Network Costs Fade, Compute Dominates: Carbon Modeling for Edge Stream Processing in the LLM Era

BRIAN RAMPRASAD, University of Toronto, Canada
EYAL DE LARA, University of Toronto, Canada

Stream processing systems are widely adopted for executing real time machine learning pipelines and are increasingly incorporating billion-parameter LLMs for inference. This shift fundamentally changes the compute-network tradeoff that governs operator placement where for example object detection (OD) pipelines at 30 fps are network-bound, Vision Language Model (VLMs) inference at 1 fps is now overwhelmingly compute-bound. We build a system dynamics model calibrated with empirical GPU benchmarks across three hardware generations to quantify how this transition impacts the carbon footprint of edge vs. cloud placement over a decade. Under a GPU Jevons paradox—where efficiency gains are consumed by richer models within two years—we find that network costs become negligible, grid carbon intensity dominates, and the edge-to-cloud operational carbon penalty widens to 2.45× over the decade, making a split strategy—lightweight filtering at the edge, heavy inference in the cloud—the only placement that consistently minimizes total emissions.

CCS Concepts: • **Hardware** → **Impact on the environment**; • **Computer systems organization** → *Cloud computing*; • **Computing methodologies** → **Distributed artificial intelligence**.

Additional Key Words and Phrases: sustainability, edge computing, stream processing, carbon footprint, system dynamics

1 Introduction

Distributed compute infrastructure, including smaller data centers and resources integrated within or near cellular networks, commonly referred to as edge computing, is increasingly being used to reduce application response times for GPU-intensive workloads such as Large Language Models (LLMs). At the same time, existing machine learning applications deployed as part of data processing pipelines are also incorporating LLMs [1, 16]. For example, Vision Language Models (VLMs) used for video analytics pipelines have evolved from lightweight object detection (OD) to LLM-enriched scene analysis. The hardware required at the edge escalates from 32 W embedded GPUs [23] to datacenter-class servers drawing over 6,000 W [24]—erasing the power asymmetry that made edge devices cheap to deploy from an energy and a carbon perspective. When we add the embodied carbon of dedicated edge hardware [5, 9] and the Power Usage Efficiency (PUE) penalty of edge sites [28] which for example have less efficient cooling systems compared to large cloud datacenters, the carbon case for edge placement requires more precise planning to choose the optimal infrastructure for application deployment.

This paper highlights the emerging challenge of precise planning in the context of stream processing applications that are widely used to serve real time machine learning workloads and shows that the carbon-optimal placement depends on what each operator *does to data*. A filter that reduces volume (e.g., OD producing bounding boxes at 1% of input) makes edge placement carbon-efficient: low-power hardware and reduced WAN traffic dominate. A compute-intensive operator (e.g., a vision-language model producing scene descriptions) demands the same datacenter GPUs regardless of location, yet no existing carbon-aware scheduler reasons about this per-operator property. Systems for batch DAGs [13], ML training [34], and request-response services [32] use compute-only carbon models. Edge stream processing frameworks [10, 31] optimize placement for performance but not carbon. No existing approaches model how an operator's data-transformation ratio couples placement to downstream network and compute costs.

We present a dynamic end-to-end carbon model for edge stream processing—to our knowledge, the first to jointly capture network energy, compute, and embodied carbon with per-operator placement. A system dynamics simulation over the full parameter space reveals three findings that, taken together, overturn the conventional wisdom:

- **The magnitude of computation drives the operator placement strategy.** For lightweight operators that *reduce* data, edge placement saves over half the carbon of cloud-only processing because low-power edge hardware and reduced WAN traffic dominate. However, for heavy operators that *expand* data (e.g., LLM analytics), edge placement *increases* carbon. (§4.1).
- **WAN energy matters less than assumed.** Published WAN energy intensity estimates span 0.001–0.03 kWh /GB [2, 22] —a 30× range that dominates the placement decision for lightweight workloads. But as workloads shift to compute-intensive LLM analytics, WAN energy becomes a rounding error, raising questions about where the community should focus empirical measurement effort (§4.2).
- **The GPU Jevons paradox erodes the edge advantage over time.** Each GPU generation lowers the cost per inference, but this enables adoption of richer models that consume the efficiency gains. Over a ten-year horizon, workload intensity grows by orders of magnitude; we conduct experiments on various AWS cloud GPUs to obtain the GPU saturation rates and parameterize our simulation with this data. The simulation results show that even aggressive edge GPU upgrades cost more carbon than cloud, and only a split strategy—lightweight filtering at the edge, heavy inference in the cloud—consistently wins (§4.3).

These results show the need for a re-evaluation of how the systems community models and measures edge-computing-based deployments as LLMs become more widely integrated into applications. Our model serves as a foundation that practitioners can extend to incorporate additional factors contributing to the carbon footprint of stream processing applications. This work builds on our prior efforts to model the carbon footprint of edge-based applications in the pre-LLM era [25]. We extend our model for the LLM era to capture the shift from lightweight Computer Vision (CV) workloads to

Authors' Contact Information: Brian Ramprasad, brianr@cs.toronto.edu, University of Toronto, Canada; Eyal de Lara, delara@cs.toronto.edu, University of Toronto, Canada.

compute-heavy pipelines, adding embodied carbon, VLM-oriented workloads, and the selectivity characteristics of stream processing operators.

