

ALBIREO: Adaptive, Energy-Efficient Inference Framework for Video Object Detection on the Edge

Amir Taherin¹ José Cano² Bin Ren³ Yanzhi Wang¹ David Kaeli¹

¹Northeastern University ²University of Glasgow ³William & Mary

ACM/IEEE Symposium on Edge Computing (SEC) 2026



arXiv:2609.29648 github.com/amirtaherin/albireo

Introduction: edge video object detection is production technology

- The per-frame detector forward pass is the **dominant cost**:
 - YOLO26x on every frame: **70 W** board power on Thor (49% GPU), **43 W** on Orin (72% GPU)
- On battery- and thermally-constrained devices, this cost decides **which perception workloads are deployable at all**
- **Each application has its own accuracy requirement**: our goal is to preserve whatever accuracy the deployed detector delivers, at lower cost

Can we skip detector invocations without giving up detection quality?



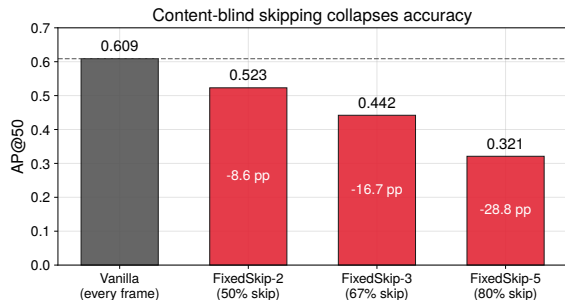
AGX Orin (60 W)

AGX Thor (130 W)

	Orin	Thor
GPU	Ampere	Blackwell
Memory	32 GB	128 GB
Peak BW (GB/s)	204.8	273
TDP	60 W	130 W

Two consecutive NVIDIA Jetson AGX generations

Motivation: video is redundant, but naive skipping fails



- **Temporal redundancy**: consecutive frames are nearly identical, so why run the detector on every one?
- A fixed schedule is **content-blind**: it skips exactly when the scene changes
- Even 50% skip costs **-8.6 pp AP@50**

The question is not *whether* to skip — it is *which* frames to skip, driven by scene content and per-object state.

Skip means **skip the detector**, not the frame: every frame still gets output boxes.

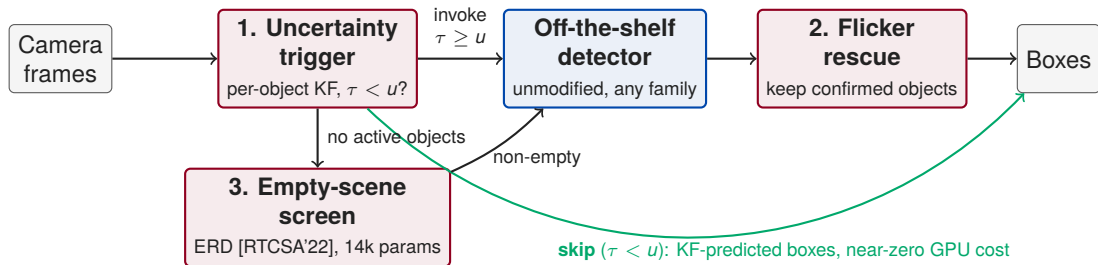
Prior adaptive methods each miss a deployment requirement

Method	DA	NDR	EM	CF	EE	POA
DFF	×	×	✓	✓	×	×
SDOF-Tracker	×	✓	×	✓	×	×
Tran et al.	✓	×	✓	×	×	×
Skip-Conv	×	×	✓	✓	×	×
PASS	×	×	✓	✓	✓	×
Statues	✓	✓	×	✓	✓	×
CTD	✓	✓	✓	✓	×	✓
Detect or Track	×	×	✓	✓	×	×
FrameHopper	✓	×	✓	✓	×	×
Henning et al.	✓	✓	✓	✓	✓	×
Contoli et al.	✓	✓	×	✓	✓	×
Compressed-domain	✓	×	✓	×	×	×
ALBIREO (ours)	✓	✓	✓	✓	✓	✓

DA = detector-agnostic NDR = no detector retraining EM = ego-motion capable CF = codec-free EE =
energy evaluated on edge SoCs POA = per-object adaptive

Feature propagation needs retraining; pixel differencing breaks under ego-motion; CTD's confidence reference goes stale.

