Budget-Adaptive Routing: Skipping the Weak When the Strong Answers Anyway

Wei Geng
Technical University of Munich
Munich, Germany
wei.geng@tum.de

Nitinder Mohan
TU Delft
Delft, Netherlands
n.mohan@tudelft.nl

Jörg Ott
Technical University of Munich
Munich, Germany
ott@in.tum.de

Abstract

Edge-cloud inference collaborations are often designed with a routing estimator[1] that decides whether to offload each frame from weak models at the edge to stronger models in the cloud. Existing systems place the routing estimator *after* the weak detector, so the weak forward pass still runs even on frames that are later offloaded. In this paper, we argue that this *weak-conditioned* design can be suboptimal when the offload budget varies. First, we present a competitive *weak-skipping* estimator (0.153 GFLOPs, ~29× lighter than the weak detector at 4.49 GFLOPs) that extracts routing signal from raw pixels, outperforming the common after-weak placement *weak-conditioned* baselines. Second, we show that neither *weak-skipping* nor *weak-conditioned* placement dominates across the full operating curve, and we propose *budget-adaptive* routing, which selects between them by offload budget via two offline-tuned thresholds. On PASCAL VOC, our *budget-adaptive* router traces the upper accuracy envelope of both fixed placements across the operating range. Our method [2] reduces per-frame latency by up to 19.1 ms (~30% lower at ρ=0.9). Besides outperforming SOTA methods, it is surprisingly **stronger than the strong** model (+1.7 pp over the strong model's peak mAP) at some operating points with far less compute.

CCS Concepts

• **Networks** → **Cloud computing**; **Network performance modeling**; • **Computer systems organization** → **n-tier architectures**; • **Computing methodologies** → *Object detection*.

Keywords

selective offloading, budget-adaptive routing, object detection, cost-accuracy trade-off

ACM Reference Format:
Wei Geng, Nitinder Mohan, and Jörg Ott. 2026. Budget-Adaptive Routing: Skipping the Weak When the Strong Answers Anyway. In *Workshop on Networks for AI Computing (SIGCOMM '26), August 17–21, 2026, Denver, CO, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3789240.3828740

[1] In this paper, in most cases we use *estimator* and *router* interchangeably.
[2] Artifacts are available at https://github.com/ViGeng/bgt-ada

ACM ISBN 979-8-4007-2467-1/26/08
https://doi.org/10.1145/3789240.3828740

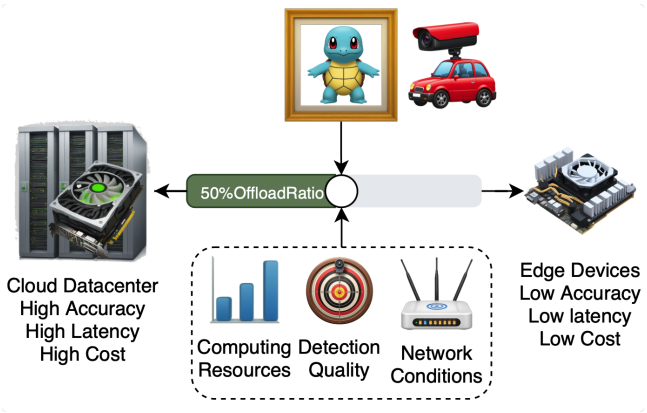

Figure 1: Selective offloading for object detection: a *local weak* detector and a *cloud strong* detector with a per-frame router under offload budget ρ.

1 Introduction

Edge devices increasingly run visual perception pipelines by offloading computation to the cloud, such as traffic cameras counting vehicles, industrial robots operating on assembly lines, and identity systems checking faces at borders. While some simply stream tasks to the edge or cloud for inference [2, 5, 8, 13, 18], others pair a *local/edge weak* detector that is fast but limited with a *strong* detector in the *cloud* that is more accurate but more costly in latency, compute, bandwidth, and energy. Selective offloading routes each frame so that the cloud complements local inference, reducing overall cost without sacrificing, and often improving, end-to-end accuracy [3, 23]. This routing decision is constrained by an offload budget that caps the fraction of frames sent to the cloud, given network conditions, compute resources, and application requirements.

A decade of selective-offloading work [3, 8, 12, 15, 23, 26, 28] runs the weak detector on every frame and feeds its output (proposal scores, top-k box statistics, learned embeddings, or intermediate activations) to a *routing estimator* that decides whether to escalate to a stronger detector in the cloud. EdgeML [23] regresses on the top-25 proposal features so that uncertain cases can be improved by strong models. DCSB [3] hand-crafts a difficult-case discriminator that directs requests to the strong model at a fixed threshold. Early-exit families such as BranchyNet [26], MSDNet [12], and SDN [15] gate *within* the weak network at intermediate layers. Despite their architectural variety, all of these methods share one structural commitment: *the weak forward pass runs on every frame, because the routing decision depends on its output or intermediate features.*

This commitment is misaligned with high-budget deployments. Safety-critical settings such as autonomous driving or security run at a high offload budget, say ≥ 30% of frames sent to the cloud, and there the weak forward pass is wasted on most frames: they are offloaded and answered by the strong model anyway, so paying for the weak pass only adds unnecessary compute and latency.

Structurally, if the estimator is moved *before* the weak model and predicts from the raw image, the serial dependency on the weak model is removed and the weak pass can be skipped for frames that will most likely be offloaded. We call this a *weak-skipping* estimator, in contrast to the *weak-conditioned* estimators of prior work (shown in Fig. 2 left and middle). To our knowledge no previous selective-offloading system has deployed it. The usual assumption is that raw-pixel features are too weak to predict detector failure. Our results challenge this: a 0.15 GFLOPs image-only *weak-skipping* estimator trained on a binary offload-utility target (Eq. (6)) is competitive with proposal-feature baselines that require the full 4.49 GFLOPs weak forward pass. We attribute this to the assumption that predicting *whether* a frame is difficult is substantially cheaper than predicting *what* it contains.

Weak-skipping is not, however, strictly better. At low offload budgets the weak forward pass is unavoidable anyway. Since the cloud rarely fires, the *weak-conditioned* estimator's richer features come essentially for free. The two placements therefore divide the operating curve into compute regimes: *weak-conditioned* and *weak-skipping* win in the low and high-budget bands, respectively, and neither dominates the curve.

