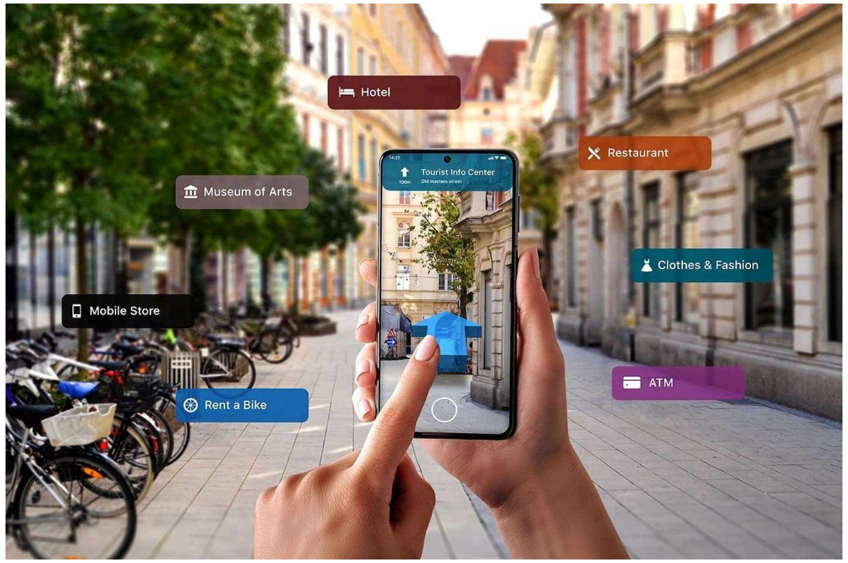
Online Resource Allocation for Edge Intelligence with Colocated Model Retraining and Inference

Huaiguang Cai, Zhi Zhou, Qianyi Huang

Sun Yat-Sen University

Presenter: Zhiwei Zhai

The Killer App for Edge Computing: Video Analytics[1]



Self-driving and smart cars — Surveillance and security — Augmented reality

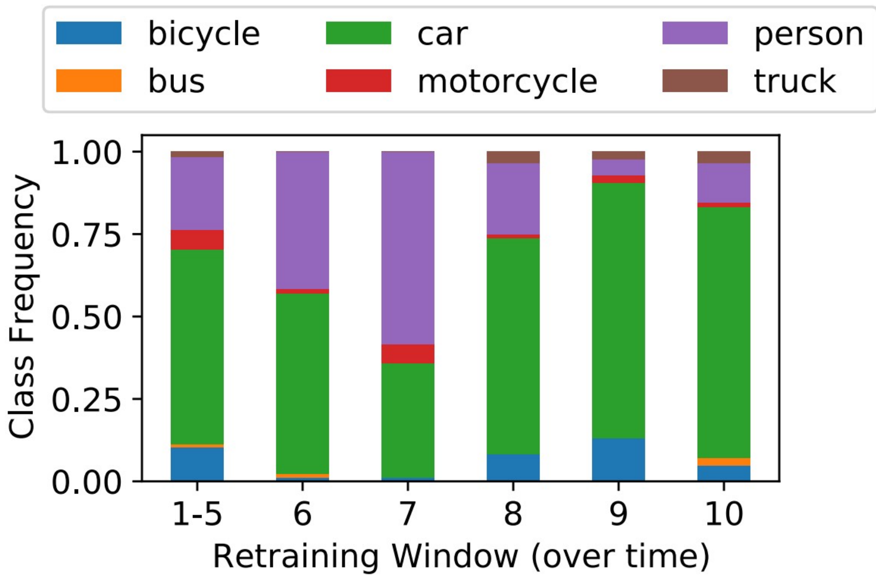
Potential benefits of edge computing for video analytics:
Providing low-latency, energy-efficient, and privacy-protecting services to users.

Credit: Google images

[1] "Real-Time Video Analytics: The Killer App for Edge Computing", in Computer, 2017.

The Model's Accuracy Suffers from Various Drifts

- Data drift: A shift in the distribution of features or labels.



Example: Class Distribution Shifts[2]

- Model drift: Compressed models have less generalization ability compared to the original models.
- Task drift: The deployed model may be applied to perform unseen tasks (e.g., fine tuning, transfer learning, embodied AI).

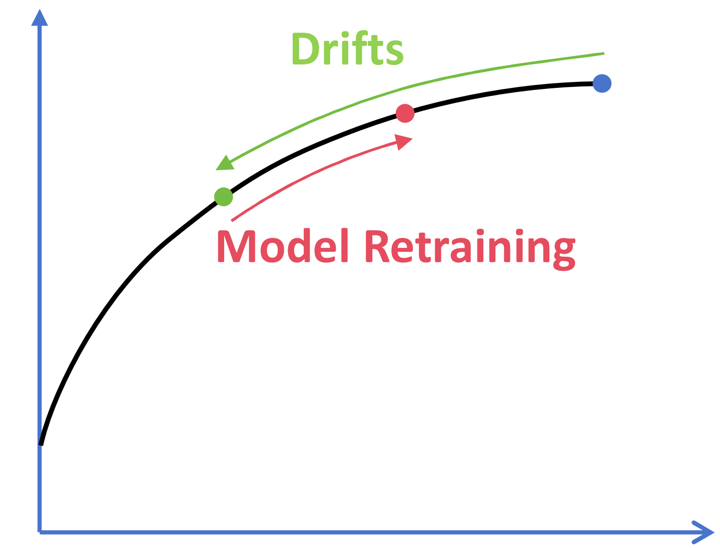
What can we do?
Retrain the model!