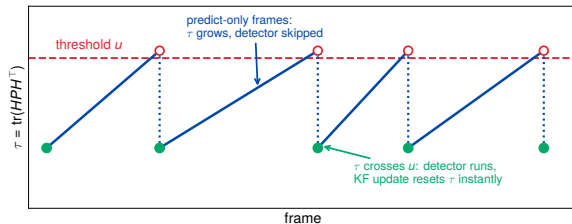
ALBIREO: a drop-in wrapper with three mechanisms



No frame is dropped — a skipped frame's boxes come from the Kalman filter instead of the detector.

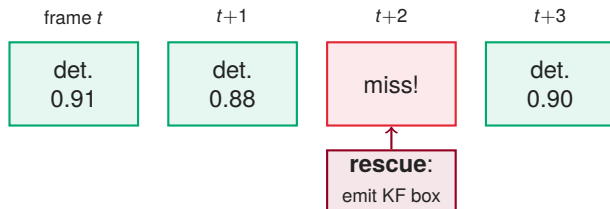
- **Detector-agnostic, no retraining, codec-free:** wraps any detector that emits boxes + confidences
- One primary knob: the uncertainty threshold u selects the accuracy–efficiency operating point

Mechanism 1: forward-looking uncertainty trigger



- Each object: a **10-D Kalman filter** (position, size, velocities, acceleration) in normalized coordinates
- Skip while every object's projected covariance trace $\tau = \text{tr}(HPH^\top)$ stays below u
- **Forward-looking:** τ grows with each prediction
- **Safety caps:**
 - Force detection after 15 predictions
 - Once every 30 frames

Mechanism 2: detector-flicker rescue



- **Detector flicker:** frames are processed independently, so an object's box can drop below the confidence threshold for a single frame
- **Rescue:** a recently confident object keeps its Kalman-predicted box, for at most 2 consecutive frames
- The **largest accuracy contributor:** AP@50 0.572 → **0.617**

Mechanism 3: empty-scene screen, and the five frame states

- **Empty scenes:** with no objects there is no covariance to consult, yet the detector would still run on every frame
- **ERD** [RTCSA'22]: a tiny empty-scene classifier (14k parameters, ~ 1.3 ms) gates the detector; the **detector itself is untouched**
- Every frame lands in exactly one of **five states**

State	Name	Computation
E	ERD-Empty	ERD only
P	KF-Predict	predicted boxes
E+I	ERD-Infer.	ERD, then detector
I	Inference	detector + update
AI	Augmented	detector + rescue