Taken together, these observations argue for an *adaptive* router that selects between *weak-skipping* and *weak-conditioned* placements as a function of the offload budget, instead of one fixed placement. We confirm it empirically on PASCAL VOC from compute, latency, and accuracy perspectives. This paper contributes:

(i) We articulate the *weak-first assumption* latent in the selective-offloading literature, formalize its hidden cost as an *implicit compute tax*. (§ 2).

(ii) We introduce the *first competitive weak-skipping* estimator for object-detection offloading: a 0.15 GFLOPs image-only estimator that matches or exceeds the strongest *weak-conditioned* baselines (§ 3.1) detection quality-wise. We also build a lightweight *weak-conditioned* estimator (XGBoost on MORIC) that outperforms current *weak-conditioned* SOTAs on routing quality at sub-ms inference cost.(Tab. 2).

(iii) Building on our *weak-skipping* and *weak-conditioned* estimators, we propose *budget-adaptive* routing, which selects between *weak-skipping* and *weak-conditioned* placements according to the deployment budget. From our simulated study, our approach outperforms SOTAs. (§ 3.2, § 4).

2 Problem Statement

2.1 Problem Formulation

Let $\mathcal{I} = (i_1, \ldots, i_N)$ be a stream of image frames, M_w a weak local detector with per-frame compute cost C_w, and M_s a strong cloud detector with per-frame cost C_s (compute plus network round-trip). For each frame i_t the router produces a binary routing decision $d_t \in \{0, 1\}$ where $d_t = 0$ means "return $M_w(i_t)$" and $d_t = 1$ means

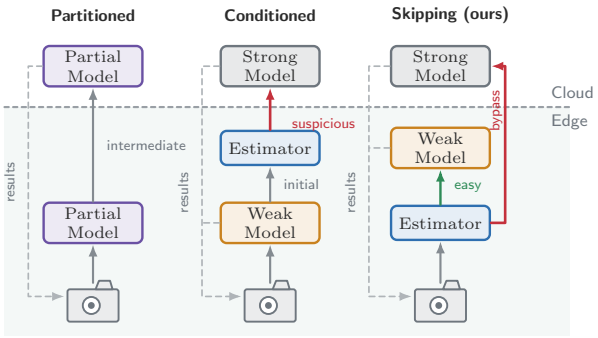

Figure 2: Routing schemes: full partitioned compute [12, 14, 15, 26] (left), *weak-conditioned* after the weak pass [3, 23] (middle), and our *weak-skipping* (right), which enables *budget-adaptive* routing. Estimator placement is the axis: it fires after the weak model (middle) or before it (right). Green marks the keep-local path, red the escalation to the cloud.

"return $M_s(i_t)$". Given a downstream detection utility $U(\cdot)$ (typically mean Average Precision, mAP), an offload budget $\rho \in (0, 1]$, and an estimator producing a per-frame score s_t, the router solves

$$\max_{d_{1:N}} \frac{1}{N}\sum_{t=1}^{N} U(i_t, M_{d_t}) \quad \text{s.t.} \quad \frac{1}{N}\sum_{t=1}^{N} d_t \le \rho. \tag{1}$$

We instantiate U as mAP@0.5, the standard PASCAL VOC metric [6], for comparability with the VOC benchmark and the *weak-conditioned* baselines we reproduce. The framework is otherwise metric-agnostic: the proxy reward ΔAP (Eq. (4)) can be computed for any U, so a stricter COCO-style AP@[.5:.95] is a drop-in substitution we leave to future work.

2.2 The Implicit Compute Tax

In every *weak-conditioned* design the weak pass executes on *every* frame, including those answered by the cloud, as an *implicit compute tax*. The expected per-frame compute is

$$T_{\text{cond}}(\rho) = C_w + C_e^{\text{cond}} + \rho \cdot C_s, \tag{2}$$

$$T_{\text{skip}}(\rho) = C_e^{\text{skip}} + (1 - \rho) \cdot C_w + \rho \cdot C_s, \tag{3}$$

where C_e^{cond} and C_e^{skip} are the estimator costs for the two placements. The tax is $\rho C_w - (C_e^{\text{skip}} - C_e^{\text{cond}})$. With our values (MobileNetV3 [11] vs ResNet50 [10], C_w=4.49, $C_e^{\text{cond}}\approx$0, C_e^{skip}=0.15, C_s=280.37 GFLOPs), the tax reaches 3.90 GFLOPs at ρ=0.9, approaching the full weak-detector cost. In addition, a *weak-conditioned* router cannot start until the weak forward pass finishes (24.70 ms here), so C_w is a serial wall-clock dependency on every frame.

The tax reflects a fallback mindset: the cloud as a backstop for the weak detector. Beside the compute and latency costs, it may also increase the jitter of the system (we leave further analysis to a follow-up work), which is undesirable for real-time applications. We don't have to pay the tax if we treat the cloud as a collaborator that complements local inference instead of a fallback, illustrated

as a branch topology in Fig. 2-right. Then the practical question is *whether an image-only lightweight estimator can catch enough routing signal without paying the tax?*

3 Method

3.1 *Weak-skipping* Estimator

Proxy metric. A *weak-skipping* estimator can only out-cheap a *weak-conditioned* one if it learns to predict *whether offloading helps*, not the contents of the frame. A naive per-frame target like ΔAP (= cloud AP − local AP), is misleading because detection AP is computed dataset-wide via a single global Precision-Recall (PR) curve, so a single frame's contribution depends on every other frame. Inspired by EdgeML [23], we use the *contextual offloading reward*: for frame i_t, the per-frame reward

$$\Delta\mathrm{AP}(i_t) \;=\; \mathrm{AP}\big(\mathrm{swap}_t \to s\big) \;-\; \mathrm{AP}\big(\text{all-weak}\big), \tag{4}$$

i.e., the change in dataset-wide AP@0.5 obtained by replacing the local detections on i_t with the cloud detections while holding all other frames at their local outputs. Computing it offline over the training set is $O(N)$ via a precomputed-IoU merge swap, and the resulting reward depends only on the dataset, not on the system at inference time. Eq. (4) produces a long-tailed signed signal that is hard to regress directly, as shown by the raw ΔAP ridge (top) of Fig. 3. We extract two estimator-friendly targets:

$$\mathrm{MORIC}^{+}(i_t) = \begin{cases} F^{+}_{\Delta\mathrm{AP}}\big(\Delta\mathrm{AP}(i_t)\big) & \Delta\mathrm{AP}(i_t) > 0, \\ 0 & \Delta\mathrm{AP}(i_t) = 0, \\ F^{-}_{\Delta\mathrm{AP}}\big(\Delta\mathrm{AP}(i_t)\big) - 1 & \Delta\mathrm{AP}(i_t) < 0, \end{cases} \tag{5}$$