[2] “Ekya: Continuous Learning of Video Analytics Models on Edge Compute Servers”, in NSDI, 2022.

Model Retraining Can Handle Drifts

- Retraining configuration adaption
- Inference configuration adaption

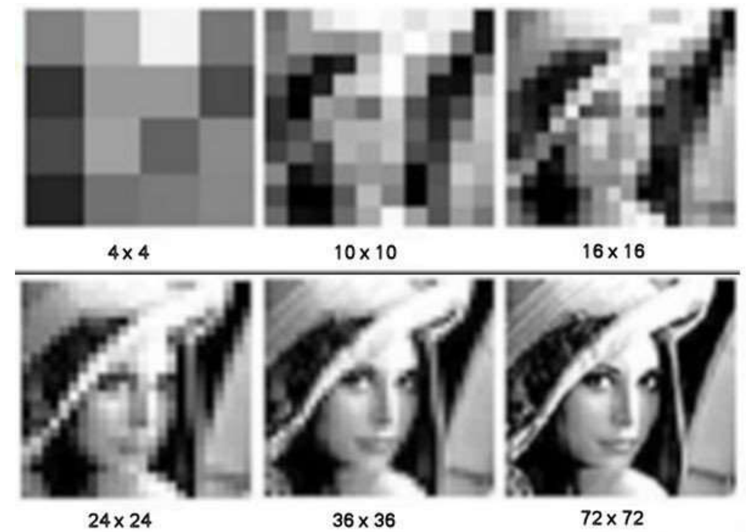
$f(x)$: Validation accuracy



x: Epochs, training data size, etc.

Where do the additional computing resources for model retraining come from?

Downgrade the inference configuration!

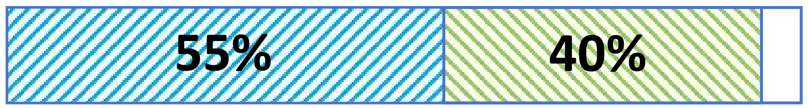


Example: Lower input resolution leads to reduced inference accuracy and resource consumption.

Credit: Google images

[3] "Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves", in IJCAI, 2015.

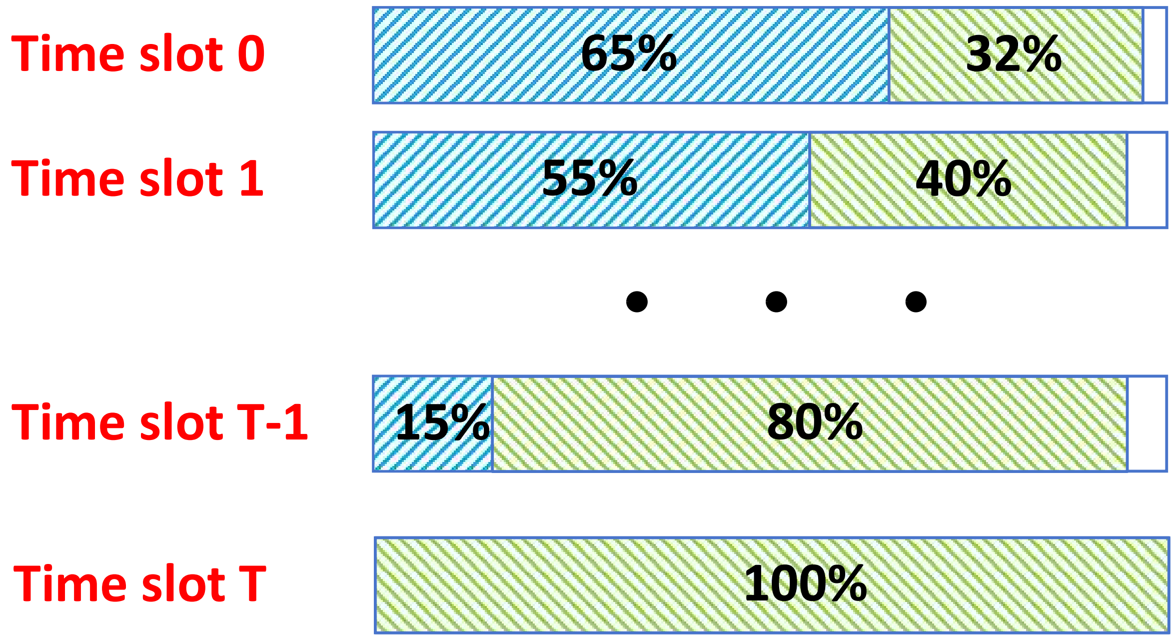
Retraining vs. Inference: Competitive Dynamics



Resource allocation on edge.

