Shoggoth: Towards Efficient Edge-Cloud Collaborative Real-Time Video Inference via Adaptive Online Learning

Liang Wang[1], Kai Lu[1], Nan Zhang[2], Xiaoyang Qu[2], Jianzong Wang[2], Jiguang Wan[1], Guokuan Li[1], Jing Xiao[2]

[1]Huazhong University of Science and Technology, China
[2]Ping An Technology (Shenzhen) Co., Ltd., China

WHERE INNOVATION BEGINS
DESIGN AUTOMATION CONFERENCE

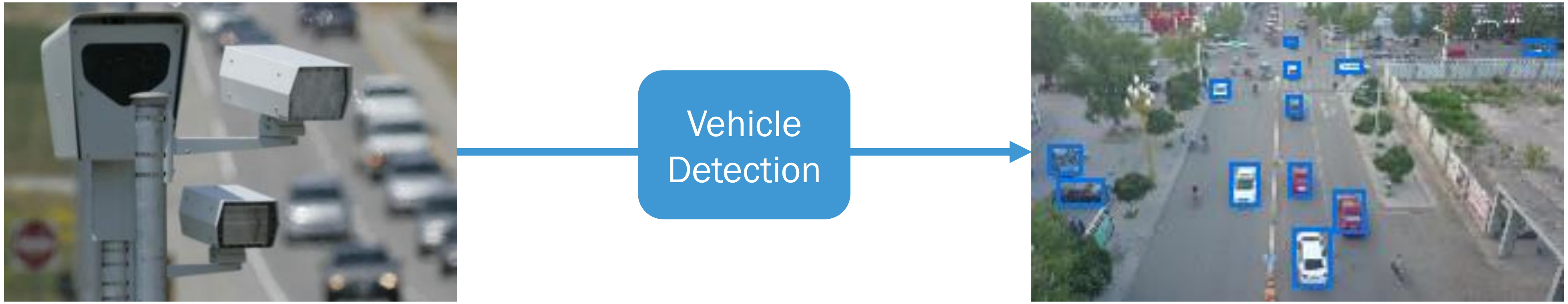
JULY 9–13, 2023
MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

Outline

- Background and Motivation
- Shoggoth
- Evaluation
- Conclusion

Background

- Video analytics is ubiquitous



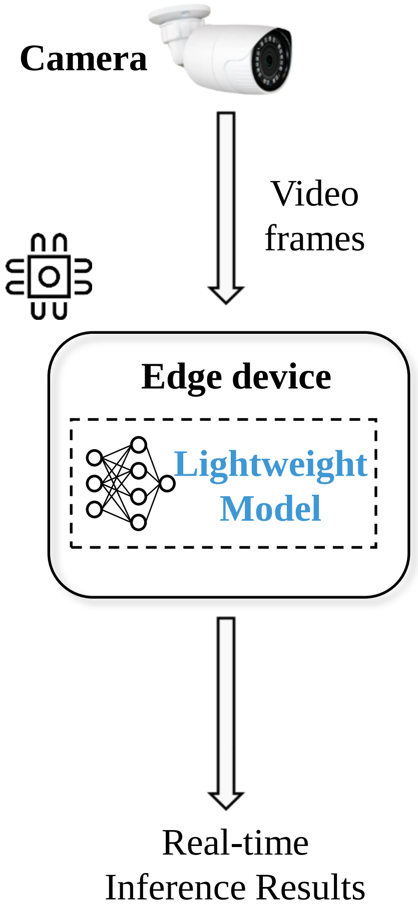
Video Analytics

- Real-time video analytics prefer edge devices
  - Reduce Latency
  - Minimize Bandwidth
  - Increase Scalability