Scene examples

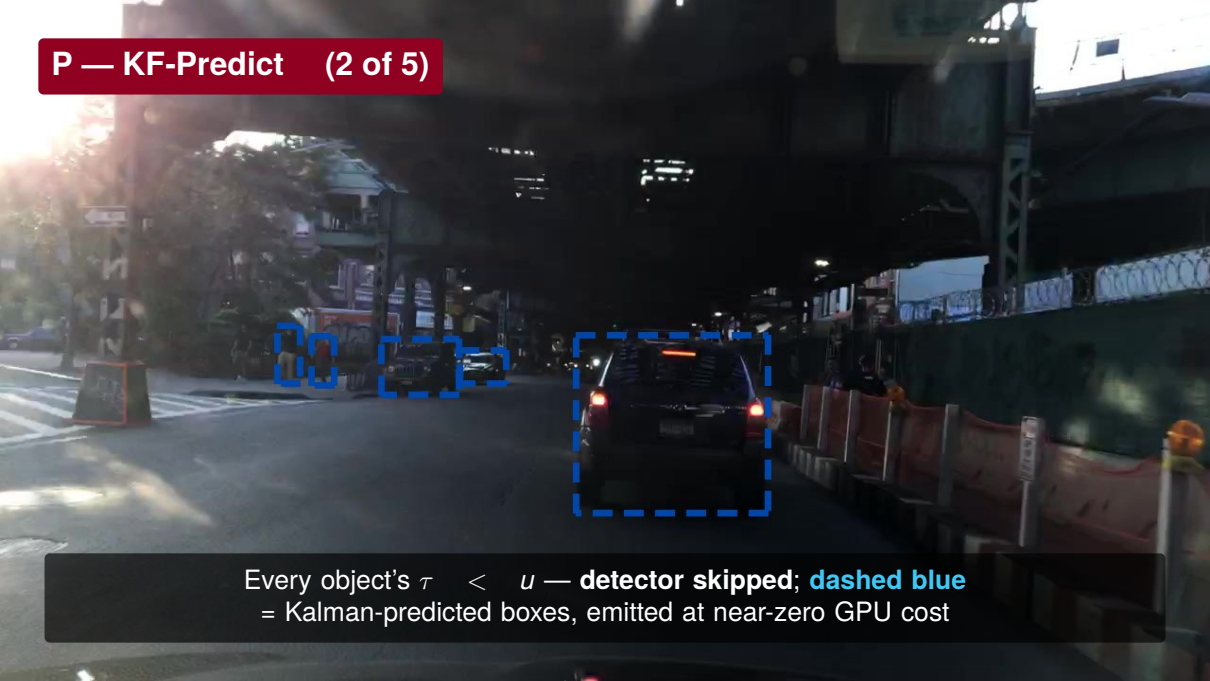
the five frame states on real BDD100K frames
(ALBIREO + YOLO26x on Thor)

I — Inference (1 of 5)



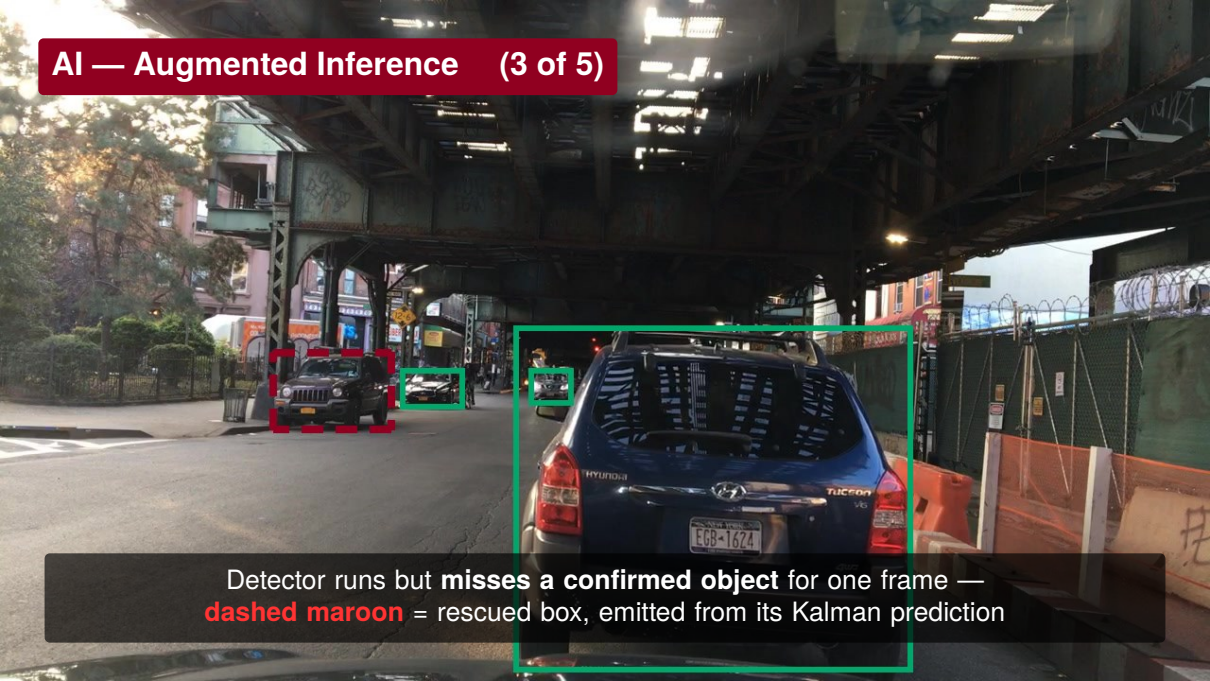
An object's τ crossed u — **detector runs**; **solid green** = detections, Kalman filters updated