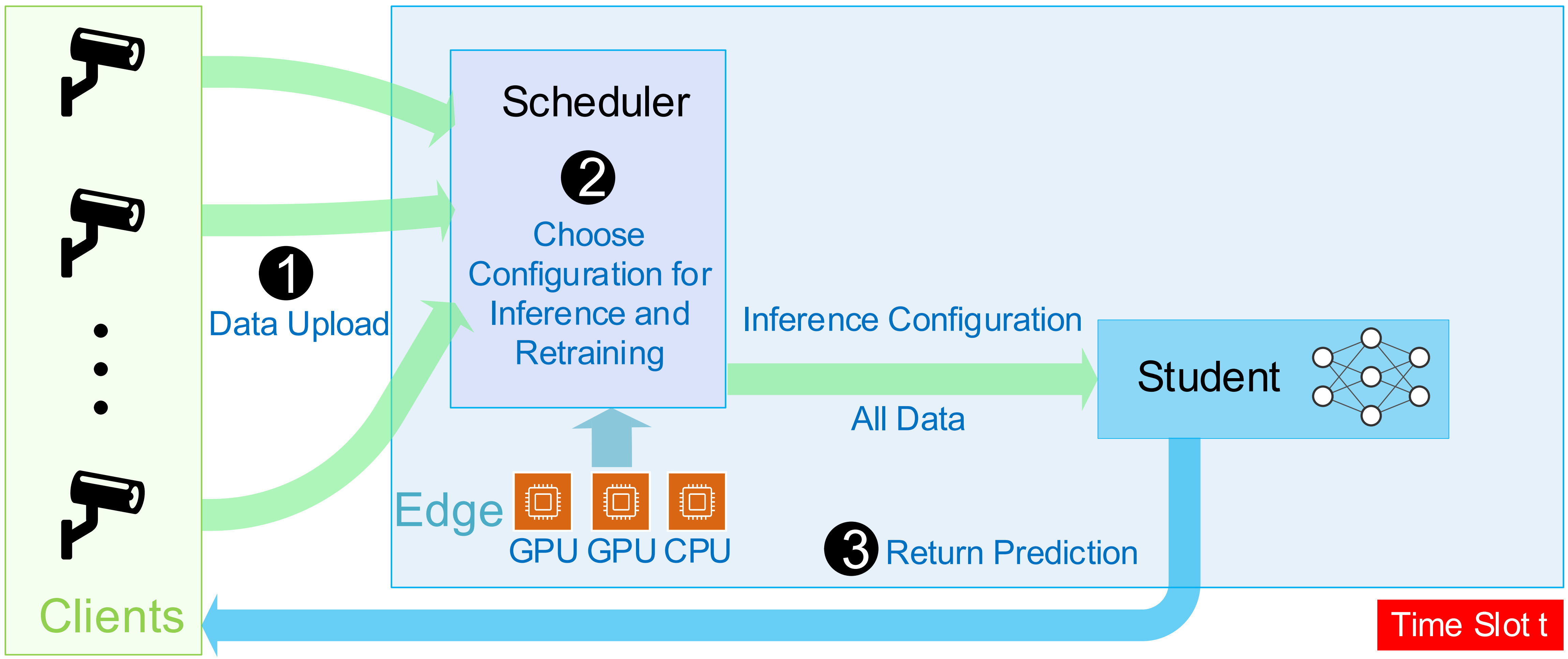
Computing resources allocated to model **retraining**.

Computing resources allocated to model **Inference**.

