SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

OmniSense: Towards Edge-Assisted Online Analytics for 360-Degree Videos

Miao Zhang, Yifei Zhu, Linfeng Shen, Fangxin Wang, **Jiangchuan Liu**

BACKGROUND

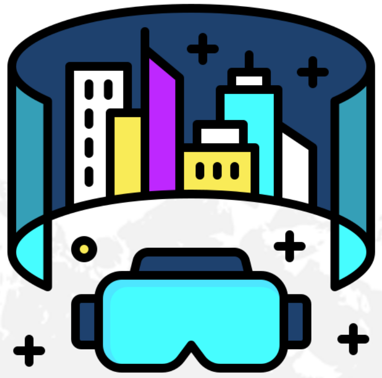
Omnidirectional cameras have become increasingly affordable

Insta360 One X2

GoPro Max

Common Usage:

Entertainment:
Record videos for human viewers



Potential Usage:

Perception and Interaction:
Analyze videos for full situational awareness without blind spots

BACKGROUND

Video Analytics (VA): Evolving From Regular Videos to 360-Degree Videos

VA for regular videos:

Challenge 1: high-accuracy VA requires running compute-intensive deep neural networks (DNNs)

Solution: offload VA tasks to resource-rich edge or cloud servers

Challenge 2: offloading video data over networks requires tremendous bandwidth

Solution: encoding configuration tuning, frame filtering, DNN model splitting

Immersive VA for 360-degree videos:

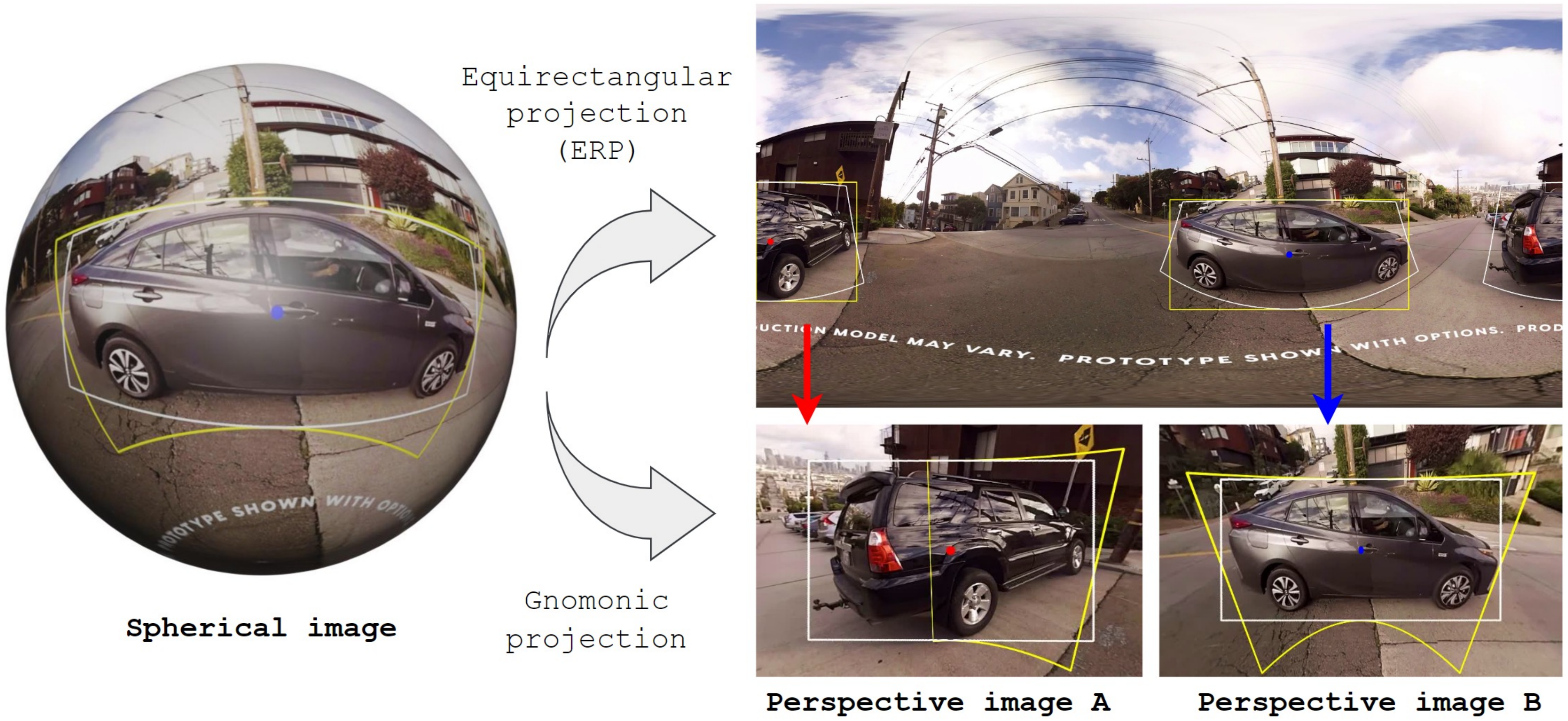
Challenge 1: 4 – 6 x large than regular videos. Much more compute-intensive and bandwidth-intensive

Challenge 2: particular geometry structure (spherical image)

BACKGROUND

Immersive VA – Potential Solution Discussion

- Direct inference on ERP images? Distortion and discontinuity can hurt accuracy.
- Analyze many perspective images (PIs) to minimize distortion while covering the entire sphere? Resource-intensive and time-consuming
- Design DNNs specialized for spherical geometry? Cannot directly benefit from advances in off-the-shelf vision models. Require extra efforts.



