) gait (45–75% of the preferred speed [48], the affected arm held 5–20° forward with 5–35% of its swing, and uneven steps), a *Parkinsonian* gait (reduced and asymmetric arm swing [49], and short, shuffling steps at a higher cadence), a *cane* (the cane arm moving in a short arc 8–20° ahead of the body, slower, with uneven steps), an *occupied arm* (0–40% of one arm's swing, held 0–20° forward, as with a bag, a phone, or a sling), and a *slow* gait (55–80% of the preferred speed, with smaller arm swing and head bounce). These people are labeled normal throughout and are never the target of an injected anomaly. Profiles are drawn from a separate random stream, so scenes without them, including every CACK-Sim scene of the other sections, are unchanged. Unlike the held-out sudden limp of Table II, which begins and ends while a person is in view and is an anomaly, these gaits last as long as a person is in view and are normal; Section VI-M evaluates both in the same crowds.

B. Real-World Benchmarks

UCSD Ped2 [1] (16 training and 12 test clips, 360×240 pixels, 10 frames per second) and CUHK Avenue [2] (16 training and 21 test videos, 640×360, 25 frames per second) are processed by the front end of Section IV-H. For ShanghaiTech [20] (13 cameras, 330 training and 107 test videos, 856×480, 24 frames per second), CACK uses the AlphaPose [41] tracks distributed with STG-NF [12], the input of pose-based methods. Every camera is calibrated on its own normal training videos. These benchmarks contain no feed tampering, so CACK models the eight person channels in 2 s windows. A frame is scored by the most anomalous person in it, as in [12]; in the *offline* setting each window scores its center sample and frame scores are smoothed with a 0.5 s Gaussian, and in the *online* setting each window scores its last sample without smoothing, which is what a live system can do. The metric is the frame-level AUROC over all test frames. Every baseline sees exactly the same windows and calibration footage, and the protocol was fixed before any result was seen.

C. Real Footage

Four real surveillance-style clips from different environments complement the benchmarks: an airport terminal filmed with a hand-held camera, a retail shop seen by a ceiling camera, a rainy street seen by a pole-mounted camera, and an atrium seen from above (Fig. 2). They carry no anomaly labels, so the study measures what can be measured without them: the kernels discovered on real gait, the realized false-alarm rate when CACK is calibrated on the first half of a clip of the same or another camera and applied to the second half, and the detection of feed tampering. Freeze, loop replay, and time-warp are applied to the real frame sequence of the two long static clips and of three CUHK Avenue training videos long enough for the feed channels (at least 20 s of history), with CACK calibrated on clean footage of the same camera; the full tracker and detector then run on the tampered feed. The retail clip also contains a natural glitch: from 98.0 s to its end the recording shows the same frame, updated about every 0.8 s, a frozen feed that we label objectively by its pixel-identical frames.

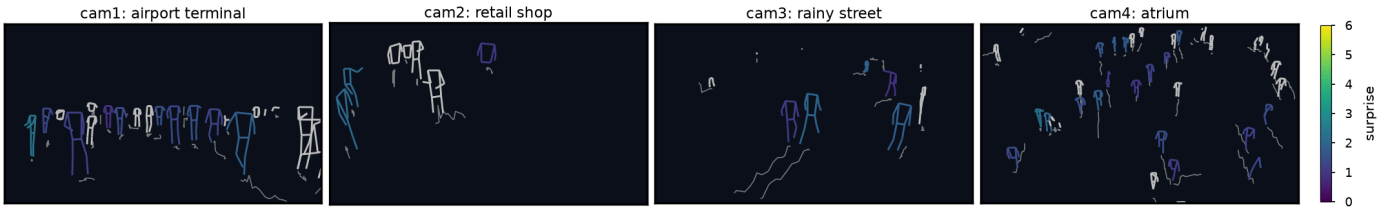


Fig. 2. The four real cameras as CACK sees them: tracked skeletons and their recent trails at the busiest scored frame, colored by window surprise (gray: not yet scored). Only derived tracks and skeletons are shown, never pixels of the footage.

D. Baselines and Implementation

All baselines see exactly the same windows and calibration footage: a robust per-channel z-score, an isolation forest [34] and a one-class SVM [35] on six summary statistics per channel, a textbook GP detector (one SE kernel per standardized channel, scored by window negative log-likelihood), and a 1-D convolutional autoencoder scored by reconstruction error. Appendix A lists every configuration.

VI. Results

A. Detection on CACK-Sim

Table III reports window-level detection as mean ± standard deviation over runs, the recall of evaluable injected events and their median detection latency at the 1% operating point, and the false-alarm rate that a threshold set on the calibration clips realizes on the normal test windows; bold marks the best value per tier.

TABLE III. WINDOW-LEVEL DETECTION, EVENT RECALL, AND REALIZED FALSE ALARMS ON CACK-SIM

Method	AUROC	AP	TPR@1%	TPR@0.1%	Event recall	Latency (s)	Realized FPR
Standard tier (10 runs)							
CACK	**0.9990 ± 0.0007**	**0.9980 ± 0.0012**	**0.991 ± 0.007**	0.940 ± 0.044	**99.2%**	1.13	2.60%
One-Class SVM	0.9985 ± 0.0012	0.9972 ± 0.0022	0.989 ± 0.009	**0.964 ± 0.020**	98.7%	0.95	1.71%
Conv autoencoder	0.9916 ± 0.0046	0.9869 ± 0.0071	0.952 ± 0.028	0.898 ± 0.047	93.3%	1.10	1.09%
Isolation Forest	0.9919 ± 0.0012	0.9795 ± 0.0031	0.879 ± 0.017	0.659 ± 0.072	84.4%	1.35	1.10%
GP likelihood	0.9883 ± 0.0063	0.9638 ± 0.0226	0.757 ± 0.171	0.699 ± 0.140	64.5%	0.76	1.27%
Z-score	0.9694 ± 0.0118	0.9125 ± 0.0381	0.592 ± 0.153	0.517 ± 0.088	51.6%	0.74	1.46%
Subtle tier (10 runs)							
CACK	**0.9922 ± 0.0018**	**0.9867 ± 0.0035**	0.944 ± 0.017	0.890 ± 0.066	85.8%	1.77	4.49%
One-Class SVM	0.9912 ± 0.0025	0.9861 ± 0.0052	**0.952 ± 0.020**	**0.913 ± 0.021**	**90.3%**	1.21	1.93%
Conv autoencoder	0.9746 ± 0.0064	0.9639 ± 0.0085	0.904 ± 0.017	0.850 ± 0.040	83.7%	1.07	1.02%
Isolation Forest	0.9780 ± 0.0023	0.9556 ± 0.0054	0.817 ± 0.026	0.642 ± 0.084	63.6%	1.44	1.02%
GP likelihood	0.9811 ± 0.0065	0.9550 ± 0.0205	0.745 ± 0.151	0.668 ± 0.099	58.0%	1.11	0.95%
Z-score	0.9391 ± 0.0071	0.8706 ± 0.0124	0.606 ± 0.028	0.586 ± 0.031	43.6%	1.31	1.11%

Event recall counts person-level events through windows that do not overlap a feed glitch; under the previous protocol, which also credited such windows, CACK's standard-tier event recall is 100.0%. Latency is the mean over runs of each run's median. Realized false alarms: rate on normal test windows at the threshold meeting a 1% budget on the calibration clips.

On the standard tier CACK reaches an AUROC of 0.9990, an AP of 0.9980, and a detection rate of 0.991 at 1% false alarms; it is the best method on AUROC, AP, and TPR@1% and behind on TPR@0.1% (one-class SVM 0.964 vs. 0.940). It detects 99.2% of evaluable injected events with a median latency of 1.13 s. At the strictest operating point the baselines order as one-class SVM (0.964), convolutional autoencoder (0.898), GP-likelihood detector (0.699), isolation forest (0.659), and per-channel z-score (0.517) at 0.1% false alarms, against 0.940 for CACK. Thresholds set on the calibration clips at a 1% budget realize 2.60% false alarms on the test clips for CACK and between 1.09% and 1.71% for the baselines. On the subtle tier, with milder events and twice the sensor noise, every method loses AUROC. CACK reaches an AUROC of 0.9922 and detects 0.944 of anomalous windows at 1% false alarms; against the one-class SVM it is ahead on AUROC and AP and behind on TPR@1% and TPR@0.1%, so the two have complementary strengths (Section VI-B). The next best baseline, the GP-likelihood detector, reaches an AUROC of 0.9811. The GP-likelihood baseline is instructive: it uses the same kind of model as CACK but scores windows by their raw likelihood on standardized channels, and it detects 0.757 of anomalous windows at 1% false alarms against 0.991 for CACK; the difference is due to warping, test design, and calibration.

B. Anomaly Types

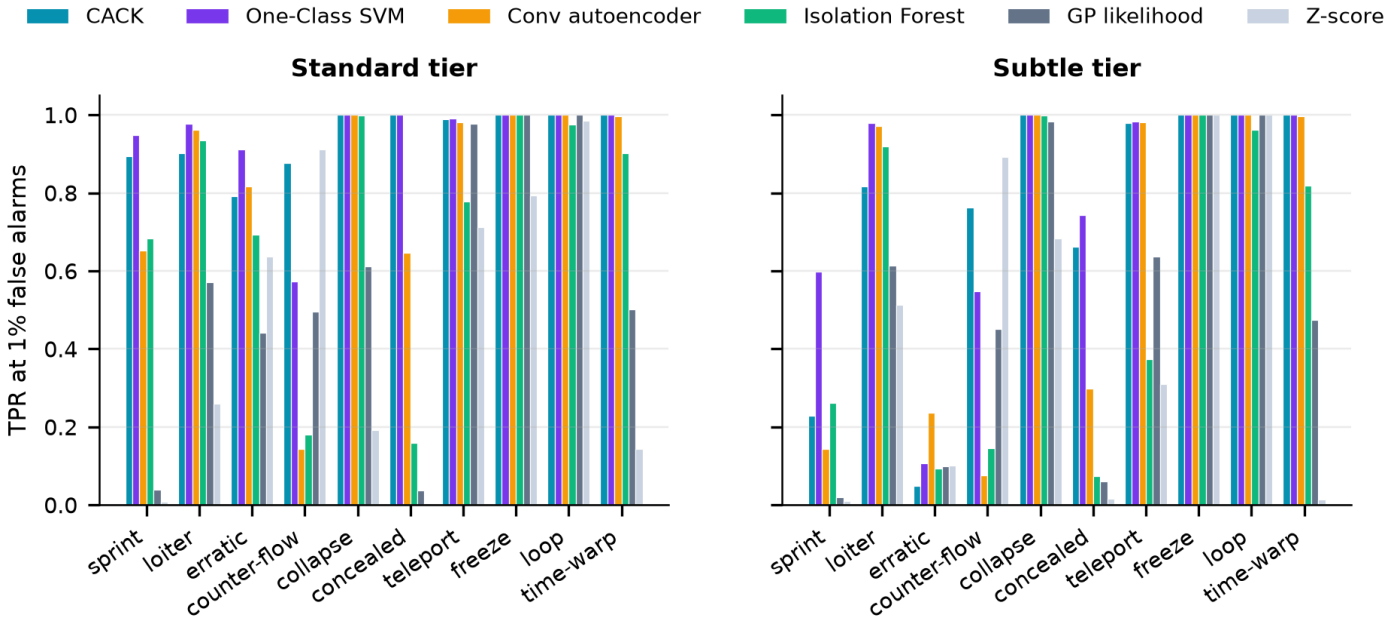


Fig. 3. Detection rate at 1% false alarms for each anomaly type and method, averaged over runs, on the standard (left) and subtle (right) tiers.