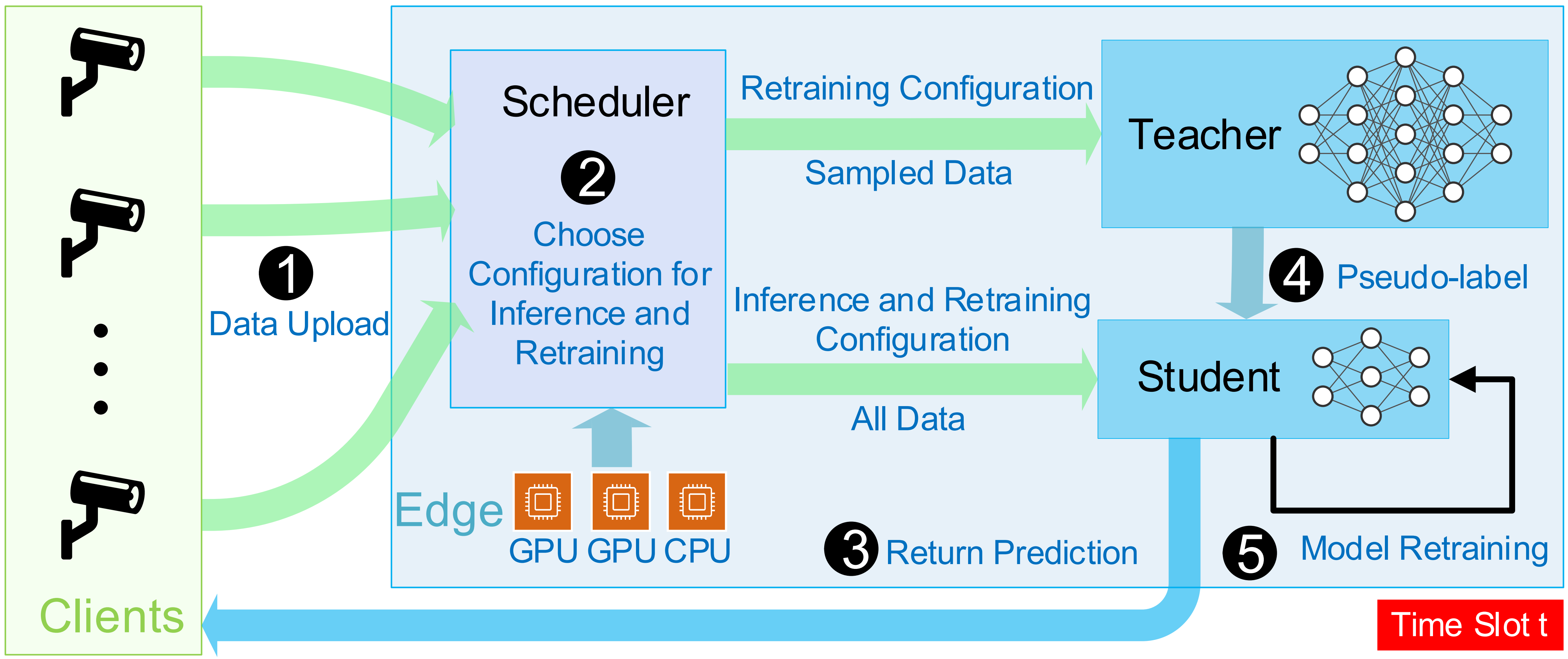


Example: A typical resource allocation process for model retraining and inference across T time slots.

Model Retraining and Inference Co-location Paradigm



Model Retraining and Inference Co-location Paradigm



Summary Thus Far

AI models are increasingly pushed to the edge to serve users.

↓

The model's accuracy suffers from various drifts.

↓

Model retraining can handle drifts.

↓

Competitive relationship between model retraining and inference.

Summary Thus Far

Central question:

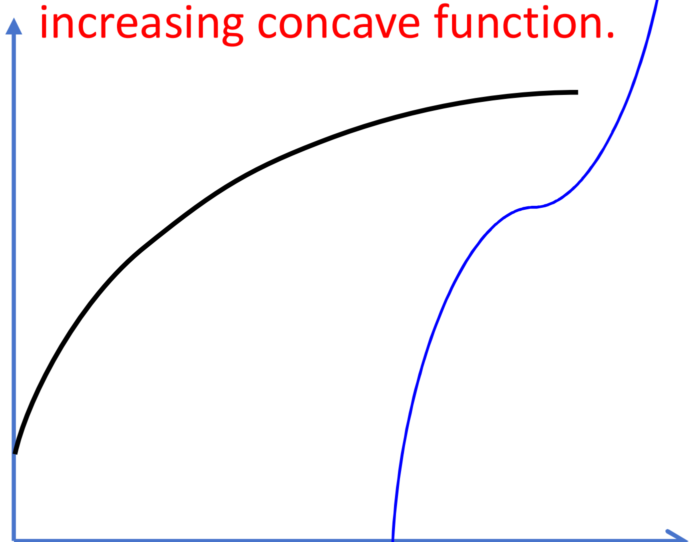
How can resources be credibly allocated for model retraining and inference co-location to optimize long-term model performance under various drifts?

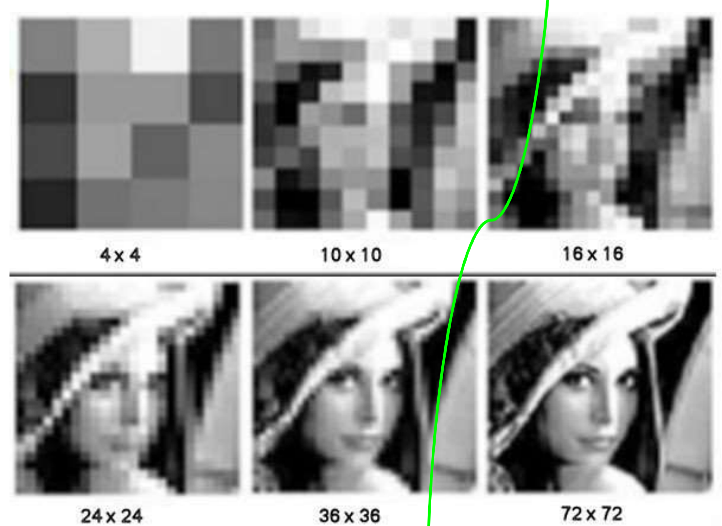
Long-term Accuracy Model and Resource Allocation Model

Objective : Optimize long-term accuracy.

$$\max_{x_i(t),y_j(t)} \sum_{t=1}^{T} f\left(\sum_{\tau=1}^{t-1} D_{(\tau)} \sum_{i=1}^{M} x_i(\tau) A_i^T \Big/ \sum_{\tau=1}^{t-1} D_{(\tau)}\right) \sum_{j=1}^{N} y_j(t) A_j^I D_{(t)}$$

$f(x)$: Validation accuracy, increasing concave function.