P — KF-Predict (2 of 5)



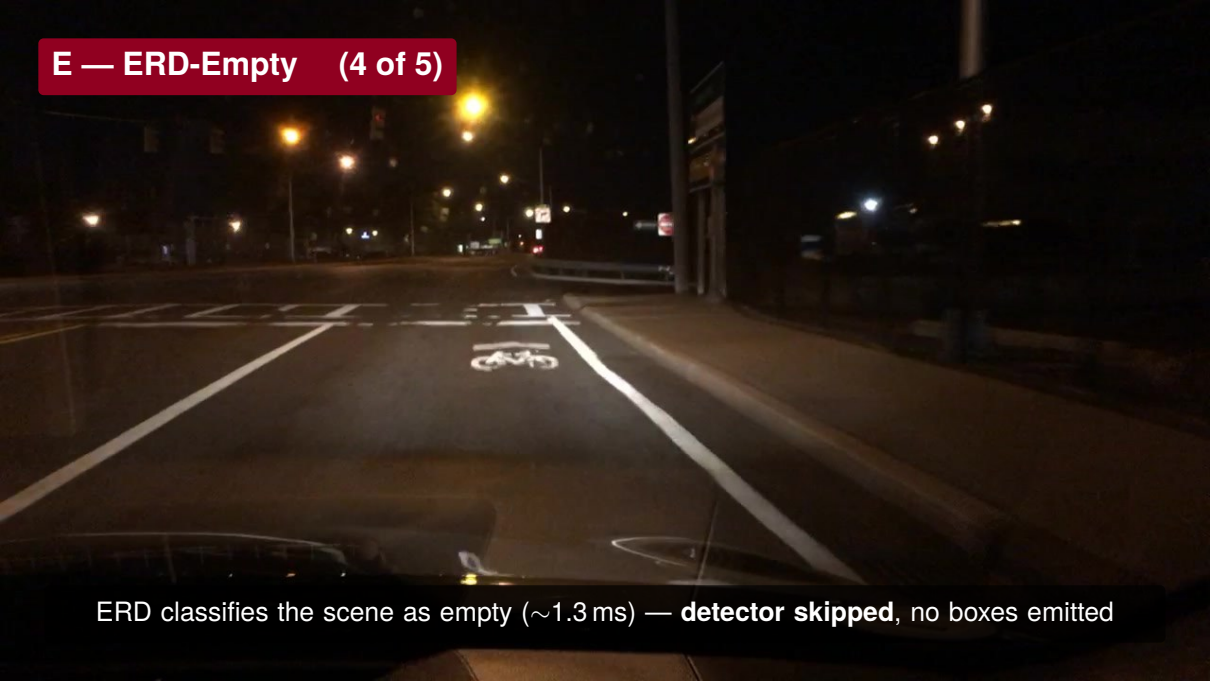
Every object's $\tau < u$ — **detector skipped**; **dashed blue**
= Kalman-predicted boxes, emitted at near-zero GPU cost

AI — Augmented Inference (3 of 5)



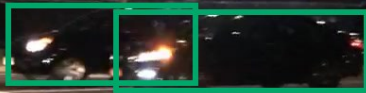
Detector runs but **misses a confirmed object** for one frame —
dashed maroon = rescued box, emitted from its Kalman prediction

E — ERD-Empty (4 of 5)