MOTIVATION

Measurement Setup

Vision Task (model): object detection (scaled-YOLOv4, which provides a set of model variants)

Video dataset: self-collected UHD 360-degree video dataset

Model Name (Index)	Model Size	Input Size
YOLOv4-Tiny-416 (1)	23 MB	416 x 416
YOLOv4-CSP-512 (2)	202 MB	512 x 512
YOLOv4-CSP-640 (3)	202 MB	640 x 640
YOLOv4-P5 (4)	271 MB	896 x 896
YOLOv4-P6 (5)	487 MB	1280 x 1280

Immersive Video Name	Video Source	Resolution	Frames
New-Orleans-drive	YouTube	7680 x 3840	2100
Expressway-drive	YouTube	5760 x 2880	2100
Chicago-drive	YouTube	7680 x 3840	2100
Sunny-walk1	Self-captured	5376 x 2688	2100
Sunny-walk2	Self-captured	5376 x 2688	2100
Cloudy-walk	Self-captured	5376 x 2688	2100

Immersive object detection criteria:

Spherical bounding box (SphBB): represented by a spherical region (SR)

$$(\theta, \phi, \Delta\theta, \Delta\phi)$$

longitude latitude vertical FoV

horizontal FoV

MOTIVATION

☐ Measurement Setup

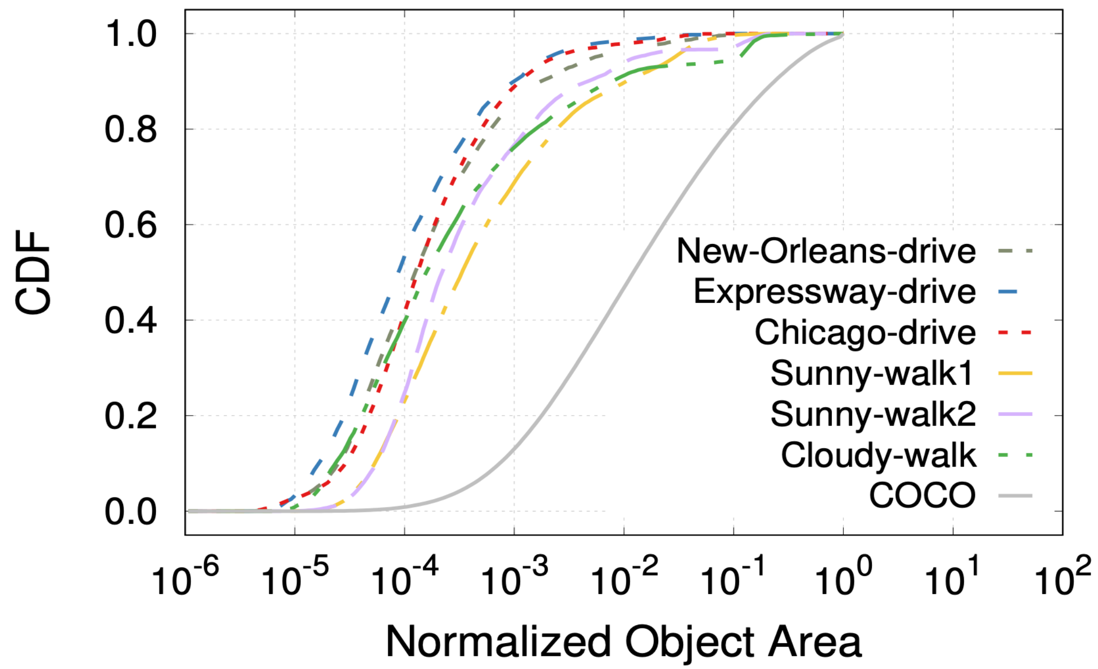
Ground-truth annotation: self-developed automated annotation method