Fig. 3 breaks detection down by anomaly type. Reality glitches are the easiest family: frozen, looped, and time-warped feeds move the feed channels by hundreds of standard deviations. The methods separate on the behavioral types and the concealed-carry proxy. Counter-flow is a persistent offset in one channel, which the level test of CACK targets directly:

CACK detects 0.88 of counter-flow windows at 1% false alarms and the one-class SVM 0.57, for which the offset is one feature among 66. Loitering, erratic motion, and sprinting move several kinematic channels at once, and there the joint model of all summaries in the one-class SVM is ahead: loitering (0.98 vs. 0.90 for CACK), erratic motion (0.91 vs. 0.79 for CACK), and sprinting (0.95 vs. 0.89 for CACK). The concealed-carry signature is found through the bilateral asymmetry channel (Fig. 4): a restricted arm breaks the anti-phase cancellation of the two arm swings, which shifts the level of the asymmetry signal and adds a periodic component that the normalcy GP does not expect.

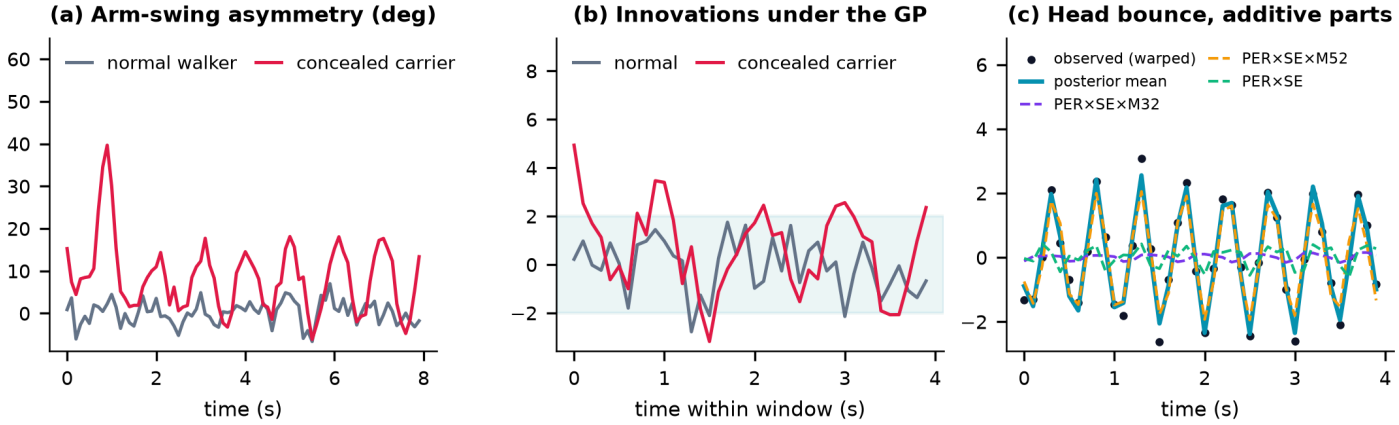


Fig. 4. (a) Arm-swing asymmetry (left plus right arm angle) of a normal walker and of a concealed carrier. (b) The same windows whitened by the normalcy GP of the channel; the innovations of the carrier leave the ±2 band. (c) A head-bounce window and the posterior mean of each additive component of its evolved kernel.

C. Real-World Benchmarks

Table IV compares CACK with the five baselines on the three public benchmarks under the shared front end, and Table V lists published results of detectors that use other inputs. On ShanghaiTech, CACK reaches 0.707 offline and 0.679 online, and the baselines range from 0.680 (convolutional autoencoder) to 0.724 (isolation forest); on Avenue, CACK reaches 0.671 offline and 0.638 online, and the baselines range from 0.672 (isolation forest) to 0.721 (GP-likelihood detector). On Ped2 every method stays between 0.468 and 0.555: its pedestrians are 21–40 pixels tall, tracking jitter gives normal walkers apparent speeds of several meters per second, and most of its anomalies are vehicles, carts, cyclists, and skateboarders that a person-centric front end either misses or cannot tell from jitter. Against published results (Table V), CACK's ShanghaiTech AUROC (70.7) is in the range of the early pose-based networks (73.4–76.1) and of several frame-based detectors (70.5–76.2), and well below STG-NF (85.9) on the same pose input; on Avenue it is at the level of the convolutional autoencoder (70.2) and below frame-based detectors (83.3–91.1). Online scoring, which uses only the past, costs between 0.028 and 0.034 AUROC.

TABLE IV. Frame-Level AUROC on Public Benchmarks (Shared Front End, Offline / Online)

Method	Ped2	Avenue	ShanghaiTech
CACK	**0.555 / 0.565**	0.671 / 0.638	0.707 / 0.679
One-Class SVM	0.510 / 0.556	0.672 / 0.659	0.705 / 0.672
Conv autoencoder	0.529 / 0.550	0.685 / 0.644	0.680 / 0.665
Isolation Forest	0.494 / 0.541	0.672 / 0.644	**0.724 / 0.694**
GP likelihood	0.468 / 0.526	**0.721 / 0.646**	0.709 / 0.691
Z-score	0.518 / 0.546	0.677 / 0.634	0.699 / 0.680

Ped2 and Avenue: Keypoint R-CNN and SORT tracks; ShanghaiTech: AlphaPose tracks. Offline: window scores at window centers, 0.5 s smoothing; online: at window ends, no smoothing. Frame AUROC over all test frames.

TABLE V. Published Frame-Level AUROC (%) of Detectors With Other Inputs

Method	Input	Ped2	Avenue	ShanghaiTech
Conv-AE [3]	frames	90.0	70.2	–
Frame prediction [4]	frames	95.4	85.1	72.8
MemAE [6]	frames	94.1	83.3	71.2
MNAD [7]	frames	97.0	88.5	70.5
HF2-VAD [8]	objects, flow	99.3	91.1	76.2
MPED-RNN [9]	pose	–	–	73.4
Multi-timescale [10]	pose	–	–	76.0
GEPC [11]	pose	–	–	76.1
STG-NF [12]	pose	–	–	85.9
CACK (offline)	tracks, pose	55.5	67.1	70.7

Values as reported in [8] (Conv-AE to HF2-VAD) and [12] (pose-based methods); protocols differ in smoothing and in how frames without people are scored, so the comparison is indicative.

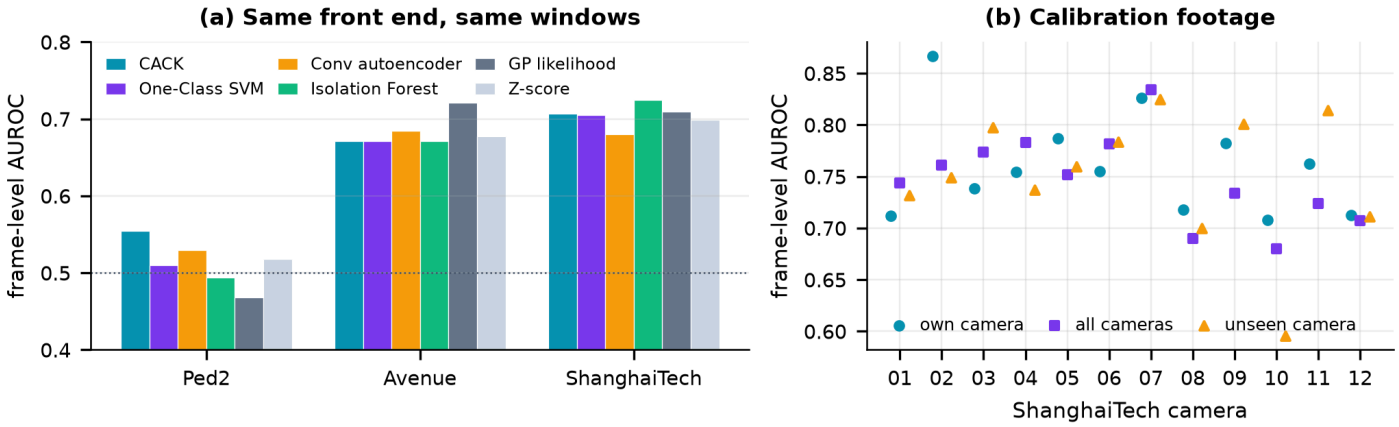


Fig. 5. (a) Frame-level AUROC (offline) of CACK and the baselines under the shared front end. (b) ShanghaiTech per camera: CACK calibrated on the camera's own training videos, on videos of all thirteen cameras, and on videos of the other cameras only (an unseen camera).

Camera environments. The thirteen ShanghaiTech cameras differ in viewpoint, lighting, and crowd, and CACK's per-camera AUROC ranges from 0.708 to 0.867 (Fig. 5b). Calibrating one model on training videos of all cameras gives 0.739, and a model that has never seen the test camera, calibrated on the other twelve, gives 0.748, against 0.707 for per-camera calibration: several cameras have only six to fifteen training videos, and pooled normal footage covers more of normal variation than a camera's own few videos. Normalcy models of person kinematics therefore transfer between cameras once the geometry is rectified, while thresholds do not (Section VI-D).

D. Real Footage

Table VI summarizes the four real cameras. Kernel evolution, run on the unlabeled clips alone, placed periodic terms where people walk freely: on the airport terminal camera, left-arm swing (1.08 s) and head bounce (0.87 s); on the atrium camera, arm-swing asymmetry (1.01 s) and head bounce (0.54, 0.46 s). Periods near one second are the stride and periods near half a second the step, the same rhythms as in CACK-Sim, whose gait was simulated independently. Where people browse or their arms are hidden, the search chose aperiodic kernels: the retail shop camera (arms seen in 68% of samples) and the rainy street camera (arms seen in 31% of samples). The search does not invent a rhythm that is not in the footage. Calibrated on the first half of a clip and applied to its second half at a nominal 1% budget, CACK realized between 3.2% and 44.6% false alarms on the same camera, and between 3.2% and 80.5% when the second half came from another camera: a minute of footage represents neither the normal variety of a scene nor the drift of its crowd well enough for a 1% threshold to hold, and a camera's threshold is meaningless on another camera.

TABLE VI. Real Footage: Perception, Kernels Discovered on Real Gait, and Realized False Alarms

Camera	Clip (s)	People	Tracks	Arms seen	Kernels on real gait (periods, s)	Realized FPR
Airport terminal	14	23	196	52%	left arm: PER×SE + C (1.08); right arm: M12; asymmetry: LIN×M52 + M32; head bounce: PER×SE (0.87)	16.0%
Retail shop	95	12	195	68%	left arm: M12; right arm: LIN×M12 + M12; asymmetry: LIN×M12 + M12 + C; head bounce: SE×RQ	9.5%
Rainy street	63	13	319	31%	left arm: RQ; right arm: RQ; asymmetry: RQ; head bounce: LIN×M12 + LIN	44.6%
Atrium	14	36	103	64%	left arm: LIN×SE×RQ; right arm: PER×SE + SE (1.77); asymmetry: LIN×PER×SE + C (1.01); head bounce: PER×SE + PER + M32 (0.54, 0.46)	3.2%

Kernels discovered with the symbolic reasoner on the whole clip (normal footage), with the periods of their periodic factors; "arms seen": share of track samples with both upper arms detected. Realized false alarms: CACK calibrated on the first half of the same camera's clip, rate on the second half at a nominal 1% budget.