$$\mathrm{OffloadBin}(i_t) = \mathbf{1}\big[\Delta\mathrm{AP}(i_t) > 0\big], \tag{6}$$

where F^{+} and F^{-} are the empirical CDFs of ΔAP over the frames where offloading strictly helps (ΔAP>0) and strictly hurts (ΔAP<0), respectively; $\mathrm{MORIC}^{+}(i_t) \in [-1, 1]$ and $\mathrm{OffloadBin}(i_t) \in \{0, 1\}$. MORIC^{+} generalises EdgeML's MORIC [23] by splitting the CDF at zero, which equalises positive and negative magnitudes and gives a symmetric loss landscape around the routing boundary. OffloadBin instead reduces the problem to a binary class label, which we train with focal loss [20] to handle the positive-class imbalance. On VOC the raw ΔAP is long-tailed with 62.6% of frames at exactly zero, 25.1% positive, and 12.3% negative (Fig. 3), so direct regression wastes capacity on the zero spike. In other words, the estimator is always lazily and blindly predicting the majority class zero, which already yields good rewards. As a result, it fails to learn the routing boundary. Both transformed targets decouple hardness from content, which is what lets raw pixels suffice. OffloadBin gives our best routing quality and MORIC^{+} is the softer-signal variant reported in the per-ratio sweep.

Architecture and calibration. A *weak-skipping* estimator $f_{\mathrm{skip}} : I \mapsto \hat{s}_{\mathrm{skip}}$ produces a routing score from the raw image alone, where $\hat{s}_{\mathrm{skip}} \in [0, 1]$ estimates $P(\Delta\mathrm{AP}{>}0)$ when trained on OffloadBin (or the predicted $\mathrm{MORIC}^{+} \in [-1, 1]$); a higher score means offloading is more likely to help (not how much to help), instantiating the generic per-frame score s_t of § 2.1. Our reference instantiation is a highly compressed MobileNetV2-Lite [25] backbone (128×128 input, 0.15 GFLOPs, 0.54 M parameters) trained on OffloadBin, though the framework is agnostic to the backbone and proxy. A thresholder π_ρ [9] then maps the score stream to binary decisions $d_t{=}\pi_\rho(\hat{s}_{\mathrm{skip}})$

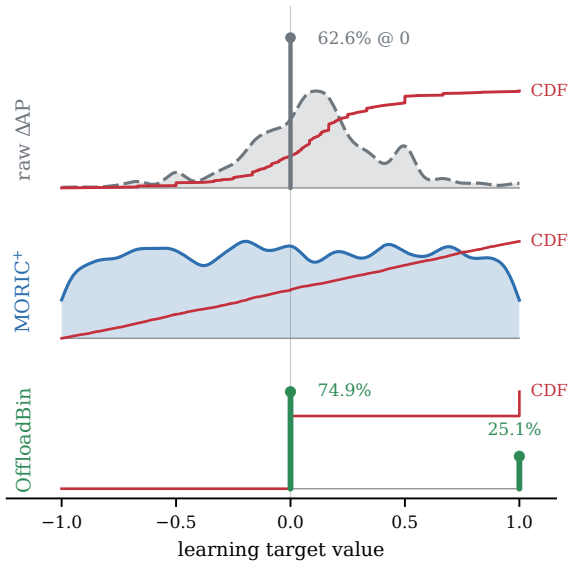

Figure 3: Per-frame learning targets on VOC test (N=3105). Top: raw ΔAP is degenerate, with 62.6% of frames exactly at 0 (stem) and short signed tails (25.1%>0, 12.3%<0). Middle: MORIC^{+} spreads this into a smooth, symmetric regression target on $[-1, 1]$. Bottom: OffloadBin ($\mathbf{1}[\Delta\mathrm{AP}{>}0]$) is the binary classification target, a ∼1:3 split (25.1% positive).

so that the offloaded fraction matches the requested budget ρ without test-set lookahead. Concretely, π_ρ keeps a running estimate of the $(1{-}\rho)$ quantile of the incoming scores and offloads any frame scoring above it, nudging the threshold as the stream drifts so the realized offload rate stays near ρ. Ratio control thus stays orthogonal to the score: any estimator emitting a calibrated per-frame score drops into the same thresholder. We leave a full treatment of the thresholder, including drift handling, finite-window error, and the resulting jitter, to follow-up work.

3.2 *Budget-adaptive* Routing

The accuracy side of the placement mirrors the compute side of § 2.2. At low budgets, *weak-skipping* wastes richer signals from the weak forward pass on most frames because most of them comes for free. At high budgets, *weak-conditioned* pays the tax on most frames because they will be offloaded anyway. Every fixed-placement router is therefore suboptimal on part of the operating curve.

Formulation. A *budget-adaptive* router holds two estimators, $f_{\mathrm{skip}} : I \mapsto \hat{s}_{\mathrm{skip}}$ and $f_{\mathrm{cond}} : (I, M_w(I)) \mapsto \hat{s}_{\mathrm{cond}}$ ($\hat{s}_{\mathrm{cond}} \in [0, 1]$, read from the image plus weak output), plus a binary arbiter $\alpha(\rho) \in \{0, 1\}$ and the same thresholder π_ρ (§ 3.1), now shared across both estimators. The per-frame decision is

$$d_t \;=\; \pi_\rho\Big(\alpha(\rho)\,\hat{s}_{\mathrm{skip}}(i_t) + (1{-}\alpha(\rho))\,\hat{s}_{\mathrm{cond}}(i_t, M_w(i_t))\Big). \tag{7}$$

$$\alpha(\rho) \;=\; \arg\max_{a \in \{0,1\}} \widehat{\mathrm{AP}}\big(\rho,\; a\,f_{\mathrm{skip}} + (1{-}a)\,f_{\mathrm{cond}}\big), \tag{8}$$

where $\widehat{\mathrm{AP}}$ is offline AP@0.5 on a held-out tuning split. Evaluated per budget, Eq. (8) makes $\alpha(\rho)$ piecewise constant with two crossovers,

splitting the operating range into three regimes:

$$\alpha(\rho) = \begin{cases} 0 & \rho < \rho_{\text{frontier}} \quad (\textit{weak-conditioned}), \\ 1 & \rho_{\text{frontier}} \le \rho < \rho_{\text{ceiling}} \quad (\textit{weak-skipping}), \\ 0 & \rho \ge \rho_{\text{ceiling}} \quad (\textit{weak-conditioned}). \end{cases} \tag{9}$$

The two thresholds are the budgets at which the winning placement flips on the tuning split. At the low crossover ρ_{frontier} the weak pass on offloaded frames turns from a near-free byproduct into pure tax, so *weak-skipping* starts to win. At the high crossover ρ_{ceiling} the fewer frames still kept local carry enough weight that the richer *weak-conditioned* signal wins again. We fit both once, offline, by sweeping ρ and reading off the two switch points (on VOC, ρ_{frontier}=0.3 and ρ_{ceiling}=0.8). At runtime the arbiter is a constant-time lookup on ρ.