Motivation

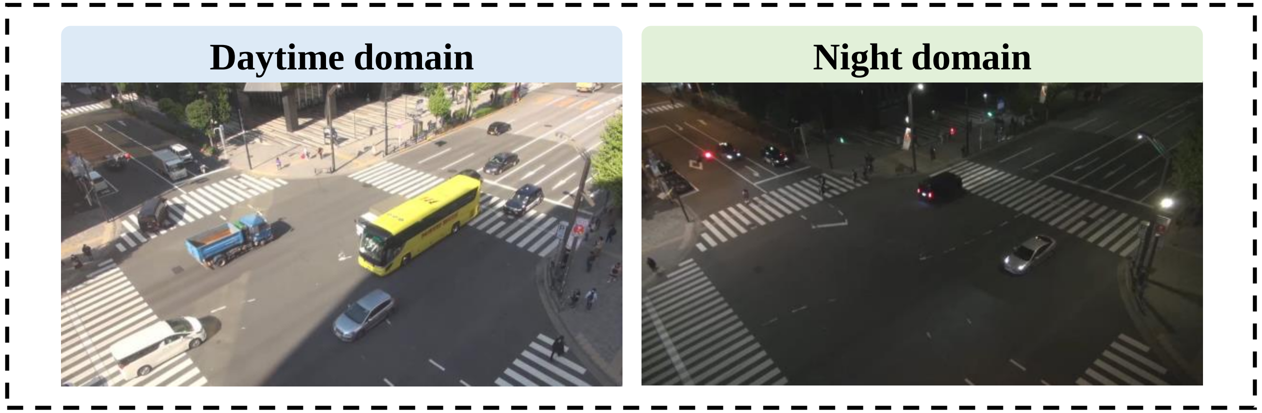
- Edge devices are resource-constrained
- Only specialized lightweight models can be deployed
  - Fewer weights and shallower architectures
  - Identify limited amount of object appearances, object classes, and scenes.
  - Vulnerable to data drift

Lightweight models on edge devices are not that accurate!

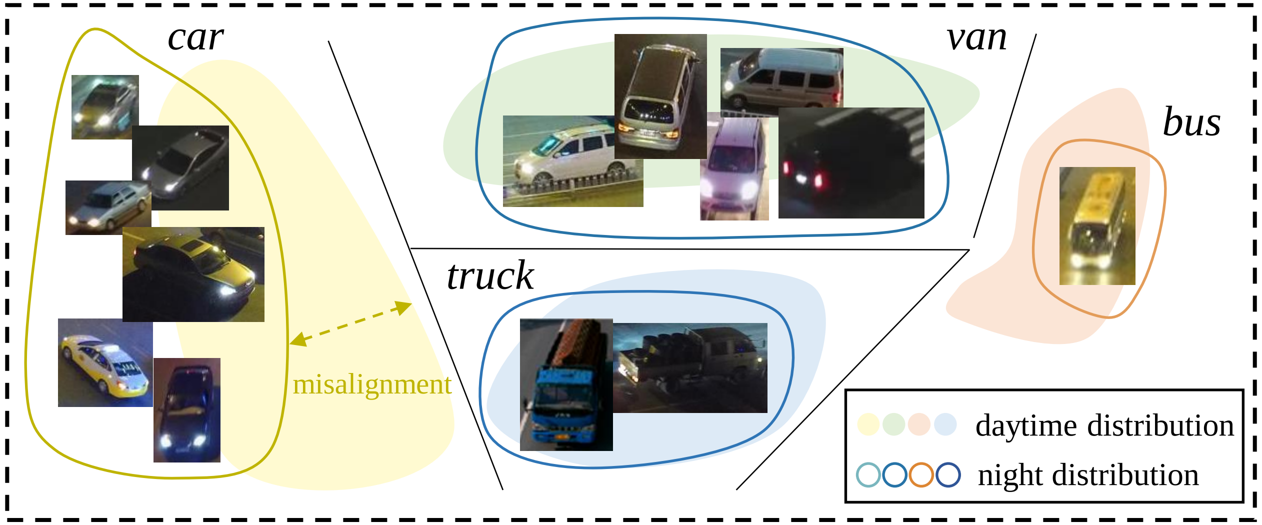


Challenge of Data Drift

- Why does data drift occur?
  - Real-time video scenes vary over time
- How does data drift lead to accuracy drop?
  - *Domain Shift* - a lightweight model trained on daytime images does not work well when it encounters night
  - *Class Distribution Shift* - the dynamic, time-evolving distribution results in objects of different class distributions are difficult to distinguish for the lightweight model