Frame source: Sunny-walk2

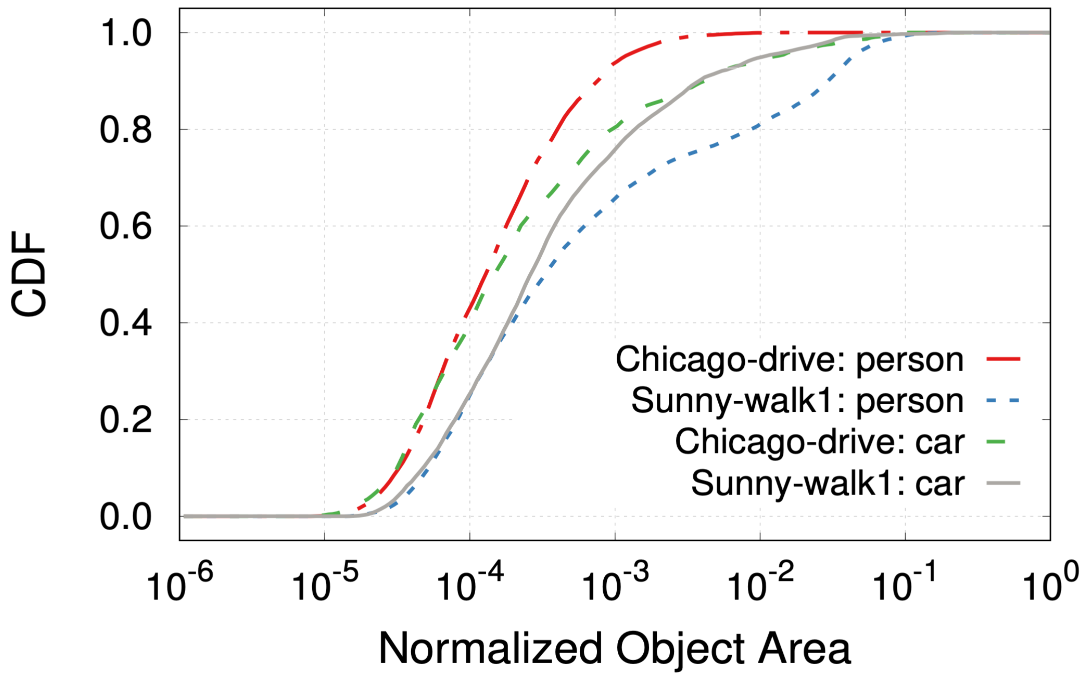
Frame source: Chicago-drive

MOTIVATION

Measurement Insights



CDFs of NOA for 360-degree videos and 2D images



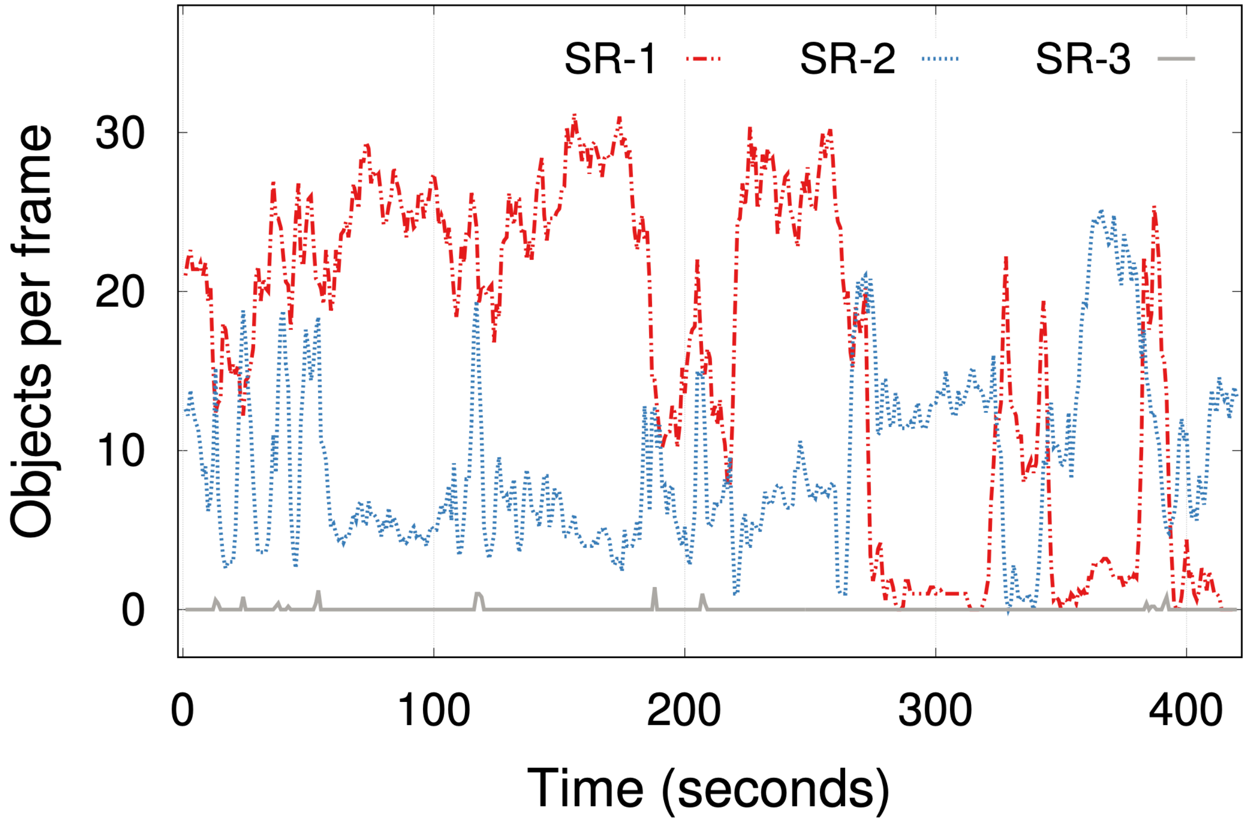
CDFs of NOA for two specific object categories

Most objects in 360-degree videos only occupy a tiny area of a frame, which cannot be handled by off-the-shelf vision models.

Both object size and category are crucial reference factors in characterizing video content.

MOTIVATION

☐ Measurement Insights



Object number variations in different spherical regions (video: New-Orleans-drive).

The spatial distribution of objects is biased and highly dynamic.