2 Background and Motivation

Carbon-aware scheduling has made real progress for batch jobs, ML training, and request-response services—workloads where the scheduling decision (when or where to run) does not change how much data crosses the network. PCAPS [13] schedules precedence-constrained data processing DAGs carbon-aware (up to 32.9% reduction); GREEN [34] defers ML training to low-carbon hours (up to 41%); CarbonEdge [33] places each edge application entirely on whichever nearby data center has the lowest grid carbon intensity (up to 78.7%). Quality Time [32] adapts LLM serving quality to grid conditions; Go with the Flow [21] models end-to-end carbon for video CDN using per-physical-hop WAN energy with CAIDA network topology—a more precise but CDN-specific WAN model than our per-GB intensity approach. None of these systems model the defining property of stream processing: operators *transform* data, creating a coupling between where an operator runs and how much data downstream operators must process. A filter placed at the edge reduces WAN transfer; an LLM operator placed at the edge may *increase* it. This coupling is absent from batch jobs (data is staged before execution), training (no downstream consumers), and CDN (data delivered unchanged). Sharma et al. [26] argue that cloud is the preferred inference location for real-time video analytics, showing that datacenter GPUs amortize network delay with faster inference than on-device Jetson execution. Our work extends this argument to the carbon dimension, showing that the same compute asymmetry that favors cloud for latency also favors it for carbon—compounded by PUE, grid intensity, and embodied cost differences that latency-focused analyses do not capture.

On the stream processing side, COSTREAM [10] and CAPSys [31] optimize per-operator placement for latency and throughput, but are carbon-unaware. Prior system dynamics modeling of edge carbon captured end-to-end emissions but used a higher kWh/GB network energy estimate that may overstate WAN savings. The difficulty in using published specifications to estimate real world performance continues to be an issue as Sinha et al. [27] identified a *utilization fallacy*: higher GPU utilization does not guarantee better carbon efficiency. We find an analogous *edge placement fallacy*: processing at the edge does not guarantee lower carbon. Xue et al. [36] show that harvesting idle cycles on already-provisioned edge devices for ML *training* can achieve 4–8× carbon reduction—but streaming inference is time sensitive so these workloads require dedicated GPUs eliminating this advantage.

No existing system combines per-operator placement awareness from stream processing with end-to-end carbon modeling that includes network energy, embodied carbon, and GPU hardware heterogeneity.

3 An End-to-End Carbon Model

We model the carbon footprint of edge stream processing using system dynamics (SD) [20], a methodology that represents systems as coupled differential equations over stocks (accumulators), flows (rates), and feedback loops. SD is the right tool here because the carbon-optimal placement depends on at least five interacting variables whose non-linear interaction defies intuition, and because deployment growth creates temporal dynamics—accumulating stocks and capacity-triggered provisioning events—that static analysis cannot capture. We instantiate the model with a concrete use case: city-based video analytics, where a fleet of cameras streams video to inference operators that can be placed at the edge (cell tower base stations), in the cloud, or split between both. Video analytics is a natural fit because it exhibits the structural properties common to edge stream processing: compute-intensive operators whose cost scales with model complexity, and a placement decision that trades network transfer against local compute. These pipelines mirror the broader trend across edge domains—from autonomous driving to industrial IoT—where foundation models are replacing purpose-built operators.

3.1 Model Structure

Table 1. Model parameters and sources (OD | LLM).

Parameter	Value	Source
WAN output ratio	1% | 10%	bbox (OD); text (LLM)
Edge PUE	1.5 (+50%)	Midpoint of 1.4–1.6 [28]
Cloud PUE	1.12 (+12%)	Lower end of 1.1–1.2 [28]
Edge CI	400 gCO_2/kWh	Electricity Maps [4]
Cloud CI	300 gCO_2/kWh	Electricity Maps [4]
Filter selectivity	50%	Reducto [14] (OD detection at 90% accuracy)
Data rate/camera	1.5 GB/hr	1080p@30fps; scales to 0.05 GB/hr at 1 fps
RAN BS power draw	84 W	Lopez-Perez [15]
WAN intensity	0.008 kWh/GB	Mid-estimate; range 0.001–0.03 [2, 22]; §4.2

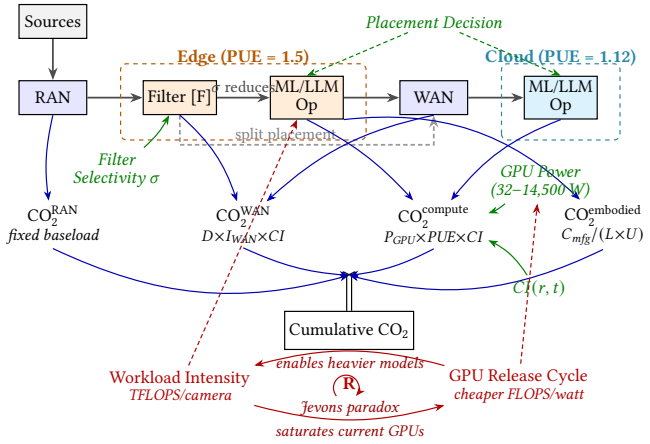

Fig. 1. End-to-end carbon model for edge stream processing.

The model considers the cellular Radio Access Network (RAN), Wide Area Network (WAN), compute, and embodied emissions, and is parameterized with recent systems benchmarks shown in Table 1 and our empirical experiments on AWS shown in Table 2. In our video analytics use case, cameras transmit frames to cell towers;

from there, video is either processed at the edge (base station) or forwarded over the WAN to a cloud datacenter.