(a) Domain Shift



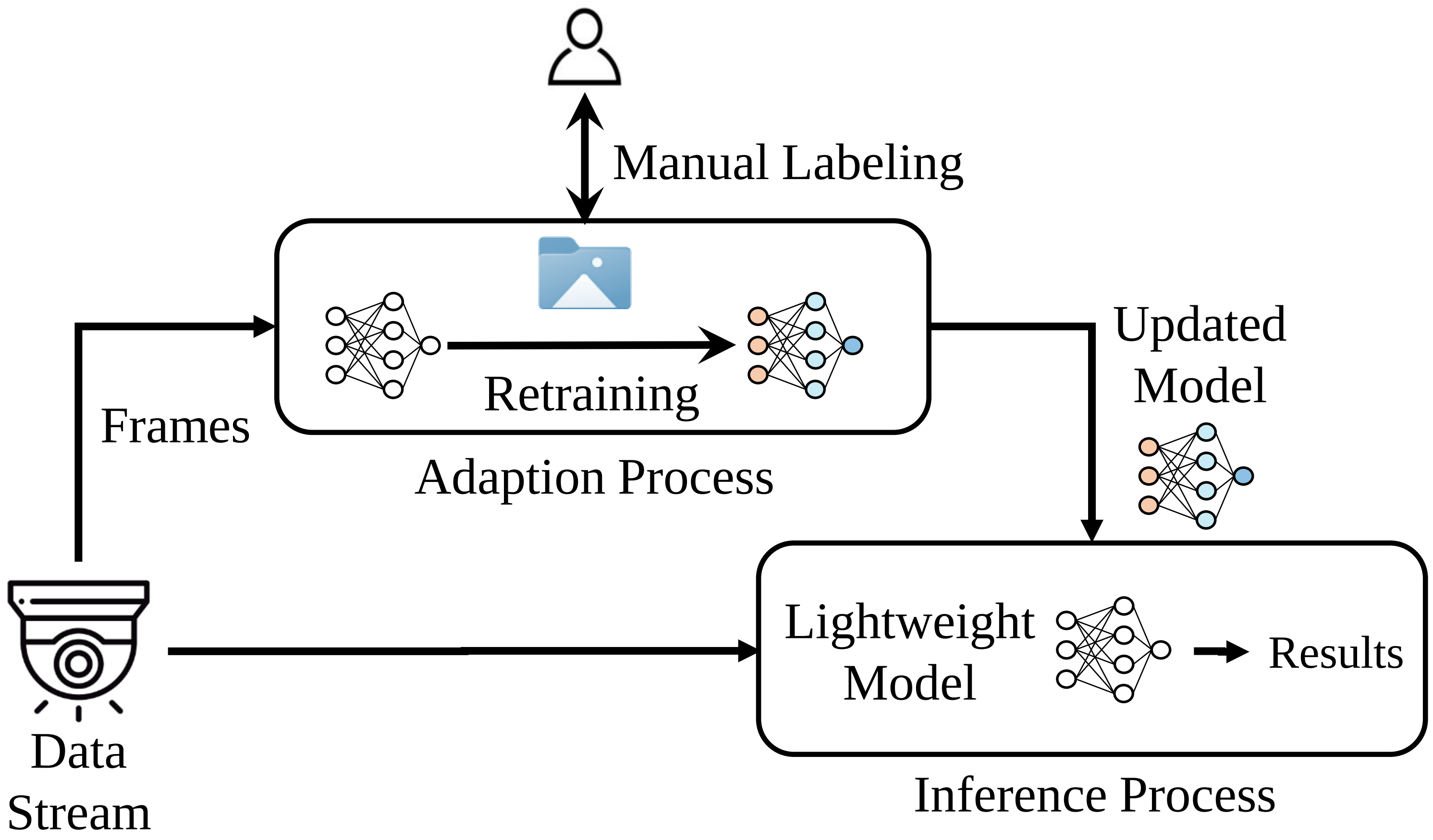
(b) Class Distribution Shift

Outline

- Background and Motivation
- Shoggoth
- Evaluation
- Conclusion

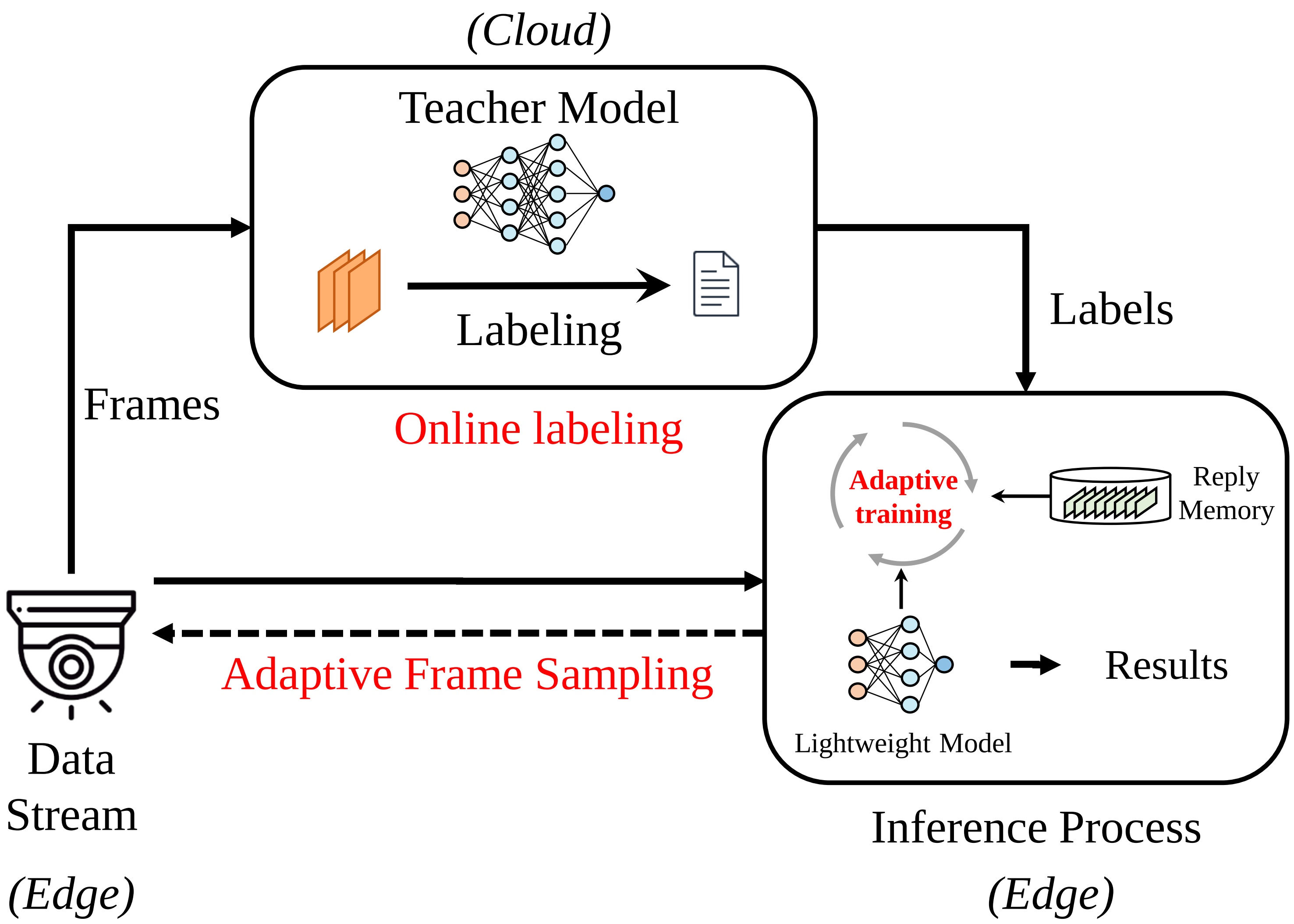
Existing Solution - Model Retraining

- Frame Sampling → Manual Labeling → Retraining → Updating Model
- Drawbacks:
  - Manual labor required
  - Not responsive
  - Hard to determine retraining frequency