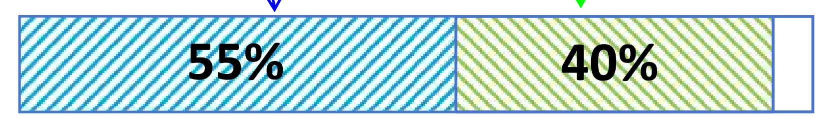


x: Average retraining configuration (such as sample ratio) before time slot t.

y: Inference configuration (such as resolution) at time slot t.

Constraint (1): Limited resource on edge.

$$D_{(t)}\sum_{i=1}^{M} C_i^T x_i(t) + D_{(t)}\sum_{j=1}^{N} C_j^I y_j(t) \le C_{(t)}, \ \forall t \in \mathrm{T}.$$



Resource allocation on edge.

Constraint (2-4): Each time slot, select only one retraining and inference configuration.

$$x_i(t) \in \{0,1\}, \quad \forall i \in \mathrm{M}, \forall t \in \mathrm{T},$$
$$y_j(t) \in \{0,1\}, \quad \forall j \in \mathrm{N}, \forall t \in \mathrm{T},$$
$$\sum_{i=1}^{M} x_i(t) = 1, \quad \forall t \in \mathrm{T},$$
$$\sum_{j=1}^{N} y_j(t) = 1, \quad \forall t \in \mathrm{T}.$$

Challenges of the Original Problem

$$\max_{x_i(t), y_j(t)} \sum_{t=1}^{T} f\left(\sum_{\tau=1}^{t-1} D_{(\tau)} \sum_{i=1}^{M} x_i(\tau) A_i^T / \sum_{\tau=1}^{t-1} D_{(\tau)}\right) \sum_{j=1}^{N} y_j(t) A_j^I D_{(t)} \qquad \text{(P)}$$

s.t. $D_{(t)} \sum_{i=1}^{M} C_i^T x_i(t) + D_{(t)} \sum_{j=1}^{N} C_j^I y_j(t) \le C_{(t)}, \ \forall t \in \mathrm{T}.$

$$x_i(t) \in \{0,1\}, \quad \forall i \in \mathrm{M}, \forall t \in \mathrm{T},$$

$$y_j(t) \in \{0,1\}, \quad \forall j \in \mathrm{N}, \forall t \in \mathrm{T},$$

$$\sum_{i=1}^{M} x_i(t) = 1, \quad \forall t \in \mathrm{T},$$

$$\sum_{j=1}^{N} y_j(t) = 1, \quad \forall t \in \mathrm{T}.$$

Challenges:

1. Time-coupled decision making.
2. Non-convex objective function.
3. Problem (P) is integer programming problem, NP-hard.
4. Analytical formula for *f* is commonly unknown in practice.

Our Solution

(P)

$$\max_{x_i(t),y_j(t)} \sum_{t=1}^{T} f\left(\sum_{\tau=1}^{t-1} D_{(\tau)} \sum_{i=1}^{M} x_i(\tau) A_i^T / \sum_{\tau=1}^{t-1} D_{(\tau)}\right) \sum_{j=1}^{N} y_j(t) A_j^I D_{(t)}$$

relaxation

$$\max_{x_i(t),y_j(t)} V_t \sum_{i=1}^{M} x_i(t) A_i^T + W_t \sum_{j=1}^{N} y_j(t) A_j^I$$

(Dt)

$$where\ V_t = L \frac{D_{\min} A_{\min}^{I}}{D_{\max}} \left(\sum_{\tau=t}^{T-1} \frac{1}{\tau} \right),$$

$$W_1 = f\left(A_{\max}^T\right) - L A_{\max}^T \ and\ W_t = f\left(A_{\max}^T\right), \forall t > 1.$$

Algorithm 1 ORRIC

Input: V_t, W_t, $U_t = \frac{C_{(t)}}{D_{(t)}}$ and four ascending lists: $\{A_i^T, i \in \mathcal{M}\}$, $\{A_j^I, j \in \mathcal{N}\}$, $\{C_i^T, i \in \mathcal{M}\}$, $\{C_j^I, j \in \mathcal{N}\}$.
Output: A pair of retraining and inference configurations.

```
Initialization: Set i = 1, j = N, i* = j* = K = 0.
while i ≤ M and j ≥ 1 do
    if C_i^T + C_j^I ≤ U_t then
        if V_t A_i^T + W_t A_j^I > K then
            i* = i; j* = j; K = V_t A_i^T + W_t A_j^I;
        i = i + 1;
    else
        j = j − 1;
return i*, j*;
```

Our solution:

1. Deal with target function of (P): Leverage the concave property of *f* and a special-designed regularization term to relax the target function to a linear function. Decouple it to every time slot, we get (Dt).
2. To deal with (Dt), we propose ORRIC. The basic idea is: first we remove all configurations that consume more resources yet yield lower profits, then searching through retraining and inference configurations pairs likely to exceed the computational resource constraint.
3. ORRIC has linear complexity and uses partial information of *f*: $f(A_{max}^T)$ and *L*, a positive lower bound of $f'(A_{max}^T)$.