4 Preliminary Evidence

We report preliminary results on PASCAL VOC [6] based on the setup in Tab. 1: the weak/strong mAP gap is small which makes routing genuinely difficult and any larger gap can make the benefits *more pronounced*. Three claims are verified:

(i) *Weak-skipping* estimators are competitive with *weak-conditioned* baselines on routing quality, and our *weak-conditioned* estimator outperforms current SOTAs (§ 4.1, Tabs. 2 and 3, Fig. 5).

(ii) The compute advantage of *weak-skipping* routing at high offload budget is real GFLOPs- and latency-wise. (§ 4.2, Fig. 4).

(iii) A *budget-adaptive* router that selects between the two placements traces the lowest cost across the operating curve and Pareto-dominates either fixed placement on the joint accuracy-compute frontier, outperforming all existing baselines (§ 4.3, Fig. 5 and Tab. 3).

Table 1: Experiment setup (PASCAL VOC).

Detectors (profiled on Nvidia A40)	
Weak M_w	`fasterrcnn_mobilenet_v3_large_fpn` [11, 19, 24]; C_w=4.49 GFLOPs / 24.70 ms; mAP = 0.760
Strong M_s	`fasterrcnn_resnet50_fpn_v2` [10, 19, 24]; C_s=280.37 GFLOPs / 44.12 ms; mAP = 0.791
Estimators (ours, § 3)	
Weak-skipping	MobileNetV2-Lite [25] on OffloadBin or MORIC$^+$
Weak-conditioned	XGBoost [4] on MORIC
Budget-adaptive	offline-tuned ρ_{frontier}, ρ_{ceiling}
Prior weak-conditioned baselines	
EdgeML [23]	proposal-level, after weak detector
DCSB [3]	fixed-rule, after weak detector
Reference points	
Trivial	always-weak, always-strong, uniformly random
Oracle	per-frame ΔAP ground-truth gain

4.1 *Weak-skipping* is Empirically Viable

Tab. 2 reports an overview across all estimators on VOC Test. Our *weak-skipping* MobileNetV2-Lite estimator trained on *OffloadBin* attains a Spearman rank correlation of 0.557 with the per-frame oracle gain, peak end-to-end mAP@0.5 of 0.808, and AUC$_\rho$ = 0.795 over the offload-budget sweep. It *outperforms* the strongest *weak-conditioned* baseline we could construct (XGBoost on MORIC: 0.472 / 0.804 / 0.794) on every metric, despite running before the weak detector and never seeing its features. Both EdgeML [23] and DCSB [3] sit well below either family. The MORIC$^+$ regression target lands between the two families and gives a useful soft signal for the smaller-trunk variant.

These results indicate that raw images are sufficient for object-detection routing. With 0.15 GFLOPs of estimator compute, we obtain a routing signal that exceeds a strong proposal-feature baseline whose signal depends on the weak model's 4.49 GFLOPs. This confirms the suspicion raised in § 1: predicting *whether* a frame is difficult is substantially cheaper than predicting *what* it contains.

Table 2: Routing quality on VOC (single-pass, seed 42, mAP at IoU 0.5). Spearman ρ_s vs. the per-frame oracle; Peak mAP and AUC$_\rho$ (area under the mAP–ρ curve) over $\rho \in [0, 1]$ with step 0.1. Green deltas on Peak mAP are relative % over always-weak (0.760). *weak-conditioned* methods additionally pay C_w=4.49 GFLOPs upstream. "–" = no continuous score.

Estimator	ρ_s	Peak mAP	AUC$_\rho$	GFLOPs
Reference points				
Always weak	0.000	0.760 0.00%=	0.760	0
Always strong	0.000	0.791 4.08%▲	0.791	0
Uniform random	0.000	–	0.777	0
Oracle (ΔAP)	–	0.827 8.82%▲	0.816	–
Weak-conditioned (after weak)				
EdgeML [23]	0.138	0.793 4.34%▲	0.784	≈ 0
DCSB [3]	0.359	0.789 3.82%▲	–	≈ 0
XGBoost [4] / MORIC *(Ours)*	0.472	0.804 5.79%▲	0.794	≈ 0
Weak-skipping (before weak)				
MV2-Lite + MORIC$^+$ *(Ours)*	0.350	0.801 5.39%▲	0.790	0.15
MV2-Lite + OffloadBin *(Ours)*	**0.557**	**0.808** 6.32%▲	**0.795**	0.15
Budget-adaptive (ρ_{frontier}=0.3, ρ_{ceiling}=0.8)				
OffloadBin ↔ MORIC *(Ours)*	–‡	**0.808** 6.32%▲	**0.796**	0.15†

† Adaptive uses f_{cond} for $\rho \le 0.2$ and $\rho \ge 0.8$, f_{skip} for $0.3 \le \rho \le 0.7$ (offline arbitration, Eq. (8)). Reported GFLOPs are the skipping-branch cost; the *weak-conditioned* branch additionally pays C_w.
‡ Spearman is undefined for an arbiter that selects a different score per ρ; both component scores' ρ_s are reported above.

4.2 The Compute Advantage is Material

Fig. 4 decomposes the expected per-frame cost for *weak-conditioned* (C) and *weak-skipping* (S) at each budget ρ into three stacked components: estimator C_e (blue), weak detector (orange), and cloud ρC_s (gray). The cloud band dominates both placements equally (ρC_s), so the saving comes entirely from the device side: in the S bars the orange weak-detector band shrinks with $(1-\rho)C_w$ instead of the constant C_w paid in C, at the cost of adding a thin blue estimator band (C_e=0.15 GFLOPs).

Compute (panel a). The GFLOP saving is monotone in ρ: from 0.3 GFLOPs at ρ=0.1 through 1.6 at ρ=0.4 and 2.5 at ρ=0.6, to 3.9 at ρ=0.9, approaching one full weak pass (4.49 GFLOPs) as $\rho \to 1$. The breakeven is very low, $\rho^* = C_e/C_w \approx 0.03$: *weak-skipping* is cheaper for essentially all operating budgets, since the estimator pays for itself once it avoids more than one weak pass in ~30 frames.

