SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

EDGE-CLOUD COLLABORATIVE MULTIMEDIA ANALYSIS

Ivan V. Bajić

School of Engineering Science
Simon Fraser University
Burnaby, BC, Canada

IEEE ICME 2022 Tutorial

SPECIAL THANKS

People @ SFU Multimedia Lab (multimedia.fas.sfu.ca) who are/have been working on collaborative intelligence

Anderson de Andrade

Lior Bragilevsky

Hyomin Choi

Robert A. Cohen

Ashiv Hans Dhondea

Yalda Foroutan

Alon Harell

Elahe Hosseini

Suemin Lee

Saeed Ranjbar Alvar

Chamani Shiranthika

Mateen Ulhaq

OVERVIEW

Introduction

- Technological trends and emerging applications
- The case for collaborative intelligence

Part 1 – Theory

- Review of information theory: entropy, mutual information, data processing inequality
- Bounds on feature compressibility

Part 2 – Practical considerations

- Error resilience
- Feature compression
- Privacy
- Scalable feature coding
- Motion analysis

Part 3 – Standardization

- JPEG AI and MPEG-VCM (Video Coding for Machines)

Introduction

TECHNOLOGICAL TRENDS

Getty Images

Internet of Things (IoT)

- 125 billion IoT devices by 2030
- Market worth >$500 billion
- By 2025, IoT data volume 80 zettabytes

  (80×10^{21} = 80,000,000,000,000,000,000,000 bytes)
- Many kinds of devices:
  - Consumer products – digital assistants, home security cameras, smart appliances, …
  - Industry 4.0 – automation, smart factories, predictive maintenance, …
  - Logistics and fleet management – vehicles, ships, drones, aircraft, …
  - Infrastructure – traffic monitoring, video surveillance, smart buildings, …

https://techjury.net/blog/how-many-iot-devices-are-there/

TECHNOLOGICAL TRENDS

Shutterstock

Fifth Generation (5G) Communication Networks

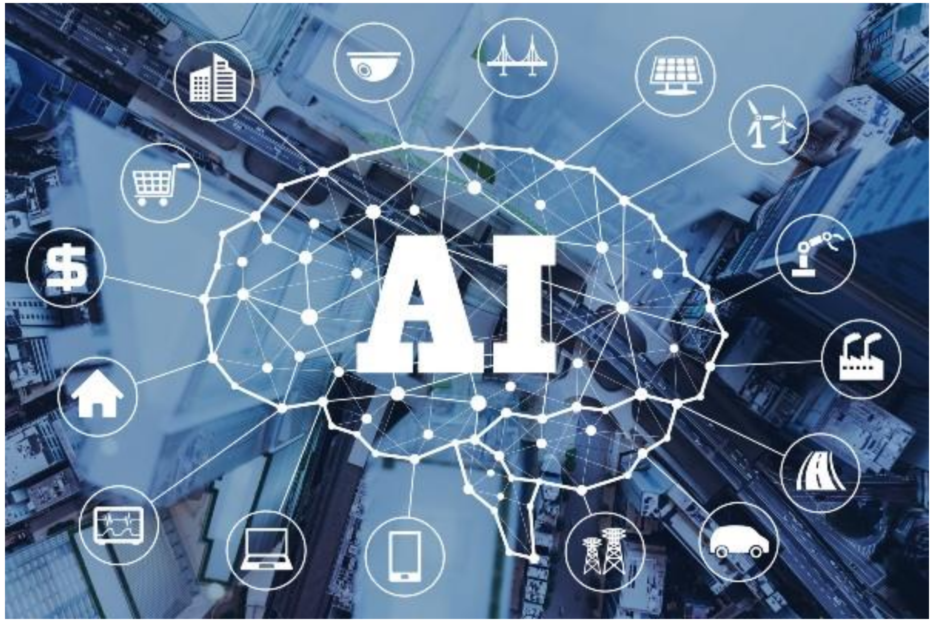
- Higher bandwidth, higher data rates
- Shorter range at higher frequencies
- Different types of cells
- Broad application areas:
  - Enhanced Mobile Broadband (eMBB) – improved services for mobile devices
  - Ultra-Reliable Low-Latency Communications (URLLC) – for “mission-critical” applications
  - Massive Machine-Type Communications (mMTC) – for “less critical” applications

Suitable for IoT applications

TECHNOLOGICAL TRENDS

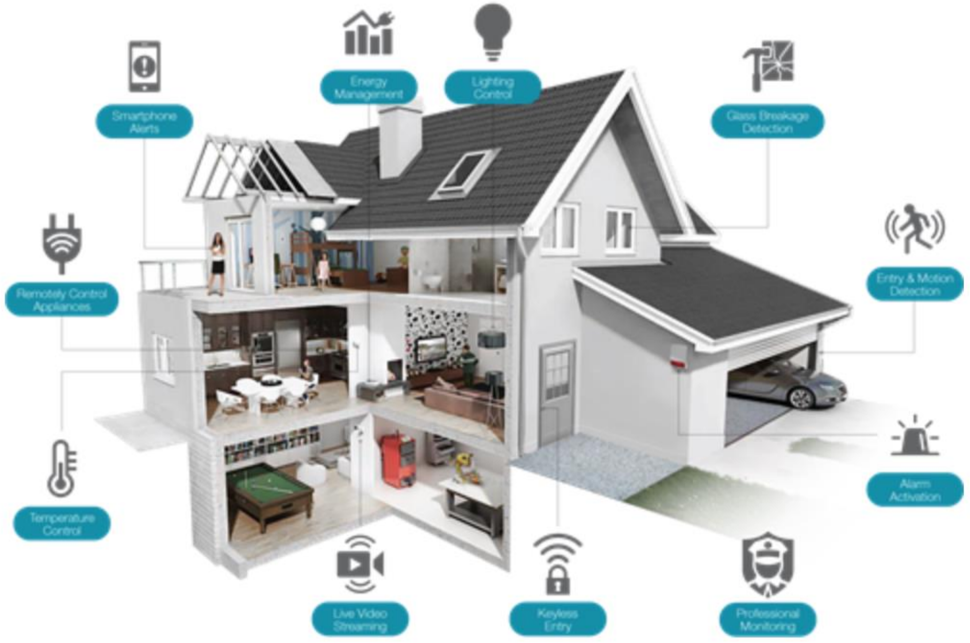
Artificial Intelligence (AI)

- Different, this time around
- Industry-driven, products on the market
- Facilitated by advances in computing technology, machine/deep learning, data availability
- Becoming indispensable in:
  - Computer vision and image processing
  - Speech and audio processing and analysis
  - Natural language processing and understanding
  - Robotics and automation
  - Medical diagnostics, …



nih.gov

EMERGING APPLICATIONS



medium.com

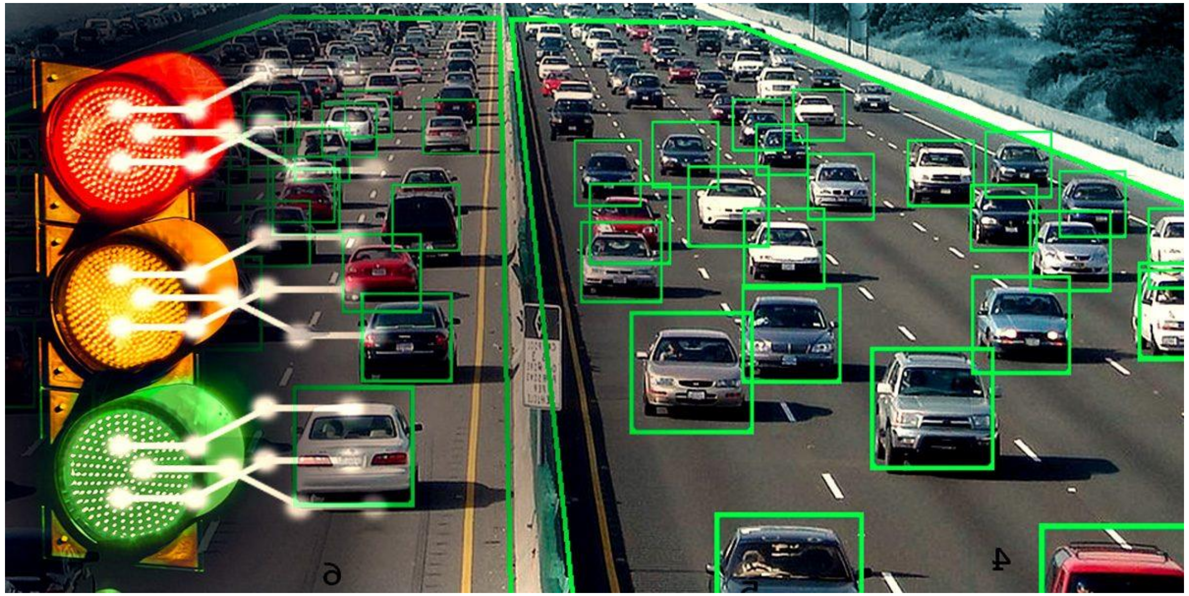
Smart home

- Many sensors around home to help make the home more comfortable, safe, and efficient
- Control entertainment, lighting, heating, cooling, security, predictive maintenance, etc.
- Smart speaker market alone > $17B by 2025
- “Silo” mode
  - Devices communicate with each other through local network, but not with outside world
  - Can still perform basic functions (control lights, security, leak detection, …)
- Connected mode
  - Full power of smart speakers (“What is the weather forecast for the weekend?”)
  - Enhanced security (“There is a fire in the neighborhood”), efficiency and comfort (“avoid highway on your way to work due to heavy traffic”), …

EMERGING APPLICATIONS

Traffic monitoring & management

- Cameras (and other sensors) along roads and intersections
- Counting vehicles, pedestrians, etc.
- Estimating their speed, traffic intensity, detecting violations and emergencies
- Control traffic lights to manage traffic



entrackr.com

- “Silo” mode
  - Each camera controls its own traffic light
- Connected mode
  - Aggregate data from multiple cameras within a neighborhood to improve awareness and make better decisions

EMERGING APPLICATIONS

Autonomous driving

- Cameras and other sensors mounted on the vehicle to help understand its surroundings
- Detecting vehicles, bikes, pedestrians, traffic lights and signs, speed bumps, etc.
- Estimated ~ 2 kWh for on-board processing of sensor data (2.5 kWh in cities)
- “Silo” mode
  - Full autonomy, but energy cost high
- Connected mode (especially appropriate in cities)
  - Save energy by offloading some of the “intelligence” to the cloud
  - Benefit from other sensors in the vicinity (e.g., children playing around the corner)

aarp.org

D. Richart, Autonomous Cars’ Big Problem: The energy consumption of edge processing reduces a car’s mileage with up to 30%, May 2019. https://medium.com/@teraki/energy-consumption-required-by-edge-computing-reduces-a-autonomous-cars-mileage-with-up-to-30-46b6764ea1b7

EMERGING APPLICATIONS

- Previous examples (and many more) make use of advanced sensing and processing capabilities of edge devices
- In many cases, the system can operate in the “silo” mode or in a connected mode
- “Silo” mode
  - Most autonomous
  - No need to communicate with the rest of the world
- Connected mode
  - Requires communication, but…
  - Enables more sophisticated applications
  - Potential for energy savings
  - Several ways to run this mode, depending on where “intelligence” is deployed

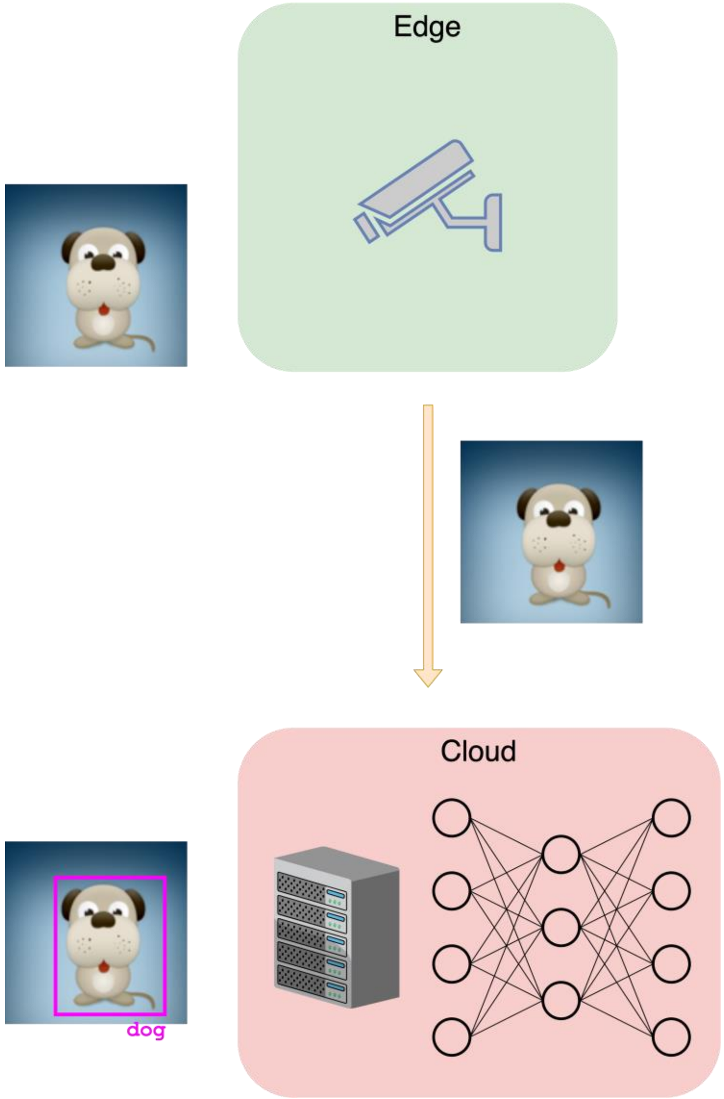
CLOUD-BASED INTELLIGENCE

The traditional approach

- Edge sensor captures the signal
- Signal transmitted to the cloud
- Analysis ("intelligence") performed in the cloud
- Result sent back to the edge (if needed) or to other systems in the cloud