ERD classifies the scene as empty (~ 1.3 ms) — **detector skipped**, no boxes emitted

E+I — ERD-Inference (5 of 5)



ERD sees the scene is no longer empty and
wakes the detector — **solid green** = detections

Workload and platforms

- **BDD100K MOT** validation split:
 - **200** real dashcam clips, ego-motion
 - $\sim 40,000$ annotated frames, **7** classes
- **Detectors** (all off-the-shelf):
 - **CNN**: YOLO11x, **YOLO26x** (primary)
 - **Transformer**: RF-DETR-Large
- **Platforms**:
 - **AGX Thor**
 - **AGX Orin**

Baselines

- **Vanilla**: detector on every frame
- **FixedSkip- N** : every N th frame
- **CTD** [ICECAL'24]: confidence-triggered detection
- **Statues** [ISLPED'24]: pixel differencing
- **ERD-only** [RTCSA'22]: empty-scene screen on every frame



code released

Finding 1: the default point improves all three axes at once

YOLO26x, BDD100K MOT, default $u = 3 \times 10^{-4}$, vs. per-frame inference

+0.8 pp

AP@50

0.609 $\rightarrow$ 0.617

—17.6 %

Energy per clip

Thor; —14.4% on Orin

—8.8 %

Frame time

Thor; —13.7% on Orin

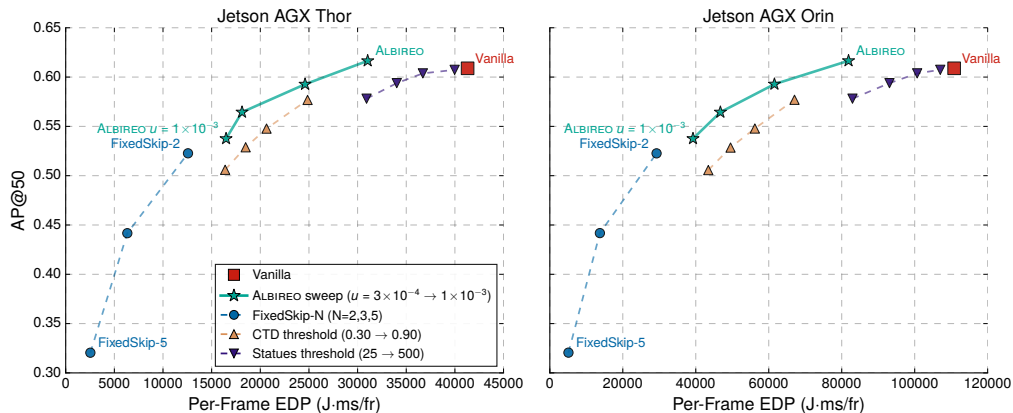
—24.9 %

Energy-delay product

Thor; —26.1% on Orin

Strict Pareto improvement over per-frame inference: higher accuracy, lower energy, lower latency — at a 19.3% skip rate.

Finding 2: the threshold sweep forms the upper Pareto frontier



- ALBIREO forms the **upper Pareto frontier**: whatever accuracy you need, it is the cheapest way to get there, and one knob (u) picks the point
- Per-frame inference pays **33% more EDP** than ALBIREO's default point, and still gets lower accuracy

Finding 3: the savings generalize — detectors, platforms, power modes

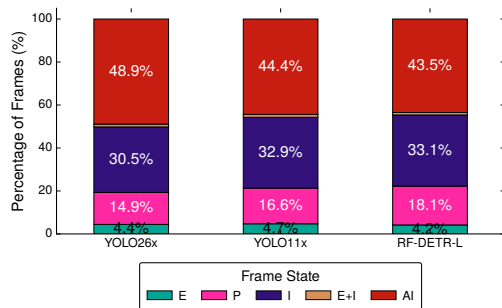
Plat.	Detector	Δ AP@50	Δ Energy	Skip
Thor	YOLO11x	−0.5 pp	−14.4%	21.3%
Thor	YOLO26x	+0.8 pp	−17.6%	19.3%
Thor	RF-DETR-L	−1.1 pp	−16.3%	22.2%
Orin	YOLO11x	−0.5 pp	−16.4%	21.3%
Orin	YOLO26x	+0.8 pp	−14.4%	19.3%
Orin	RF-DETR-L	−1.2 pp	−12.1%	22.2%