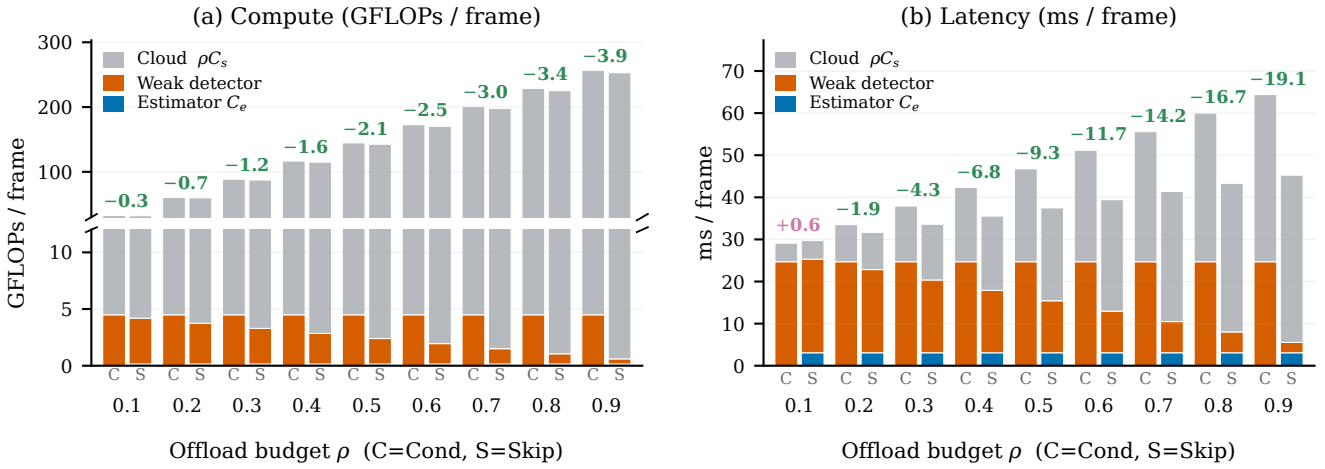

Figure 4: Expected per-frame cost vs. offload budget ρ on VOC. Each bar stacks estimator C_e (blue), weak detector (orange), and cloud ρC_s (gray). Paired bars are C (*weak-conditioned*, weak on every frame) and S (*weak-skipping*, weak on only the $(1-\rho)$ kept frames); each pair's label is the cond−skip saving (pink when *weak-skipping* costs more). (a) Compute (GFLOPs, broken y-axis; C_e too small to see). (b) Latency (ms), which adds the serial weak-pass dependency.

Latency (panel b). The wall-clock picture differs because it captures a *serial* dependency. A *weak-conditioned* router cannot produce a score until the weak forward pass finishes (24.70 ms), whereas a *weak-skipping* router runs f_{skip} (3.08 ms) in its place on offloaded frames. At ρ=0.1 *weak-skipping* is 0.6 ms *slower* (pink label) because the estimator overhead slightly exceeds the small saving from skipping only 10% of weak passes. The crossover sits at $\rho^*_{\text{ms}}\approx$0.12. Beyond it the saving grows near-linearly, up to ~30% end-to-end latency reduction. For latency-critical deployments with higher ρ , this saving alone justifies the *weak-skipping* placement.

4.3 *Budget-adaptive* Achieves Upper Envelope

(1) Neither fixed placement dominates the operating curve. Fig. 5 shows *weak-conditioned* (XGBoost on MORIC) mAP leads at low and very high budgets ($\rho\leq$0.2 and $\rho\geq$0.8, by 0.1–0.4 pp), while *weak-skipping* (MV2-Lite + OffloadBin) dominates in the mid-budget regime ($\rho\in\{0.3,\ldots,0.7\}$). Tab. 3 shows that two crossovers bracket the skipping band: at $\rho_{\text{frontier}}\approx$0.3 the weak forward pass on offloaded frames becomes a pure tax and skipping it yields both compute savings and better frame selection; at $\rho_{\text{ceiling}}\approx$0.8 the few remaining local frames carry enough weight that the richer *weak-conditioned* signal again dominates the per-frame compute saved by skipping, also shown in Fig. 5 frame selection quality-wise.

(2) Budget-adaptive traces the upper envelope of both placements and strictly dominates all baselines. A two-threshold offline arbiter (ρ_{frontier}=0.3, ρ_{ceiling}=0.8) produces the per-budget maximum (last row of Tab. 3, teal curve in Fig. 5): it picks *weak-conditioned* at $\rho\leq$0.2 and $\rho\geq$0.8, *weak-skipping* at $0.3\leq\rho\leq0.7$. This envelope achieves the **highest** peak mAP@0.5 of any router, 0.808, which is +1.7 pp over the strong model itself, and the highest AUC$_\rho$=0.796 (Tab. 2). Compared with prior *weak-conditioned* SOTAs, the *budget-adaptive* system leads EdgeML [23] by 0.7–2.1 pp across the full budget range (the strong model itself leads the weak model by only 3.1 pp). The

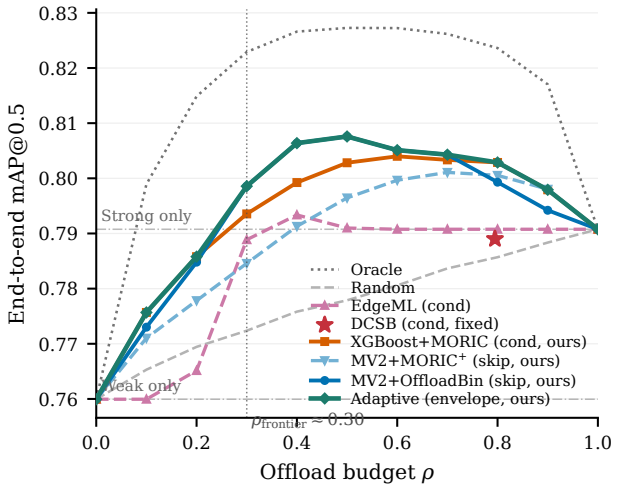

Figure 5: End-to-end mAP@0.5 vs. offload budget ρ on VOC. *budget-adaptive* (teal) is the pointwise upper hull of *weak-skipping* (blue) and *weak-conditioned* (orange), with crossovers at $\rho_{\text{frontier}}\approx$0.3, $\rho_{\text{ceiling}}\approx$0.8.

widest gap is at ρ=0.2 (0.786 vs. 0.765) and the narrowest is at ρ=0.9 (0.798 vs. 0.791) where EdgeML's native threshold saturates to always-offload. DCSB [3] is a fixed binary rule locked to a single operating point ($\rho\approx$0.79, mAP= 0.789). At the same budget the *budget-adaptive* router reaches 0.803, and its peak (0.808) exceeds DCSB by 1.9 pp while offering continuous budget tunability. The envelope sits 1.9 ~ 2.9 pp below the offline oracle, bounding what routing quality alone can recover.