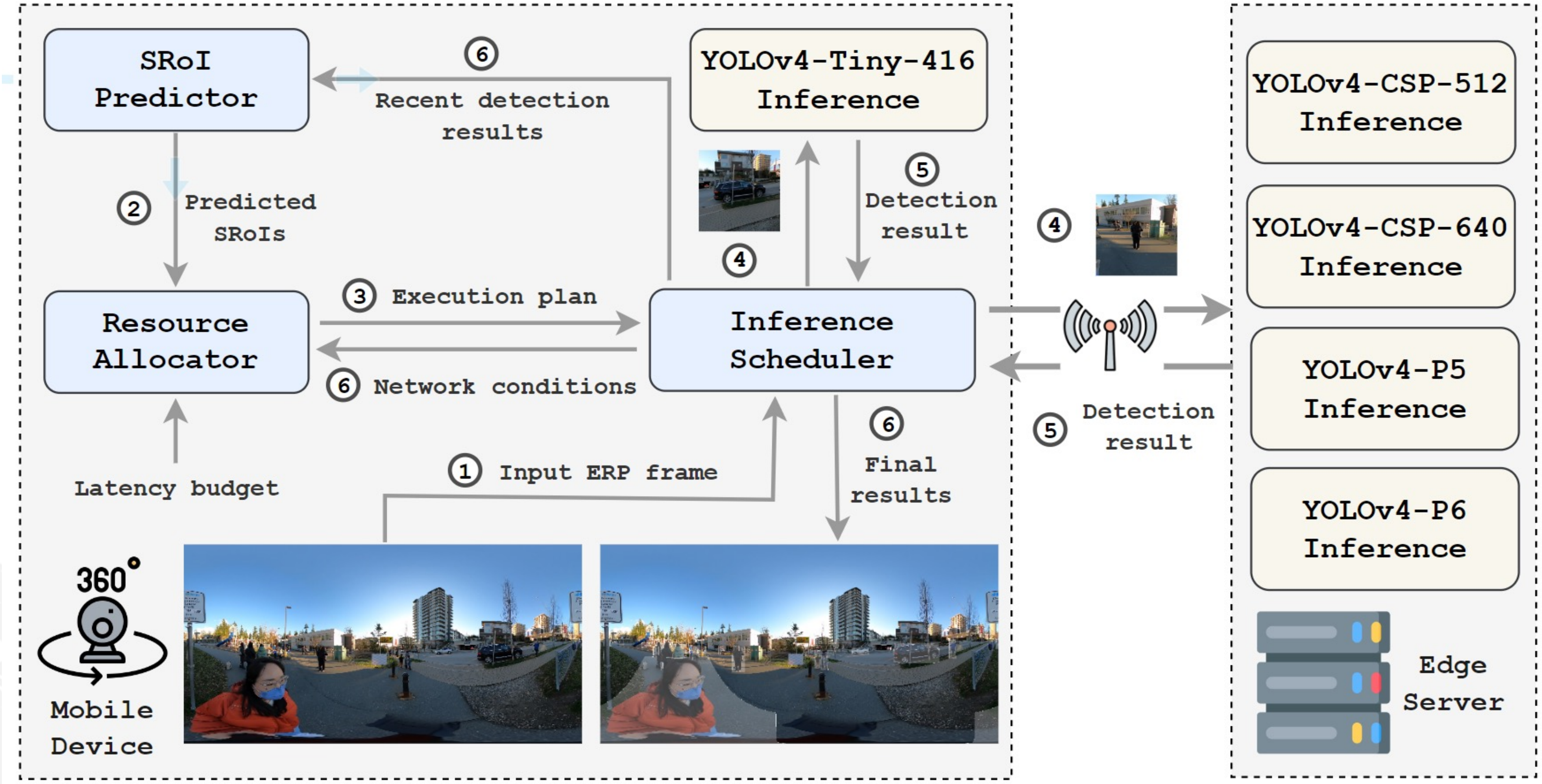
Prune useless pixels before offloading.

Apply different models to different SRs.

Adapt to content variations.

SYSTEM DESIGN

Overview of OmniSense



SYSTEM DESIGN

☐ Lightweight SRoI Prediction

Motivation: consecutive frames have the smallest content differences.

Solution: Propose a lightweight spherical RoI (SRoI) prediction algorithm based on the most recent detection results. Design a spherical object discovery mechanism to discover new objects.

Algorithm 1: SRoI Prediction Algorithm

Input: f; γ; $\mathcal{O}$ (detected objects of the most recent δ frames)
Output: A set of predicted SRoIs $\mathcal{R}$

Initialize SRoI sets $\mathcal{S} \leftarrow \emptyset$, $\mathcal{S}' \leftarrow \emptyset$;
Get the number of all historical objects $N \leftarrow |\mathcal{O}|$;
foreach *object* $o \in \mathcal{O}$ **do**
  if *o can be covered by an $f \times f$ FoV* **then**
    $merged \leftarrow False$;
    foreach *SRoI* $s \in \mathcal{S}$ **do**
      $hFoV, vFoV \leftarrow$ merged horizontal and vertical FoVs for the set $s.objects \cup \{o\}$;
      if $hFoV < f$ *and* $vFoV < f$ **then**
        $s.objects \leftarrow s.objects \cup \{o\}$;
        $s.FoV \leftarrow (hFoV, vFoV)$;
        $merged \leftarrow True$; $break$;
    if *not* $merged$ **then**
      $new\_s \leftarrow$ create a new SRoI with o ;
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{new\_s\}$;
  else
    Create a new special SRoI s' with o ;
    $s'.center \leftarrow o.center$; $s'.FoV \leftarrow \gamma \times o.FoV$;
    Calculate content characteristics $s'.ccv$ based on o ;
    $s'.\alpha \leftarrow 1 \;/\; N$;
    $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{s'\}$;
foreach *SRoI* $s \in \mathcal{S}$ **do**
  Calculate SRoI center $s.center$ according to $s.FoV$;
  Calculate $s.ccv$ based on $s.objects$;
  $s.\alpha \leftarrow |s.objects| \;/\; N$;
  $s.FoV \leftarrow (f, f)$;
$\mathcal{R} \leftarrow \mathcal{S}' \cup \mathcal{S}$;
return $\mathcal{R}$;

SYSTEM DESIGN

☐ Content-Specific Model Performance Estimation

Model inference latency : offline profiling

Model accuracy estimation: Accuracy varies with the analyzed image content.

Quantify the detection capability of a model. Define the general accuracy vector (gav):

$$\boldsymbol{A_i} = [a_i^{s1}, \cdots, a_i^{sn}, a_i^{m1}, \cdots, a_i^{mn}, a_i^{l1}, \cdots, a_i^{ln}]$$

Quantify the content characteristics of an SRoI. Define the content characteristics vector (ccv):