Detector-agnostic: accuracy within ± 1.2 pp, energy down 12–18%, in all six configurations

Board	Mode	Vanilla (J/clip)	ALBIREO (J/clip)	Δ Energy
Orin	15 W	1417	1166	−17.7%
Orin	30 W	1094	917	−16.2%
Orin	50 W	847	714	−15.7%
Thor	70 W	920	776	−15.7%
Thor	90 W	920	779	−15.3%
Thor	120 W	779	654	−16.1%

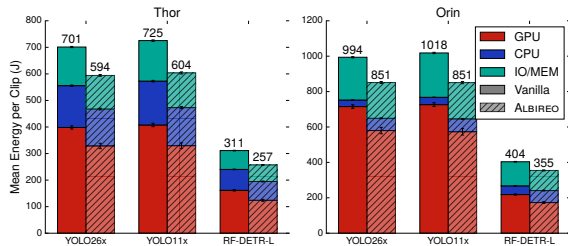
Budget-independent: savings hold in every power mode, even at 15 W

Finding 4: the frame-state ledger — every frame accounted for



- **Skip rate (E + P):** 19–22% across all three detectors
- **Rescue at work (AI):** almost half of detector frames carry a rescued box
- **Identical on Orin and Thor:** the policy is deterministic, so the savings are platform-portable

Finding 5: the savings are GPU savings — with a CPU caveat



- **GPU savings:** GPU energy drops 17.5–19%, more than the total reduction
- **The caveat:** bookkeeping runs on the CPU; on Orin, CPU power **doubles**
- **Deployment lesson:** skipping trades GPU work for CPU work

Concluding remarks

1. **Which-frame-to-skip is a per-object state question:** a forward-looking Kalman covariance trigger beats fixed cadence and stale-confidence gating
2. **The default operating point is not a trade-off:** it improves accuracy, energy, and latency together over per-frame inference, and the threshold sweep forms the upper Pareto frontier
3. **Drop-in by construction:** any detector, no retraining, no codec metadata, robust to ego-motion
4. **Open source:** code and evaluation pipeline are released

Thank you — questions?



Paper
[arXiv:2609.29648](https://arxiv.org/abs/2609.29648)



Code
github.com/amirtaherin/albireo

taherin.a@northeastern.edu



Workload and platforms

- **BDD100K MOT** validation split:
 - **200** real dashcam clips, ego-motion
 - ~40,000 annotated frames, **7** classes
- **Detectors** (all off-the-shelf, conf 0.50):
 - YOLO11x, **YOLO26x** (primary) — CNN
 - RF-DETR-Large — transformer
- **Platforms**: Jetson **AGX Orin** (JetPack 6.2) and **AGX Thor** (JetPack 7.1)

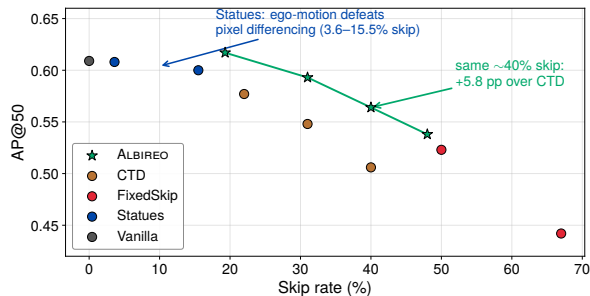
Baselines and metrics

- **5 baseline families**: Vanilla (every frame), FixedSkip- N , CTD [ICECAI'24], Statues [ISLPED'24], ERD-only [RTCSA'22]
- **Metrics**: AP@50, mAP@50:95, frame time, clip energy, per-frame EDP, skip rate, frame states
- **Power**: tegrastats at 100 ms, trapezoidal energy integration; paired Vanilla/ALBIREO runs, same 200 clips, same order, seed 42