Total CO_2 emissions are the sum of four components:

$$CO_2^{\text{total}} = CO_2^{\text{WAN}} + CO_2^{\text{RAN}} + CO_2^{\text{compute}} + CO_2^{\text{embodied}} \quad (1)$$

WAN emissions use a per-GB energy intensity model:

$$CO_2^{\text{WAN}} = D_{\text{WAN}} \times I_{\text{WAN}} \times CI \quad (2)$$

where D_{WAN} is the data volume traversing the backhaul (GB/month), I_{WAN} is the WAN energy cost per GB (kWh/GB), and CI is the grid carbon intensity (gCO_2/kWh). Aslan et al. [2] estimated 0.06 kWh/GB in 2015 with a halving period of roughly two years, but this rate has slowed: Malmodin et al. [18] report that global network traffic grew 600% between 2015 and 2022 while energy grew only 18–64%. Mytton et al. [22] and Guennebaud and Bugeau [8] argue the true marginal cost is near zero because network equipment draws largely fixed power; however, shared infrastructure must still be amortized across its users. We adopt 0.008 kWh/GB, the approximate geometric mean of the marginal (∼0.001) and average (∼0.03) endpoints—the natural midpoint for a parameter that spans orders of magnitude. Edge placement reduces D_{WAN} by filtering data before the backhaul hop—the WAN savings scale directly with filter selectivity. In our model, the edge filter is a lightweight change-detection gate (e.g., Reducto [14]) that discards redundant frames before they reach the inference operator; only frames that pass the filter are transmitted and processed, so filtering reduces both WAN volume and cloud compute demand.

RAN emissions use the 5G massive MIMO power model from Lopez-Perez et al. [15]. RAN cost is fixed regardless of placement because cameras always upload to the base station; power is allocated proportionally to the deployment's share of base station (BS) capacity.

Compute emissions:

$$CO_2^{\text{compute}} = \frac{N_{\text{cam}} \cdot (1-\sigma) \cdot P_{\text{GPU}}}{C_{\text{cam/GPU}}} \times \text{PUE} \times CI(r,t) \times t \quad (3)$$

where N_{cam} is the number of cameras, σ the filter selectivity, P_{GPU} the per-GPU power, $C_{\text{cam/GPU}}$ the cameras served per GPU, PUE the power usage effectiveness, and t the operating time. We set the edge PUE to 1.5 and cloud PUE to 1.12 reflecting that hyperscale operators achieve near-best-case efficiency while edge sites typically do not. [28]

Embodied emissions are sunk at provisioning:

$$CO_2^{\text{embodied}} = \sum_{\text{tier}} N_{\text{new}}^{\text{tier}} \times C_{\text{mfg}}^{\text{tier}} \quad (4)$$

where $N_{\text{new}}^{\text{tier}}$ is the number of servers provisioned and $C_{\text{mfg}}^{\text{tier}}$ is the manufacturing carbon per server for each hardware tier. When a server is deployed, its full C_{mfg} is charged immediately to recognize the sunk costs [3].

Temporal dynamics. The model tracks five quantities over a 10-year horizon: (i) camera fleet size (10%/yr growth); (ii) grid carbon intensity, declining asymmetrically (edge 4%/yr, cloud 8%/yr); (iii) a multi-tier GPU fleet spanning five generations; (iv) workload complexity driven by the GPU Jevons paradox (48%/yr); and (v) model provisioning when new GPUs are released.

The complete model is shown in Figure 1. Starting at the Source in the top left, the streaming pipeline sends data to the filter scheduler that places operators across edge and cloud; filter selectivity σ determines how much data reaches the WAN, and the dashed path shows split placement (filter at edge, ML/LLM in cloud). Each pipeline segment generates one of four CO_2 components—RAN, WAN, Compute, and Embodied (middle)—which accumulate into cumulative CO_2 (bottom). The reinforcing loop **R** (red) captures the GPU Jevons paradox: each GPU generation delivers cheaper inference, enabling heavier models that saturate the hardware and drive demand for the next generation.

4 Evaluation

We implement our model using PySD with a Vensim-compatible model definition and evaluate two representative video analytics applications that span the compute-intensity spectrum:

- **Video Object Detection** (data-reducing): 1080p dashcam video at 1.5 GB/hr per camera from the Berkeley Deep Drive 100K (BDD100K) driving dataset [37], spanning diverse weather, lighting, and scene conditions. An OD produces bounding boxes (∼1% of input volume), reducing downstream data. We model three generations of YOLO detectors—YOLOv5s, YOLOv8m, and YOLOv11x—each running at 30 fps.
- **LLM-Enriched Analytics** (compute-intensive): The same dashcam video, but each frame is sent to a vision-language model with the prompt *"Describe the scene in this image. Note any vehicles, pedestrians, and notable activity in 2–3 sentences."* The model produces a ∼80-token natural-language scene description (∼10% of input volume), replacing compact bounding boxes with rich semantic text, e.g.: *"The image captures a view from inside a vehicle, looking out at a black Chevrolet Suburban parked in a lot beneath a large, curving concrete overpass. The scene is quiet and devoid of pedestrians, with the focus on the SUV and the imposing infrastructure of the highway above."* We use two open vision-language models—Qwen3-VL-8B (8B parameters) and Gemma 3 12B (12B parameters)—representing successive size classes in the Jevons workload trajectory (years 6–8 and 8–10 respectively). Qwen3-VL-8B requires 8–18 TFLOPS per image [12, 35]—500–1,000× more compute than YOLOv5s (0.017 TFLOPS) [11].

Although both workloads reduce output volume, LLM analytics requires orders of magnitude more compute per frame, shifting the balance between network and compute costs. Our evaluation progresses from mechanism to scale: we first show how this shift changes the placement decision (§4.1), then expose sensitivity to contested WAN parameters (§4.2), and finally we model a 953 camera deployment derived from the CVPR 2024, 8th AI City Challenge [29] over 10 years under the GPU Jevons paradox (§4.3).

4.1 Same Application, Two Carbon Profiles

Figure 2 compares all three placement strategies for both applications. The results are starkly different.

Table 2. Saturation throughput and measured GPU power via `nvidia-smi` on AWS VMs; system overhead (CPUs, DRAM, NICs) is excluded.