A natural frozen feed. The shop recording freezes at 98.0 s (Fig. 6). Calibrated on 3–60 s of the same camera, CACK flags the freeze at its onset, with feed surprises up to 83. A threshold at a 0.1% budget on the calibration footage also raised false breaches on 29.5% of the clean samples before the freeze; the two-tier gate raised none and still flagged the freeze at its onset. Table VII reports the tampering trials. With the threshold at a 0.1% budget on clean footage, CACK detects 100% of freezes, 72% of loop replays, and 72% of time-warps, but that threshold also breaches on 9.6% of clean samples; under the two-tier gate false breaches fall to 0.0% and detection to 65%, 22%, and 56%. Freezes are the clearest (median peak feed surprise 17) and loop replays the subtlest (median peak 7): a replay shows a crowd that behaves normally, broken only at its two splices. As in simulation, a calibrated threshold on a minute of real footage is too permissive for the crowd-level channels, and a fixed, overwhelming-evidence gate trades subtle glitches for zero false breaches.

TABLE VII. FEED TAMPERING INJECTED INTO REAL VIDEO (56 TRIALS, 5 FEEDS)

Tampering	Trials	Detected (calibrated)	Latency (s)	Detected (gate)	Latency (s)	Feed evidence
Freeze	20	100%	0.0	65%	1.2	76%
Loop replay	18	72%	0.0	22%	0.0	69%
Time-warp	18	72%	2.1	56%	3.0	60%
False breaches (clean)		*9.6%*		*0.0%*		

Detected: an alarm on the feed channels within the tampered span plus one window. Calibrated: the 0.1% budget on clean footage of the same camera; gate: the console's two-tier rule (surprise ≥ 12). Feed evidence: share of alarmed windows in the span whose strongest evidence is a feed channel.

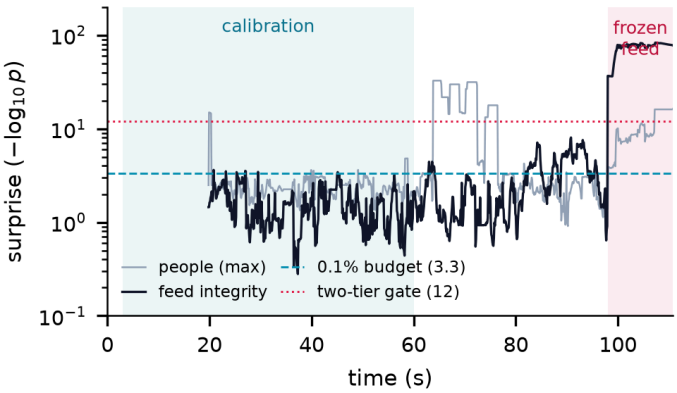


Fig. 6. The retail-shop camera. Feed-integrity and person evidence per sample, with CACK calibrated on the shaded early footage; the recording freezes at 98.0 s.

E. Ablations: Kernel Discovery and Statistical Calibration

TABLE VIII. ABLATIONS OF CACK ON CACK-SIM AND REAL BENCHMARKS (DIFFERENCE TO THE FULL SYSTEM IN PARENTHESES)

Variant	CACK-Sim standard		CACK-Sim subtle		Realized FPR	Real benchmarks	
	AUROC	TPR@1%	AUROC	TPR@1%		ShanghaiTech	Avenue
Full system	0.9990	0.991	0.9922	0.944	2.60%	0.707	0.671
Fixed squared-exponential kernel	0.9989 (−0.0001)	0.990 (−0.001)	0.9923 (+0.0001)	0.943 (−0.001)	2.83%	0.707 (0.000)	0.676 (+0.005)
Best single base kernel by BIC	0.9989 (−0.0001)	0.990 (−0.001)	0.9907 (−0.0015)	0.943 (−0.001)	2.55%	–	–
Evolution without the reasoner	0.9990 (0.0000)	0.991 (0.000)	0.9922 (0.0000)	0.945 (+0.001)	2.42%	–	–
Evolution with detection-aware fitness	0.9989 (−0.0001)	0.992 (+0.001)	0.9926 (+0.0004)	0.948 (+0.004)	2.52%	–	–
No marginal warping	0.9985 (−0.0005)	0.985 (−0.006)	0.9912 (−0.0010)	0.936 (−0.008)	2.46%	0.709 (+0.002)	0.700 (+0.029)
Without the level test	0.9979 (−0.0011)	0.969 (−0.022)	0.9769 (−0.0153)	0.912 (−0.032)	2.63%	0.725 (+0.018)	0.669 (−0.002)
Pooled energy instead of T^2/Q split	0.9989 (−0.0001)	0.991 (0.000)	0.9912 (−0.0010)	0.942 (−0.002)	2.48%	–	–
Max |z| instead of EVT surprise	0.9987 (−0.0003)	0.984 (−0.007)	0.9890 (−0.0032)	0.920 (−0.024)	1.54%	0.697 (−0.010)	0.693 (+0.022)
Gaussian tails instead of EVT	0.9978 (−0.0012)	0.984 (−0.007)	0.9889 (−0.0033)	0.920 (−0.024)	1.54%	–	–
Empirical tails, no EVT extrapolation	0.9799 (−0.0191)	0.000 (−0.991)	0.9547 (−0.0375)	0.000 (−0.944)	5.23%	–	–
Split (out-of-sample) calibration	0.9988 (−0.0002)	0.989 (−0.002)	0.9907 (−0.0015)	0.932 (−0.012)	2.16%	0.705 (−0.002)	0.684 (+0.013)
Plus Gaussian evidence fusion	0.9990 (0.0000)	0.991 (0.000)	0.9922 (0.0000)	0.944 (0.000)	2.85%	–	–
4 s windows on real benchmarks	–	–	–	–	–	0.698 (−0.009)	0.642 (−0.029)

Real benchmarks: frame-level AUROC (offline) under the shared front end; –: not run in that setting. Realized FPR: false-alarm rate on normal test windows at a 1% budget set on the calibration clips.

Table VIII changes one component of CACK at a time. *Tests.* Removing the level test changes detection at 1% false alarms by −2.2 points on the standard tier and −3.2 points on the subtle tier; splitting the Mahalanobis energy into structured and residual parts performs within 0.0010 AUROC of a pooled energy test with the same sidedness. *Calibration.* Aggregating raw |z| without any tail calibration changes detection at 0.1% false alarms by −10.1 and −5.6 points on the two tiers; assuming Gaussian tails changes detection at 0.1% false alarms by −76.6 and −11.4 points on the two tiers; using empirical tails without extreme-value extrapolation changes detection at 0.1% false alarms by −94.0 and −89.0 points on the two tiers. Without a fitted tail, a test's surprise cannot exceed the logarithm of the number of calibration windows, so the scores of many windows tie at that ceiling and a low false-alarm threshold separates little; the extreme-value tail is what lets the 44 tests keep ranking beyond the calibration sample. Calibrating the tests on held-out clips (split calibration) changes AUROC by −0.0002 and the realized false-alarm rate from 2.60% to 2.16%. *Warping.* Marginal warping changes detection at 1% false alarms by +0.6 and +0.8 points on the standard and subtle tiers (positive values favor warping), and by −0.029 AUROC on Avenue, so on real tracker output the warp learned on normal footage does not always help. *Kernels.* Every kernel variant, from a fixed squared-exponential kernel to evolution with a detection-aware fitness, lies within 0.0015 AUROC of the full system on CACK-Sim and within 0.005 on the real benchmarks; adding a Gaussian fusion of the whole evidence vector leaves AUROC unchanged at the reported precision. On the short clips of the real benchmarks, 4 s windows change AUROC by between −0.095 and −0.009 against 2 s windows.

F. Anomaly Classification

TABLE IX. CLASS HEADS ON THE SAME EVIDENCE (MACRO-F1 ON ANOMALOUS TEST WINDOWS)

Class head	Standard	Subtle
Softmax (CACK)	0.934 ± 0.012	0.915 ± 0.017
Softmax, signed evidence only	0.919 ± 0.015	0.895 ± 0.019
Softmax, one labelled clip	0.932 ± 0.015	0.912 ± 0.014
Nearest centroid	0.912 ± 0.010	0.857 ± 0.024
Random forest	**0.953 ± 0.019**	**0.955 ± 0.010**
Channel-group rule	0.924 ± 0.014	0.909 ± 0.012

The class head of CACK reaches a macro-F1 of 0.934 on the standard tier and 0.915 on the subtle tier, with an accuracy of 96.8% on windows that CACK flags at the 1% operating point (Fig. 7). On the same evidence (Table IX, standard tier), dropping the magnitude block of the evidence changes it by −0.015, training on one labeled clip instead of two changes it by −0.002, a nearest-centroid rule changes it by −0.022, a random forest changes it by +0.020, and an expert rule on where the evidence lies changes it by −0.010. A random forest names classes more accurately; CACK keeps the linear head because its decision decomposes exactly into per-test contributions. When one type is withheld from the head's training, its windows are still named with the right class for loiter (0.85), erratic (0.90), counter-flow (0.89), collapse (0.98), and loop (0.85), but not for sprint (0.18), teleport (0.01), freeze (0.38), and time-warp (0.15): the head names a new type correctly when its evidence resembles that of a sibling type of the same class, not from an abstract notion of behavior or glitch. (Concealed carry has no sibling type, so it cannot be named once withheld.)

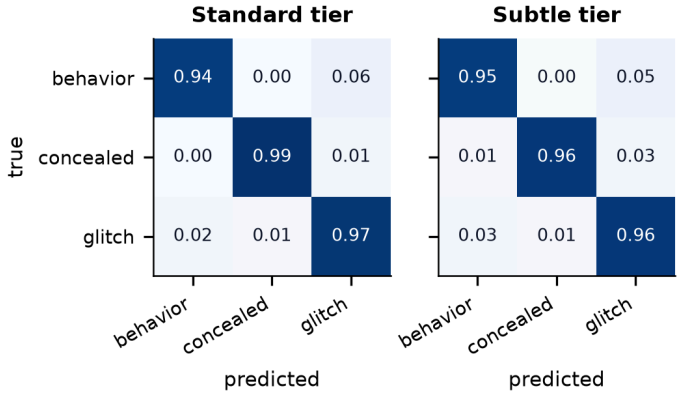


Fig. 7. Row-normalized confusion matrices of the class head of CACK on anomalous test windows.

G. Anomaly Types Unseen by Every Component

TABLE X. Detection Rate at 1% False Alarms on the Four Held-Out Types

Method	Sudden limp	Stagger	Shake	Stutter	Event recall
CACK	**0.93**	0.01	**1.00**	**1.00**	67.7%
One-Class SVM	0.40	0.04	**1.00**	**1.00**	60.1%
Conv autoencoder	0.25	**0.10**	1.00	**1.00**	56.8%
Isolation Forest	0.10	0.04	0.99	0.78	41.8%
GP likelihood	0.02	0.02	0.67	**1.00**	29.7%
Z-score	0.02	0.01	0.54	0.60	24.7%
CACK names class	*0.95*	*0.87*	*0.98*	*0.97*	

Standard tier; windows that overlap a feed glitch are excluded for the two behavioral types. Last row: share of CACK's class-head labels that name the intended class.

Table X reports the four held-out types. They were added to the simulator after every component of CACK was fixed and are absent from calibration, kernel search, and the class head. CACK detects sudden limp (0.93), shake (1.00), and stutter (1.00) at 1% false alarms but misses stagger (0.01), which no evaluated method detects well; the best baseline on it reaches 0.10 (convolutional autoencoder). The class head, trained on the ten known types only, names sudden limp 0.95, stagger 0.87, shake 0.98, and stutter 0.97 of the time with the intended class (behavior for the sudden limp and the stagger, glitch for shake and stutter).