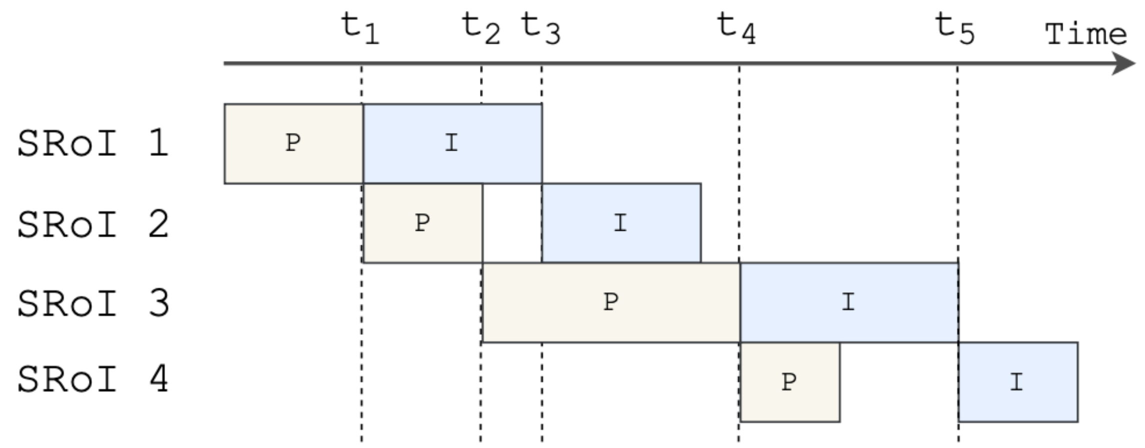
$$\boldsymbol{P_j} = [p_j^{s1}, \cdots, p_j^{sn}, p_j^{m1}, \cdots, p_j^{mn}, p_j^{l1}, \cdots, p_j^{ln}]$$

Estimate detection accuracy of model i on SRoI j : $\boldsymbol{A_i} \cdot \boldsymbol{P_j}$

☐ Latency-Constrained Model Allocation

$$\max_{x_{i,j}} \sum_{j \in \mathcal{R}} \sum_{i \in \mathcal{M}} A_{i,j} \cdot x_{i,j}$$

$$s.t. \begin{cases} \mathcal{L}(\mathcal{X}) \le T, \quad \mathcal{X} = \{x_{i,j} \mid i \in \mathcal{M}, j \in \mathcal{R}\} \\ \sum_{i \in \mathcal{M}} x_{i,j} = 1, \quad \forall j \in \mathcal{R} \\ x_{i,j} \in \{0, 1\}, \quad \forall i \in \mathcal{M}, \quad \forall j \in \mathcal{R} \end{cases}$$



Algorithm 2: Dynamic Programming Algorithm

Input: $\{A_{i,j}\}$; $\{d_{i,j}\}$; $\{d^P_{i,j}\}$; $\{d^I_{i,j}\}$; T
Output: The optimal execution plan

$\mathcal{S}(1) \leftarrow \emptyset$;
foreach *model* $i \in \mathcal{M}$ **do**
  if $d_{i,1} \le T$ **then**
    $\mathcal{S}(1) \leftarrow \mathcal{S}(1) \cup \{(A_{i,1},\ d^P_{i,1},\ d_{i,1},\ [i])\}$;
for $j = 1$ *to* $r - 1$ **do**
  $\mathcal{S}(j+1) \leftarrow \emptyset$;
  foreach *quaternion* $(v, t^P, t, m\_list) \in \mathcal{S}(j)$ **do**
    foreach *model* $i \in \mathcal{M}$ **do**
      $cur\_t \leftarrow \max(t^P + d_{i,j+1},\ t + d^I_{i,j+1})$;
      if $cur\_t \le T$ **then**
        $cur\_v \leftarrow v + A_{i,j+1}$;
        $cur\_t^P \leftarrow t^P + d^P_{i,j+1}$;
        $m\_list.append(i)$;
        $\mathcal{S}(j+1) \leftarrow$
         $\mathcal{S}(j+1) \cup \{(cur\_v, cur\_t^P, cur\_t, m\_list)\}$;
  Remove dominated execution plans from $\mathcal{S}(j+1)$;
Return the execution plan with the highest v in $\mathcal{S}(r)$;

EVALUATION

System Implementation

Mobile device: Nvidia Jetson TX2; Edge server: a desktop with an Nvidia GeForce GTX 1080Ti

The mobile device and server are connected by ASUS AC 1900 router.

OpenCV for Image and video operations; ZeroMQ for data transmission

Experimental Setup

Videos and models: same as in the measurement study

Networks: shape the traffic to typical 5G mobile throughputs

Performance metric: Spherical mAP; Mean end-to-end (E2E) latency

Baselines: ERP; CubeMap

Latency budgets: from 500 ms to 4,500 ms based on baselines

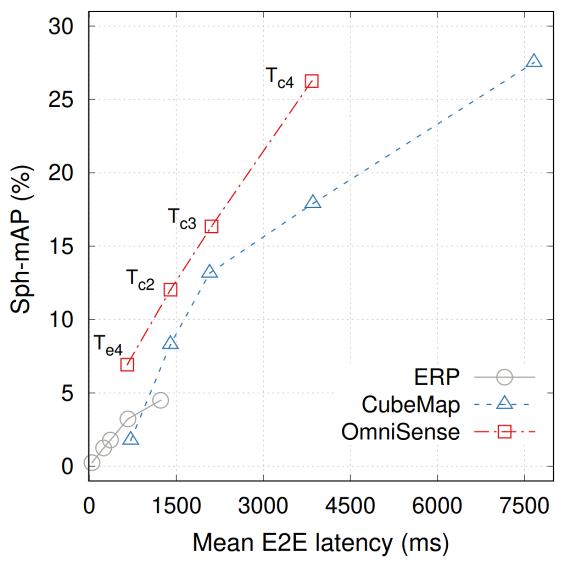
EVALUATION

Evaluation Results

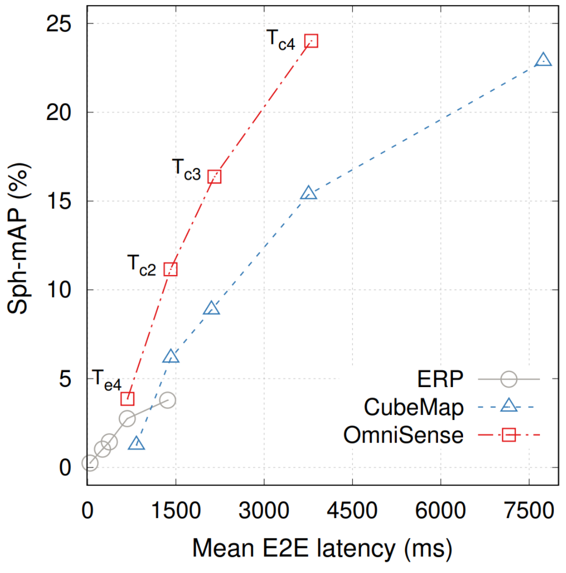
Overall performance comparisons of various methods on different videos