GPU	AWS Instance	Model	Sat. (req/s)	cam/GPU	Power (W)
A100 (8×)	p4de.24xlarge	Qwen3-VL-8B	159	20	2,853
A100 (8×)	p4de.24xlarge	Gemma 3 12B	78	10	3,017
B200 (8×)	p6-b200.48xlarge	Qwen3-VL-8B	291	36	4,287
B200 (8×)	p6-b200.48xlarge	Gemma 3 12B	237	30	6,125
B300 (8×)	p6-b300.48xlarge	Qwen3-VL-8B	289	36	4,182
B300 (8×)	p6-b300.48xlarge	Gemma 3 12B	244	30	6,050

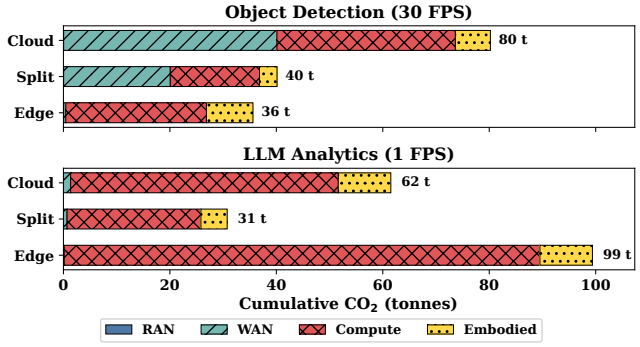

Fig. 2. Carbon breakdown by placement strategy (edge, cloud, split) for 953 cameras over 12 months.

For Object Detection, **Cloud** (80 t) is dominated by WAN (40 t) and compute (34 t): 4 A100 servers handle all 953 cameras, but shipping raw video is costly. **Edge** (36 t) narrowly wins: low-power Jetsons eliminate WAN (0.4 t of bounding-box output), but require 159 devices whose aggregate compute (26 t) and embodied carbon (9 t) nearly offset the savings. **Split** (40 t) halves both WAN and cloud compute by filtering 50% of frames at the edge, but its residual WAN cost (20 t) keeps it slightly above edge.

For LLM Analytics (1 fps), **cloud** (62 t) provisions 6 A100 servers for all 953 cameras (50 t compute, 10 t embodied); at 1 fps, WAN is negligible (1 t vs. 40 t for OD at 30 fps). **Edge** (99 t) is the worst option at 1.6× cloud: the same 6 A100 servers at edge PUE 1.5 and dirtier grid (400 vs. 300 gCO_2/kWh) inflates compute carbon to 89 t. **Split** (31 t) wins at 50% savings: the edge filter reduces cloud demand to 477 effective cameras, cutting compute carbon to 25 t and embodied to 5 t. The data-transformation property of the operator—not just its compute cost—determines the carbon-optimal placement. Despite running at 1 fps instead of 30, LLM analytics produces a comparable total carbon footprint to OD — but with a fundamentally different composition: compute and embodied carbon dominate instead of WAN. Even a modest increase in LLM frame rate would dramatically increase total emissions and change the composition mix.

Cloud compute scales as $(1 - \sigma)$ (Eq. 3), so filter savings are linear in σ—but the absolute impact depends on the operator's cost per frame. For LLM workloads, even a conservative filter ($\sigma = 20\%$) eliminates substantial compute carbon because each filtered frame avoids an expensive VLM inference. For OD, the same σ saves far less because per-frame compute is cheap. Pushing σ higher demands a more sophisticated edge filter (e.g., a lightweight classifier rather than simple motion detection), which increases edge hardware requirements and risks accuracy loss—so practitioners face a diminishing-returns tradeoff. Our baseline of 50% from Reducto [14] is a moderate, empirically validated operating point; the key insight is that split placement is most valuable precisely where compute costs are highest, and even modest filtering delivers outsized savings for LLM workloads.

4.2 The WAN Is Less Relevant Now

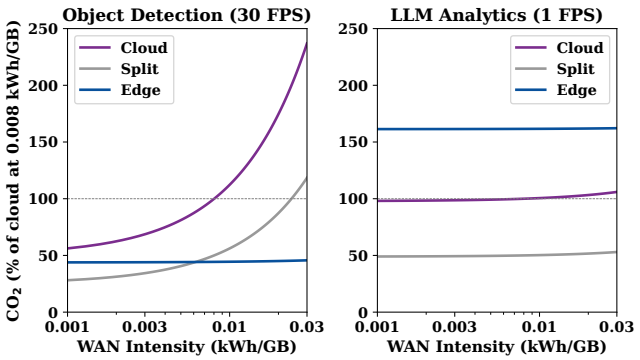

Fig. 3. Total CO_2 by placement strategy (edge, cloud, split) as WAN energy intensity sweeps from 0.001 to 0.03 kWh/GB for 953 cameras over 12 months. Each panel normalizes to cloud CO_2 at 0.008 kWh/GB.

Published estimates of WAN energy intensity have been a persistent source of uncertainty in placement studies. We ask: does the choice of WAN value change the placement decision?

Figure 3 sweeps WAN intensity across the full range for all three placement strategies. For OD (left), the answer is *yes*—the WAN dispute is decisive. Cloud CO_2 rises from 55% to over 230% of the baseline as WAN intensity increases, crossing both split and edge. At Mytton's low estimate, cloud is competitive; at Aslan's high estimate, edge wins by more than 4×. The placement decision depends entirely on which published value one believes. For LLM Analytics (right), the answer is *no*—the WAN dispute is irrelevant. All three lines are nearly flat because compute cost dominates: VLM compute dwarfs WAN energy, so WAN intensity becomes a rounding error. Edge is consistently the worst option (~160% of cloud) due to higher PUE (1.5 vs. 1.12) and dirtier grid (400 vs. 300 gCO_2/kWh). This result holds at any frame rate: increasing LLM fps scales both compute and WAN proportionally, so the normalized placement ranking is unchanged.