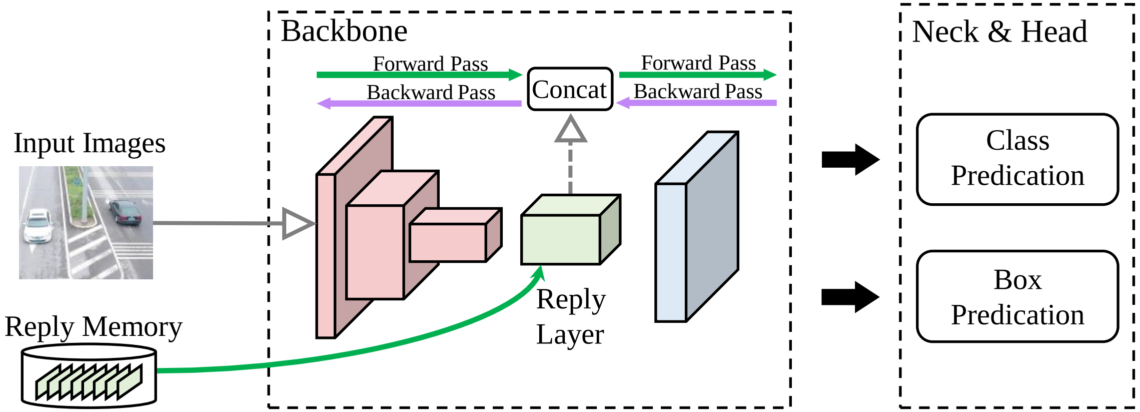
Shoggoth - Adaptive Online Learning

- Online labeling *(Cloud)*
  - Label sample frames with the large teacher model
- Adaptive training *(Edge)*
  - Fine-tune the lightweight model
  - Adapt for scene change
- Adaptive frame sampling
  - Adjust the sampling rate
  - Increase robustness and reduce bandwidth



Adaptive Training

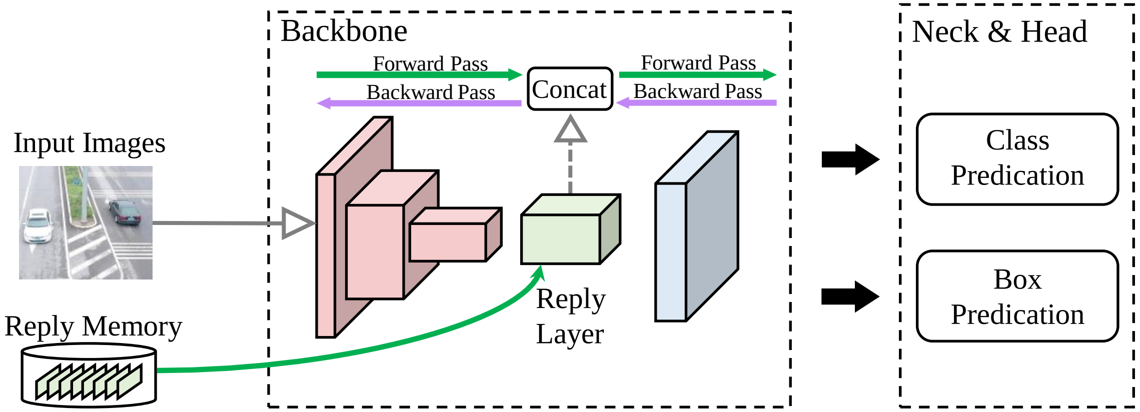
- Execute on edge devices
- Address catastrophic forgetting
- Key Insights:
  - Forgetting occurs in the classification head, needs tuning for accuracy
  - Early layers stable and reusable post sufficient pre-training
  - Replay memory stores activation volumes, not raw input images



Adaptive Training Schema of Object Detector

Adaptive Training

- Replay Memory Management
  - Memory updates occur after each adaptive training
  - Random current image subset replaces random replay memory subset
  - All images stored if memory isn't full
  - Equal chance of batch sampling storage in memory
- Training Control
  - Constant replay ratio in each mini-batch
    - Every training batch contains N images and the replay memory includes M images, in a mini-batch of size K, only $\frac{K \times N}{N+M}$ images need to travel across the red layers
  - Weights frozen after first batch to address slow-down



Adaptive Training Schema of Object Detector

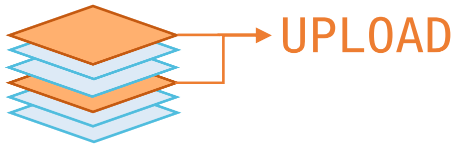
Adaptive Frame Sampling

- Models need adaptive training frequency to handle scene variations stably
- Frame sampling rate impacts training frequency
- Adaptive frame sampling adjusts the rate based on –
  - Degree of video scene changes
  - Inference accuracy
  - Resource usage

Adaptive Frame Sampling

- ϕ: the change rate over time for video frames
- α: the inference accuracy of the current model
- λ: the resource utilization of the edge device