19.8 % - 114.6 % relative accuracy improvement

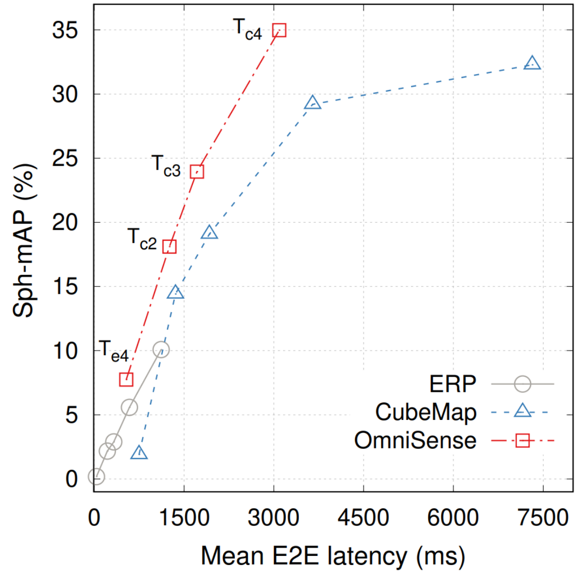
2.0 x – 2.4 x speedups



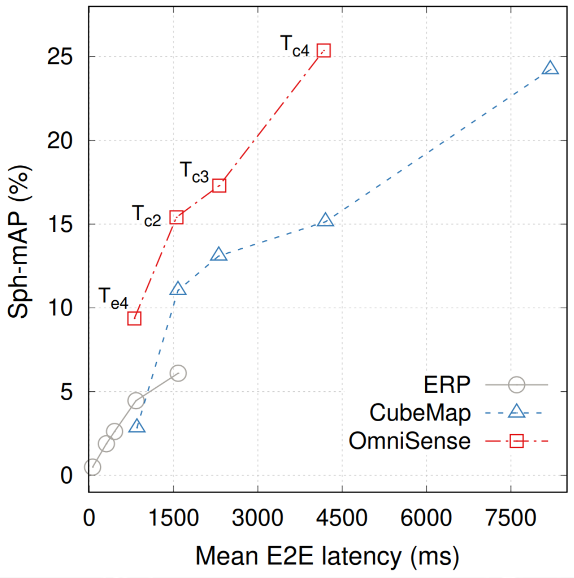
(a) Video: New-Orleans-drive



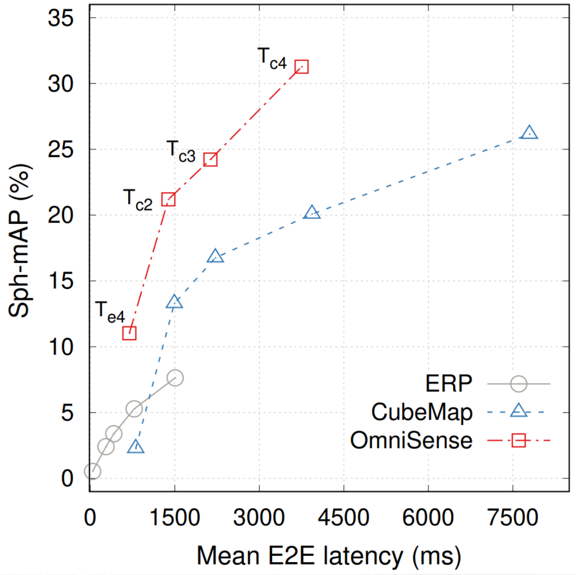
(b) Video: Chicago-drive



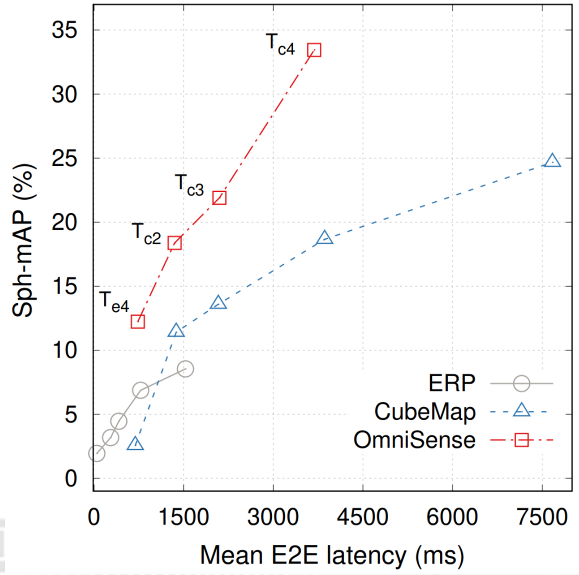
(c) Video: Expressway-drive



(d) Video: Sunny-walk1



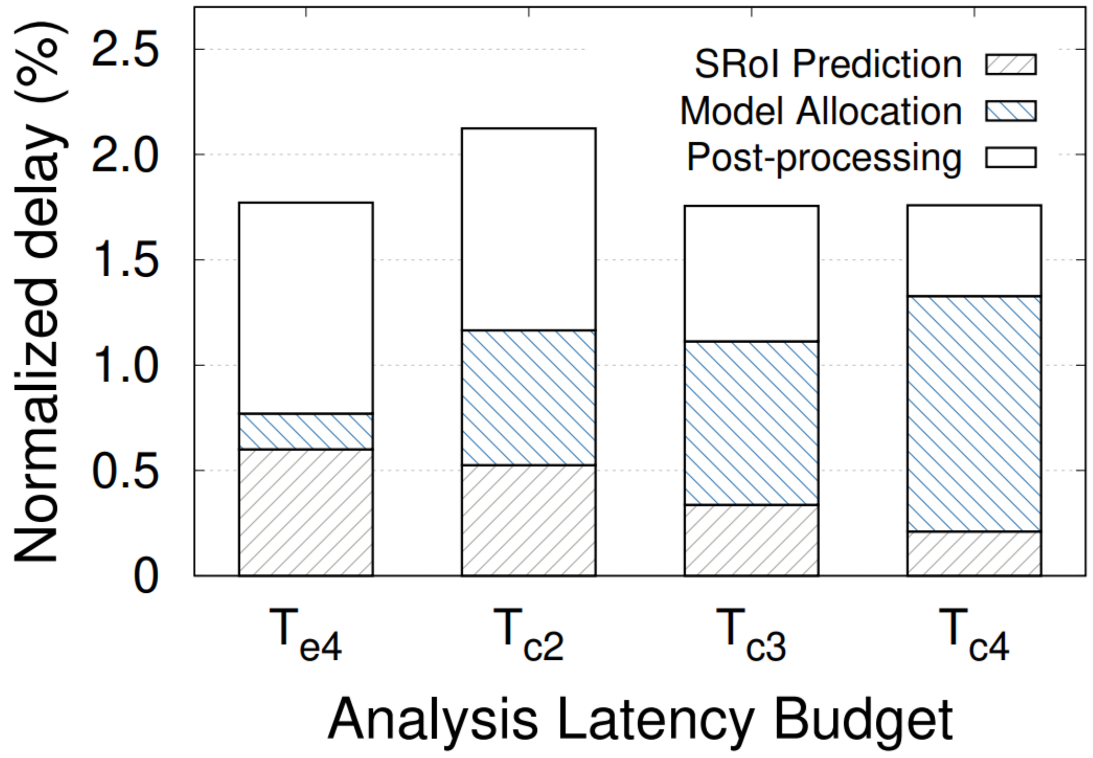
(e) Video: Sunny-walk2