4.3 The GPU Jevons Paradox

The preceding experiments use a small camera fleet with fixed workloads and static hardware. Real deployments grow: cameras are added, models get richer, and hardware must keep up. Each GPU hardware generation has delivered more performance per watt, and this efficiency gain lowers the cost per inference, which drives practitioners to adopt more powerful models—which in turn consume the efficiency gains and more as shown in Figure 4.

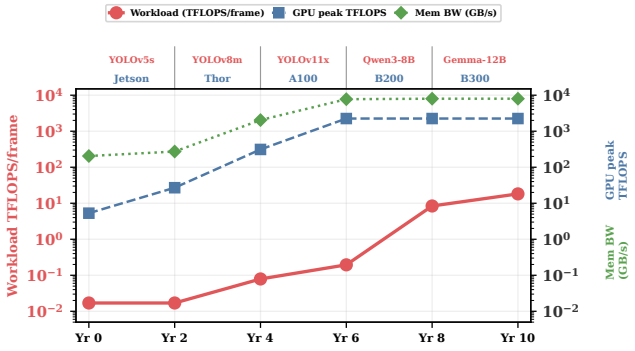

Fig. 4. The GPU Jevons paradox: workload demand per camera grows 1,000× as practitioners adopt richer models (YOLOv5s → Gemma 3 12B)

This is the GPU Jevons paradox [17] applied to edge AI: the reinforcing loop runs from GPU efficiency → cheaper inference → richer models adopted → higher FLOPS demand per camera → need for more or better GPUs as depicted in the lower section of our carbon model in Figure 1.

The goal of this evaluation is to show the impact of the paradox on deployment decisions that are available and the resulting carbon footprint of those decisions after 10 years. We consider six different strategies that cover likely decisions such as deploying stream processing operators exclusively at edge data centers or the cloud, upgrading GPUs more frequently, etc. The model adoption cadence follows the pattern of growth in relation to the reduction in inference cost driven by the paradox. We model the paradox impact as a reinforcing loop with a continuously growing workload intensity: TFLOPS per frame grows from YOLOv5s (0.017) through successive YOLO v5,v8,v11 generations to VLMs: Qwen3-VL-8B and Gemma 3 12B (8–18 TFLOPS) over 10 years as shown in Table 3. The workload demonstrates the transition from simple OD to LLM inference.

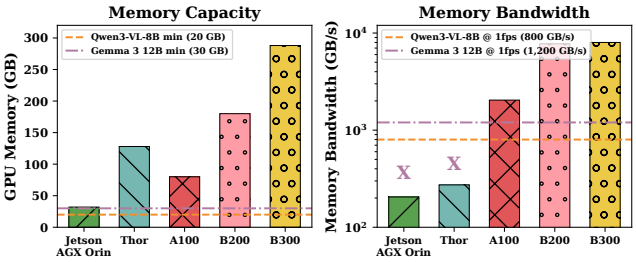

Fig. 5. Edge GPUs can *load* Qwen3-VL-8B (memory ≥ 20 GB) but cannot *serve* it: memory bandwidth below 800 GB/s (the minimum for 1 fps with 50-token output) makes real-time VLM inference infeasible on Jetson and Thor.

The OD runs at 30 fps (real-time); LLM scene analysis runs at 1 fps, due to the memory-bandwidth limits of autoregressive decoding (see Fig. 5) and the periodic nature of scene description. To obtain realistic Edge OD throughput (cameras per GPU) the model uses the end-to-end pipeline measurements from RegenHance [30], which include video decode, preprocessing, and scheduling overhead, giving for example ~6 cameras per Jetson at 30 fps for YOLOv5s and we apply the same methodology to all YOLO models used. We source the GPU power requirements for the Jetson and Thor from NVIDIA's product information at 32 W at the system level (Jetson AGX Orin, 2022) to 75 W (Jetson Thor, 2025) [23]. For LLM workloads we use A100, B200, and B300 and source energy consumption directly from our AWS benchmark experiments shown in Table 2. Embodied carbon for each tier is sourced from published LCAs (Jetson: 55 kg [9]; A100 8-GPU: 1,644 kg [5]; B200 8-GPU: 2,274 kg from NVIDIA's HGX B200 Product Carbon Footprint) and estimated via die-level scaling otherwise (Thor: ~150 kg; B300: ~2,948 kg, scaled from B200 by HBM ratio).

Placement Strategies:
The workloads for the simulation consist of the same initial 953 cameras [29] used in the preceding experiment now growing at ~10%/yr [19] (conservatively; China's unit growth exceeds 37%/yr [6]). The six strategies are shown in Table 3 over 10 years.

Table 3. Fleet deployment by strategy (GPUs at each time point; yr 2–10 are end-of-era peaks including Jevons growth).

		GPUs deployed					
	Strategy	**Yr 0**	**Yr 2**	**Yr 4**	**Yr 6**	**Yr 8**	**Yr 10**
	Workload	YOLOv5s		YOLOv8m	YOLOv11x	Qwen3-8B	Gemma-12B
	Cameras	*953*	*1,151*	*1,393*	*1,686*	*2,040*	*2,469*
S1	Upgrade edge	160 J	418 J	506 T	1,552 E	128 B2	184 B3
S2	Edge filter	16 C	40 C	240 C	776 C	112 C	280 C
S3	Cloud year 2	160 J	418 J	472 C	1,552 C	224 C	560 C
S4	Cloud day 1	32 C	80 C	472 C	1,552 C	224 C	560 C
S5	A100 edge (elastic)	32 E	80 E	472 E	1,552 E	224 E	560 E
S6	A100 edge (upfront)	1,552 E	1,552 E	1,552 E	1,552 E	1,552 E	1,552 E

J=Jetson 32W, T=Thor 75W, C=A100 cloud, E=A100 edge, B2=B200, B3=B300.
All GPU counts in table are singles, servers (A100/B200/B300) have 8X.