Challenges:

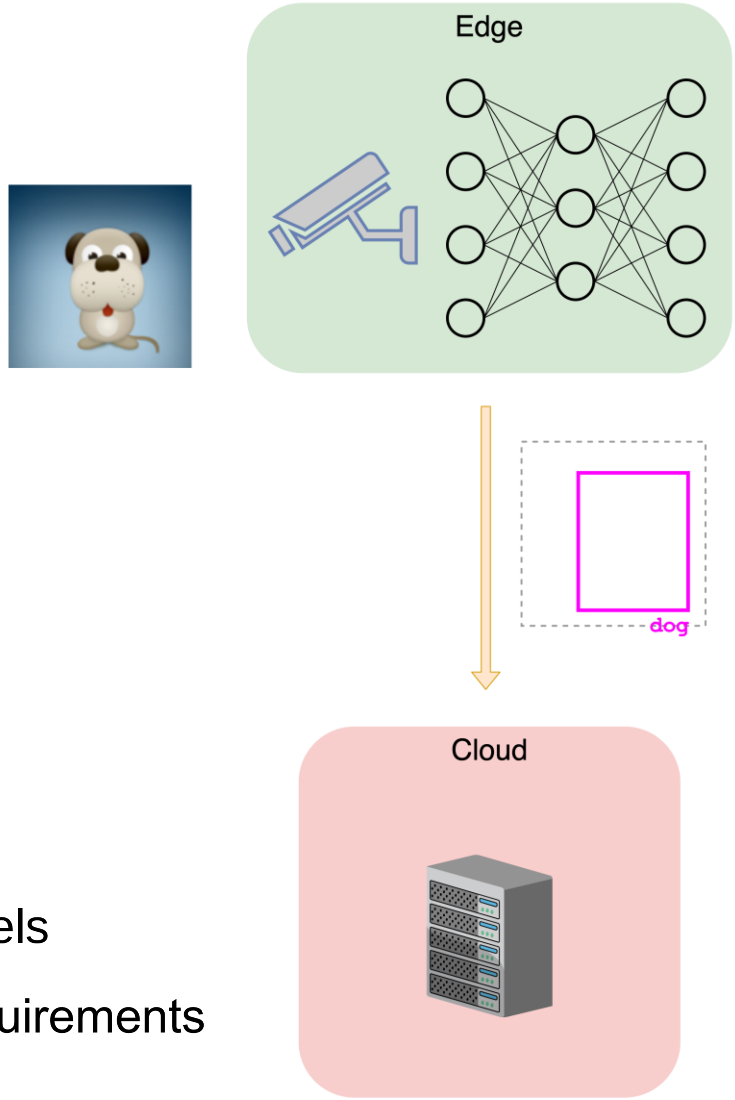
- Concerns over privacy
- Does not take full advantage of capabilities of modern edge devices



EDGE-BASED INTELLIGENCE

The new approach

- Analysis ("intelligence") performed at the edge
- Only the result sent to the cloud, could also operate in "silo" mode
- Makes the edge device "smart"
- Addresses some privacy concerns



Challenges:

- Can be energy-intensive (at the edge)
- Model complexity limited by the resources of the edge device
  - Cloud will always be able to host larger, more complex models
- What if more then one type of analysis ("task") is needed, or requirements change over time?

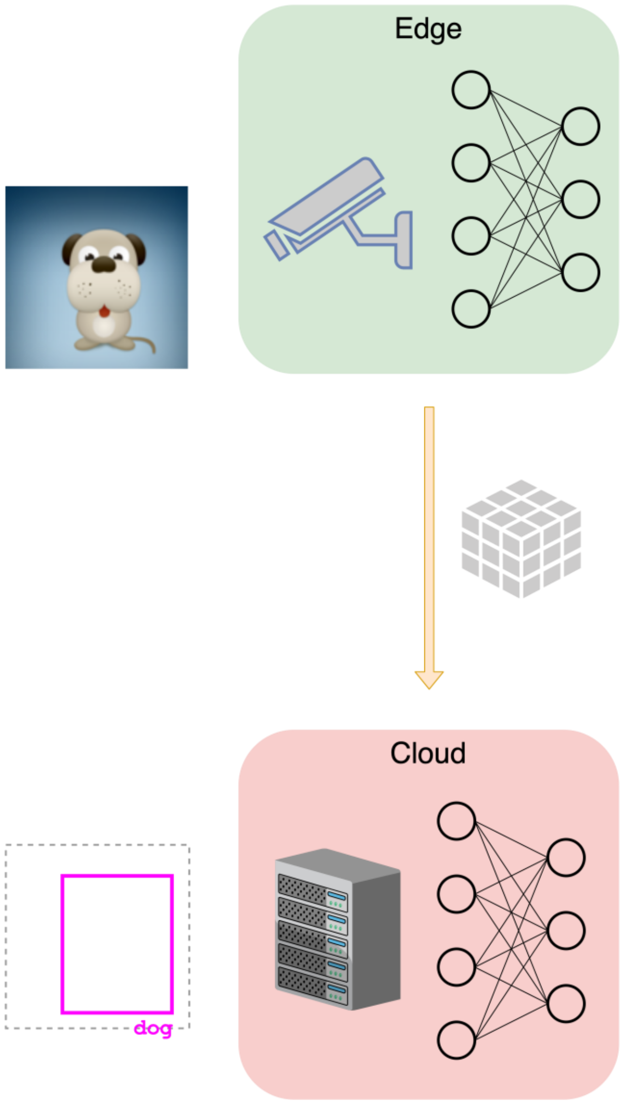
EDGE-CLOUD COLLABORATIVE INTELLIGENCE

The future approach

- Covers the spectrum between cloud-only and edge-only extremes
- Part of “intelligence” at the edge, other part at the cloud
- Signal features sent to the cloud, analysis completed there
- Able to address privacy concerns
- Able to scale to available resources

Challenges:

- Requires new science and engineering to understand tradeoffs
- Lack of clear design guidelines (true for all AI)



THE CASE FOR COLLABORATIVE INTELLIGENCE

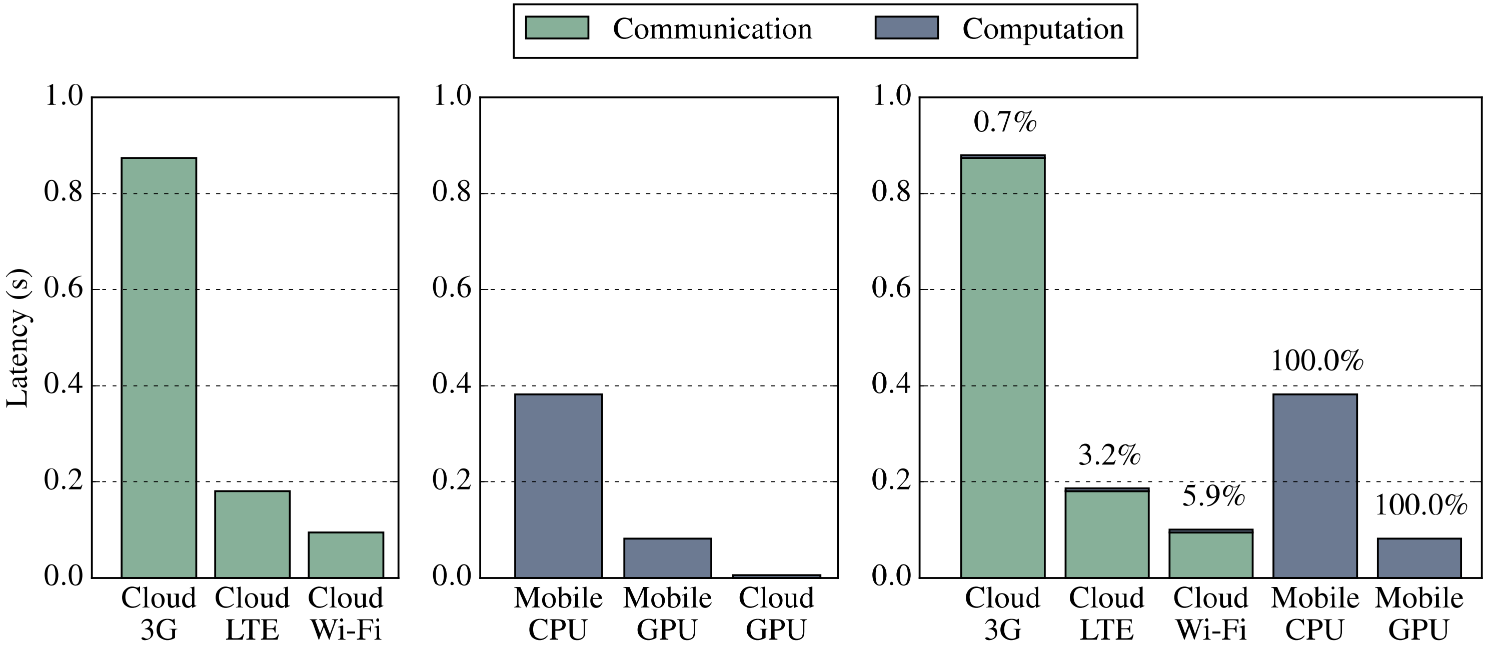
Neurosurgeon study

- Measured energy (@ edge device) and latency for cloud-based, edge-based, and distributed model deployment
- Considered both CPU (Arm Cortex A15) and GPU (NVIDIA Kepler) @ edge
- Considered various models and applications
  - Image classification
  - Face recognition
  - Handwritten digit recognition
  - Speech recognition
  - Speech tagging
  - …

Y. Kang et al., "Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge," Proc. ACM ASPLOS'17, pp. 615–629, 2017.

THE CASE FOR COLLABORATIVE INTELLIGENCE

Some results from the Neurosurgeon study



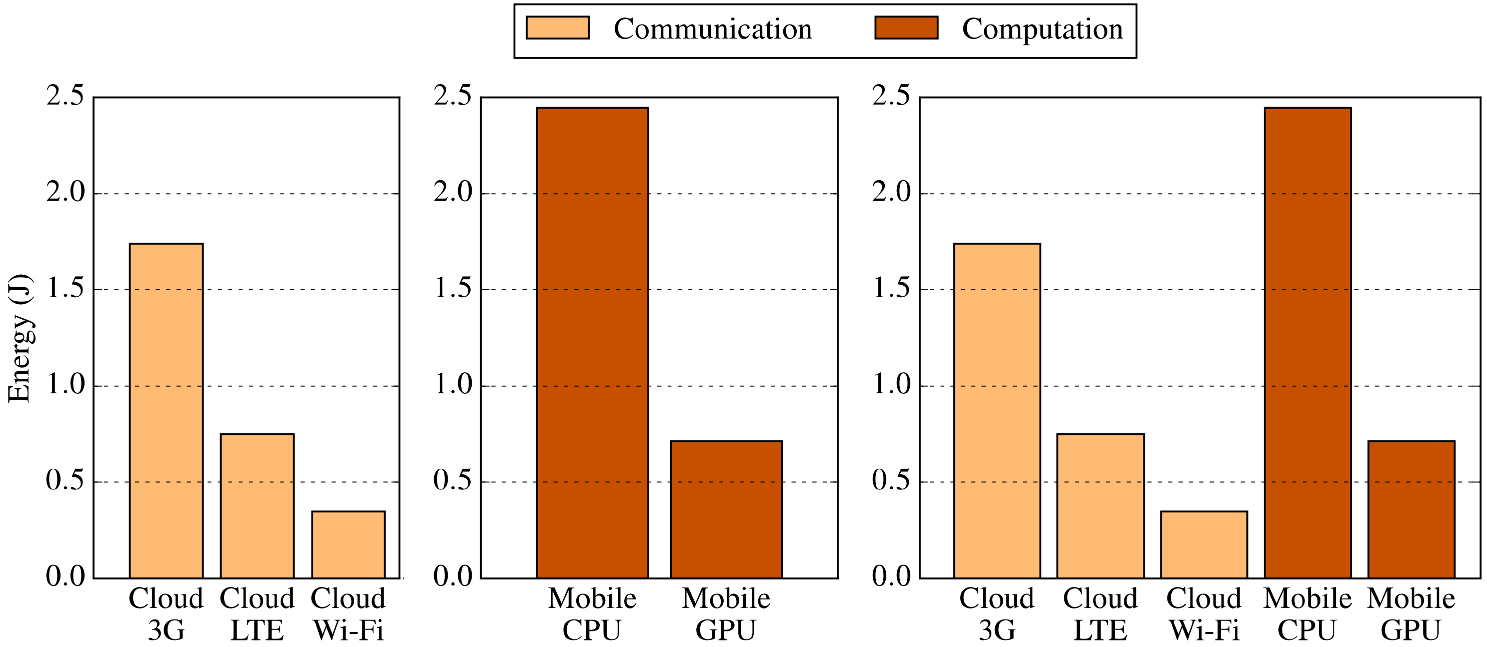
(a) Communication (b) Computation (c) End-to-end

Overall latency depends on type of connection and resources available at the edge device