code released

Backup: why the baselines fall short



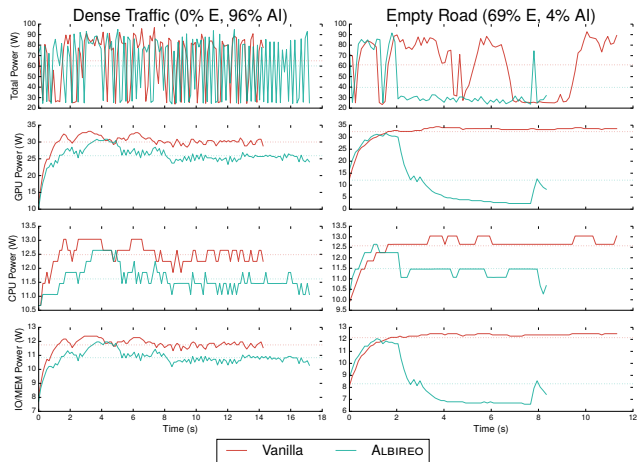
- **FixedSkip**: content-blind — cheap, but leaves the target accuracy region entirely beyond 50% skip
- **CTD**: confidence vs. the *last detection* goes stale across consecutive skips: at ~40% skip, ALBIREO leads by **+5.8 pp** AP@50
- **Statues**: pixel differencing under ego-motion almost never skips (3.6–15.5%)
- **ERD-only**: screening without per-object state *adds* 28% energy — the classifier runs on every frame, the detector still runs on 92%

Backup: ablation — component contribution

Variant	KF	Lifecycle	AP@50	Skip (%)
no-rescue variants	—	—	0.572–0.578	—
ALBIREO (full)	10D CA	rescue + ERD	0.617	19.3

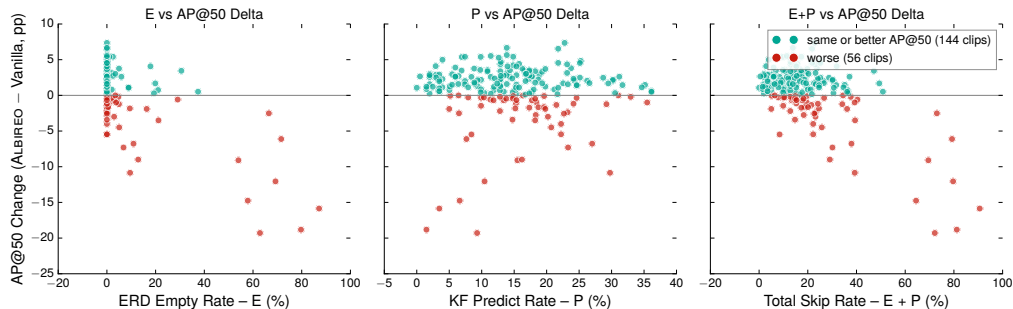
Rescue is the largest single AP@50 contributor. Full ablation table in the paper.

Backup: where the savings come from — gating the GPU



Skipping gates the GPU burst (CPU/I/O stay flat); dense traffic (96% AI) vs. empty road (69% E): savings track scene content.

Backup: adaptation, not uniform cadence — the per-clip view



Same or better AP@50 than Vanilla on 144 of 200 clips (72%); on the rest, the median drop is below 2 pp — and lower energy on 82–95% of clips ($p < 0.001$, paired). Large negative deltas concentrate on high-E clips with few ground-truth objects, where AP is sensitive to single misses.

Backup: stress test beyond the operating region

- At $u = 3 \times 10^{-3}$: **55% energy savings**, but AP@50 drops to 0.46 — outside the target region $\text{AP@50} \geq 0.50$
- Energy savings vs. accuracy is fully tunable through the single threshold u
- Composable with quantization and model compression (orthogonal: they cut per-inference cost, ALBIREO cuts the number of inferences)