S1 – Progressive edge upgrade: Edge hardware is upgraded every two years to match GPU generations: Jetson >Thor >A100 >B200 >B300. Each upgrade matches the next workload era, starting with YOLOv5s on Jetson and ending with Gemma 3 12B on B300.
S2 – Split (edge filter + cloud inference): A lightweight edge filter (e.g., motion detection) discards ~50% of frames; only selected frames are sent to cloud A100 servers for full inference. This halves cloud compute demand at the cost of WAN transfer for the filtered stream.
S3 – Jetson start, cloud at year 2: Deploys Jetsons for the first two years of OD, then migrates all inference to cloud A100 servers from year 2 onward.
S4 – Cloud from day 1: All inference runs on elastically provisioned cloud A100 servers for the entire 10 years.
S5 – Edge A100, elastic scaling: Same A100 hardware as S4, but deployed at the edge. Hypothetically assumes cloud-like elastic scaling (servers added/removed as demand changes) to isolate the effect of location: only PUE and grid carbon intensity differ from S4.
S6 – Edge A100, upfront provisioning: The realistic edge scenario: physical hardware must be purchased and installed before deployment. The entire peak fleet is provisioned at month 0 to cover the worst-case demand (end of the YOLOv11x era, when Jevons decay has maximally eroded capacity).

4.4 Experimental Results

Figure 6 compares the six strategies. All curves flatten at year 6 when the workload transitions from OD (30 fps) to VLM inference (1 fps): the 30× frame rate reduction temporarily lowers emissions despite the shift to heavier models, but fleet growth and Jevons decay restore the upward trend within two years. Jevons decay in practice manifests as operators raising image resolution, or adding inference passes on detected regions, more complex queries requested—changes that steadily erode cameras-per-server within a fixed model generation.

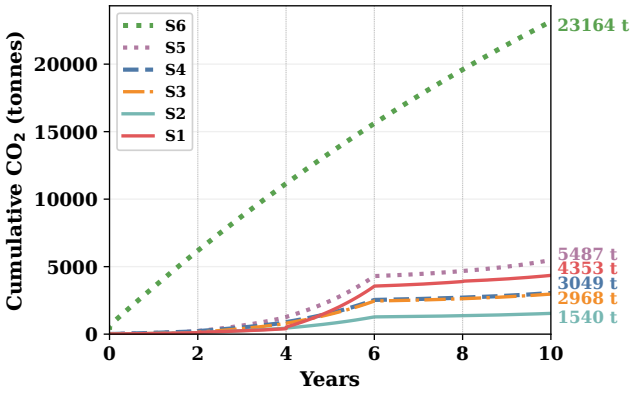

Fig. 6. The GPU Jevons paradox for a camera deployment growing at 10%/yr.

Split placement (**S2**, 1,540 t) is the clear winner. A lightweight edge filter discards ∼50% of frames before they reach the cloud, halving compute demand regardless of whether the downstream operator is YOLOv5s or Gemma 3 12B. This saving is remarkably stable: S2 reduces emissions by 49–50% at every two-year checkpoint. WAN transfer for the filtered stream adds only 146 t (10% of S2's total)—a negligible cost for halving compute. The remaining five strategies all suffer from the same structural problem: operational compute accounts for 80–97% of emissions, and any strategy that places heavy inference at the edge pays a compounding penalty from higher PUE (1.5 vs. 1.12) and grid carbon intensity (400 vs. 300 gCO_2/kWh) that widens from 1.79× to 2.45× over the decade. Edge-only strategies (S1, S5, S6) emit 1.4–7.6× more than cloud regardless of whether hardware is upgraded (S1), elastically scaled (S5), or provisioned upfront (S6)—the PUE×CI gap dominates. Cloud-only strategies (S3, S4) avoid this penalty but cannot reduce the compute itself; only edge filtering (S2) achieves both.

5 Discussion

Limitations. Our model is parameterized with empirically measured GPU throughput using AWS cloud GPUs and published infrastructure data; real-world deployments may differ. We fix grid carbon intensity at 400 gCO_2/kWh (edge) and 300 gCO_2/kWh (cloud), representative of US averages from Electricity Maps [4]. In practice, CI varies by location: a cloud region on a cleaner grid would widen the gap further in favor of cloud and split placement, while placing cloud in a coal-heavy region could raise cloud CI above edge CI, narrowing or inverting the advantage. However, the structural asymmetry persists: hyperscale operators choose low-CI regions and invest in renewable PPAs [7], while edge sites are constrained to wherever the cameras are deployed, giving operators less control over grid carbon. Our model's asymmetric decarbonization rates (cloud 8%/yr vs. edge 4%/yr) reflect this structural difference. The key finding—that compute dominates WAN as workloads shift to LLMs—holds regardless of absolute CI values, since it is driven by the orders-of-magnitude increase in GPU memory and FLOPS required per frame.

Future work. These findings point to a new design principle: carbon-aware stream processing must jointly optimize operator placement, hardware selection, and embodied carbon amortization—a problem no existing scheduler addresses. Follow-up directions include: (1) *carbon-aware stream processing schedulers* in production frameworks (Flink, Kafka Streams) that use the filter threshold for per-operator placement; (2) *carbon-aware model selection* between LLM sizes (8B vs. 70B) based on grid conditions, extending [32] to pipelines where quality cascades through stages; and (3) *carbon budgets for streaming* that amortize against operational savings [3]. (4) *fine grained runtime carbon attribution* incorporating GPU utilization fluctuations, excessive reservations, and tenant reuse, all of which may affect embodied and operational carbon.

References

[1] [n. d.]. Overview — nightlies.apache.org. https://nightlies.apache.org/flink/flink-agents-docs-latest/docs/get-started/overview/. [Accessed 19-05-2026].