H. Behavioral Anomalies Versus Video-Feed Abnormalities

CACK separates the two families at three levels. In the evidence, feed channels are medians over everyone in view, so a single person cannot move them and a feed fault moves them for everyone. In the class head, glitch windows are named behavior 2.1% of the time and behavior windows glitch 5.6% of the time on the standard tier (Fig. 7): a frozen or time-warped feed makes every person appear to stop or sprint, so person evidence resembles behavior and only the feed channels tell the two apart. In the alarm, the two-tier gate admits feed evidence only when it is overwhelming. On real video, the strongest evidence of the alarmed windows inside a tampered span pointed at the feed channels 76.3% of the time for freezes, 69.1% of the time for replays, and 60.1% of the time for time-warps; replays and time-warps also disturb the tracker at their splices and so produce person-level evidence as well, and feed-level breaches are therefore decided on feed evidence alone.

I. Kernel Discovery: Better Models, Not Better Detectors

Table XI and Fig. 8 summarize the kernels discovered on the standard tier. Evolution finds strongly structured kernels wherever the data are rhythmic: products of periodic, squared-exponential, and Matérn terms for the arm swings, their asymmetry, and head bounce, with periods near the simulated stride (1.06 s) and step (0.54 s). The exact expression varies between runs; 10 of 10 standard-tier runs place a periodic term in the kernel of every gait channel. The largest gains over a squared-exponential kernel are for head bounce (0.56), right arm swing (0.35), left arm swing (0.35) nats per observation. On real footage the same search found the corresponding rhythms where people walk freely (Section VI-D).

The reasoner reaches a better model sooner (Fig. 9): after one generation its BIC gain over the best base kernel is 181.4 per 1000 observations against 133.2 for genetic operators alone, and 228.4 against 229.8 after the last generation.

Yet Table VIII shows that detection barely depends on which kernel is used. The reason is structural. The one-step predictor of a GP is linear and therefore scale-equivariant: a model that has learned a rhythm also predicts a louder, quieter, or shifted version of that rhythm almost perfectly after one cycle, so amplitude and offset anomalies survive in only a few whitened directions, which is what the structured-energy and level tests are designed to read. Better model evidence is valuable for explanation (Section VI-J) and for structural anomalies, but on its own it does not translate into detection power; the test design does.

TABLE XI. KERNELS DISCOVERED ON THE STANDARD TIER

Channel	Evolved kernel (median run)	With PER	Gain vs. SE (nats/obs.)
Walking speed	PER×SE×RQ + PER×RQ×M12 + SE	10/10	0.100 ± 0.026
Turn rate	PER×RQ×M52 + PER×RQ	10/10	0.076 ± 0.025
Counter-flow index	SE×RQ×M12 + M32 + C	5/10	0.008 ± 0.002
Kinematic jump	PER	10/10	0.003 ± 0.000
Left arm swing	PER×SE×M52 + PER×RQ + PER	10/10	0.351 ± 0.013
Right arm swing	PER×SE×M32 + PER×SE×M52 + PER×M32	10/10	0.354 ± 0.011
Arm-swing asymmetry	PER×RQ×M12 + PER×SE + PER	10/10	0.030 ± 0.002
Head bounce	PER×SE×RQ + PER×SE×M52 + PER×SE	10/10	0.561 ± 0.015
Crowd speed	PER×SE + PER×M12 + M52	8/10	0.015 ± 0.007
Frame motion energy	PER×SE×M32 + PER×SE + M52	10/10	0.055 ± 0.007
Frame novelty	PER×RQ×M52 + RQ	7/10	0.012 ± 0.006

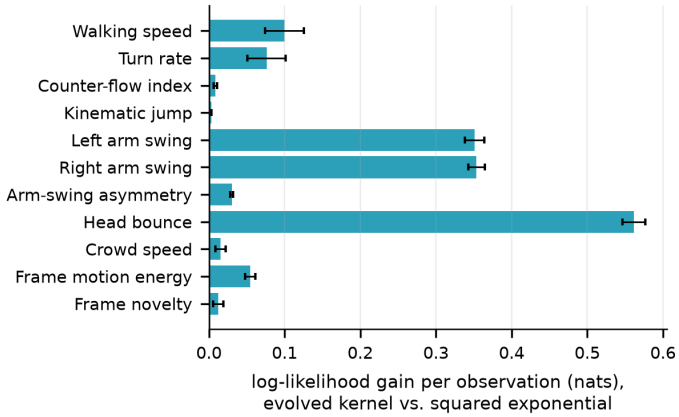


Fig. 8. Model evidence gained by kernel evolution: log-likelihood gain per observation of the evolved kernel over a squared-exponential kernel, per channel (mean ± standard deviation over runs, standard tier).

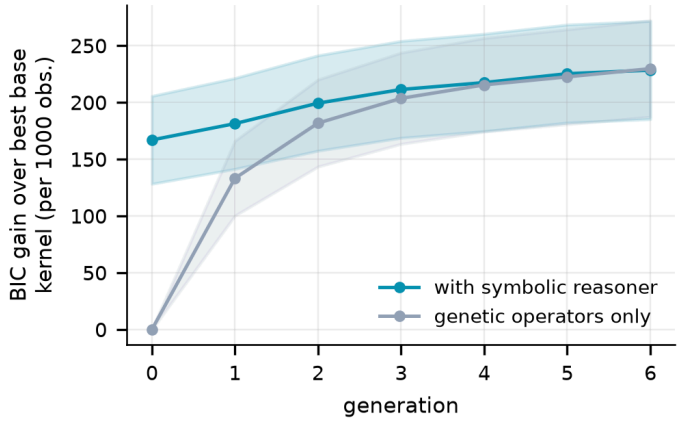


Fig. 9. BIC gain over the best single base kernel as evolution proceeds, with and without the symbolic reasoner (mean ± 2 standard errors over runs and channels).

J. Explanations

Explanations follow the tests that fired, for example a person-level alert, *"Out-of-place behaviour (100%): Head bounce: its level is shifted down (<-100 sigma); Turn rate: its level is shifted up (+41.0 sigma)"*, and a feed-level alert, *"Reality glitch (83%): Frame novelty: an abrupt, unpredictable step (>100 sigma); Frame motion energy: an abrupt, unpredictable step (>100 sigma)"* (the most anomalous windows of two standard-tier runs). On real footage the explanations were equally specific: the frozen shop feed was reported as "Frame motion energy: an abrupt, unpredictable step (46.4 sigma)", and the strongest one-sided arm evidence in the shop (arm-swing asymmetry shifted up by about ten standard deviations) belonged to a shopper reaching for a display with one arm while carrying a bag with the other, a benign posture that the concealed-carry proxy cannot tell apart from carrying an object and a reminder that its alerts need human review.

K. Real-Time Performance

TABLE XII. COST OF EACH STAGE (INTEL(R) CORE(TM) ULTRA 9 275HX, ONE CORE; PERCEPTION ON AN NVIDIA GEFORCE RTX 5080 LAPTOP GPU)

Stage	Measured	Complexity
Score one window (Python, single call)	598 µs (p99 1066)	$\mathcal{O}(CW^2)$
Score windows in batches (Python)	73,909 windows/s	$\mathcal{O}(CW^2)$ each
Score a 50-person frame (Python)	1.26 ms (p95 1.91)	$\mathcal{O}(NCW^2)$
Score one window (browser engine)	29 µs (p99 79)	$\mathcal{O}(CW^2)$
Stream features, 47 people (browser)	0.33 ms per frame	$\mathcal{O}(N \log N + NL)$
Track a 50-person frame	0.90 ms	$\mathcal{O}(N^3)$ worst case
Detect people and pose (GPU)	12.4 frames/s (11 people per frame)	per person
Calibrate a camera (two 2-min clips, kernel evolution)	41 s	$\mathcal{O}(CG\lambda IRW^3)$
Recalibrate for a new camera (four clips, kernels kept)[1]	14 s	$\mathcal{O}(MCW^2)$
Model size per camera	299 kB	$\mathcal{O}(CW^2)$

Measured on an otherwise idle machine by experiments/timing.py, except [1], measured during the CACK-Sim benchmark with ten runs in parallel. $C = 11$ channels, $W = 40$ samples.

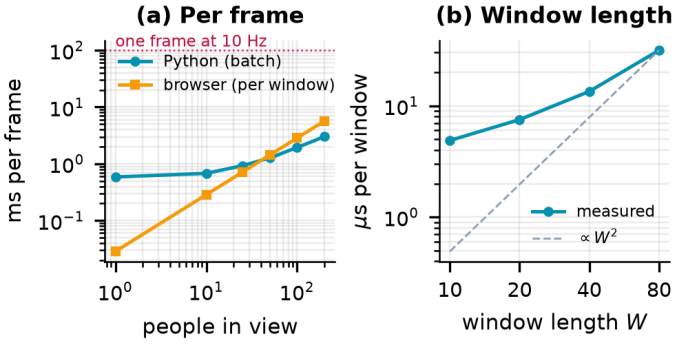


Fig. 10. (a) Time to score everyone in view once (one window per person) with the Python engine (batched) and the browser engine (window by window). (b) Scoring time per window against the window length, with a W^2 reference.

Table XII lists the cost of every stage. Scoring every person of a 50-person camera once per 100 ms frame takes 1.26 ms on one core, 1.3% of the frame budget, and 3.01 ms for 200 people (Fig. 10a); in batches one core scores 73,909 windows per second, enough for about 148 fifty-person cameras at 10 Hz. The cost per window grows as W^2 for long windows (8, 13, and 31 µs per window for $W = 20$, 40, and 80; Fig. 10b) and is dominated by a fixed overhead per window below about $W = 20$. The browser engine scores a window in 29 µs (median; 32,419 windows per second), so the console runs four cameras in one browser tab. In a real deployment the perception front end, not CACK, sets the cost: the GPU detector used here processes about ten frames per second, and a real-time system would use a faster detector and tracker in front of the same CACK engine.

L. Camera Environments and Live Deployment

Other sites. Moved to the other CACK-Sim site of the same tier without recalibration, CACK keeps an AUROC of 0.9592 (against 0.9990 on its own site), but a threshold set at a 1% budget on its own site raises false alarms on 63.4% of normal windows there: the scale of every test belongs to a scene. Recalibrating the flow field, the warps, and the test calibration on the new site's normal clips, with the discovered kernels kept, restores an AUROC of 0.9990 with 2.17% realized false alarms, in 14 s instead of 90 s for a full calibration with kernel evolution.

Running CACK live exposed the same problem. Table XIII measures it with the browser engine on six unseen normal crowds of the plaza layout. With a single threshold over all evidence, false alarms at a nominal 1% budget were 16.1% after calibrating on one two-minute clip, 7.4% after two, and 4.2% after four: a camera has a single crowd-level series, so its calibration sees few independent realizations of crowd-level variation. The two-tier alarm of Section VII, which scores people on their own channels and admits feed evidence only when it is overwhelming, reduces this to 1.7%, 2.1%, and 1.0%. With three calibration clips, as in the console, the realized false-alarm rates are close to the nominal budgets.

TABLE XIII. Console Deployment Study: False Alarms on Unseen Crowds

Layout	1% budget	0.5% budget	0.1% budget	False breaches
Plaza	1.02%	0.54%	0.12%	0.00%
Concourse	0.96%	0.59%	0.05%	0.00%

Browser engine, six unseen normal crowds per layout, three calibration clips. Person-level false alarms per window at each budget, and the share of frames with a false feed-wide integrity breach.

In a five-minute run of the full console logic on four feeds (about 194 people in view) with an event injected every 7 to 14 seconds, 26 of 28 injected events raised an alert within 20 s, all 8 feed-wide breaches corresponded to injected feed glitches (0 false), and alerts not attributable to an injected event occurred 1.8 times per minute across all four feeds; some of these are people reacting to injected events, such as walkers swerving around a collapse.