Adjusting
the sampling rate



Outline

- Background and Motivation
- Shoggoth
- Evaluation
- Conclusion

Evaluation Setup

- Real-time video object detection as our evaluated workload
- Comparisons
  - *Edge-Only*: Edge model without video-specific customization; all inferences on the edge device
  - *Cloud-Only*: All frames uploaded to the cloud for detection and results
  - *Prompt*: Shoggoth without adaptive sampling; fixed 2 fps sampling rate; model adaptation happens promptly and regularly
  - *Adaptive Model Streaming (AMS)*: Knowledge distillation in the cloud for model adaption; updated student model sent to edge device

Datasets	UA-DETRAC, KITTI (Car only) and Waymo Open
DNN models	YOLOv4 with Resnet18 backbone (Edge) Mask R-CNN with ResNeXt-101 (Cloud)
Platforms	NVIDIA Jetson TX2 (Edge) NVIDIA V100 GPU (Cloud)
Metrics	uplink and downlink bandwidth mAP@0.5 (mean Average Precision, and Intersection over Union = 0.5)

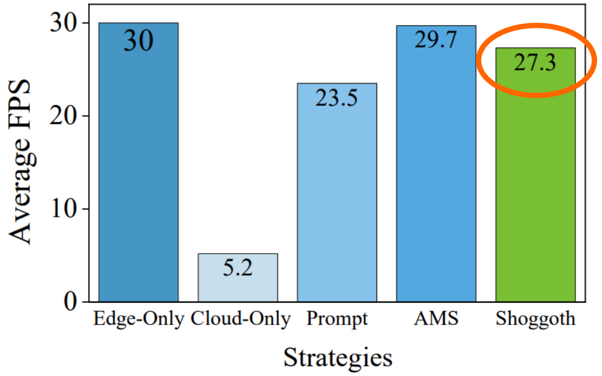
Overall Improvements

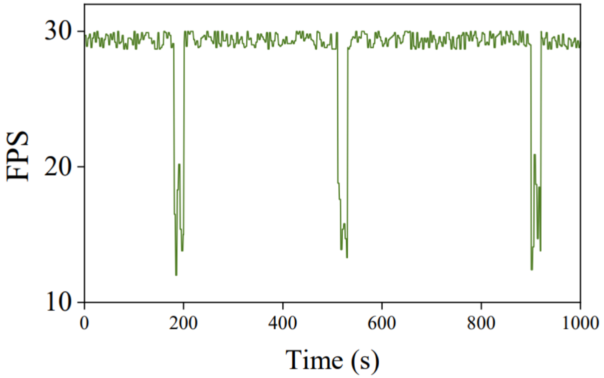
- Comparison of different strategies on three datasets
  - 15%–20% accuracy improvement compared to the edge-only
  - require 24× less uplink bandwidth to achieve similar accuracy to the cloud-only

Dataset	Metric	Edge-Only	Cloud-Only	Prompt	AMS	Shoggoth
UA-DETRAC [20]	Up/Down Bandwidth (Kbps)	0/0	3257/3539	303/22	151/226	**135/10**
	mAP@0.5 (%)	34.2	58.9	48.3	51.6	**53.5**
KITTI [21]	Up/Down Bandwidth (Kbps)	0/0	2184/2437	179/10	94/203	**91/5**
	mAP@0.5 (%)	56.8	78.0	71.4	72.8	**74.7**
Waymo Open [22]	Up/Down Bandwidth (Kbps)	0/0	2687/2880	278/15	127/207	**112/8**
	mAP@0.5 (%)	47.5	64.7	61.5	59.1	**61.9**

Impact of Adaptive Training

- Average FPS overall for different strategies (left) and FPS over time (right)





- mAP (%) and training time (in seconds) of different methods

Method	mAP	Training Time		
		Forward	Backward	Overall
Ours (Baseline)	53.5	17.8	0.8	18.6
Input	49.6	536.2	31.6	567.8
Completely Freezing	50.7	17.8	0.7	18.5
Conv5_4	52.3	20.2	5.8	26.0
No Replay Memory	45.6	95.7	6.2	101.9