Table 3: End-to-end mAP@0.5 on VOC across offload budgets $\rho \in \{0.1, \ldots, 0.9\}$ (single seed). Each cell is the accuracy at that ρ. ▲ beats and • matches the always-strong model (0.791), an unmarked cell is below it. Bold is the column best. On the *budget-adaptive* row a superscript marks the selected branch (S = *weak-skipping*, C = *weak-conditioned*).

ρ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Reference									
Weak only (constant)	0.760	0.760	0.760	0.760	0.760	0.760	0.760	0.760	0.760
Strong only (constant)	0.791	0.791	0.791	0.791	0.791	0.791	0.791	0.791	0.791
Random offloader	0.765	0.769	0.772	0.776	0.778	0.781	0.784	0.786	0.788
Oracle (ΔAP)	0.799 ▲	0.815 ▲	0.823 ▲	0.827 ▲	0.827 ▲	0.827 ▲	0.826 ▲	0.824 ▲	0.817 ▲
Weak-conditioned									
EdgeML [23]	0.760	0.765	0.789	0.793 ▲	0.791 •	0.791 •	0.791 •	0.791 •	0.791 •
XGBoost on MORIC *(Ours)*	**0.776**	**0.786**	0.794 ▲	0.799 ▲	0.803 ▲	0.804 ▲	0.803 ▲	**0.803** ▲	**0.798** ▲
Weak-skipping									
MV2 + MORIC^{+} *(Ours)*	0.771	0.778	0.785	0.791 •	0.796 ▲	0.800 ▲	0.801 ▲	0.801 ▲	**0.798** ▲
MV2 + OffloadBin *(Ours)*	0.773	0.785	**0.799** ▲	**0.806** ▲	**0.808** ▲	**0.805** ▲	**0.804** ▲	0.799 ▲	0.794 ▲
Budget-adaptive (regime in superscript)									
MV2/OffloadBin ↔ XGB/MORIC *(Ours)*	**0.776**^{C}	**0.786**^{C}	**0.799**^{S} ▲	**0.806**^{S} ▲	**0.808**^{S} ▲	**0.805**^{S} ▲	**0.804**^{S} ▲	**0.803**^{C} ▲	**0.798**^{C} ▲

(3) The dominance extends to the joint accuracy-compute frontier inside the skipping band. For $0.3 \leq \rho \leq 0.7$, *budget-adaptive* inherits *weak-skipping*'s compute profile: 2.1 GFLOPs/frame less than any *weak-conditioned* method at ρ=0.5 and 3.0 GFLOPs/frame less at ρ=0.7 (the top of the band; Fig. 4a), with wall-clock savings reaching 14.2 ms (~26% reduction) at ρ=0.7 (Fig. 4b). At $\rho < \rho_{\text{frontier}}$ the weak pass runs on nearly all frames regardless, so switching to *weak-conditioned* costs no incremental compute. At $\rho \geq \rho_{\text{ceiling}}$ *budget-adaptive* has the option to trade the skipping compute saving for the 0.4 pp accuracy lift of *weak-conditioned*. Inside the skipping band the *budget-adaptive* router is therefore never worse in accuracy *and* never worse in compute than either fixed placement alone.

5 Related Work

Selective-offloading admits a clean taxonomy along one architectural axis: *when does the estimator fire, relative to the weak model?*

***Weak-conditioned* estimators.** Shown in Fig. 2-middle, the estimator fires *after* the weak model and consumes its outputs. EdgeML [23] regresses on the top-25 proposal features and DCSB [3] hand-crafts a difficult-case discriminator. Confidence thresholds and learned detection embeddings [28] also belong to this class. They exploit rich signals but pay *implicit compute tax* unconditionally.

Mid-network gates and early exit. This is essentially *weak-conditioned* estimators. BranchyNet [26], MSDNet [12], and SDN [15] fuse the estimator into the weak network and gate at intermediate layers. They reduce average weak-model cost on easy frames but do not skip the "Early Pass" entirely. Neurosurgeon [14] and SPINN [16, 17, 22] partition or progressively split a single network across device and cloud, which is related but distinct.

Cascaded inference. Classical cascades [27] and modern cascade detectors [1] escalate on uncertainty. As detailed in § 3.2, they use the cheap stage to *produce predictions* whereas *budget-adaptive* routing uses it to *route*.

***Weak-skipping* estimators.** The estimator fires *before* the weak model from the raw image. Early image-complexity heuristics and learned image-level routers in our work belong to this class. They allow the weak pass to be skipped entirely on offloaded frames but have often been considered too weak for detection routing, a premise challenged by our results.

6 Conclusion and Future Work

Estimator placement is a design axis the selective-offloading literature has implicitly fixed. We exhibited both a lightweight image-only *weak-skipping* estimator and a *weak-conditioned* estimator that outperform the SOTAs that we could reproduce from the literature, on both routing quality and compute. We showed the placement choice is regime-dependent and built a budget-aware *budget-adaptive* router that traces the upper accuracy envelope of both placements on VOC while saving compute and latency within its skipping band.

Several directions remain. A **real edge-cloud testbed** would replace our offline simulation with measured round-trips [7] on real-world hardwares. **Testing beyond VOC** such as COCO[21] would probe how far the raw-pixel routing signal carries. Beyond mAP@0.5, stricter AP@[.5:.95], class-weighted or downstream-task utility will be experimented. A **skip-then-reconsider** cascade would run the *weak-skipping* estimator first and, on frames kept local where the weak pass executes anyway, apply a *weak-conditioned* estimator to revisit the offload decision on the now-available weak features at no extra detector cost. A **head-to-head with early-exit routers** across weak-model prefix depths would chart when a shallow prefix already rivals the image-only *weak-skipping* estimator.

Acknowledgments

This work was supported by the Dutch National Growth Fund "Future Network Services".