M. Gait Diversity and Fairness

In the benchmark every pedestrian walks with a typical gait. Table XIV and Fig. 11 audit CACK on crowds in which 20% of people walk with one of the six natural gaits of Section V-A (3,392 such people over 20 runs, with the protocol of Section V-A otherwise unchanged). The alarm threshold is fixed at 1% false alarms on the normal windows of people with a typical gait, so any excess on other people is a disparity. Calibrated on typical crowds, CACK flags people for how they walk: on the standard tier, 100.0% of the windows of people with a post-stroke gait raise an alarm, 99.7% with a Parkinsonian gait, 100.0% with a cane, and 96.8% with an occupied arm, and the class head names at least 99% of these alarms concealed carry. At least 99% of them come from the arm-swing asymmetry, the channel that finds concealed carriers. This is what the proxy looks for: an arm that is held or barely swings breaks the anti-phase cancellation of the two arms, whatever the reason. Gaits that leave both arms swinging are flagged less, but still more than a typical gait: 41.8% of windows for a limp and 5.7% for a slow gait. Both are flagged mostly through head bounce (87% and 62% of their alarms), whose step rhythm a limp makes uneven and a slow gait weaker.

Two remedies act on different parts of the problem. *Representative calibration*, on footage in which 20% of people walk differently, teaches the population model that such gaits are normal and lowers the worst group from 100.0% to 2.2% of windows. *Person-referenced gait channels* (Section IV-I) compare each person with their own earlier walk. On their own they cannot help while a person is new to the view, when the gait is still tested against a population model that has never seen it, and the worst group stays at 44.6%. On top of representative calibration they change the false alarms little (worst group 2.3%) but recover detection: concealed carriers are found in 0.343 of their windows instead of 0.177, as their security touches stand out against their own restricted swing, and the detection rate at the operating point rises from 0.928 to 0.939. This gait-inclusive configuration flags at most 24% of the people in any group at least once while they are in view, against 9% of typical walkers; slower walkers stay in view about 1.3–1.5 times as long and so have more chances to be flagged. Like those of typical walkers, the false alarms that remain come mostly from path and feed channels rather than gait: 69% for post-stroke walkers, 53% from the counter-flow index and turn rate alone, as slower walkers are overtaken and jostled more. It still detects 94% of behavioral events and 100% of glitches.

The price is paid by the concealed-carry proxy. Window-level detection of carriers falls from 1.000 to 0.343, although 96% of carriers are still flagged at least once, mostly through their security touches, which are a change in how a person moves rather than a way of walking. This is not a shortcoming of the remedy but a property of the proxy: in a representative crowd a restricted arm swing is ordinary, and arm swing alone cannot separate a carrier from someone carrying a bag. The benefit also depends on how well calibration footage reflects the people a camera sees: when only 5% of people in it walk differently, the worst group rises to 3.8%, or 3.9% with person-referenced gait channels. Fair treatment of people with

disabilities requires such evaluation on representative data [50]; the audit is a single command in the released engine (`python -m cack audit`).

A gait-inclusive detector must still notice a *change* in how someone walks. Every run also scores two held-out clips of the same crowds in which typical walkers suddenly start to limp or stagger for 6–14 s (the held-out behaviors of Section VI-G, never seen in calibration or training), at the same threshold. CACK as published flags 90% of the windows of a sudden limp and 100% of the episodes; the gait-inclusive configuration flags 31% and 68%. Part of the sensitivity to a sudden change of gait is lost, through representative calibration alone (4% of windows), which teaches the population model that uneven steps and a head nod are normal, whereas person-referenced gait channels on typical calibration footage keep it (86% of windows, 100% of episodes). A deployment that must notice such a change, for example after a fall or an injury, needs an explicit change test on each person's gait, which the person-referenced mode provides only once a person has been in view for about ten seconds. The held-out stagger stays largely undetected in both configurations (2% and 1% of windows).

TABLE XIV. FALSE ALARMS BY NATURAL GAIT AND DETECTION OF INJECTED EVENTS

Method	False alarms on normal windows, by natural gait							Events detected		
	Typical	Limp	Post-stroke	Parkinsonian	Cane	Occupied arm	Slow	Behavior	Concealed	Glitch
Standard tier (10 runs)										
As published	1.0%	41.8%	100.0%	99.7%	100.0%	96.8%	5.7%	94%	**100%**	**100%**
Person-referenced gait	1.0%	27.3%	44.6%	29.7%	32.3%	38.4%	3.5%	**95%**	**100%**	**100%**
Representative calibration	1.0%	2.0%	**2.2%**	1.2%	**1.3%**	1.7%	2.0%	94%	80%	**100%**
Both (gait-inclusive)	1.0%	**1.9%**	2.3%	**1.1%**	1.4%	**1.1%**	**1.9%**	94%	96%	**100%**
Under-represented calibration (5%)	1.0%	3.6%	2.9%	1.2%	3.8%	3.1%	2.1%	93%	76%	**100%**
Under-represented (5%) + person-ref.	1.0%	3.5%	3.9%	**1.1%**	2.6%	1.8%	2.1%	94%	96%	**100%**
Subtle tier (10 runs)										
As published	1.0%	14.9%	98.5%	89.4%	99.1%	94.8%	12.5%	66%	**98%**	**100%**
Person-referenced gait	1.0%	5.7%	36.2%	27.3%	32.8%	34.2%	6.0%	**70%**	97%	**100%**
Representative calibration	1.0%	1.3%	3.1%	1.3%	2.4%	1.2%	2.0%	63%	51%	**100%**
Both (gait-inclusive)	1.0%	**1.0%**	**2.9%**	**1.2%**	**2.3%**	**0.7%**	**1.8%**	66%	81%	**100%**
Under-represented calibration (5%)	1.0%	2.0%	7.3%	1.7%	8.9%	5.5%	2.1%	60%	45%	**100%**
Under-represented (5%) + person-ref.	1.0%	1.8%	5.8%	1.4%	5.1%	2.1%	2.2%	64%	78%	**100%**

Threshold at 1% false alarms on normal windows of people with a typical gait. False alarms: share of normal windows that raise an alarm (mean over runs). Events: share of injected events with at least one alarm. Representative footage contains 20% of people with an atypical gait, under-represented footage 5%; bold marks the best value per column and tier.

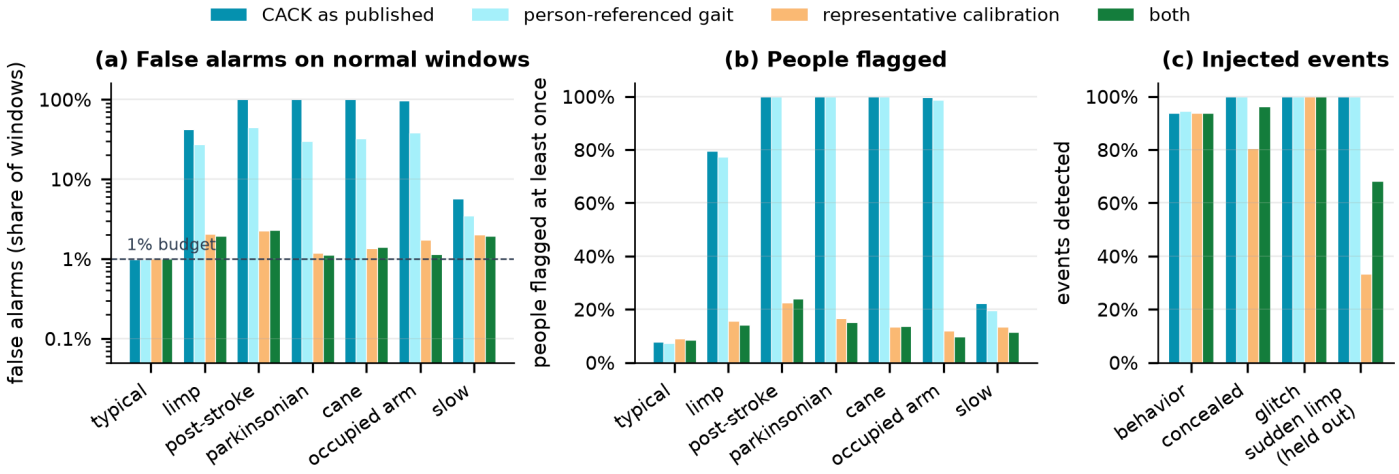


Fig. 11. Gait-diversity audit on the standard tier, at 1% false alarms for people with a typical gait: (a) false alarms on normal windows by natural gait (log scale; bars at the floor are below 0.05%), (b) people flagged at least once while in view, and (c) injected events detected, with the episodes of the held-out sudden limp in separate clips of the same crowds.

N. Calibration Without Curated Footage

Section VI-L moved CACK to the other CACK-Sim site with four curated normal clips of it. A newly installed camera has none, only its live stream. Table XV moves CACK with that stream instead, under the transfer protocol of Section VI-L (12 runs): four unlabeled two-minute clips of the new site with sparse anomalies (7% of windows anomalous) or with the dense anomalies of the test clips (24%), or a single minute with the four events of a sparse clip (15%). Every alarm threshold is set at its budget on the footage that a method calibrated on, as a deployment would set it. To keep the study's cost down, every run uses a fixed squared-exponential kernel, which detects on par with evolved kernels (Section VI-E); the method itself keeps whatever kernels were discovered (Section IV-K). Recalibrating on the stream as if it were

normal, the obvious shortcut, fails: on the standard tier it detects 0.523 of anomalous windows when anomalies are sparse and 0.144 when they are dense, against 0.996 after recalibration on curated footage, because the anomalies have entered the tails of the very tests that should flag them. Self-calibration uses no label and matches the detection of curated footage when anomalies are sparse: an AUROC of 0.9987 against 0.9990, and 0.996 of anomalous windows and 99% of events detected against 0.996 and 99% (subtle tier: AUROC 0.9921 against 0.9920). It does so at more realized false alarms, 4.0% against 2.5% at a nominal 1% (subtle tier: 5.7% against 5.2%), because the windows it sets aside (8.0% of the stream) include some of the unusual normal windows that shape the tails.

When 24% of the windows are anomalous, a single pass of the first step leaves many anomalies below its threshold, which is why it runs twice (Section IV-K): self-calibration then sets aside 22.4% of the windows and detects 0.912 of anomalous windows and 98% of events at 3.2% false alarms, below the oracle that removes anomalies by their labels (0.993 and 3.2%), where naive recalibration detects 0.144. Neither makes a single minute of footage enough: recalibrated as if normal and self-calibrated, CACK realized 17.3% and 62.1% false alarms, the shortfall seen on the real cameras (Section VI-D); setting windows aside leaves even less of a minute to calibrate on. Self-calibration removes the need for curated footage, not for enough of it.

TABLE XV. Moving CACK to the Other Site With Curated Footage or With Its Unlabeled Live Stream

Calibration footage	Standard tier				Subtle tier			
	AUROC	FPR	TPR	Events	AUROC	FPR	TPR	Events
No footage of the new site								
None (thresholds of the old site)	0.9579	63.59%	0.987	98.6%	0.9229	61.08%	0.972	94.6%
Curated normal footage								
Recalibrated	0.9990	2.51%	0.996	99.3%	0.9920	5.19%	0.969	93.0%
Live stream, sparse anomalies (7% of windows)								
Recalibrated as if normal	0.9806	0.43%	0.523	86.0%	0.9519	0.35%	0.427	62.6%
Self-calibrated	0.9987	3.96%	0.996	99.3%	0.9921	5.74%	0.968	94.1%
Oracle: anomalies removed by labels	0.9987	2.96%	0.995	99.3%	0.9935	2.09%	0.964	91.8%
Live stream, dense anomalies (24% of windows)								
Recalibrated as if normal	0.9312	0.15%	0.144	45.6%	0.8585	0.11%	0.119	44.3%
Self-calibrated	0.9885	3.22%	0.912	97.9%	0.9845	2.58%	0.909	89.9%
Oracle: anomalies removed by labels	0.9985	3.19%	0.993	99.3%	0.9933	3.06%	0.970	93.6%
One minute of live stream with the same four events (15% of windows)								
Recalibrated as if normal	0.9259	17.30%	0.780	88.8%	0.8980	7.27%	0.662	69.5%
Self-calibrated	0.9734	62.12%	0.999	100.0%	0.9223	54.86%	0.988	98.8%

Mean over runs. AUROC on the new site's test clips; FPR: realized false alarms on their normal windows at a nominal 1% budget; TPR: anomalous windows detected at that threshold; events: injected events with at least one alarm. Every threshold is set on the footage the method calibrated on. The oracle removes anomalous windows by their labels and bounds what any unsupervised trimming can reach.