Insights from ORRIC

With different V_t and W_t, ORRIC can convert to several heuristic algorithms for different resource environments.

$$\max_{x_i(t), y_j(t)} V_t \sum_{i=1}^{M} x_i(t) A_i^T + W_t \sum_{j=1}^{N} y_j(t) A_j^I$$

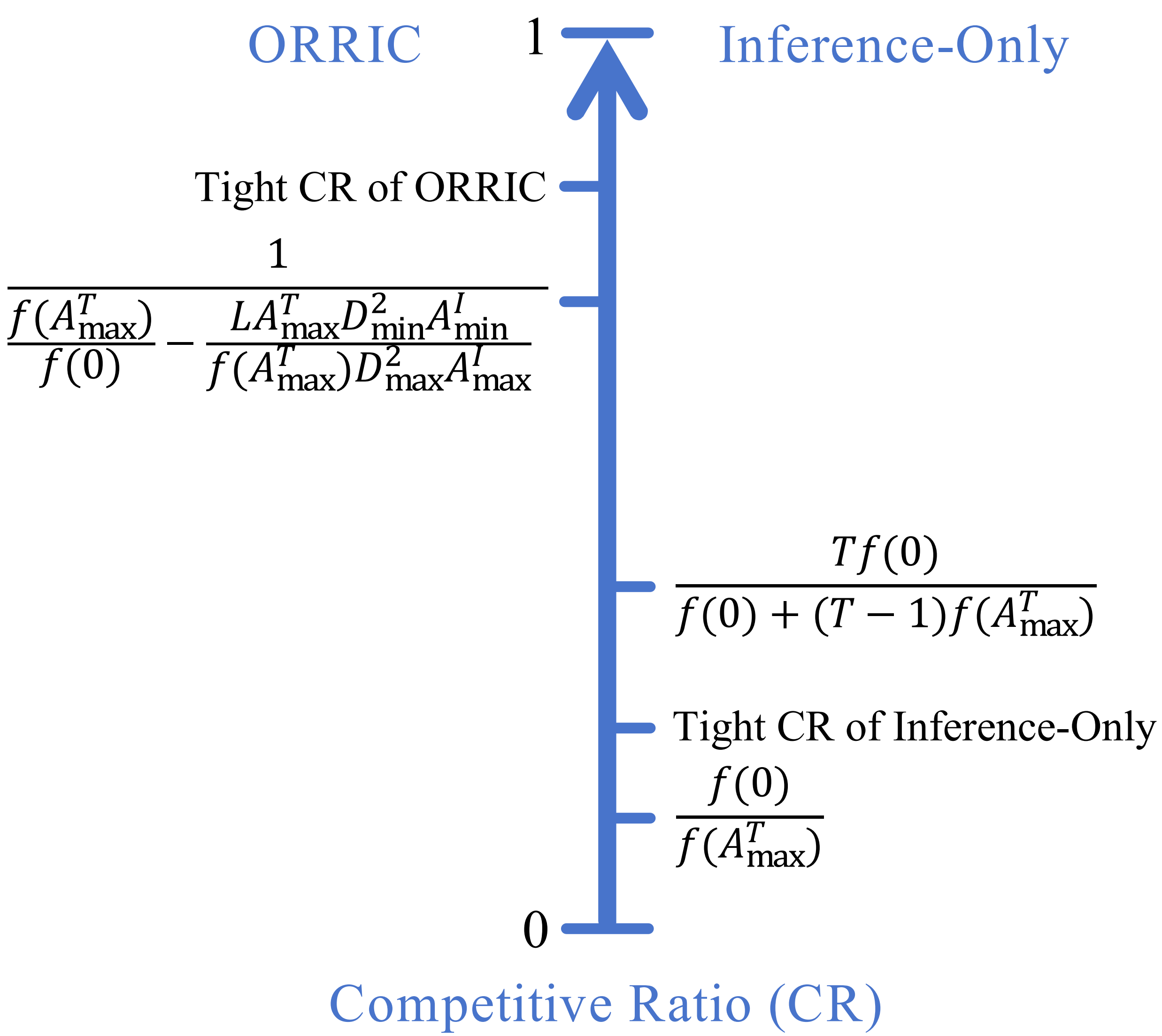
$$where\ V_t = L \frac{D_{\min} A_{\min^I}}{D_{\max}} \left(\sum_{\tau=t}^{T-1} \frac{1}{\tau} \right), \qquad \text{(Dt)}$$

$$W_1 = f\left(A_{\max}^T\right) - L A_{\max}^T \ and\ W_t = f\left(A_{\max}^T\right), \forall t > 1.$$

Resources	T is Large	T is Small
Sufficient	Knowledge-Distillation	
Limited	Focus-Shift	Inference-Greedy
Scarce	Inference-Only	

1) Knowledge-Distillation: The teacher model imparts knowledge to the student model without considering resource consumption.
2) Inference-Greedy: Prioritize using a higher configuration for inference and utilize the remaining resources for retraining.
3) Focus-Shift: Shift the focus from retraining to inference as time passes.
4) Inference-Only: This algorithm is actually the traditional computing paradigm that deploys a trained model and then performs inference.