Y. Kang et al., "Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge," Proc. ACM ASPLOS'17, pp. 615–629, 2017.

THE CASE FOR COLLABORATIVE INTELLIGENCE

Some results from the Neurosurgeon study



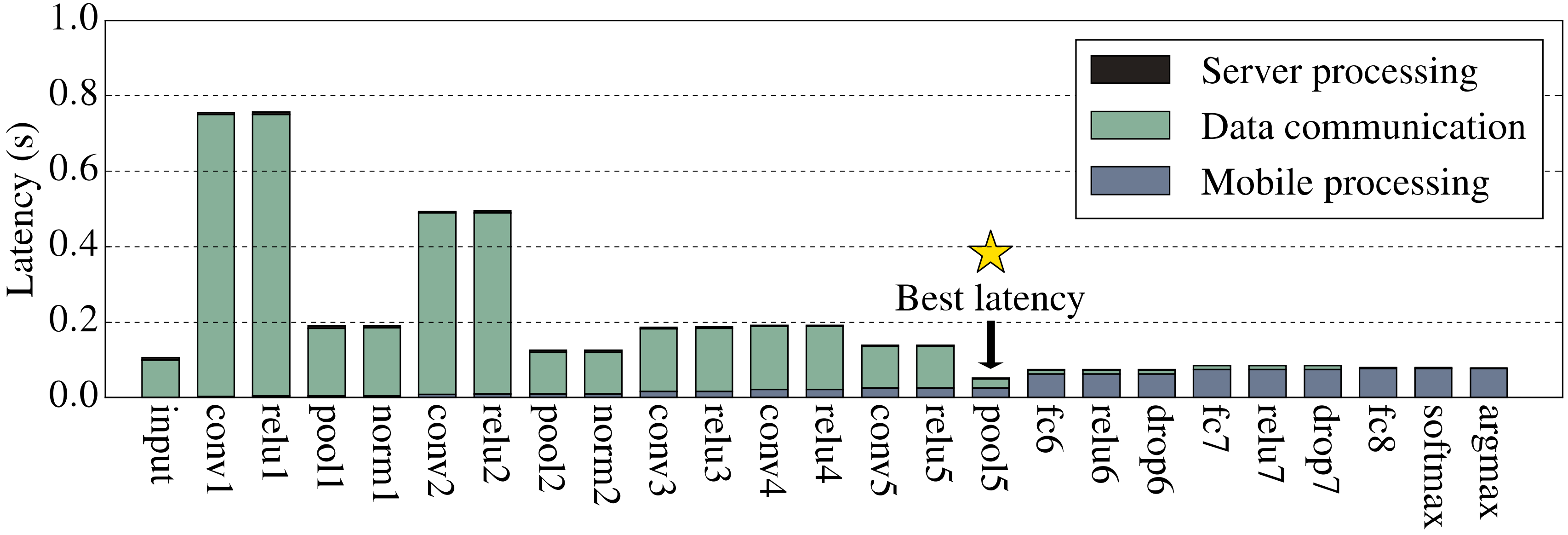
(a) Communication **(b) Computation** **(c) Total**

Energy @ edge device also depends on type of connection and resources available at the edge

Y. Kang et al., "Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge," Proc. ACM ASPLOS'17, pp. 615–629, 2017.

THE CASE FOR COLLABORATIVE INTELLIGENCE

Some results from the Neurosurgeon study



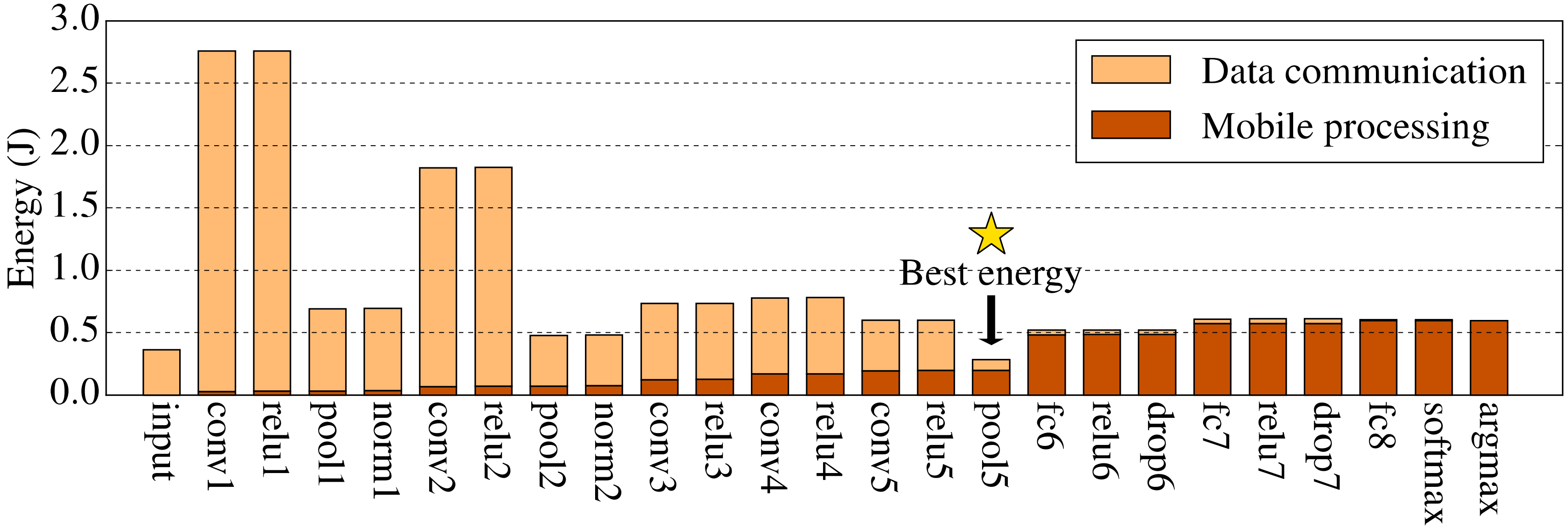
(a) AlexNet latency

When considering end-to-end latency, running part of the model @ edge and remainder in the cloud often the best solution (above: using edge GPU and WiFi connection)

Y. Kang et al., "Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge," Proc. ACM ASPLOS'17, pp. 615–629, 2017.

THE CASE FOR COLLABORATIVE INTELLIGENCE

Some results from the Neurosurgeon study



(b) AlexNet energy consumption

When considering energy @ edge, running part of the model @ edge and remainder in the cloud often the best solution (above: using edge GPU and WiFi connection)

Y. Kang et al., "Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge," Proc. ACM ASPLOS'17, pp. 615–629, 2017.

THE CASE FOR COLLABORATIVE INTELLIGENCE

Some conclusions from the Neurosurgeon study

- In terms of end-to-end latency and energy @ edge, it is often best to run part of the AI model at the edge, and remainder in the cloud → collaborative intelligence
- Optimal partition depends on many factors:
  - ○ The architecture of the AI model
  - ○ Hardware @ edge
  - ○ Type of connection
  - ○ …
- Optimal partitioning point for energy might be different from that for latency

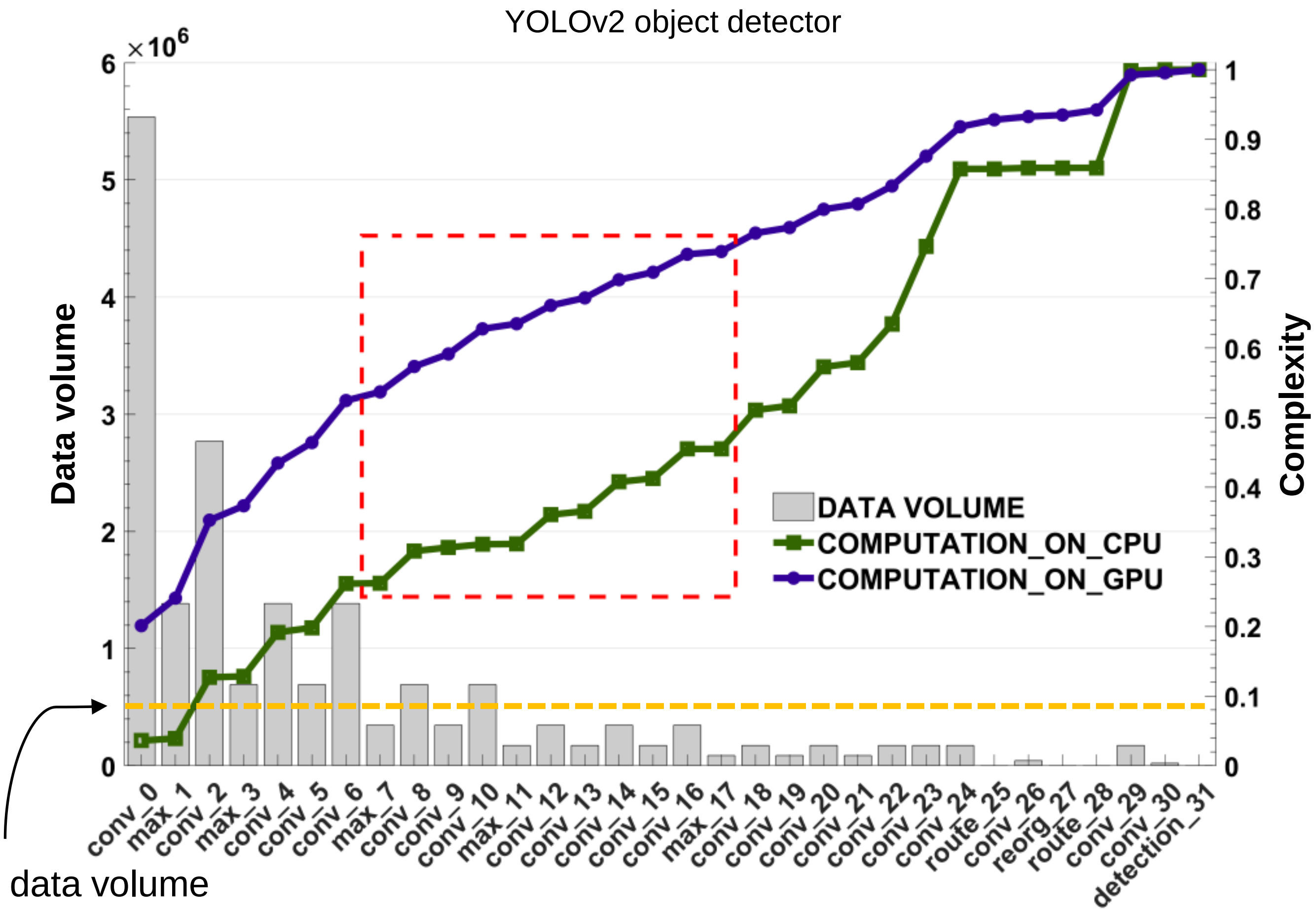
Y. Kang et al., "Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge," Proc. ACM ASPLOS'17, pp. 615–629, 2017.

THE CASE FOR COLLABORATIVE INTELLIGENCE

Explaining the Neurosurgeon results:

- In many deep models, data volume decreases towards the output
- Less data → fewer bits to send
- Fewer bits → less energy used by radio
- Energy saved on radio may compensate for energy spent on extra computation

- Bits to be sent are the key!


YOLOv2 object detector

Input data volume

H. Choi and I. V. Bajić, "Deep feature compression for collaborative object detection," Proc. IEEE ICIP'18, pp. pp. 3743-3747, 2018.

Questions?

Part 1

Theory

REVIEW OF RELEVANT INFORMATION THEORY

Entropy

- Let X be a discrete random variable taking on values x in some sample space $\mathcal{X}$
- The entropy of X (in bits) is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} p(X = x) \cdot \log_2 p(X = x)$$

- Entropy is a measure of uncertainty (randomness)
- Entropy is the limit of lossless compressibility
- Examples:
    - Fair coin: $\mathcal{X} = \{\text{Heads}, \text{Tails}\}, \quad p(X = \text{Heads}) = p(X = \text{Tails}) = 1/2, \quad H(X) = 1$ bit
    - Fair die: $\mathcal{X} = \{1, 2, 3, 4, 5, 6\}, \quad p(X = 1) = \cdots = p(X = 6) = 1/6, \quad H(X) = \log_2 6 = 2.58$ bits

T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd Edition, Wiley, 2006.

REVIEW OF RELEVANT INFORMATION THEORY

Mutual information

- Let X and Y be discrete random variables taking on values in sample spaces $\mathcal{X}$ and $\mathcal{Y}$
- The mutual information (MI) between X and Y (in bits) is defined as

$$I(X;Y) = \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p((X,Y)=(x,y)) \cdot \log_2 \frac{p((X,Y)=(x,y))}{p(X=x)\cdot p(Y=y)}$$

- MI is a measure of statistical dependence (linear or nonlinear) between X and Y
- MI is the amount of information that X carries about Y, and vice versa
- Examples:
  - X and Y independent $\Leftrightarrow$ $I(X;Y)=0$
  - $I(X;X)=H(X)$: mutual information between X and itself is its own entropy

T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd Edition, Wiley, 2006.

REVIEW OF RELEVANT INFORMATION THEORY

Markov chain

- A sequence of random variables $X \rightarrow Y \rightarrow Z$ is a Markov chain if Z is conditionally independent of X, given Y

$$\begin{aligned} p(x,y,z) &\overset{\text{always}}{=} p(x) \cdot p(y|x) \cdot p(z|y,x) \\ &\underset{\text{if Markov chain}}{=} p(x) \cdot p(y|x) \cdot p(z|y) \end{aligned}$$

- If Z is a function of Y, i.e. $Z = f(Y)$, then $X \rightarrow Y \rightarrow Z$ is a Markov chain
  - Since Z is computed from Y, it does not depend on X (when Y is given)

T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd Edition, Wiley, 2006.