Impact of Adaptive Frame Sampling

- Sensitivity to different sampling rates

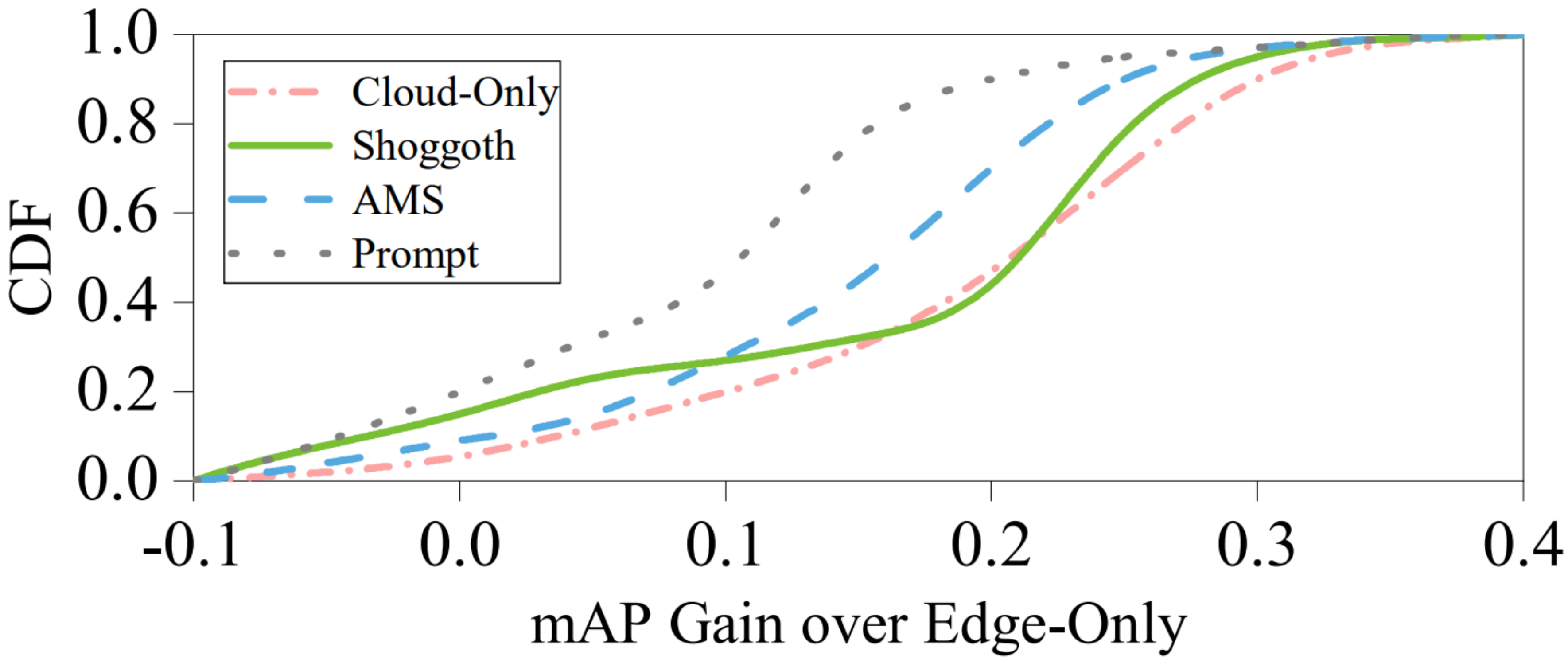
$rate \rightarrow$	0.1	0.2	0.4	0.8	1.6	2.0	Adaptive
Up BW (Kbps)	19	36	61	122	249	307	135
Average IoU	0.483	0.524	0.556	0.623	0.612	0.597	0.640

High Sampling Rates Cause Overfitting

Best Accuracy

Cumulative Distribution of mAP Improvement

- CDF of mAP gain vs. Edge-Only across all frames for other strategies



Outline

- Background and Motivation
- Shoggoth
- Evaluation
- Conclusion

Conclusion

- Shoggoth is an efficient edge-cloud collaborative architecture designed to improve inference performance on real-time videos of changing scenes
  - Online knowledge distillation – enhance model accuracy suffering from data drift
  - Adaptive training – adapt models under limited computational power
  - Adaptive sampling – increase robustness and reduce bandwidth
- Outperform state-of-the-art solutions in the trade-off between low latency and high accuracy

Thanks!