Insights from Compititive Results

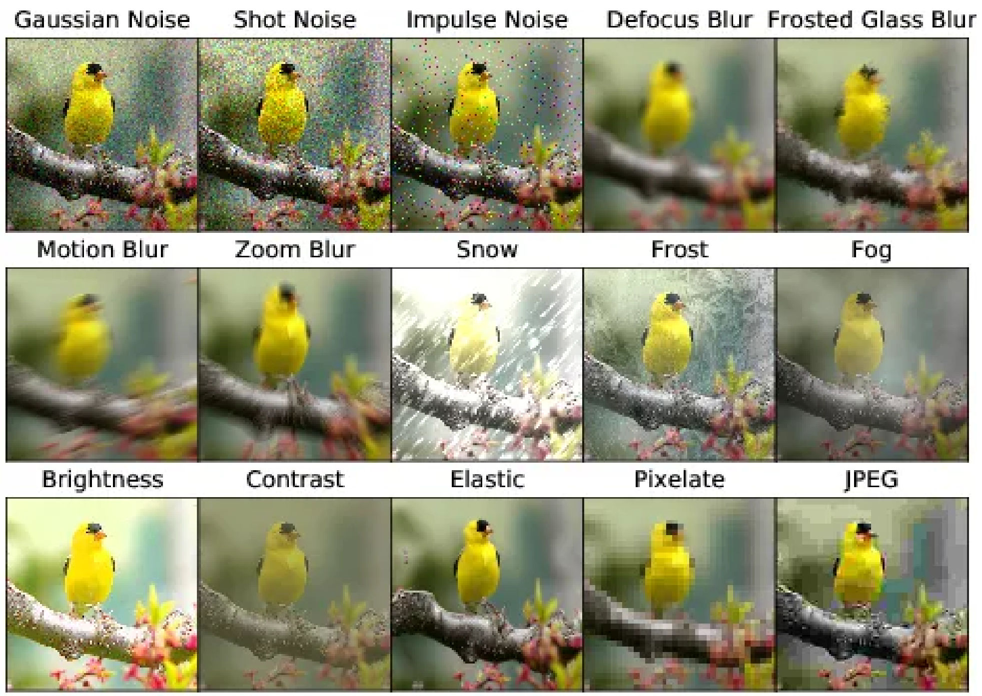


Definition: For a maximization problem, the competitive ratio (or CR) ***c*** of algorithm ***ALG*** is defined as $\boldsymbol{c} \leq \boldsymbol{ALG/OPT}$ for every input ***I***, where ***OPT*** represents the optimal offline algorithm with complete knowledge of future information. ***c*** higher, ***ALG*** better.

Corollary 1: When $T > (f(A^T_{max}) - f(0))/(\alpha f(0))$, the tight competitive ratio of ORRIC is strictly better (bigger) than the tight competitive ratio of Inference-Only.

Insights: When drift occurs for a sufficiently lengthy time, the worst-case performance of the ***Model Retraining and Inference Co-location paradigm*** is strictly better than that of the traditional ***Inference-Only paradigm***.

Evaluation Setup



Dataset: CIFAR-10-C

Setup: We treat these corruptions as imitations of data drift. We first train MobileNetV2 (student model) and ResNet50 (teacher model) on the training set of CIFAR-10, then test them on CIFAR-10-C.

Inference configuration: different resolutions of input images (32*32, 28*28, 24*24, or 20*20). A_j^I is the model's normalized accuracy on the CIFAR-10 test dataset when using different input resolutions (with the largest number being 1), C_j^I is the coresoponding MACs.

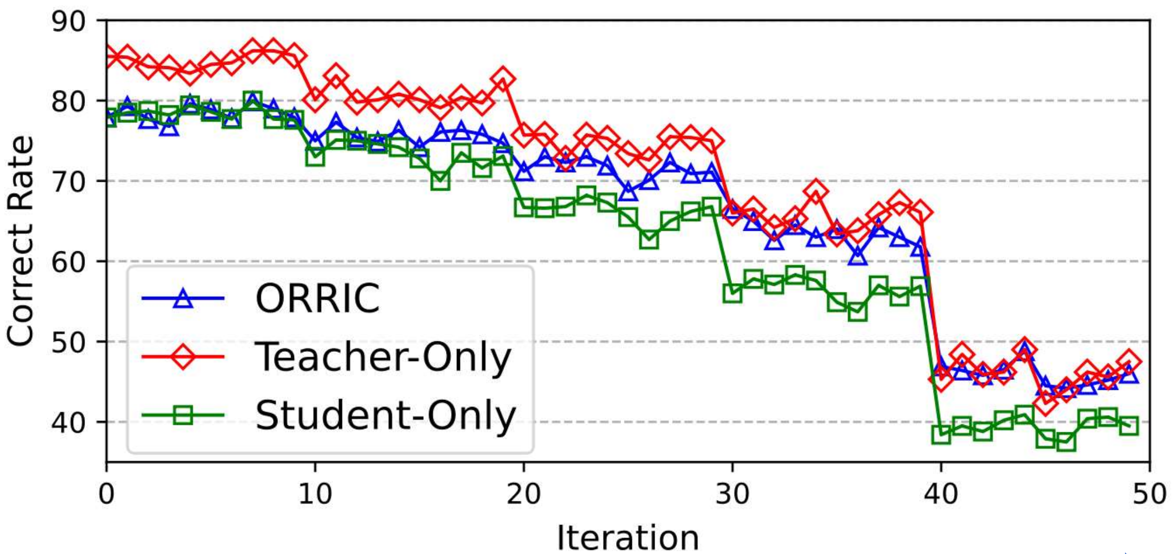
Retraining configuration: different sampling ratios of uploaded data at the t-th time slot (0, 0.1, 0.2, 0.3, 0.5, 1.0), with training for only 1 epoch. C_i^T is the coresoponding MACs, and A_i^T is propotianl to C_i^T (with the largest number normalized to 1).