O. Feed Integrity Under Realistic Tracking and Informed Tampering

CACK-Sim keeps every person's identity across a feed glitch, as a perfect tracker would; a real tracker does not (Section IV-L). Table XVI therefore tampers normal CACK-Sim crowds with the attacks of the threat model and scores every tampered clip twice: with the perfect tracker's identities, and with those of a real tracker, which gives a new identity to everyone whose position leaves its constant-velocity prediction by more than 0.5 m (20 runs of the benchmark's tiers and sites, three normal test clips each, 840 attacks per tracker). Four detectors are calibrated on the same four normal clips per run: the scene channels in person windows, as published; the same channels in frame windows; the identity-free channels in frame windows; and the frame-level maximum of the last two. An attack counts as detected when a frame of its span, plus one window, breaches the integrity gate (a surprise of 12); the AUROC compares the peak of a tampered span with the peaks of clean spans of the same length, which needs no threshold. Behind the perfect tracker, the published detector breaches the gate for 100% of the naive glitches (freeze, loop, and time-warp, as in Section VI-B), 100% of holds, and 100% of slow motion, but for 49% of replays and no splices: their footage is older than the eight seconds of lags of frame novelty, and only the brief seams at their start and end remain. The identity-free channels breach it for all attacks. Behind a real tracker, the published detector also loses the loop (100% of loops breach the gate behind the perfect tracker, 0% behind a real one): at the splice every track restarts, and while the new tracks are younger than one window no person window exists in which the feed could be tested, so the feed is scored in only 1% of a loop's frames. Testing the scene channels in frame windows does not help (0% of loops, scored in 3% of the frames), because frame motion and frame novelty are themselves undefined until the new tracks have a past. The identity-free channels, which no tracker can change, breach the gate for all attacks (AUROC 1.000, against 0.592 for the published detector). No detector breached the gate on the clean test clips. At the 0.1% budget, both overshoot (the published detector breached 1.87% of the clean frames and the identity-free channels 1.38%): a feed is a single series whose calibration sees few independent realizations of crowd-level variation (Section VI-L), which is why feed breaches answer to the gate.

TABLE XVI. FEED TAMPERING OF CACK-SIM CROWDS BEHIND A PERFECT AND A REAL TRACKER

Detector	Attacks breaching the integrity gate							AUROC	False breaches	
	Freeze	Loop	Time-warp	Replay	Hold	Slow motion	Splice		Gate	0.1%
Perfect tracker (identities kept across a splice, as in CACK-Sim)										
Scene channels, person windows (published)	**100%**	**100%**	**100%**	49%	**100%**	**100%**	0%	0.794	0.00%	1.87%
Scene channels, frame windows	**100%**	**100%**	**100%**	49%	**100%**	**100%**	0%	0.798	0.00%	2.34%
Identity-free channels	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	1.000	0.00%	1.38%
Both: frame-level maximum	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	1.000	0.00%	2.61%
Real tracker (a new identity wherever a position jumps)										
Scene channels, person windows (published)	**100%**	0%	**100%**	0%	**100%**	**100%**	0%	0.592	0.00%	1.87%
Scene channels, frame windows	**100%**	0%	**100%**	0%	**100%**	**100%**	0%	0.601	0.00%	2.34%
Identity-free channels	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	1.000	0.00%	1.38%
Both: frame-level maximum	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	1.000	0.00%	2.61%

Share of attacks breaching the integrity gate (surprise ≥ 12) within their span plus one window, over both tiers; AUROC: peak of a tampered span against peaks of clean spans of the same length. False breaches: share of the clean test clips' frames breaching the gate, and the 0.1% budget set on the calibration clips. The identity-free channels are identical behind both trackers, so their rows are too.

Varying the strength of the informed attacks on the standard tier, behind a real tracker: the published detector breached the gate for no replays of footage 5 to 90 s old, and frame echo for all of those within its one-minute memory and, beyond it, where only the two seams betray a replay, for all. The identity-free channels breached it for all holds with jitter from 1 to 30 cm, all time-warps from 1.1 to 3×, and all slow motion from 0.25 to 0.9×: CACK-Sim's crowds walk far more regularly than real ones, and a jittered hold cannot imitate walking, whose displacement over two frames is twice that over one.

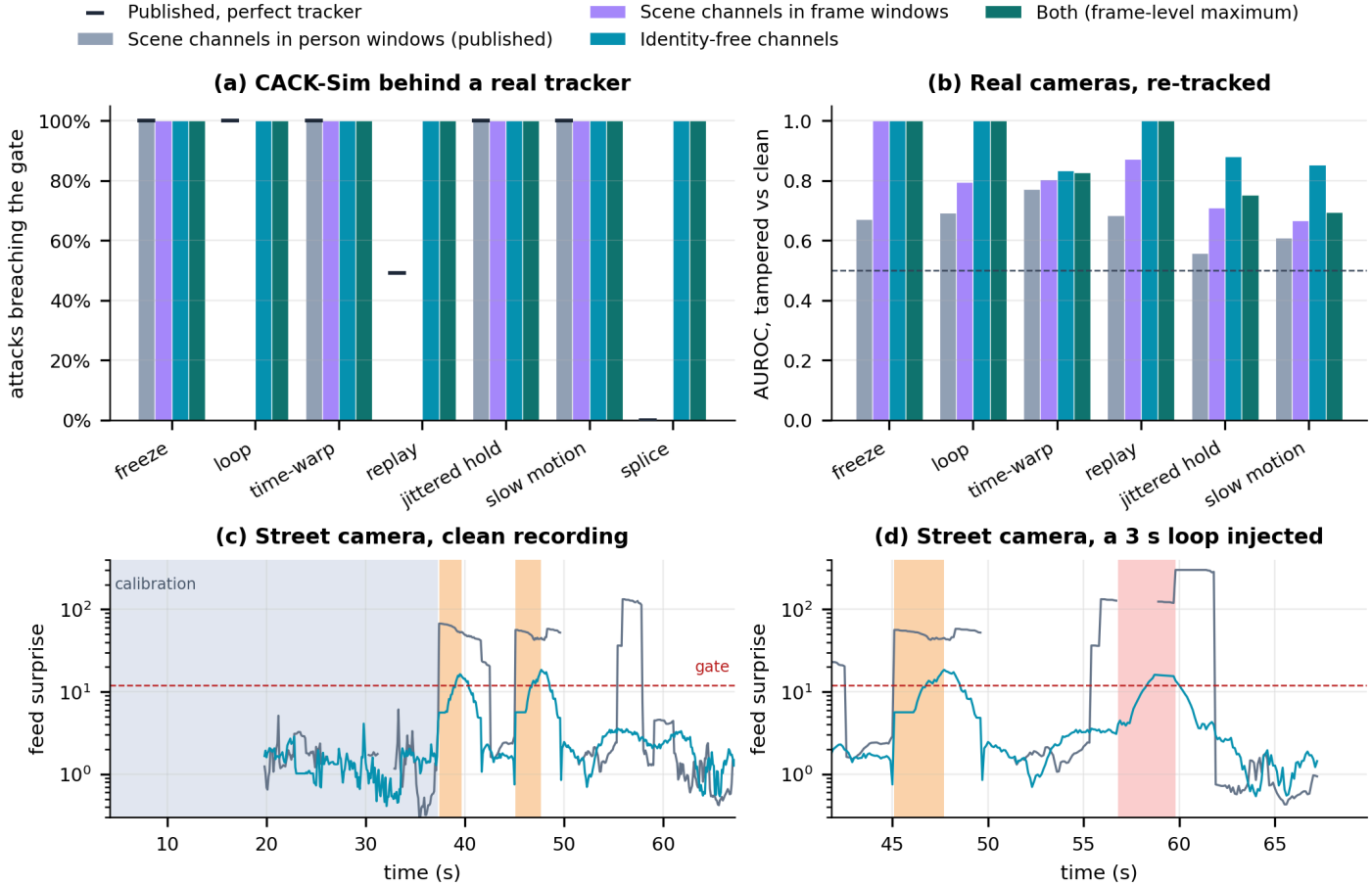


Fig. 12. Feed integrity. (a) Attacks breaching the integrity gate in CACK-Sim with a real tracker's identities (both tiers); dashes: the published detector behind the perfect tracker. (b) The two real cameras, tampered before tracking and re-tracked: AUROC of tampered against clean spans. (c) Feed surprise of the street camera's recording for the published detector (gray) and the identity-free channels (blue), with the calibration footage (gray band), the two natural stutters (orange) and the integrity gate (dashed); (d) the same camera with a loop injected (red band) and re-tracked. The published detector's breaches from 55.4 s on precede the loop: it breaches the gate there in the clean recording too.

Real footage. The two long static cameras of Section VI-D (shop and street) were rebuilt from their console exports, which hold every tracked person's box and keypoints at 10 Hz, tampered before tracking, and re-tracked by the front end's tracker, with the protocol of Section VI-D. With the paper's calibration, the published detector on the rebuilt detections agrees with the published results on 92% of the 24 trials at the 0.1% budget and 79% under the gate (correlation of the log peak surprises 0.88); the rebuilt detections lack the detector's sub-pixel precision and the tracks it dropped. The rebuilt detections also expose two natural glitches that Section VI-D did not identify: the street camera's recording repeats whole frames, every box and keypoint of 12 to 19 people, from 37.4 to 39.7 s and 45.1 to 47.7 s. The first lies in the footage on which Section VI-D calibrated the street camera, and with that calibration the published detector raised no breach at all during the second. Calibrated on the footage before them, every detector breaches the gate within 1.7 s of the onset of both stutters and of the shop's frozen ending.

Over the 139 attacks on both cameras, the identity-free channels breach the gate for all freezes, all loops, and all replays, and never on either camera's clean footage; they separate tampered from clean spans with an AUROC of 0.93, against 0.67 for the published detector. The published detector breaches the gate for 77% of the attacks, but also on 5.3% and 15.6% of the shop's and the street's clean frames: its marginal warps, learned from a few hundred values of a single feed series, extrapolate small excursions into breaches, and with supported warps alone its clean breaches vanish on both cameras. Fast-forward, holds, and slow motion of the real crowds mostly stay below the gate for the identity-free channels (12%, 0%, and 0% breach it), although they leave evidence (AUROC 0.83, 0.88, and 0.85): real crowds vary their pace far more than simulated ones, and a two-fold fast-forward moves the self-normalized pace channel by a median of 0.6 of its own standard deviations on the real cameras, against 21 in CACK-Sim.