References

[1] Zhaowei Cai and Nuno Vasconcelos. 2018. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6154–6162.

[2] Christopher Canel, Thomas Kim, Giulio Zhou, Conglong Li, Hyeontaek Lim, David G Andersen, Michael Kaminsky, and Subramanya Dulloor. 2019. Scaling Video Analytics on Constrained Edge Nodes. In *Proceedings of Machine Learning and Systems*, A. Talwalkar, V. Smith, and M. Zaharia (Eds.), Vol. 1. 406–417. https://proceedings.mlsys.org/paper_files/paper/2019/file/6bcfac823d40046dca25ef6d6d59cc3f-Paper.pdf

[3] Zhiqiang Cao, Zhijun Li, Yongrui Chen, Heng Pan, Youbing Hu, and Jie Liu. 2023. Edge-cloud collaborated object detection via difficult-case discriminator. In *2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 259–270.

[4] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[5] Tiffany Yu-Han Chen, Lenin Ravindranath, Shuo Deng, Paramvir Bahl, and Hari Balakrishnan. 2015. Glimpse: Continuous, real-time object recognition on mobile devices. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*. 155–168.

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision* 88, 2 (2010), 303–338.

[7] Wei Geng, Oguz Kagan Altas, David Guzman, Giovanni Bartolomeo, Nitinder Mohan, and Joerg Ott. 2025. Poster: KUT: Towards Lightweight On-path Network Assessment for Edge Orchestration. In *Proceedings of the 21st International Conference on emerging Networking EXperiments and Technologies*. 9–11.

[8] Wei Geng, Xiang Su, Nitinder Mohan, Jörg Ott, and Pan Hui. 2026. SMOOTH: Scalable Multitask Offloading with Backbone Sharing. In *2026 IFIP Networking Conference (IFIP Networking)*. IFIP, 1–10.

[9] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR, 1321–1330.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[11] Andrew Howard, Mark Sandler, Bo Chen, et al. 2019. Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[12] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844* (2017).

[13] Junchen Jiang, Ganesh Ananthanarayanan, Peter Bodik, Siddhartha Sen, and Ion Stoica. 2018. Chameleon: scalable adaptation of video analytics. In *Proceedings of the 2018 conference of the ACM special interest group on data communication*. 253–266.

[14] Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang. 2017. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *ACM SIGARCH Computer Architecture News* 45, 1 (2017), 615–629.

[15] Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. 2019. Shallow-deep networks: Understanding and mitigating network overthinking. In *International conference on machine learning*. PMLR, 3301–3310.

[16] Stefanos Laskaridis, Stylianos I Venieris, Mario Almeida, Ilias Leontiadis, and Nicholas D Lane. 2020. SPINN: Synergistic progressive inference of neural networks over device and cloud. In *Proceedings of the 26th annual international conference on mobile computing and networking*. 1–15.

[17] En Li, Zhi Zhou, and Xu Chen. 2018. Edge intelligence: On-demand deep learning model co-inference with device-edge synergy. In *Proceedings of the 2018 workshop on mobile edge communications*. 31–36.

[18] 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 on the applications, technologies, architectures, and protocols for computer communication*. 359–376.

[19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2117–2125.

[20] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[22] Yoshitomo Matsubara, Marco Levorato, and Francesco Restuccia. 2022. Split computing and early exiting for deep learning applications: Survey and research challenges. *Comput. Surveys* 55, 5 (2022), 1–30.

[23] Jiaming Qiu, Ruiqi Wang, Brooks Hu, Roch Guérin, and Chenyang Lu. 2024. Optimizing edge offloading decisions for object detection. In *2024 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 164–177.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[25] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4510–4520.

[26] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. 2016. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*. IEEE, 2464–2469.

[27] Paul Viola and Michael Jones. 2001. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, Vol. 1. Ieee, I–I.

[28] Qingyuan Wang, Barry Cardiff, Antoine Frappé, Benoit Larras, and Deepu John. 2024. Tiny models are the computational saver for large models. In *European Conference on Computer Vision*. Springer, 163–182.

A Supplementary Evidence

All numbers below are from the same single-pass VOC evaluation (seed 42) used in § 4, profiled on the setup of Tab. 1. Peak mAP is the maximum end-to-end accuracy over the offload-budget ρ sweep. mAP@0.5 is the VOC metric and AP@[.5:.95] (COCO-style; written AP_C in table headers) is the stricter metric. These appendices backs and justifies three design choices made in § 3 (the learning target, the estimator backbone, and the claim that routing signal is recoverable from the raw image) and probe robustness to a stricter accuracy metric.

A.1 Backbone Ablation

Table 4: Backbone ablation for the *weak-skipping* estimator, controlling for the learning target: both trunks use the identical OffloadBin/focal target and differ only in the backbone (VOC test, seed 42). GFLOPs and parameters are for the estimator alone. Trunks we trained on other targets are not directly comparable and are omitted.

Backbone (OffloadBin/focal)	GFLOPs	Par. (M)	ρ_s	Peak mAP
MobileNetV2-Lite	0.153	0.54	**0.557**	**0.808**
EfficientNet-B0-Lite	0.176	0.85	0.235	0.795

Our backbone sweep is small, so we report it only briefly. Tab. 4 holds the learning target fixed at OffloadBin/focal and varies the trunk alone: the larger EfficientNet-B0-Lite does not improve routing over the compact MobileNetV2-Lite, with a markedly lower rank correlation and no gain in peak mAP. This is consistent with the premise of § 1 that detecting difficulty needs little capacity, though a broader sweep is left to future work.

A.2 Learning-Target Ablation

§ 3.1 reduces the long-tailed per-frame ΔAP to a binary *OffloadBin* label trained with focal loss, rather than regressing a continuous reward. Because the backbone and the target must be chosen jointly, a chicken-and-egg dependency, Tab. 5 fixes the backbone first and sweeps the target. OffloadBin wins by a wide margin (ρ_s=0.557,

Table 5: Learning-target ablation on the fixed MobileNetV2-Lite *weak-skipping* backbone (VOC test, seed 42). ρ_s is the Spearman correlation between the estimator score and the per-frame oracle gain (not the offload budget ρ), and ratio-err is the mean absolute gap between realized and requested offload fractions. Exact target definitions are in our released artifacts. Rows are sorted by ρ_s within each group, best per column in bold.