[2] Joshua Aslan, Kieren Mayers, Jonathan G. Koomey, and Chris France. 2018. Electricity Intensity of Internet Data Transmission: Untangling the Estimates. *Journal of Industrial Ecology* 22, 4 (2018), 785–798. doi:10.1111/jiec.12630

[3] Noman Bashir, Varun Gohil, Anagha Belavadi Subramanya, Mohammad Shahrad, David E. Irwin, Elsa Olivetti, and Christina Delimitrou. 2024. The Sunk Carbon Fallacy: Rethinking Carbon Footprint Metrics for Effective Carbon-Aware Scheduling. In *Proceedings of the 2024 ACM Symposium on Cloud Computing (SoCC '24)*. ACM, 542–551. doi:10.1145/3698038.3698542

[4] Electricity Maps. 2026. Real-time and Historical Carbon Intensity Data. https://www.electricitymaps.com Accessed: April 2026.

[5] Jonas Falk, Clémence Lannou, and Sébastien Travadel. 2025. More than Carbon: Cradle-to-Grave Environmental Impacts of GenAI Training on the NVIDIA A100 GPU. *arXiv preprint arXiv:2509.00093v3* (2025). https://arxiv.org/abs/2509.00093

[6] Jinmei Feng, Hong Ma, Mingzhi Xu, and Wei You. 2026. Keeping an Eye on the Villain: Assessing the Impact of Surveillance Cameras on Crime. *Journal of Development Economics* 178 (2026), 103430. doi:10.1016/j.jdeveco.2025.103430 SSRN working paper August 2024. Reports China camera growth: 20M (2017) to 200M (2019) to 700M (2023).

[7] Google. 2025. *2025 Environmental Report.* Technical Report. Google. https://sustainability.google/google-2025-environmental-report/

[8] Gaël Guennebaud and Aurélie Bugeau. 2024. Energy consumption of data transfer: Intensity indicators versus absolute estimates. *Journal of Industrial Ecology* 28, 4 (2024). doi:10.1111/jiec.13513

[9] Debajyoti Halder, Deboparna Banerjee, Akash Mani, Anshul Gandhi, and Erez Zadok. 2025. How Carbon Metrics Impact Device Selection. In *CarbonMetrics Workshop (co-located with ACM SIGMETRICS)*. ACM. doi:10.1145/3764944.3764968

[10] Tiemo Bang Heinrich, Carsten Binnig, Manisha Luthra, and Boris Koldehofe. 2024. COSTREAM: Learned Cost Models for Operator Placement in Edge-Cloud Environments. In *Proceedings of the 40th IEEE International Conference on Data Engineering (ICDE '24)*. IEEE. doi:10.1109/ICDE60146.2024.00162

[11] Muhammad Hussain. 2023. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines* 11, 7 (2023), 677. doi:10.3390/machines11070677 Comprehensive GFLOPS comparison across YOLO variants.

[12] Yizhang Jin, Jian Li, Yexin Liu, Tianjun Gu, Kai Wu, Zhengkai Jiang, Muyang He, Bo Zhao, Xin Tan, Zhenye Gan, Yabiao Wang, Chengjie Wang, and Lizhuang Ma. 2025. Efficient Multimodal Large Language Models: A Survey. *Visual Intelligence* 3 (2025), 27. doi:10.1007/s44267-025-00099-6 Reports LLaVA-1.5 + Vicuna-13B at 18.2 TFLOPS per inference.

[13] Adam Lechowicz, Rohan Shenoy, Noman Bashir, Mohammad Hajiesmaili, Adam Wierman, and Christina Delimitrou. 2025. Carbon- and Precedence-Aware Scheduling for Data Processing Clusters. In *Proceedings of the ACM SIGCOMM 2025 Conference (SIGCOMM '25)*. ACM, 1241–1244. doi:10.1145/3718958.3750478

[14] Yuanqi Li, Arthi Padmanabhan, Pengzhan Zhao, Yufei Wang, Guoqing Harry Xu, and Ravi Netravali. 2020. Reducto: On-Camera Filtering for Resource-Efficient Real-Time Video Analytics. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '20)*. ACM. doi:10.1145/3387514.3405874

[15] David Lopez-Perez, Adrián De Domenico, Nicola Piovesan, Gang Ye, Sadegh Bakhshi, and Rittwik Jana. 2022. A Survey on 5G Radio Access Network Energy Efficiency: Massive MIMO, Lean Carrier Design, Sleep Modes, and Machine Learning. *IEEE Communications Surveys & Tutorials* 24, 1 (2022), 653–697. doi:10.1109/COMST.2022.3142532

[16] Duo Lu, Siming Feng, Jonathan Zhou, Franco Solleza, Malte Schwarzkopf, and Uğur Çetintemel. 2025. VectraFlow: Integrating Vectors into Stream Processing. In *15th Annual Conference on Innovative Data Systems Research (CIDR'25). To appear. Based on the provided PDF p23-lu. pdf. Amsterdam, The Netherlands.*

[17] Alexandra Sasha Luccioni, Emma Strubell, and Kate Crawford. 2025. From Efficiency Gains to Rebound Effects: The Problem of Jevons' Paradox in AI's Polarized Environmental Debate. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency (FAccT '25)*. ACM. doi:10.1145/3715275.3732007

[18] Jens Malmodin, Nina Lövehagen, Pernilla Bergmark, and Dag Lundén. 2024. ICT sector electricity consumption and greenhouse gas emissions – 2020 outcome. *Telecommunications Policy* 48, 3 (2024), 102701. doi:10.1016/j.telpol.2023.102701

[19] MarketsandMarkets. 2025. *Video Surveillance Market – Global Forecast to 2031*. Technical Report. MarketsandMarkets Research. https://www.marketsandmarkets.com/Market-Reports/video-surveillance-market.html CAGR 7.8% (2025–2031).