REVIEW OF RELEVANT INFORMATION THEORY

Data processing inequality (DPI)

- If $X \rightarrow Y \rightarrow Z$ is a Markov chain, then

$$I(X;Y) \geq I(X;Z)$$

- Downstream variable (Z) has no more information about input (X) than an upstream variable (Y)
    - Processing cannot increase (mutual) information

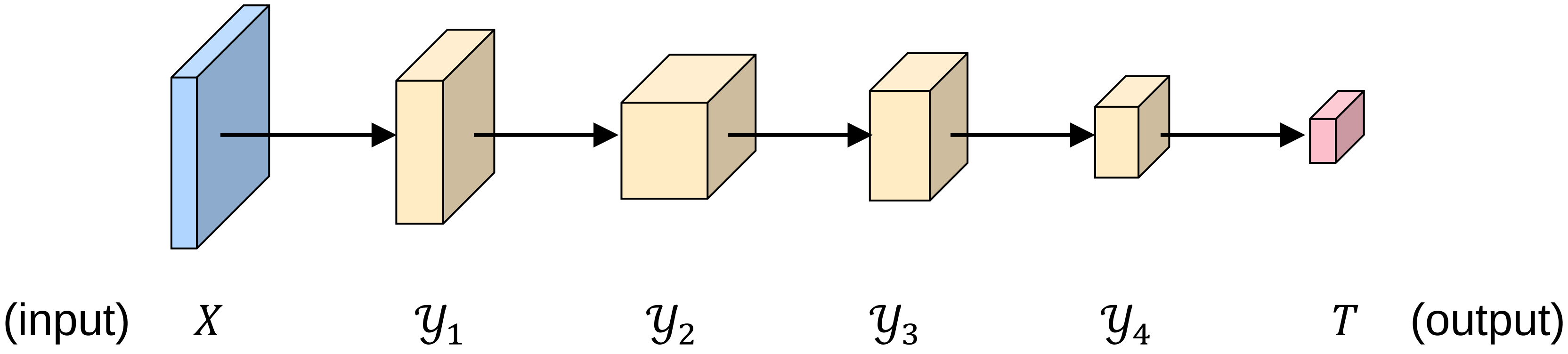
- Extended version of DPI: if $X \rightarrow Y \rightarrow Z \rightarrow W$ is a Markov chain, then

$$I(Y;Z) \geq I(X;W)$$

T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd Edition, Wiley, 2006.
R. W. Yeung, *A First Course in Information Theory*, Springer, 2006.

NEURAL NETWORK LAYERS FORM MARKOV CHAINS

- $\mathcal{Y}_i$ = output of the i-th layer in a feedforward neural network

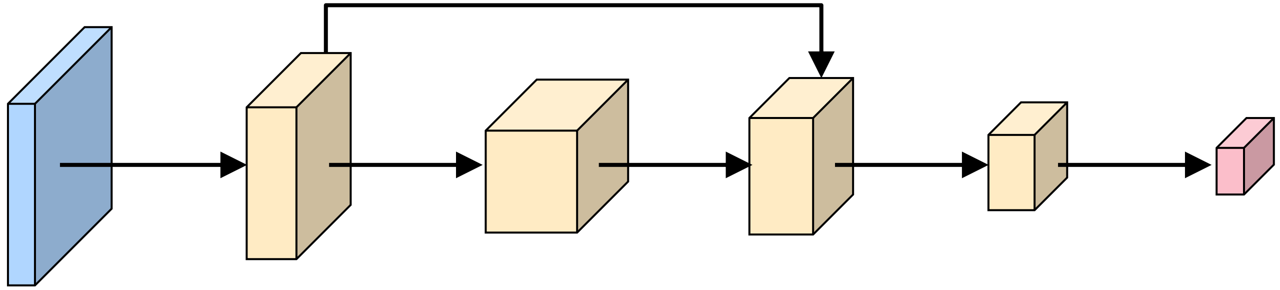


- $X \rightarrow \mathcal{Y}_1 \rightarrow \mathcal{Y}_2 \rightarrow \mathcal{Y}_3 \rightarrow \mathcal{Y}_4 \rightarrow T$ is a Markov chain
  - So is any chain $X \rightarrow \mathcal{Y}_i \rightarrow \mathcal{Y}_j \rightarrow T$ for $i < j$
  - True for dense layers, convolutional layers, pooling layers, etc.

N. Tishby and N. Zaslavsky, “Deep Learning and the Information Bottleneck Principle,” Proc. IEEE Information Theory Workshop (ITW), Mar. 2015.

NEURAL NETWORK LAYERS FORM MARKOV CHAINS

- What about skip connections?



(input) X $\mathcal{Y}_1$ $\mathcal{Y}_2$ $\mathcal{Y}_3$ $\mathcal{Y}_4$ T (output)

- $X \rightarrow \mathcal{Y}_1 \rightarrow \mathcal{Y}_2 \rightarrow \mathcal{Y}_3$ is **not** a Markov chain
  - Y_3 depends on both $\mathcal{Y}_2$ and $\mathcal{Y}_1$, not just $\mathcal{Y}_2$
  - However, $X \rightarrow \mathcal{Y}_1 \rightarrow \mathcal{Y}_3$ is a Markov chain
  - Markovity still holds “across” skip connections, but not “under” them

LOSSLESS FEATURE COMPRESSIBILITY

Claim: In a non-generative feedforward neural network, in terms of lossless compression, intermediate features are at least as compressible as the network's input.

Proof (sketch):

- Let $\mathcal{Y} = \{\mathcal{Y}_i : 1 \leq i \leq L\}$ be a set of some intermediate layer outputs (features)
- Decompose mutual information between input X and $\mathcal{Y}$ as

$$\begin{aligned} I(X;\mathcal{Y}) &= H(\mathcal{Y}) - H(\mathcal{Y} \mid X) \\ &= H(\mathcal{Y}) \end{aligned}$$

0, because $\mathcal{Y}$ is a function of X

- Note that $X \to X \to \mathcal{Y}$ is a Markov chain and apply DPI

$$H(X) = I(X;X) \geq I(X;\mathcal{Y}) = H(\mathcal{Y})$$

- So, $H(\mathcal{Y})$ is no larger than $H(X)$ $\implies$ features $\mathcal{Y}$ at least as compressible (losslessly) as input X

H. Choi and I. V. Bajić, "Scalable image coding for humans and machines," IEEE Trans. Image Processing, vol. 31, pp. 2739-2754, 2022.

LOSSLESS FEATURE COMPRESSIBILITY

- Intermediate features being more compressible than the input is good news for collaborative intelligence!
  - Bits saved on radio will help compensate for extra computation
  - End-to-end latency can be reduced

- But lossless compressibility is very limiting
  - Lossy compression gives much higher compression ratios
  - Practical image, video, audio compression are all lossy
  - Can we extend this result to lossy compression?

REVIEW OF RELEVANT INFORMATION THEORY

Rate-distortion function

- Let X be a random variable and $\hat{X}$ be its “quantized” version according to some conditional probability distribution $p(\hat{x} \mid x)$
- Let $d(\hat{x}, x)$ be a distortion metric – how much $\hat{x}$ differs from x
- For a given distortion level D, define set $\mathcal{P}_X(D)$ of conditional distributions as

$$\mathcal{P}_X(D) = \{p(\hat{x} \mid x) \ : \ \underbrace{p(x) \cdot p(\hat{x} \mid x) \cdot d(\hat{x}, x)}_{\mathbb{E}\left[d(\hat{X}, X)\right]} \le D\}$$

- Rate-distortion (RD) function for X is given by

$$R_X(D) = \min_{p(\hat{x} \mid x) \in \mathcal{P}_X(D)} I(X; \hat{X})$$

- $R_X(D)$ is the minimum rate (in bits) at which you can encode X without incurring distortion $> D$

T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd Edition, Wiley, 2006.

LOSSY FEATURE COMPRESSIBILITY

- In order to use RD theory in our case, we need some modifications

$$\begin{tikzcd} X \arrow[r, "g"'] \arrow[rr, bend left, "f"] & \mathcal{Y} \arrow[r, "h"'] & T \end{tikzcd}$$

- When we compress input X, we care about what happens to the output T

$$\mathcal{P}_X(D) = \left\{p(\hat{x} \mid x) \;\; : \mathbb{E}\left[d\left(f(\hat{X}), f(X)\right)\right] \leq D\right\}$$

- Similarly, when we compress features $\mathcal{Y}$, we care about what happens to the output T

$$\mathcal{P}_{\mathcal{Y}}(D) = \left\{p(\hat{y} \mid y) \;\; : \mathbb{E}\left[d\left(h(\hat{\mathcal{Y}}), h(\mathcal{Y})\right)\right] \leq D\right\}$$

LOSSY FEATURE COMPRESSIBILITY

- We can now define the RD function for the input

$$R_X(D) = \min_{p(\hat{x}\,|\,x)\,\in\,\mathcal{P}_X(D)} I(X;\hat{X})$$

and the RD function for the features

$$R_{\mathcal{Y}}(D) = \min_{p(\hat{y}\,|\,y)\,\in\,\mathcal{P}_{\mathcal{Y}}(D)} I(\mathcal{Y};\hat{\mathcal{Y}})$$

- In both cases, distortion is measured at the output of the network
- Distortion metric can be any metric appropriate for the network’s task, e.g.
  - Mean Squared Error for regression tasks
  - Cross-entropy or accuracy for classification tasks
  - …

LOSSY FEATURE COMPRESSIBILITY

Claim: In a non-generative feedforward neural network, in terms of lossy compression, intermediate features are at least as compressible as the network’s input.

$$R_Y(D) \leq R_X(D)$$

Proof (sketch):

- Let D be given and let $p^*(\hat{x} \mid x)$ be optimal for input compression (achieves $R_X(D)$)
- Draw inputs $X \sim p(x)$ and process each input x in two ways as follows

$$\begin{tikzcd}
 & & y \\
x \arrow[urr, "g"] \arrow[r, "p^*(\hat{x} \mid x)"'] & \hat{x} \arrow[r, "g"'] & \tilde{y}
\end{tikzcd}$$

- For each x, obtain y and $\tilde{y}$
- Define $q(\tilde{y} \mid y)$ by pairing up y and $\tilde{y}$

H. Choi and I. V. Bajić, ”Scalable image coding for humans and machines," IEEE Trans. Image Processing, vol. 31, pp. 2739-2754, 2022.

LOSSY FEATURE COMPRESSIBILITY

Proof (sketch, continued):

- Show $q(\tilde{y}|\, y) \in \mathcal{P}_{\mathcal{Y}}(D)$, i.e., satisfies distortion constraint for D
  - Easy to show because $q(\tilde{y}|\, y)$ is derived from $p^*(\hat{x} \mid x) \in R_X(D)$, which satisfies distortion constraint for D
- Apply DPI to Markov chain $\tilde{\mathcal{Y}} \rightarrow \hat{X} \rightarrow X \rightarrow \mathcal{Y}$ to show

$$I\left(\mathcal{Y};\tilde{\mathcal{Y}}\right) \leq \ I(X;\hat{X})$$

- When $p^*(\hat{x} \mid x)$ is used to generate $\hat{X}$, the above inequality becomes

$$I\left(\mathcal{Y};\tilde{\mathcal{Y}}\right) \leq R_X(D)$$

- So we have found one distribution $q(\tilde{y}|\, y) \in \mathcal{P}_{\mathcal{Y}}(D)$ that achieves $I\left(\mathcal{Y};\tilde{\mathcal{Y}}\right)$ below $R_X(D)$. Therefore

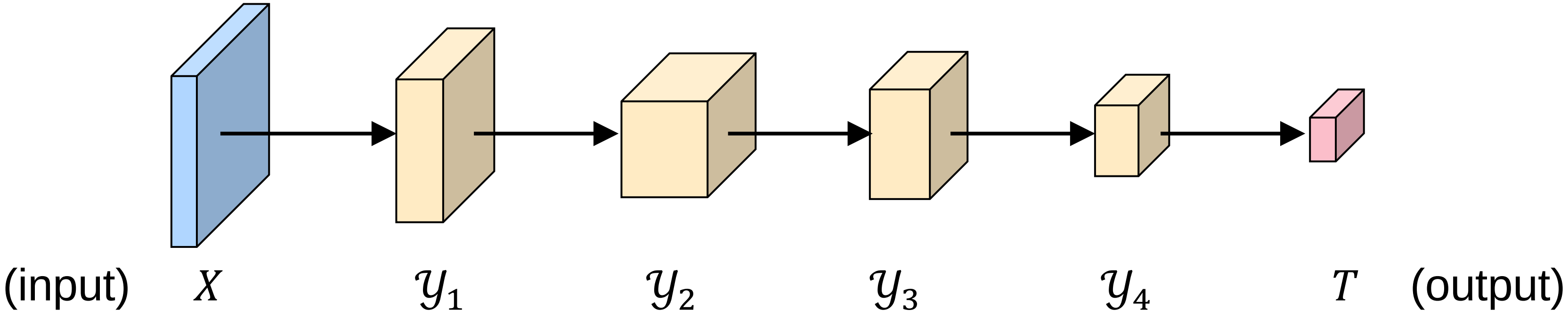
$$R_{\mathcal{Y}}(D) = \min_{p(\hat{y} \mid y) \in \mathcal{P}_{\mathcal{Y}}(D)} I(\mathcal{Y};\hat{\mathcal{Y}}) \ \leq \ R_X(D)$$

H. Choi and I. V. Bajić, "Scalable image coding for humans and machines," IEEE Trans. Image Processing, vol. 31, pp. 2739-2754, 2022.