$f(A_{max}^T)$ is set as the model's accuracy on the cifar-10 test dataset using the best inference configuration and L is set as 0.01.

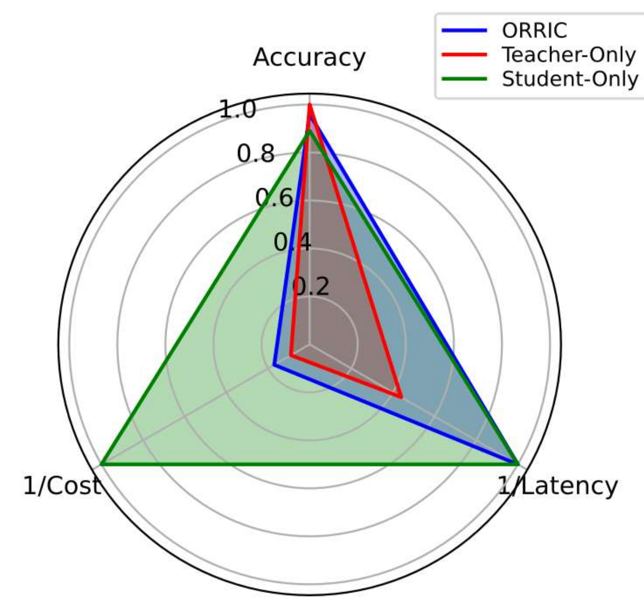
Our Code: https://github.com/caihuaiguang/ORRIC.

Evaluation Results

Model(Resolution)	MACs (M)	Latency (μs)	original	brightness	contrast	defocus blur	elastic transform	fog	frost	gaussian blur	gaussian noise	glass blur	impulse noise	jpeg compression
MobileNetV2 (20*20)	6.35	7.54	44.93	42.60	23.28	40.47	39.25	27.64	39.46	38.41	42.97	40.33	41.35	42.95
MobileNetV2 (24*24)	6.71	8.37	59.38	54.41	28.09	51.26	49.94	37.42	48.49	48.08	55.71	50.18	53.04	57.10
MobileNetV2 (28*28)	7.45	10.15	73.29	67.94	38.33	63.21	62.48	49.68	59.17	59.23	**64.53**	62.21	**60.38**	69.31
MobileNetV2 (32*32)	7.94	10.51	**79.57**	**76.00**	**47.52**	**71.08**	**71.91**	**62.74**	**62.70**	**67.02**	56.28	**62.90**	57.38	**74.71**
ResNet50 (20*20)	65.76	17.41	54.50	49.20	32.26	50.71	49.00	39.31	44.19	48.99	52.23	49.99	49.99	53.04
ResNet50 (24*24)	68.96	19.29	71.95	66.25	40.68	62.58	61.52	50.54	60.49	58.75	68.26	62.61	64.58	69.64
ResNet50 (28*28)	82.01	24.08	79.02	74.19	42.74	66.58	66.79	55.34	66.95	61.60	72.89	68.07	**66.01**	75.72
ResNet50 (32*32)	86.37	24.09	**86.13**	**83.21**	**55.34**	**73.97**	**76.59**	**70.41**	**76.09**	**68.40**	**72.94**	**70.55**	62.42	**82.43**
ORRIC	-	-	79.24	79.06	52.19	72.08	72.35	67.20	70.96	67.51	68.44	64.90	58.99	75.70



Severity level of corruption are becoming higher.



Accuracy-Cost-Latency trade-off comparison.

Future Direction: Modeling and Algorithm Design

1. Modeling of the model retraining and inference co-location paradigm.
    - $f(x)$ analytic expression (related research: learning curve).
    - Other assumption: Current model performance is only related to past data within a time window (e.g. in-context learning).
    - Multi task.
2. Algorithm design.
    - Close loop algorithm. Bandit algorithm.
    - Tighter comptitive ratio (must greater than inference only algorithm).

Future Direction: On-device Model Retraining and Inference Co-location

- Exiting researches on model retraining and inference co-location typically deploy the model on edge or cloud.
- Model retraining and inference co-location on devices holds promise for enhanced privacy protection, reduced bandwidth usage and personalized AI models.
- Famous works like TensorFlow Lite, PyTorch Mobile and MNN mainly focus on model inference on devices, and there is little code available for model retraining and inference co-location.

Thank you!

Our Code: https://github.com/caihuaiguang/ORRIC.