TABLE XVII. Feed Tampering of the Two Real Cameras, Re-Tracked (139 Attacks)

Detector	Breaching the gate / AUROC						False breaches		Natural glitches
	Freeze	Loop	Time-warp	Replay	Hold	Slow motion	Gate	0.1%	
Scene channels, person windows (published)	100% / 0.67	78% / 0.69	76% / 0.77	80% / 0.68	50% / 0.56	70% / 0.61	10.45%	21.69%	3/3
Scene channels, frame windows	100% / 1.00	59% / 0.79	36% / 0.80	25% / 0.87	5% / 0.71	5% / 0.67	0.00%	13.55%	3/3
Identity-free channels	100% / 1.00	100% / 1.00	12% / 0.83	100% / 1.00	0% / 0.88	0% / 0.85	0.00%	1.95%	3/3
Both: frame-level maximum	100% / 1.00	100% / 1.00	36% / 0.83	100% / 1.00	5% / 0.75	5% / 0.69	0.00%	12.61%	3/3

Per attack: share breaching the integrity gate, and the AUROC of tampered against clean spans. False breaches: share of clean test frames breaching the gate, and the 0.1% budget, averaged over the two cameras. Natural glitches: the three spans of repeated frames in the recordings after calibration (the shop's frozen ending, two stutters of the street camera) whose span breaches the gate. Attacks within a window of a natural glitch are left out.

VII. The CACK Console

The console at https://4waiz.github.io/cack/ runs CACK in the browser (Fig. 13). A JavaScript port of the simulator, the feature streams, the GP engine (Cholesky factorization and Jacobi eigendecomposition), the four tests, the tail calibration, and the classifier agrees with the Python engine to within 4.6×10^{-12} on kernel matrices, gradients, and test statistics. With simulated cameras, it records three normal clips per camera at boot and calibrates warps, test scales, and tails on them, then streams four CCTV-style feeds whose pedestrians are drawn from the very signals CACK analyzes, so a pinned arm is visibly stiff and a frozen feed is visibly frozen. Alerts carry their explanation and whether an event was actually injected; a track inspector plots every channel against its GP one-step forecast band; and a Kernel Lab evolves the kernel of a channel in a Web Worker and hot-swaps it into the running detector.

Real footage. The console also replays the four real cameras of Section V-C. Every 10 Hz sample of a camera's tracked people (positions, arm swings, and head bounce, and, for drawing, their boxes and skeletons) is streamed through the same feature streams and detector, with a model calibrated on that camera's normal footage (2 s windows; the eight person channels for the two 14 s clips, all eleven for the others), so the track inspector, the explanations, and the Kernel Lab work on real people. Streamed through the browser engine, the real cameras reproduce the Python engine on every window: the same 6,482 windows, with surprises within 2.4×10^{-12} and identical alarms. Feeds can be frozen, replayed, or time-warped; the feed channels are tested on every frame, whether or not anyone in view is fully tracked, and the retail camera's recorded frozen ending raises a feed-integrity breach when the replay reaches it. On real people an alarm is named by the channel that raised it (unusual movement, a track discontinuity, or the feed), never by the concealed-carry proxy (Section VIII). Where the recording itself may be shown, the boxes and skeletons are drawn over it; otherwise the tracked skeletons are drawn alone.

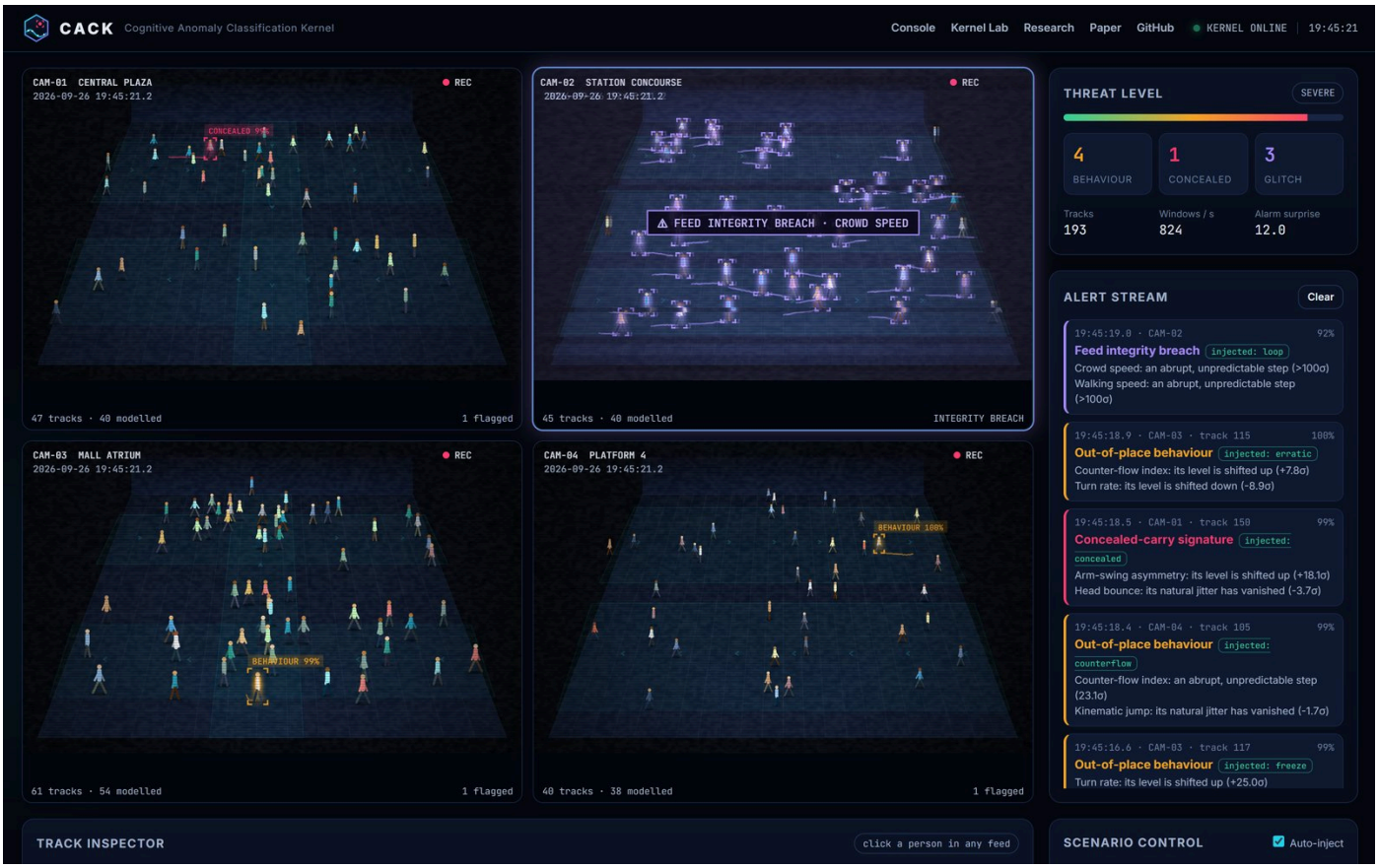


Fig. 13. The CACK console: four live feeds with class-coded detections, a feed-wide integrity breach, the threat panel, and the stream of explained alerts.

Two-tier alarm. Because a feed channel is shared by everyone in view, a natural crowd fluctuation that crosses a person-level threshold would flag the whole feed at once. The console therefore flags people on their own channels at the calibrated budget and raises a feed-wide integrity breach only on overwhelming feed evidence (a surprise of at least 12), which real glitches exceed by a wide margin (Section VI-D and VI-L).

VIII. Limitations and Responsible Use

Accuracy on public benchmarks. On ShanghaiTech and Avenue, CACK is on par with classical detectors under the same front end and with early pose-based networks, but well below the best pose-based [12] and pixel-based [8] methods. On UCSD Ped2, whose pedestrians are 21–40 pixels tall and whose anomalies are mostly vehicles, carts, cyclists, and skateboarders, tracking jitter gives normal walkers apparent speeds of several meters per second and no method reaches useful accuracy with a person-centric front end. CACK's contribution is interpretability, calibration, feed integrity, and speed, not state-of-the-art frame-level accuracy.

Calibration transfer. A threshold calibrated on about a minute of one camera does not hold on the next minute of the same camera, and still less on another camera (Section VI-D). Deployments need longer and more varied calibration footage per camera, recalibration when a camera moves, and the separate feed gate. Self-calibration on the live stream (Section VI-N) removes the need for curated footage but not for enough of it: on a single minute it realized 62.1% false alarms at a nominal 1%.

Real footage. The four real clips are short, carry no anomaly labels, and are publicly available recordings that were not made for this study. The analysis uses only the tracks and skeletons derived from them, and the figures show no pixels of the footage; the console replays the recordings under their tracks. Their results are a deployment study, not an accuracy benchmark.

Feed integrity. The identity-free channels remove the dependence on the tracker, not the limits of kinematic evidence: fast-forward, holds, and slow motion of slow, irregular real crowds mostly stay below the integrity gate; a replay older than the one-minute echo memory is betrayed only by its seams; an attacker who can synthesize a convincing crowd is beyond the threat model; and the real-footage evaluation rests on two cameras rebuilt from their console exports (Section VI-O).

Concealed carry. The concealed-carry class is a stylized kinematic proxy. Restricted arm swing and gait asymmetry are also produced by injury, disability, carrying a bag or a phone, and culture; on real footage the strongest one-sided arm evidence came from a shopper reaching for a display with one arm while carrying a bag with the other (Section VI-J). Section VI-M quantifies this: calibrated on typical crowds, CACK raises an alarm on at least 99.6% of the simulated people with a post-stroke or Parkinsonian gait, a cane, or an

occupied arm while they are in view, and the gait-inclusive configuration that removes this disparity also removes most of the proxy's window-level detections (1.000 to 0.343), because in a representative crowd a restricted arm swing is ordinary. The console therefore never names a real person a concealed carrier. Such cues are probabilistic and must never be the sole basis for action against a person.

Bias and oversight. Vision systems can perform unequally across groups [51]. Any deployment would need a legal basis, consent where required, independent evaluation on representative data, human review of every alert, and monitoring for disparate impact. The explanations of CACK are designed to support, not replace, that review.

Scope. CACK models one track at a time plus feed-level summaries; it does not reason about groups, objects, or long-range intent, its windows trade latency against sensitivity to slow behaviors, and it misses smooth, moderate anomalies such as the held-out stagger, which no evaluated method detects well.

IX. Conclusion

This paper presented CACK, a system that pairs a small set of physically meaningful signals with warped Gaussian-process normalcy models to detect, classify, and explain crowd and video-feed anomalies at interactive speed. On simulated footage it detects ten anomaly types and, on the standard tier, three of the four types that no component was designed for; on real footage it recovers the rhythms of human gait without labels, catches frozen, replayed, and time-warped feeds, and scores a window in microseconds on a CPU; on public benchmarks it is on par with classical detectors under the same front end and below the best deep methods, whose accuracy it does not claim to match.

Three lessons generalize beyond CACK. First, discovering kernels that explain normal data far better (up to 0.56 nats per observation, for head bounce) did not by itself buy detection accuracy, because a linear predictor that captures a rhythm also absorbs scaled or shifted versions of it; the tests have to be designed around what a better model makes visible. Second, calibrated thresholds belong to a camera and its footage: without recalibration they did not transfer to another site (63.4% false alarms at a nominal 1%) or even to the second minute of a real clip, whereas recalibrating on normal footage alone, with the discovered kernels kept, restored an AUROC of 0.9990 in 14 s. A new camera does not even need curated footage: self-calibration on its unlabeled live stream, which sets aside what a robustly adapted detector clearly flags, matched the detection of curated recalibration (0.9987 against 0.9990 AUROC with sparse anomalies), where recalibrating on the stream as if it were normal detected 0.523 of anomalous windows instead of 0.996. Third, crowd-level integrity channels are a single series per camera and must be gated separately from person-level evidence: under the two-tier gate, tampering injected into real video raised no false breach on clean footage.