DEEPER MEANS MORE COMPRESSIBLE

Claim: In a non-generative feedforward neural network, deeper layers are more compressible.

$$H(\mathcal{Y}_i) \le H(\mathcal{Y}_j) \quad \text{and} \quad R_{\mathcal{Y}_i}(D) \le R_{\mathcal{Y}_j}(D) \quad \text{for } i < j$$



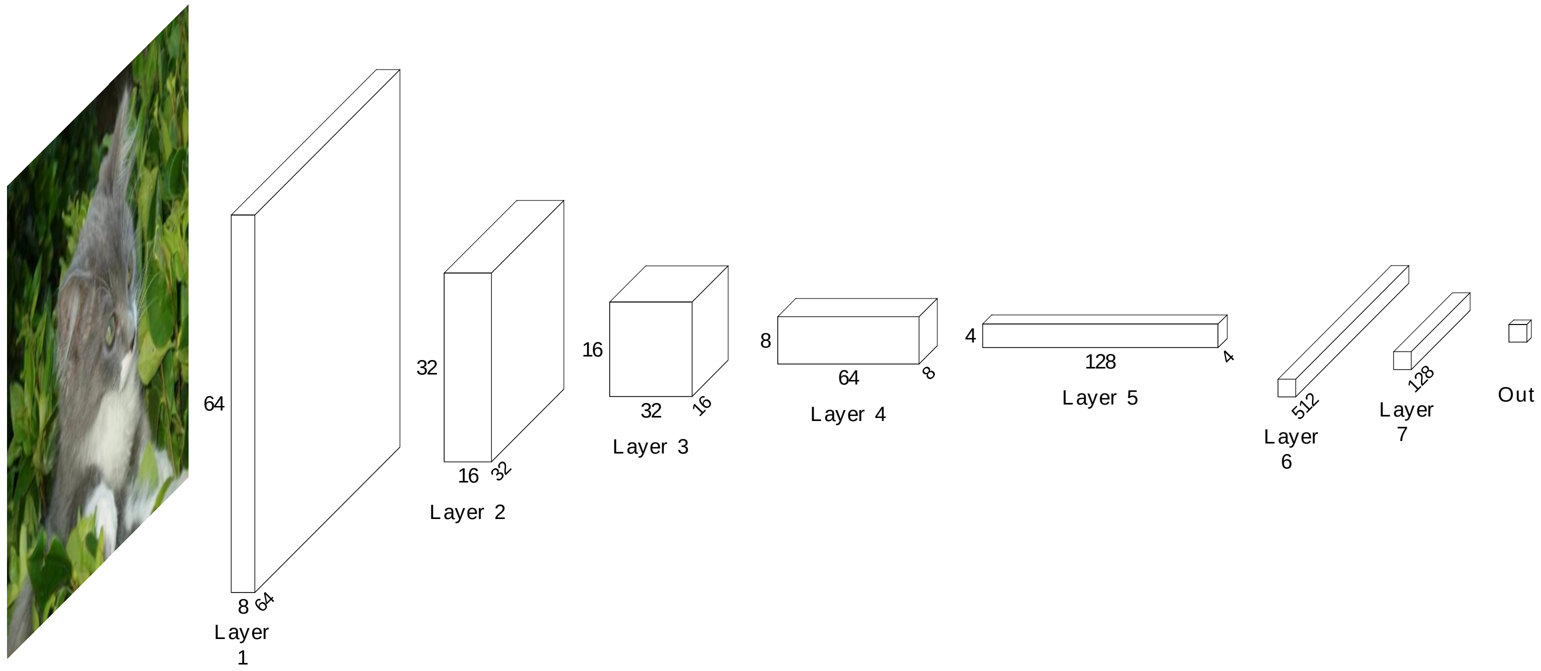
Proof (sketch): Follows from previous proofs by replacing X with $\mathcal{Y}_i$ and $\mathcal{Y}$ with $\mathcal{Y}_j$

SUMMARY OF FEATURE COMPRESSIBILITY

- Theory shows that intermediate features are at least as compressible as the network’s input
- This is true for any non-generative feedforward network, regardless of what its task is
- When optimally compressed, fewer bits will be sent in a collaborative intelligence approach compared to conventional cloud-based approach
- This bit saving, if large enough, will lead to lower latency and pay off for extra computation
- But:
  - Theory talks about limits; practical codecs might be far from those limits
  - Theory shows what is possible, but not how to get there
  - Ideal for grant proposals 😀

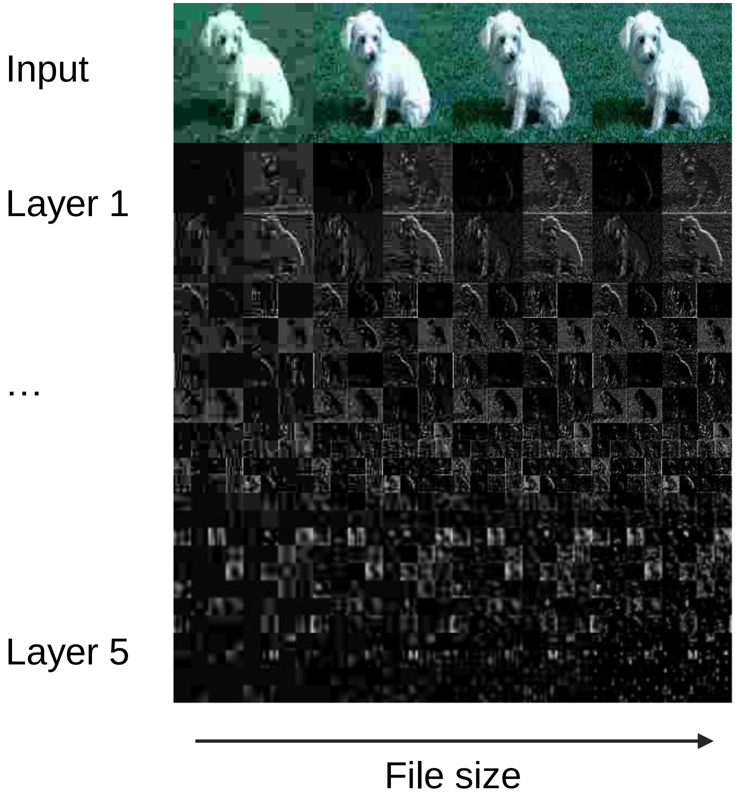
- What can we expect from practical (i.e., non-optimal) codecs?

TOY EXAMPLE OF FEATURE COMPRESSIBILITY

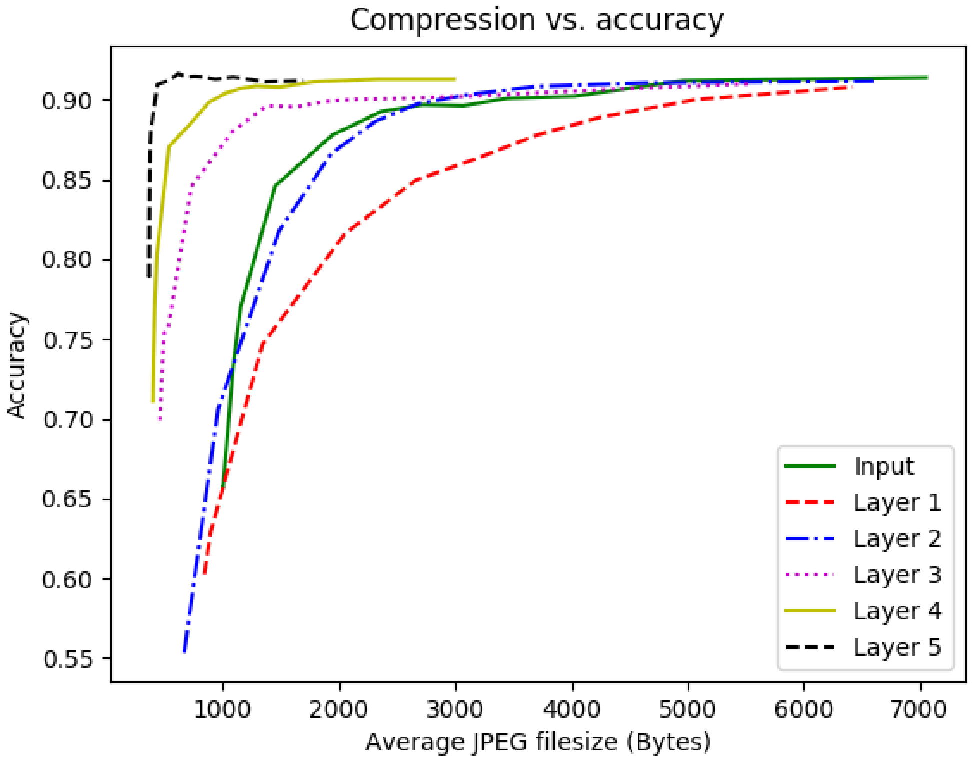


- A simple convolutional neural network (CNN) for cats vs. dogs classification
- Trained on Kaggle’s cats vs. dogs dataset
- Goal: compare input compression vs. feature compression in terms of resulting classification accuracy

TOY EXAMPLE OF FEATURE COMPRESSIBILITY



Features tiled into an image and compressed using JPEG



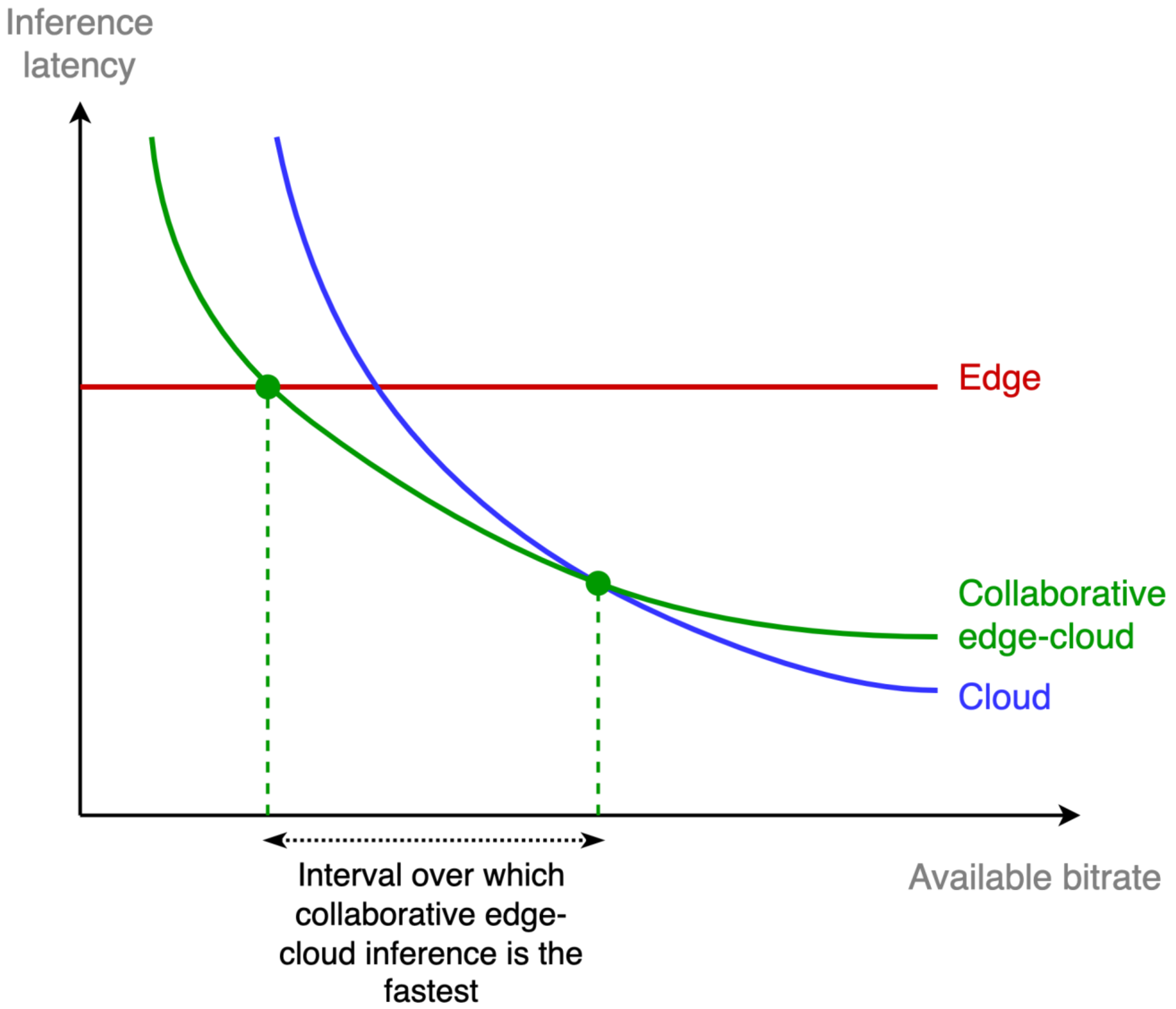
Feature compression better than input compression starting with layer 3 – why?

If we had an optimal encoder, this would already happen at layer 1

IMPLICATIONS FOR INFERENCE LATENCY

- Cloud has more powerful hardware than edge device
- Feature transmission takes fewer bits than input transmission

  ⟹ CI will have lower inference latency over some intermediate range of upload bitrates



Interval over which collaborative edge-cloud inference is the fastest

M. Ulhaq and I. V. Bajić, “Shared mobile-cloud inference for collaborative intelligence,” arXiv:2002.00157, demo at NeurIPS'19, Vancouver, BC, Dec. 2019.