Target / loss	Description	ρ_s	Peak mAP	Peak AP_C	ratio-err
Classification target (focal)					
OffloadBin	Binary $\mathbf{1}[\Delta AP>0]$: does the cloud strictly help this frame? Focal loss handles the positive-class imbalance.	**0.557**	**0.808**	**0.609**	0.009
TopQuartile	Binary: is the frame's gain in the top quartile of ΔAP? A rarer, harder positive class than OffloadBin.	0.389	0.800	0.606	0.007
Regression target					
$MORIC^+$	Signed empirical CDF of ΔAP, split at zero onto $[-1, 1]$; this is our reward (Eq. (5)).	0.350	0.801	0.607	0.018
HighIoUGain	Continuous per-frame gain proxy that up-weights tightly-localised, high-IoU matches.	0.345	0.794	0.607	0.008
$MORIC^+$ (quantile)	The $MORIC^+$ target, fit with a quantile (pinball) regression loss.	0.313	0.801	0.608	0.010
F1Gain	Continuous per-frame proxy: the change in detection F1 when the frame is offloaded.	0.306	0.791	0.605	0.006
$MORIC^+$ (wing)	The $MORIC^+$ target, fit with a wing regression loss.	0.303	0.799	0.606	0.024
RescueRatio	Continuous proxy for the share of a frame's missed objects that the cloud recovers.	0.291	0.792	0.605	0.004
WorstCaseGain	Continuous proxy that emphasises the frame's worst-case detection loss.	0.281	0.796	0.607	0.010
SigMORIC	The $MORIC^+$ reward passed through a sigmoid squashing.	0.273	0.797	0.605	0.015
MORIC★	The $MORIC^+$ reward under an alternative CDF reshaping.	0.271	0.797	0.605	0.015
Φ-MORIC	The $MORIC^+$ reward under a Φ-based CDF reshaping.	0.263	0.797	0.605	0.015
RescueRatio (wing)	The RescueRatio target, fit with a wing regression loss.	0.169	0.791	0.605	0.011

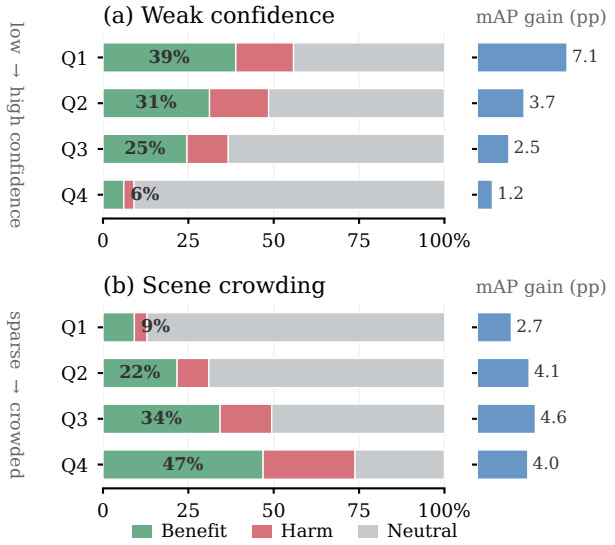

Figure 6: Where offloading helps, by weak-detector stratum (VOC test, seed 42, N=3105). Each bar splits a quartile's frames into Benefit (offloading raises ΔAP), Harm (lowers it), and Neutral (unchanged). The dominant neutral mass shows the benefit is a sparse partition. (a) Across weak-confidence quartiles benefit falls 6.4× (Q1→Q4). (b) Across scene crowding (weak detection count) it rises 5.1×. The right strip is the per-frame weak→strong mAP gain in points (pp), largest where benefit concentrates.

against 0.389 for the next-best target and 0.169 ∼ 0.350 for the continuous-regression variants) and tops both accuracy metrics. Budget tracking is uniformly tight, so this gap reflects the target itself rather than calibration. The result confirms the § 3.1 argument: collapsing *whether offloading helps* into a binary label decouples hardness from content and lets raw pixels suffice, whereas regressing the signed magnitude wastes capacity on the 62.6% zero spike (Fig. 3).

A.3 Where Offloading Helps

A *weak-skipping* estimator presumes that *which frames benefit from the cloud* is both structured and recoverable from the raw image. Fig. 6 speaks to the first half: the benefit is sharply concentrated, with the lowest weak-confidence quartile offload-beneficial 6.4× more often than the highest (38.9% vs. 6.1%) and the most crowded scenes 5.1× more often than the sparsest (46.9% vs. 9.2%), tabulated in full in Tab. 6. The same strata carry the largest weak→strong mAP headroom. These strata are defined by weak-detector outputs, however, so these views *localise* where offloading helps rather than showing the signal is recoverable *before* the weak pass. We assume that low confidence and crowding are correlates of scene complexity

Table 6: Offload benefit broken down by weak-detector stratum on the VOC test set (seed 42, N=3105 frames). We split the frames into quartiles Q1–Q4 of each weak-detector summary, so that Q1 contains the least-confident or sparsest scenes and Q4 the most-confident or most-crowded. The Benefit, Harm, and Neutral columns give the fraction of frames in each stratum for which offloading respectively raises, lowers, or leaves ΔAP unchanged, and they sum to one. The final two columns, mAP_w and mAP_s, report the mean per-frame mAP of the weak and strong detectors within the stratum.

Stratum	Q	Benefit	Harm	Neutral	mAP_w	mAP_s
Weak conf. (mean)	Q1	0.389	0.169	0.443	0.750	0.821
	Q2	0.312	0.173	0.515	0.847	0.884
	Q3	0.246	0.121	0.633	0.895	0.920
	Q4	0.061	0.030	0.911	0.961	0.973
Weak det. count	Q1	0.092	0.037	0.872	0.928	0.955
	Q2	0.217	0.093	0.690	0.864	0.905
	Q3	0.343	0.151	0.507	0.831	0.877
	Q4	0.469	0.269	0.262	0.784	0.824

that is plausibly legible from raw pixels and is established directly by the *weak-skipping* estimator's measured routing quality (Tab. 5).

The benefit structure dictates the target. Tab. 5 and Fig. 6 are two views of one phenomenon. The offload benefit is a sparse *partition*, not a smooth magnitude field: a dominant neutral mass (62.6% of frames at ΔAP=0; median gain 0 in *every* stratum) around a minority of beneficial frames (≤47% even at best) in low-confidence, crowded scenes. The learnable question is thus *whether* a frame lies in that region, not *by how much* it gains, which is why Tab. 5 ranks both classification targets (OffloadBin ρ_s=0.557, TopQuartile 0.389) above every regression variant ($\rho_s \leq 0.350$), and why routing reduces to recognising a region of image space that a cheap raw-image trunk can read.