The gait-diversity audit of Section VI-M adds a fourth lesson: a detector calibrated on typical crowds flags people for how they walk, in up to 100.0% of the windows of people whose natural gait restricts an arm. Representative calibration footage together with person-referenced gait channels brought every group to at most 2.3% while keeping behavior and glitch detection, and the audit showed that arm swing alone cannot separate a concealed carrier from someone carrying a bag: any carry signature needs evidence of what a person does, not of how they walk. The same remedy lowered the detection of a sudden limp from 90% to 31% of its windows: fairness to people who always walk differently and sensitivity to someone who starts to must be designed for together.

A feed check is only as independent as its evidence. CACK's scene channels relied on track identities that a real tracker breaks exactly when a feed is spliced, so behind a realistic tracker the published configuration let every loop, replay, and splice pass in CACK-Sim. Identity-free integrity channels, which compare each frame's detections with earlier ones by nearest neighbors and are tested in frame windows, breached it for all the naive and informed attacks in CACK-Sim, separated tampered from clean real footage with an AUROC of 0.93 against 0.67, and found two stutters in a real recording that had passed for normal footage.

The most important next steps are longer and more varied calibration footage per camera and real recordings with anomaly labels, including concealed-carry and tampering footage recorded with consent; a perception front end that is robust to small, fast objects; and interaction channels between people. Further directions include detection-aware kernel search that optimizes the tests themselves, conformal calibration pooled across cameras, and a live evaluation of the language-model reasoner.

Appendix A: Experimental Configuration

Table XVIII lists every configuration used in this paper.

TABLE XVIII. Experimental Configuration

Item	Setting
CACK-Sim windows	W = 40 samples at 10 Hz (4 s), stride 5; 11 channels
Real benchmarks	W = 20 samples (2 s), 8 person channels; test stride 1; offline: center samples, 0.5 s Gaussian smoothing
Marginal warp	121 quantiles over ±3σ, linear tails at the smaller of tail slope and core scale
GP fit	L-BFGS-B, 300 iterations, 2 starts plus a short-lengthscale start; noise variance in $[10^{-3}, 10]$
Kernel evolution	population 8, 6 generations, 4 crossovers, 2 mutations, 2 reasoner proposals per generation; at most 3 terms of 3 factors; BIC fitness
Fitting windows	at most 1500 non-overlapping normal windows per channel
Structured subspace	eigen-directions whose signal variance exceeds 4× the noise and the spectral median; at least 4 residual directions
Tail calibration	empirical body to the 95th percentile, exponential peaks-over-threshold tail (≥10 exceedances)
Class head	softmax regression, L2 = 0.01, class-balanced, L-BFGS-B, 500 iterations
CACK-Sim protocol	4 calibration, 2 labeled, 4 test, 2 held-out-type clips of 120 s per run; 5 seeds, 2 sites, 2 tiers
Front end	Keypoint R-CNN (ResNet-50 FPN, COCO), person score ≥ 0.5, input side 800 px; SORT, IoU ≥ 0.2, 3 missed frames, tracks ≥ 5 samples
Rectification	pedestrian height h(y) = a + by, robust line fit; person height 1.7 m; stabilization above 0.25 px of background motion per sample
Real-footage study	calibration on 60% of clean footage; feed channels need ≥ 3 people; alarms at a 0.1% budget or the two-tier gate (surprise ≥ 12)
Baselines	isolation forest 300 trees; one-class SVM ν = 0.05, RBF; autoencoder 25 epochs, Adam 10^{-3}; GP likelihood SE per channel
Gait-diversity study	20% of pedestrians with an atypical natural gait in test clips; calibration footage with 0%, 20% or 5%; person reference: median of ≥ 5 earlier non-overlapping windows whose path and feed surprise is below 4; threshold at 1% false alarms for people with a typical gait; 2 held-out sudden-gait clips per run
Self-calibration study	the transfer protocol of Section VI-L with a fixed SE kernel; live streams of four unlabeled 120 s clips with one behaviour, one carrier, one teleport and one freeze each, or with the test clips' anomalies, or one 60 s clip; windows set aside at the original 0.1% threshold, two rounds; thresholds on the footage kept
Integrity study	4 calibration and 3 test clips of 120 s per run, 2 attacks of each kind per test clip, 2.5–4 s (freeze) or 3–5 s long; real tracker: a new identity 0.5 m off the constant-velocity prediction; echo memory 60 s, lags beyond 8 s with 16 detections per frame; frame windows: kernels fitted at half-window stride, warp support 10 values, tails declustered by runs of one window; real cameras rebuilt from their console exports with 0.5 px dither, calibrated on the first 60% of their normal footage before any repeated frame; gate 12 or a 0.1% budget

All settings are the defaults of the released code; every experiment is reproduced by the scripts in experiments/ (benchmark.py, real_benchmark.py, real_footage.py, timing.py, gait_fairness.py, self_calibration.py, integrity_study.py) from fixed seeds.

References

[1] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1975–1981.

[2] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 FPS in MATLAB," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2013, pp. 2720–2727.

[3] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 733–742.

[4] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection – A new baseline," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6536–6545.

[5] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6479–6488.

[6] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. van den Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, 2019, pp. 1705–1714.

[7] H. Park, J. Noh, and B. Ham, "Learning memory-guided normality for anomaly detection," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 14360–14369.

[8] Z. Liu, Y. Nie, C. Long, Q. Zhang, and G. Li, "A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, 2021, pp. 13568–13577.

[9] R. Morais, V. Le, T. Tran, B. Saha, M. Mansour, and S. Venkatesh, "Learning regularity in skeleton trajectories for anomaly detection in videos," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11988–11996.

[10] R. Rodrigues, N. Bhargava, R. Velmurugan, and S. Chaudhuri, "Multi-timescale trajectory prediction for abnormal human activity detection," in *Proc. IEEE Winter Conf. Applications of Computer Vision (WACV)*, 2020, pp. 2615–2623.

[11] A. Markovitz, G. Sharir, I. Friedman, L. Zelnik-Manor, and S. Avidan, "Graph embedded pose clustering for anomaly detection," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10536–10544.

[12] O. Hirschorn and S. Avidan, "Normalizing flows for human pose anomaly detection," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, 2023, pp. 13499–13508.

[13] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[14] D. Duvenaud, J. R. Lloyd, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani, "Structure discovery in nonparametric regression through compositional kernel search," in *Proc. 30th Int. Conf. Machine Learning (ICML)*, 2013, pp. 1166–1174.

[15] J. R. Lloyd, D. Duvenaud, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani, "Automatic construction and natural-language description of nonparametric regression models," in *Proc. 28th AAAI Conf. Artificial Intelligence*, 2014, pp. 1242–1250.

[16] G. Kronberger and M. Kommenda, "Evolution of covariance functions for Gaussian process regression using genetic programming," in

Computer Aided Systems Theory – EUROCAST 2013, Lecture Notes in Computer Science, vol. 8111. Berlin, Germany: Springer, 2013, pp. 308–315.

[17] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, "Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2909–2917.

[18] M. Kemmler, E. Rodner, E.-S. Wacker, and J. Denzler, "One-class classification with Gaussian processes," *Pattern Recognition*, vol. 46, no. 12, pp. 3507–3518, 2013.

[19] H. Kim and Y. W. Teh, "Scaling up the automatic statistician: Scalable structure discovery using Gaussian processes," in *Proc. 21st Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2018, pp. 575–584.

[20] W. Luo, W. Liu, and S. Gao, "A revisit of sparse coding based anomaly detection in stacked RNN framework," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 341–349.

[21] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, 1995.

[22] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 935–942.

[23] E. Snelson, C. E. Rasmussen, and Z. Ghahramani, "Warped Gaussian processes," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 16, 2004, pp. 337–344.

[24] A. G. Wilson and R. P. Adams, "Gaussian process kernels for pattern discovery and extrapolation," in *Proc. 30th Int. Conf. Machine Learning (ICML)*, 2013, pp. 1067–1075.

[25] J. Lehman, J. Gordon, S. Jain, K. Ndousse, C. Yeh, and K. O. Stanley, "Evolution through large models," arXiv:2206.08896, 2022.

[26] E. Meyerson, M. J. Nelson, H. Bradley, A. Gaier, A. Moradi, A. K. Hoover, and J. Lehman, "Language model crossover: Variation through few-shot prompting," arXiv:2302.12170, 2023.

[27] B. Romera-Paredes, M. Barekatain, A. Novikov, *et al.*, "Mathematical discoveries from program search with large language models," *Nature*, vol. 625, pp. 468–475, 2024.

[28] R. C. Suwandi, F. Yin, J. Wang, R. Li, T.-H. Chang, and S. Theodoridis, "Adaptive kernel design for Bayesian optimization is a piece of CAKE with LLMs," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2025, arXiv:2509.17998.

[29] J. E. Jackson and G. S. Mudholkar, "Control procedures for residuals associated with principal component analysis," *Technometrics*, vol. 21, no. 3, pp. 341–349, 1979.

[30] J. F. MacGregor and T. Kourti, "Statistical process control of multivariate processes," *Control Engineering Practice*, vol. 3, no. 3, pp. 403–414, 1995.

[31] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouët, "Anomaly detection in streams with extreme value theory," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2017, pp. 1067–1075.

[32] S. Bates, E. Candès, L. Lei, Y. Romano, and M. Sesia, "Testing for outliers with conformal p-values," *The Annals of Statistics*, vol. 51, no. 1, pp. 149–178, 2023.

[33] M. Goswami, C. Challu, L. Callot, L. Minorics, and A. Kan, "Unsupervised model selection for time-series anomaly detection," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2023.

[34] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 413–422.

[35] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[36] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, Art. no. 15, 2009.

[37] D. Damen and D. Hogg, "Detecting carried objects in short video sequences," in *Proc. European Conf. Computer Vision (ECCV)*, 2008, pp. 154–167.

[38] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting duplication," in *Proc. 9th ACM Workshop Multimedia and Security*, 2007, pp. 35–42.

[39] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 2961–2969.

[40] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, 2016, pp. 3464–3468.

[41] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "AlphaPose: Whole-body regional multi-person pose estimation and tracking in real-time," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7157–7173, 2023.

[42] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[43] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image and Video Processing*, vol. 2008, Art. no. 246309, 2008.

[44] C. A. T. Ferro and J. Segers, "Inference for clusters of extreme values," *J. Royal Statistical Society: Series B (Statistical Methodology)*, vol. 65, no. 2, pp. 545–556, 2003.

[45] F. Tramèr, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 1633–1645.

[46] S. Milani, M. Fontani, P. Bestagini, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro, "An overview on video forensics," *APSIPA Trans. Signal and Information Processing*, vol. 1, e2, 2012.

[47] C. A. Okoro, N. D. Hollis, A. C. Cyrus, and S. Griffin-Blake, "Prevalence of disabilities and health care access by disability status and type among adults – United States, 2016," *Morbidity and Mortality Weekly Report*, vol. 67, no. 32, pp. 882–887, 2018.

[48] J. Perry, M. Garrett, J. K. Gronley, and S. J. Mulroy, "Classification of walking handicap in the stroke population," *Stroke*, vol. 26, no. 6, pp. 982–989, 1995.

[49] M. D. Lewek, R. Poole, J. Johnson, O. Halawa, and X. Huang, "Arm swing magnitude and asymmetry during gait in the early stages of Parkinson's disease," *Gait & Posture*, vol. 31, no. 2, pp. 256–260, 2010.

[50] A. Guo, E. Kamar, J. W. Vaughan, H. Wallach, and M. R. Morris, "Toward fairness in AI for people with disabilities: A research roadmap," *ACM SIGACCESS Accessibility and Computing*, no. 125, 2020, arXiv:1907.02227.

[51] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Proc. Conf. Fairness, Accountability and Transparency (FAT*)*, 2018, pp. 77–91.