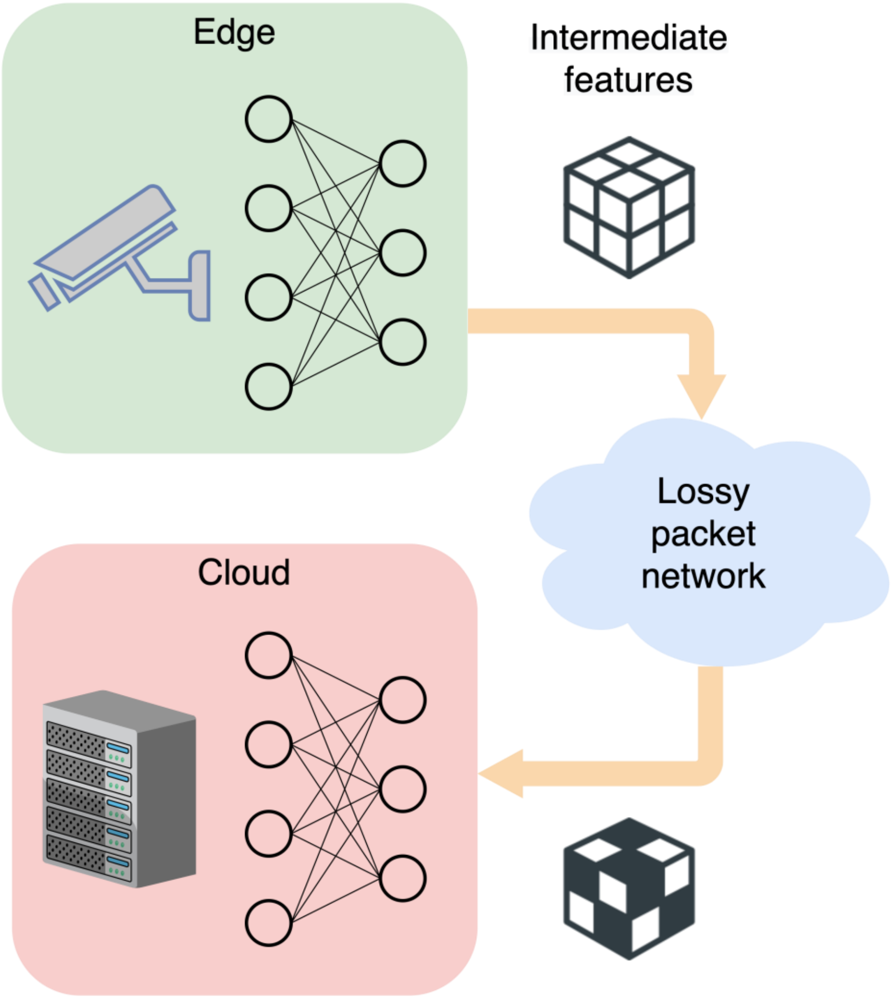
Questions?

Part 2

Practical considerations

ERROR RESILIENCE

- In CI, intermediate features are transmitted over imperfect channel
- Bit errors at physical layer → packet loss at application layer
    - What is the impact on inference accuracy?
    - How can we recover lost features?
- First idea:
    - Use existing tensor completion (imputation) approaches



TENSOR COMPLETION

- Well-known tensor completion methods for visual data:
  - Simple Low-Rank Tensor Completion (SiLRTC)
  - High-Accuracy Low-Rank Tensor Completion (HaLRTC)
  - Fused Canonical Polyadic (FCP) Decomposition
- Key assumption:
  - Tensor lies in a low-rank manifold (fewer degrees of freedom than tensor elements)
- Operationalization of the key assumption:
  - Use Singular Value Decomposition (SVD) of unfolded tensor to find this manifold (SiLRTC and HaLRTC)
  - Use CP Decomposition (CPD) of unfolded tensor to find this manifold (FCP)

J. Liu et al., “Tensor completion for estimating missing values in visual data,” IEEE TPAMI, vol. 35, no. 1, pp. 208–220, Jan. 2013.
Y. Wu et al., "A fused CP factorization method for incomplete tensors," IEEE TNNLS, vol. 30, no. 3, pp. 751–764, Mar. 2019.

TENSOR COMPLETION

- Advantage of existing methods:
  - Generic – don’t need to know how tensor was created in the first place
- Downsides:
  - Decompositions (SVD and CPD) are computationally expensive
  - Iterative – need to perform expensive decompositions in each iteration
- In the case of CI, we know how the tensor is generated
  - Can we use this knowledge to develop better tensor completion methods?

MODEL-BASED TENSOR COMPLETION

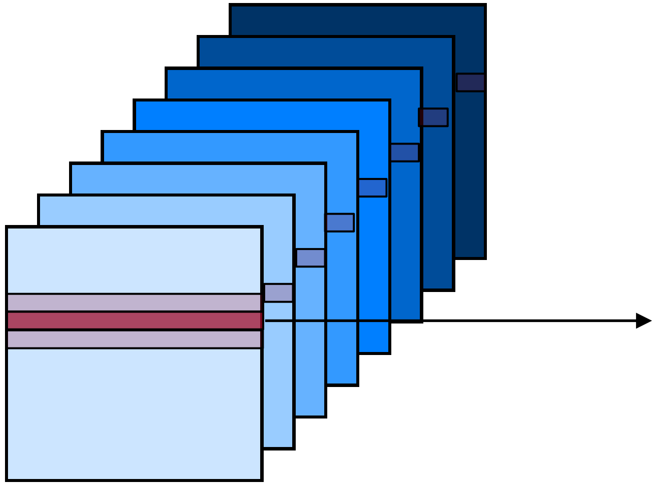
- Key idea:
  - Model the dependence among data in a feature tensor
- We know the process by which the feature tensor is generated (DNN front-end)
- We also know the kind of input data (e.g., images) from which feature tensor was generated
- This knowledge should give us some ways of modeling the feature tensor

- First attempt:
  - Adaptive Linear Tensor Completion (ALTeC)
  - Assume a certain linear relationship among feature tensor data

L. Bragilevsky and I. V. Bajić, "Tensor completion methods for collaborative intelligence," IEEE Access, vol. 8, pp. 41162-41174, Feb. 2020.

ALTEC – ADAPTIVE LINEAR TENSOR COMPLETION

- Packetization of tensor data



One row = one packet

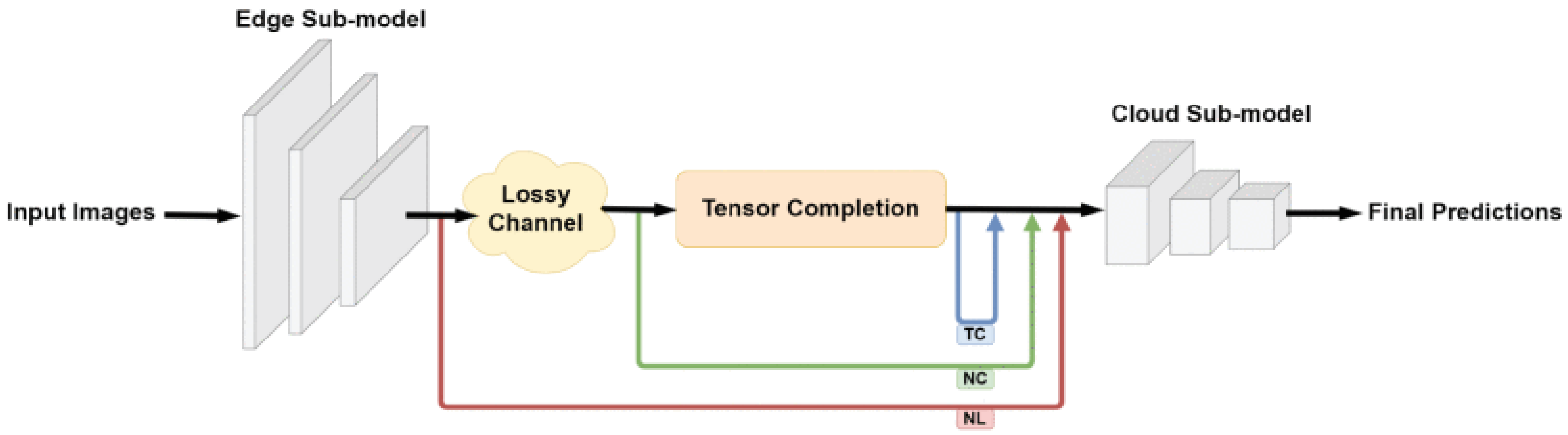
- Assume linear relationship among rows
  - Each row is approximately a linear combination of co-located rows in other channels and two spatial neighbors (top and bottom) in the same channel
- Let $\mathbf{x}_i^{(c)}$ be the i-th row in channel c

$$\mathbf{x}_i^{(c)} \approx \sum_{j \neq c} w_i^{(j)} \mathbf{x}_i^{(j)} + w_{i-1}^{(c)} \mathbf{x}_{i-1}^{(c)} + w_{i+1}^{(c)} \mathbf{x}_{i+1}^{(c)}$$

- Obtain the weights $w_i^{(j)}$ on a training set

L. Bragilevsky and I. V. Bajić, "Tensor Completion Methods for Collaborative Intelligence," IEEE Access, vol. 8, pp. 41162-41174, Feb. 2020.

ALTEC – ADAPTIVE LINEAR TENSOR COMPLETION



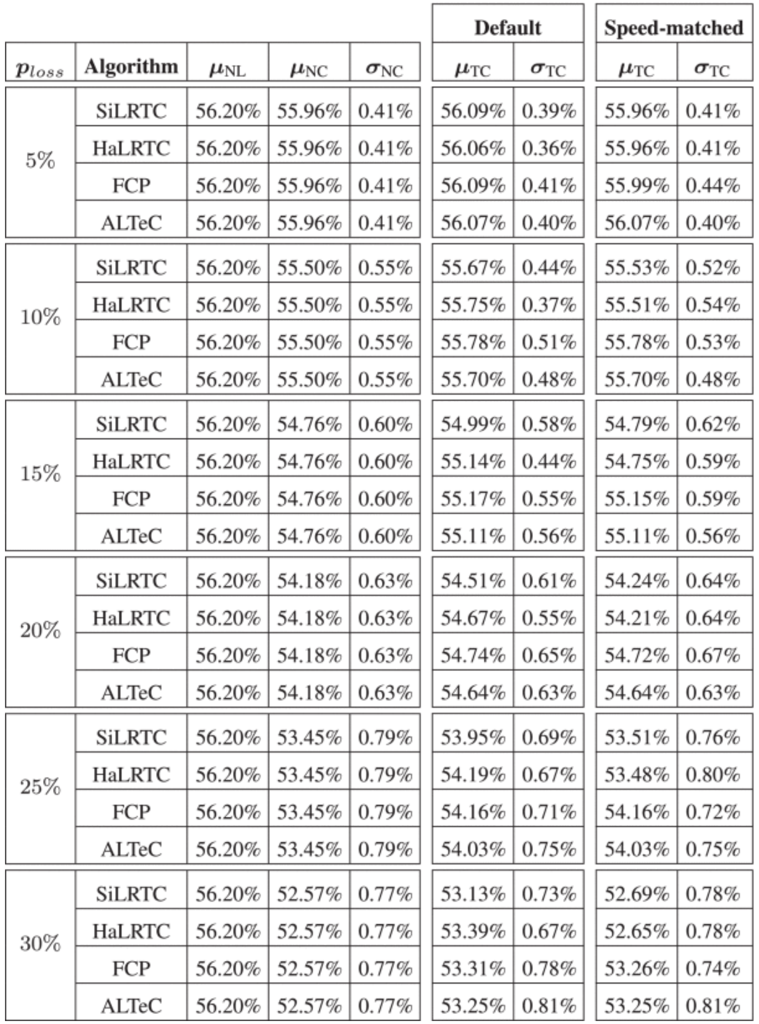
- Experimental setup
  - NL – no loss (tensor with all data used by the back-end)
  - NC – no completion (missing tensor values replaced by zeros)
  - TC – tensor completion (various methods used to estimate missing values)

L. Bragilevsky and I. V. Bajić, "Tensor Completion Methods for Collaborative Intelligence," IEEE Access, vol. 8, pp. 41162-41174, Feb. 2020.

ALTEC – ADAPTIVE LINEAR TENSOR COMPLETION

Results on VGG-16

- Two configurations tested:
  - Default: each method runs until convergence
  - Speed-matched: iterative methods run as many iterations they can up to ALTeC execution time
- Conclusions:
  - No significant difference between methods (t-test)
- Reason:
  - VGG-16 uses Rectified Linear Unit (ReLU) activation
  - Feature tensors have many zeros
  - Easy to recover, all methods do reasonably good job