[20] Donella H. Meadows. 2008. *Thinking in Systems: A Primer*. Chelsea Green Publishing.

[21] Jorge Murillo, Walid A. Hanafy, David Irwin, Ramesh Sitaraman, and Prashant Shenoy. 2026. Go with the Flow: Analyzing the Carbon Footprint of Green Streaming. In *Proceedings of the 17th ACM International Conference on Future and Sustainable Energy Systems (e-Energy '26)*. ACM. https://lass.cs.umass.edu/papers/pdf/eenergy26-go_with_flow.pdf

[22] David Mytton, Dag Lundén, and Jens Malmodin. 2024. Network energy use not directly proportional to data volume: The power model approach for more reliable network energy consumption calculations. *Journal of Industrial Ecology* 28, 4 (2024), 966–980. doi:10.1111/jiec.13512

[23] NVIDIA. 2026. NVIDIA IGX Thor: Industrial-Grade Edge AI Platform. https://www.nvidia.com/en-us/edge-computing/products/igx/ Accessed: April 2026. IGX T7000 with RTX PRO 5000 Blackwell dGPU delivers up to 5,581 FP4 TFLOPS at 300+W.

[24] NVIDIA. 2026. NVIDIA, T-Mobile, and Partners Integrate Physical AI Applications on AI-RAN Ready Infrastructure. https://nvidianews.nvidia.com/news/nvidia-t-mobile-and-partners-integrate-physical-ai-applications-on-ai-ran-ready-infrastructure Deploying RTX PRO 4500/6000 Blackwell GPUs at cell sites for edge AI.

[25] Brian Ramprasad, Hong Kai Chen, Alexandre da Silva Veith, and Eyal de Lara. 2021. Sustainable Computing on the Edge: A System Dynamics Perspective. In *Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications (HotMobile '21)*. ACM, 127–133. doi:10.1145/3446382.3448607

[26] Pragya Sharma, Hang Qiu, and Mani Srivastava. 2025. Cloud Is Closer Than It Appears: Revisiting the Tradeoffs of Distributed Real-Time Inference. In *2025 34th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 1–9.

[27] Prasoon Sinha, Dimitrios Liakopoulos, Ruihao Li, and Neeraja J. Yadwadkar. 2025. The Utilization Fallacy and the Real Drivers of Carbon-Efficient Inference Serving. In *ACM SIGENERGY Energy Informatics Review*, Vol. 5. ACM. Presented at HotCarbon '25.

[28] Uptime Institute. 2024. *Uptime Institute Global Data Center Survey Results 2024*. Technical Report. Uptime Institute. https://uptimeinstitute.com/resources/research-and-reports/uptime-institute-global-data-center-survey-results-2024

[29] Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Pranamesh Chakraborty, et al. 2024. The 8th AI City Challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 7261–7272. doi:10.1109/CVPRW63382.2024.00722 953 cameras, 1080p at 30fps, 2,491 identities, 100M+ bounding boxes.

[30] Weijun Wang, Liang Mi, Shaowei Cen, Haipeng Dai, and Xiaoming Fu. 2025. Region-based Content Enhancement for Efficient Video Analytics at the Edge. In *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI '25)*. USENIX Association. https://arxiv.org/abs/2407.16990 E2E throughput: Jetson AGX Orin ∼20 FPS, A100 ∼200 FPS for OD (6–9× lower than MLPerf raw inference).

[31] Yuanli Wang, Lei Huang, Zikun Wang, Vasiliki Kalavri, and Ibrahim Matta. 2025. CAPSys: Contention-Aware Task Placement for Data Stream Processing. In *Proceedings of the 20th European Conference on Computer Systems (EuroSys '25)*. ACM. doi:10.1145/3689031.3696085

[32] Philipp Wiesner, Dennis Grinwald, Philipp Weiss, Patrick Wilhelm, Ramin Khalili, and Odej Kao. 2025. Quality Time: Carbon-Aware Quality Adaptation for Energy-Intensive Services. In *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems (e-Energy '25)*. ACM, 415–422. doi:10.1145/3679240.3734614

[33] Li Wu, Walid A. Hanafy, Abel Souza, Khai Nguyen, Jan Harkes, David Irwin, Mahadev Satyanarayanan, and Prashant J. Shenoy. 2025. CarbonEdge: Leveraging Mesoscale Spatial Carbon-Intensity Variations for Low Carbon Edge Computing. In *Proceedings of the 34th International Symposium on High-Performance Parallel and Distributed Computing (HPDC '25)*. ACM, 12:1–12:13. doi:10.1145/3731545.3731576

[34] Kaiqiang Xu, Decang Sun, Han Tian, Junxue Zhang, and Kai Chen. 2025. GREEN: Carbon-efficient Resource Scheduling for Machine Learning Clusters. In *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI '25)*. USENIX Association, 999–1014. https://www.usenix.org/conference/nsdi25/presentation/xu-kaiqiang

[35] Fuzhao Xue, Fan Chen, Jiangchao Yao, Dian Li, Bing Liu, Ya Zhang, and Yanfeng Wang. 2025. LVPruning: Language-Guided Vision Token Pruning for Multimodal Large Language Models. In *Findings of the Association for Computational Linguistics: NAACL 2025*. https://arxiv.org/abs/2501.13652 LLaVA-1.5 baseline: 8.38 TFLOPS on VQA tasks.

[36] Leyang Xue, Meghana Madhyastha, Randal Burns, Myungjin Lee, and Mahesh K. Marina. 2025. Towards Decentralized and Sustainable Foundation Model Training with the Edge. In *ACM SIGENERGY Energy Informatics Review*, Vol. 5. ACM. Presented at HotCarbon '25.

[37] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. 2020. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2636–2645. doi:10.1109/CVPR42600.2020.00271