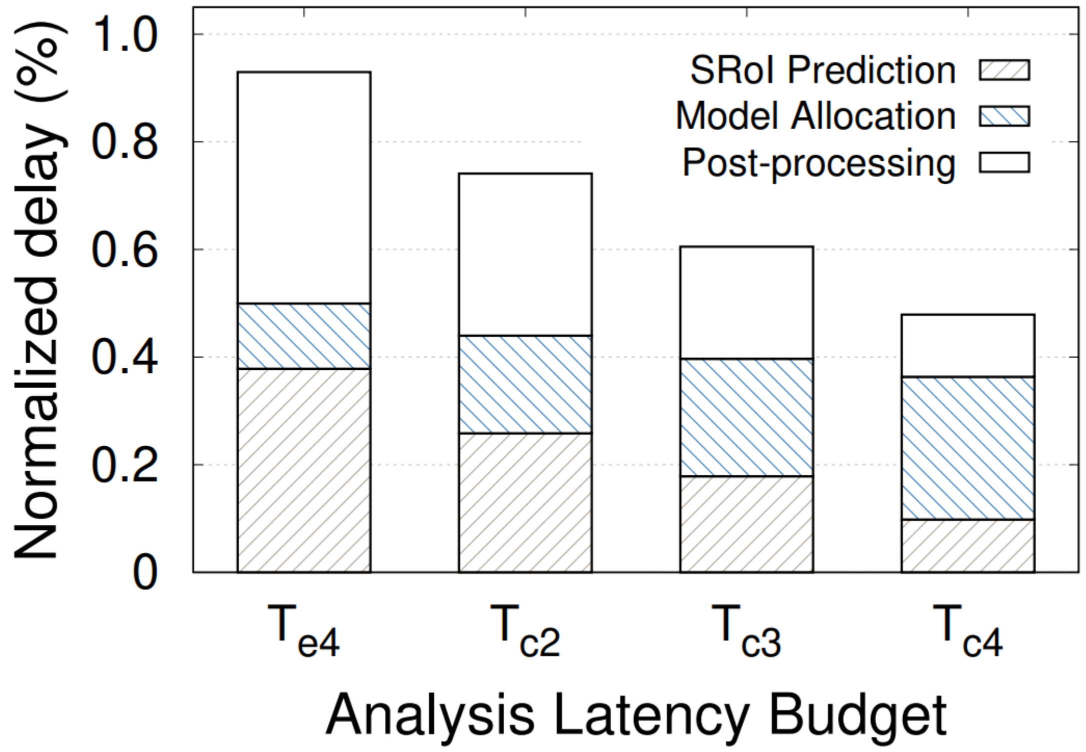
(f) Video: Cloudy-walk

EVALUATION

Evaluation Results



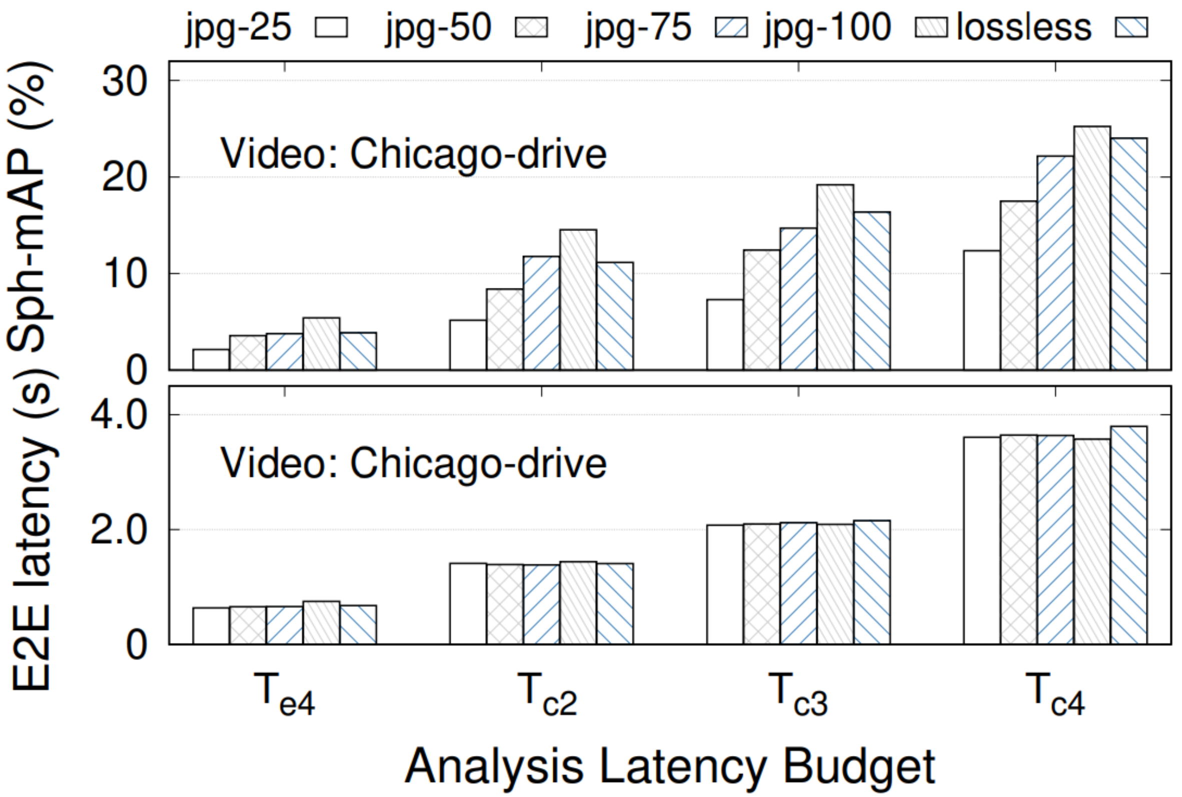
(a) video: Chicago-drive



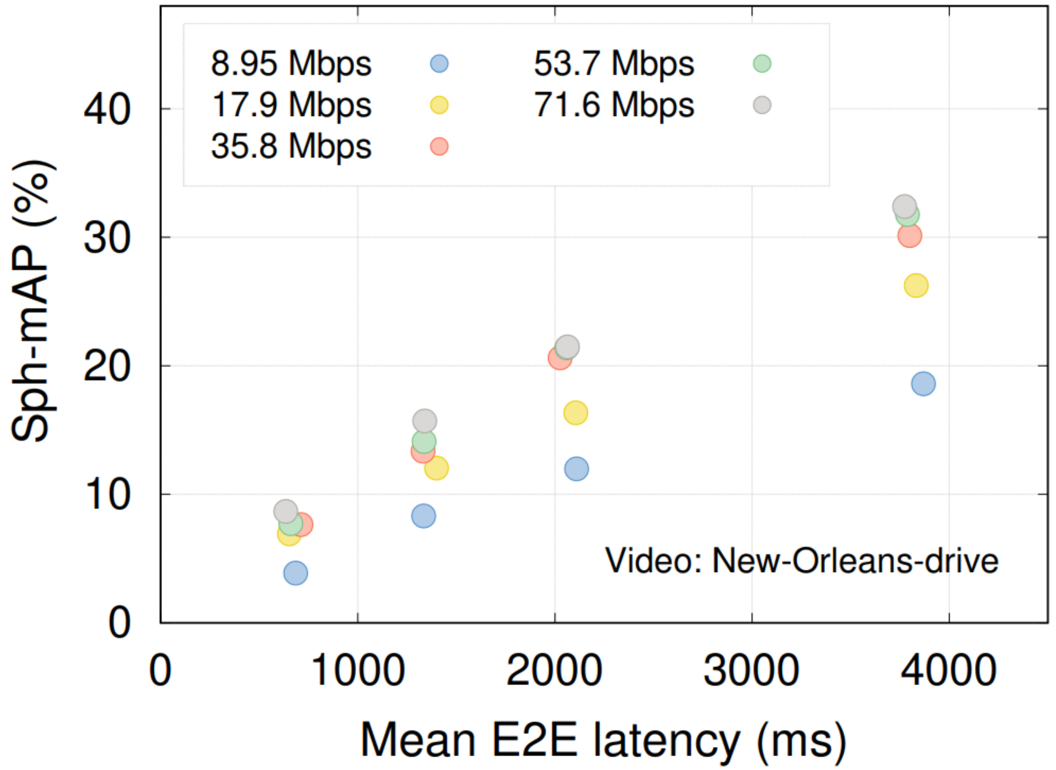
(b) video: Sunny-walk2

System overhead on the mobile device. The delays are normalized by the corresponding mean E2E latencies.

Evaluation Results



(a) Perspective image compression quality



(b) Network bandwidth

Sensitivity to the compression quality of perspective images and network throughput

SUMMARY

- Immersive video analytics will be essential in unlocking the full potential of 360-degree videos.
- Our analysis of 360-degree content characteristics reveals new resource-saving opportunities in online analytics.
- OmniSense achieves low-latency and high-accuracy immersive video analytics by fine-grained content-aware resource adaptation.

THANK YOU

Q & A