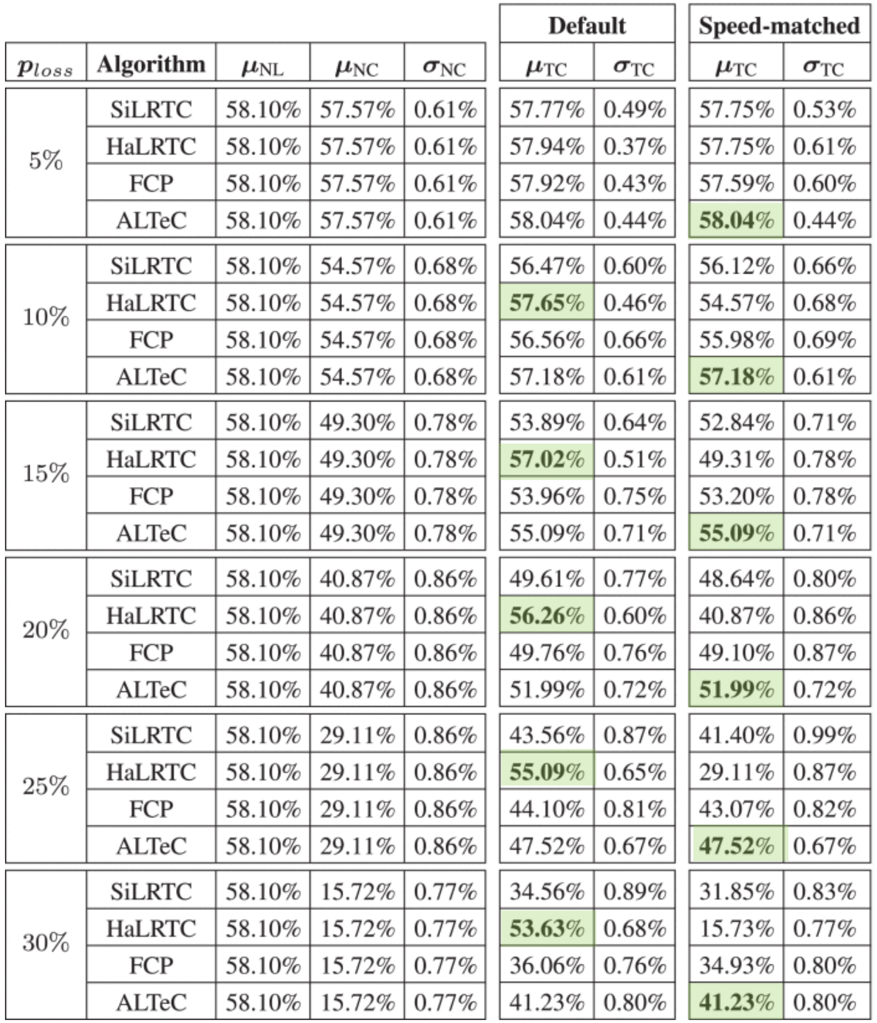
					Default		Speed-matched	
p_{loss}	Algorithm	μ_{NL}	μ_{NC}	σ_{NC}	μ_{TC}	σ_{TC}	μ_{TC}	σ_{TC}
5%	SiLRTC	56.20%	55.96%	0.41%	56.09%	0.39%	55.96%	0.41%
	HaLRTC	56.20%	55.96%	0.41%	56.06%	0.36%	55.96%	0.41%
	FCP	56.20%	55.96%	0.41%	56.09%	0.41%	55.99%	0.44%
	ALTeC	56.20%	55.96%	0.41%	56.07%	0.40%	56.07%	0.40%
10%	SiLRTC	56.20%	55.50%	0.55%	55.67%	0.44%	55.53%	0.52%
	HaLRTC	56.20%	55.50%	0.55%	55.75%	0.37%	55.51%	0.54%
	FCP	56.20%	55.50%	0.55%	55.78%	0.51%	55.78%	0.53%
	ALTeC	56.20%	55.50%	0.55%	55.70%	0.48%	55.70%	0.48%
15%	SiLRTC	56.20%	54.76%	0.60%	54.99%	0.58%	54.79%	0.62%
	HaLRTC	56.20%	54.76%	0.60%	55.14%	0.44%	54.75%	0.59%
	FCP	56.20%	54.76%	0.60%	55.17%	0.55%	55.15%	0.59%
	ALTeC	56.20%	54.76%	0.60%	55.11%	0.56%	55.11%	0.56%
20%	SiLRTC	56.20%	54.18%	0.63%	54.51%	0.61%	54.24%	0.64%
	HaLRTC	56.20%	54.18%	0.63%	54.67%	0.55%	54.21%	0.64%
	FCP	56.20%	54.18%	0.63%	54.74%	0.65%	54.72%	0.67%
	ALTeC	56.20%	54.18%	0.63%	54.64%	0.63%	54.64%	0.63%
25%	SiLRTC	56.20%	53.45%	0.79%	53.95%	0.69%	53.51%	0.76%
	HaLRTC	56.20%	53.45%	0.79%	54.19%	0.67%	53.48%	0.80%
	FCP	56.20%	53.45%	0.79%	54.16%	0.71%	54.16%	0.72%
	ALTeC	56.20%	53.45%	0.79%	54.03%	0.75%	54.03%	0.75%
30%	SiLRTC	56.20%	52.57%	0.77%	53.13%	0.73%	52.69%	0.78%
	HaLRTC	56.20%	52.57%	0.77%	53.39%	0.67%	52.65%	0.78%
	FCP	56.20%	52.57%	0.77%	53.31%	0.78%	53.26%	0.74%
	ALTeC	56.20%	52.57%	0.77%	53.25%	0.81%	53.25%	0.81%

L. Bragilevsky and I. V. Bajić, "Tensor Completion Methods for Collaborative Intelligence," IEEE Access, vol. 8, pp. 41162-41174, Feb. 2020.

ALTEC – ADAPTIVE LINEAR TENSOR COMPLETION

Results on ResNet-34

- Two configurations tested:
  - Default: each method runs until convergence
  - Speed-matched: iterative methods run as many iterations they can up to ALTeC execution time
- Conclusions:
  - HaLRTC usually best in default config, ALTeC best in speed-matched scenario (t-test)
- Reason:
  - ResNet-34 uses Leaky Rectified Linear Unit (ReLU) activations → feature tensors have fewer zeros
  - Differences between methods more obvious



					Default		Speed-matched	
p_{loss}	Algorithm	μ_{NL}	μ_{NC}	σ_{NC}	μ_{TC}	σ_{TC}	μ_{TC}	σ_{TC}
5%	SiLRTC	58.10%	57.57%	0.61%	57.77%	0.49%	57.75%	0.53%
	HaLRTC	58.10%	57.57%	0.61%	57.94%	0.37%	57.75%	0.61%
	FCP	58.10%	57.57%	0.61%	57.92%	0.43%	57.59%	0.60%
	ALTeC	58.10%	57.57%	0.61%	58.04%	0.44%	**58.04%**	0.44%
10%	SiLRTC	58.10%	54.57%	0.68%	56.47%	0.60%	56.12%	0.66%
	HaLRTC	58.10%	54.57%	0.68%	**57.65%**	0.46%	54.57%	0.68%
	FCP	58.10%	54.57%	0.68%	56.56%	0.66%	55.98%	0.69%
	ALTeC	58.10%	54.57%	0.68%	57.18%	0.61%	**57.18%**	0.61%
15%	SiLRTC	58.10%	49.30%	0.78%	53.89%	0.64%	52.84%	0.71%
	HaLRTC	58.10%	49.30%	0.78%	**57.02%**	0.51%	49.31%	0.78%
	FCP	58.10%	49.30%	0.78%	53.96%	0.75%	53.20%	0.78%
	ALTeC	58.10%	49.30%	0.78%	55.09%	0.71%	**55.09%**	0.71%
20%	SiLRTC	58.10%	40.87%	0.86%	49.61%	0.77%	48.64%	0.80%
	HaLRTC	58.10%	40.87%	0.86%	**56.26%**	0.60%	40.87%	0.86%
	FCP	58.10%	40.87%	0.86%	49.76%	0.76%	49.10%	0.87%
	ALTeC	58.10%	40.87%	0.86%	51.99%	0.72%	**51.99%**	0.72%
25%	SiLRTC	58.10%	29.11%	0.86%	43.56%	0.87%	41.40%	0.99%
	HaLRTC	58.10%	29.11%	0.86%	**55.09%**	0.65%	29.11%	0.87%
	FCP	58.10%	29.11%	0.86%	44.10%	0.81%	43.07%	0.82%
	ALTeC	58.10%	29.11%	0.86%	47.52%	0.67%	**47.52%**	0.67%
30%	SiLRTC	58.10%	15.72%	0.77%	34.56%	0.89%	31.85%	0.83%
	HaLRTC	58.10%	15.72%	0.77%	**53.63%**	0.68%	15.73%	0.77%
	FCP	58.10%	15.72%	0.77%	36.06%	0.76%	34.93%	0.80%
	ALTeC	58.10%	15.72%	0.77%	41.23%	0.80%	**41.23%**	0.80%

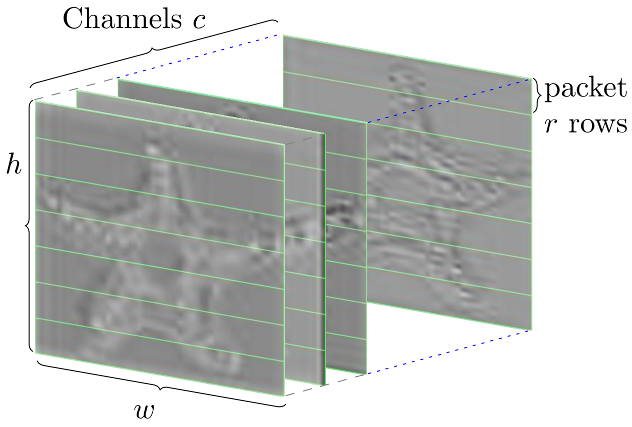
L. Bragilevsky and I. V. Bajić, "Tensor Completion Methods for Collaborative Intelligence," IEEE Access, vol. 8, pp. 41162-41174, Feb. 2020.

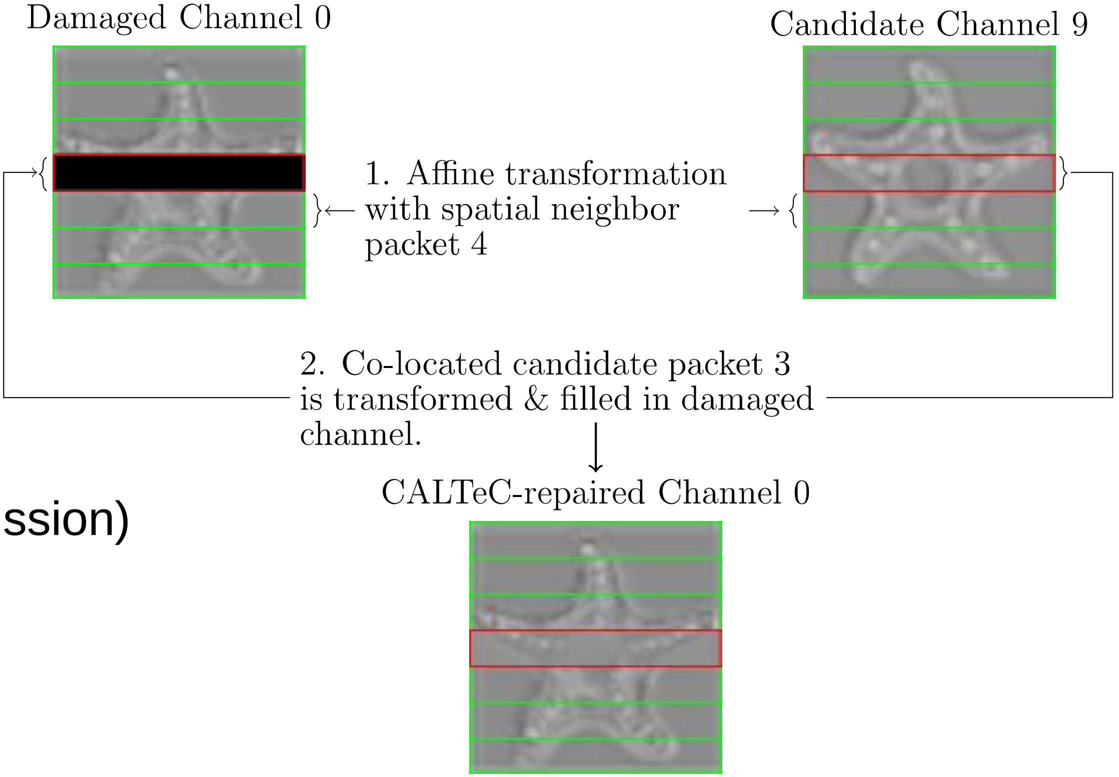
CALTEC – CONTENT-ADAPTIVE LINEAR TENSOR COMPLETION

- ALTeC was very fast and fairly accurate – best in speed-matched tests, second-best in unrestricted tests
- However, it was content-agnostic
  - “Adaptive” in ALTeC refers to spatial adaptation – different rows in a feature tensor have different coefficients $w_i^{(j)}$
  - But dependence of features on the input (content) is not being exploited
- Improvement: Content-Adaptive Linear Tensor Completion (CALTeC)
  - Recovery of missing data depends on the content – no pre-training
  - But might be slower than ALTeC

A. Dhondea, R. A. Cohen, and I. V. Bajić, “CALTeC: Content-adaptive linear tensor completion for collaborative intelligence,” IEEE ICIP, Sep. 2021.

CALTEC – CONTENT-ADAPTIVE LINEAR TENSOR COMPLETION





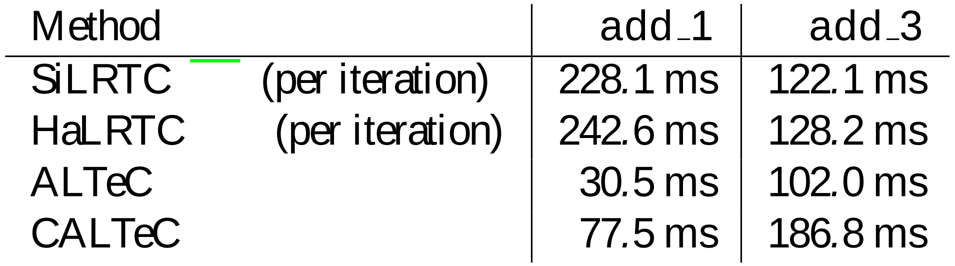
1. Affine transformation with spatial neighbor packet 4

2. Co-located candidate packet 3 is transformed & filled in damaged channel.

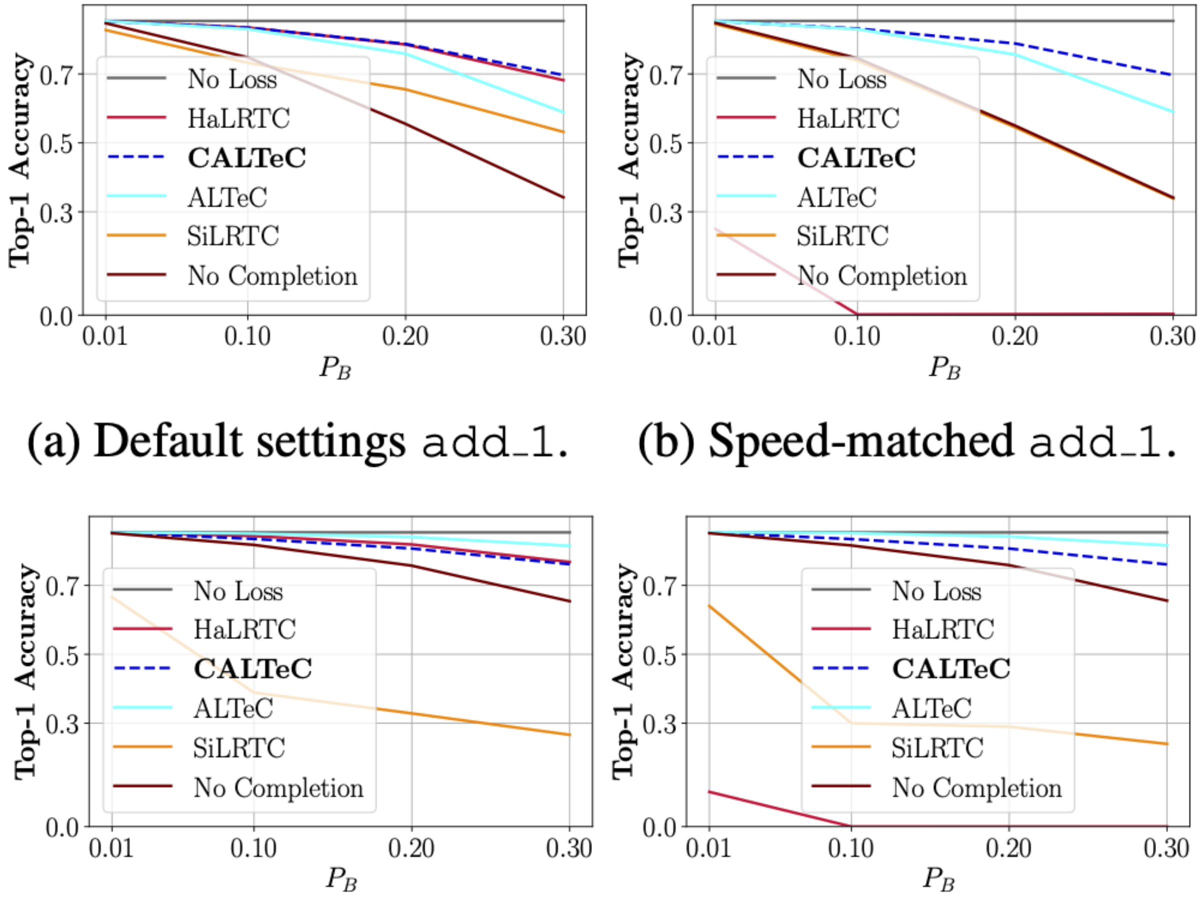
- 8 rows per packet (similar to JPEG image transmission)
- Find the channel with
  - Available co-located packet
  - Most similar available neighboring packets
- Estimate affine transform by matching neighbors
- Apply the affine transformation to co-located packet and use this as estimate of missing one

A. Dhondea, R. A. Cohen, and I. V. Bajić, “CALTeC: Content-adaptive linear tensor completion for collaborative intelligence,” IEEE ICIP, Sep. 2021.

CALTEC – CONTENT-ADAPTIVE LINEAR TENSOR COMPLETION



Method		add_1	add_3
SiLRTC	(per iteration)	228.1 ms	122.1 ms
HaLRTC	(per iteration)	242.6 ms	128.2 ms
ALTeC		30.5 ms	102.0 ms
CALTeC		77.5 ms	186.8 ms



(a) Default settings add_1. (b) Speed-matched add_1.

(c) Default settings add_3. (d) Speed-matched add_3.

- Experiments on ResNet-18
- CALTeC slower than ALTeC, but still much faster than SiLRTC and HaLRTC
- Best on `add_1` tensors, second-best (after ALTeC) on `add_3` tensors

A. Dhondea, R. A. Cohen, and I. V. Bajić, "CALTeC: Content-adaptive linear tensor completion for collaborative intelligence," IEEE ICIP, Sep. 2021.

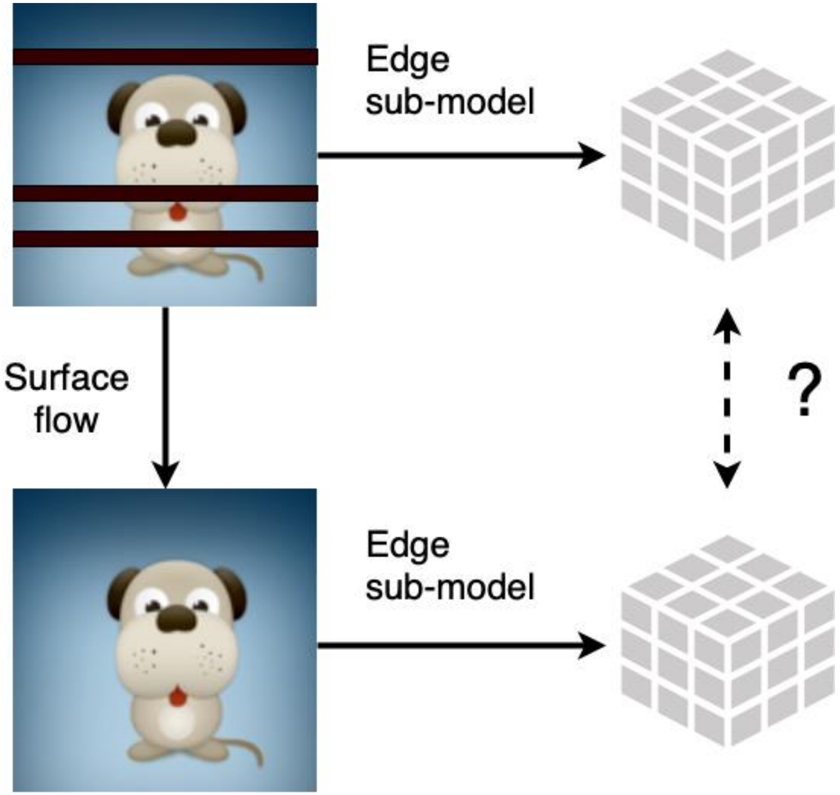
TENSOR COMPLETION BY PDE-BASED INPAINTING

- We know PDE-based inpainting works well for images
- A popular PDE model for inpainting:

$$\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} = 0$$

  - ○ I – image intensity; t – iteration
  - ○ (v_x, v_y) – surface flow

- If this model works well in the input space, what it its equivalent in the latent space?
- How does the above PDE change as I is transformed through the network's front-end (edge sub-model)?



M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-Stokes, fluid dynamics, and image and video inpainting," Proc. IEEE CVPR, 2001.

TENSOR COMPLETION BY PDE-BASED INPAINTING

Common operations in convolutional networks:

1. Convolution
2. Nonlinear activation
3. Batch normalization
4. Pooling
   - Max pooling
   - Mean pooling
   - Learnt pooling (strided convolution)

- Examine the effect of each of these on the surface flow PDE

I. V. Bajić, “Latent space inpainting for loss-resilient collaborative object detection,” Proc. IEEE ICC, Montreal, Canada, Jun. 2021.

TENSOR COMPLETION BY PDE-BASED INPAINTING

- When input image I is convolved with kernel f, the resulting flow equation is

$$\frac{\partial}{\partial x}(f * I)u_x + \frac{\partial}{\partial y}(f * I)u_y + \frac{\partial}{\partial t}(f * I) = 0$$

where (u_x, u_y) is the new flow field

- Convolution and differentiation commute:

$$f * \underbrace{\left(\frac{\partial I}{\partial x}u_x + \frac{\partial I}{\partial y}u_y + \frac{\partial I}{\partial t}\right)}_{\text{same flow equation as in input space}} = 0$$

I. V. Bajić, “Latent space inpainting for loss-resilient collaborative object detection,” Proc. IEEE ICC, Montreal, Canada, Jun. 2021.

TENSOR COMPLETION BY PDE-BASED INPAINTING

- When input image I passes through nonlinear activation $\sigma(\cdot)$, the resulting flow equation is

$$\frac{\partial\, \sigma(I)}{\partial x} u_x + \frac{\partial\, \sigma(I)}{\partial y} u_y + \frac{\partial\, \sigma(I)}{\partial t} = 0$$

where (u_x, u_y) is the new flow field

- Using the chain rule of differentiation:

$$\sigma'(I) \cdot \underbrace{\left(\frac{\partial I}{\partial x} u_x + \frac{\partial I}{\partial y} u_y + \frac{\partial I}{\partial t} \right)}_{\text{same flow equation as in input space}} = 0$$

I. V. Bajić, “Latent space inpainting for loss-resilient collaborative object detection,” Proc. IEEE ICC, Montreal, Canada, Jun. 2021.

TENSOR COMPLETION BY PDE-BASED INPAINTING

- It can be shown that the flow equation is (approximately) preserved through other processing layers commonly found in convolutional neural networks
  - Details in [1]

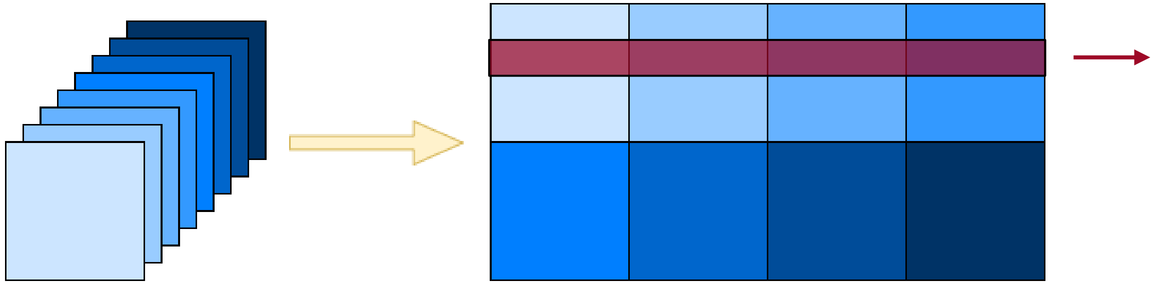
- Hence, an input-space surface flow solver should be able to do a good job in the latent space too

- Some popular solvers:
  - “Navier-Stokes” [2]
  - “Telea” [3]

[1] I. V. Bajić, “Latent space inpainting for loss-resilient collaborative object detection,” Proc. IEEE ICC, Montreal, Canada, Jun. 2021.
[2] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, “Navier-Stokes, fluid dynamics, and image and video inpainting,” CVPR 2001.
[3] A. Telea, “An image inpainting technique based on the fast marching method,” J. Graphics Tools, 2004.

TENSOR COMPLETION BY PDE-BASED INPAINTING

- Feature tensor packetization:



8 rows = packet

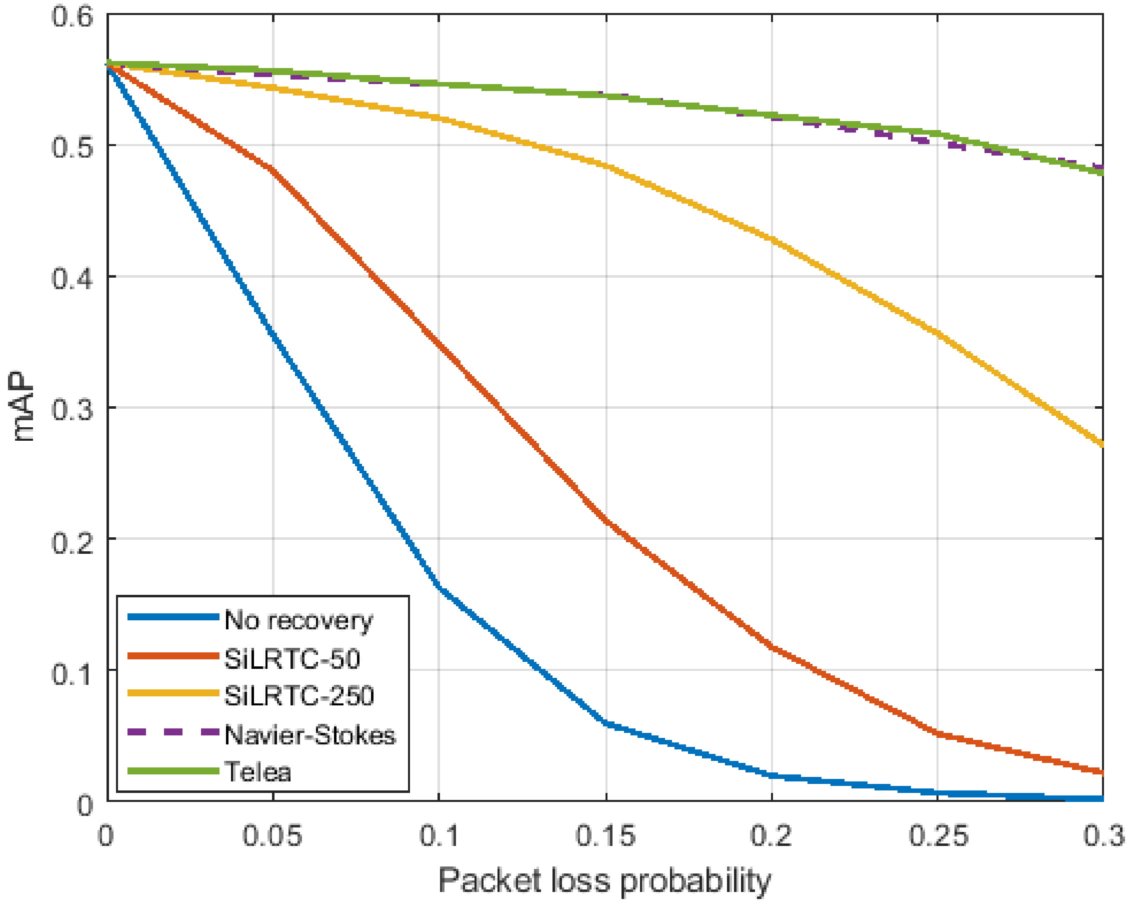
- DNN model: YOLOv3 (object detector) split at layer 12

- Channel model: i.i.d. packet loss

- Dataset: COCO 2017 validation

J. Redmon and A. Farhadi, “YOLOv3: an incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.

TENSOR COMPLETION BY PDE-BASED INPAINTING

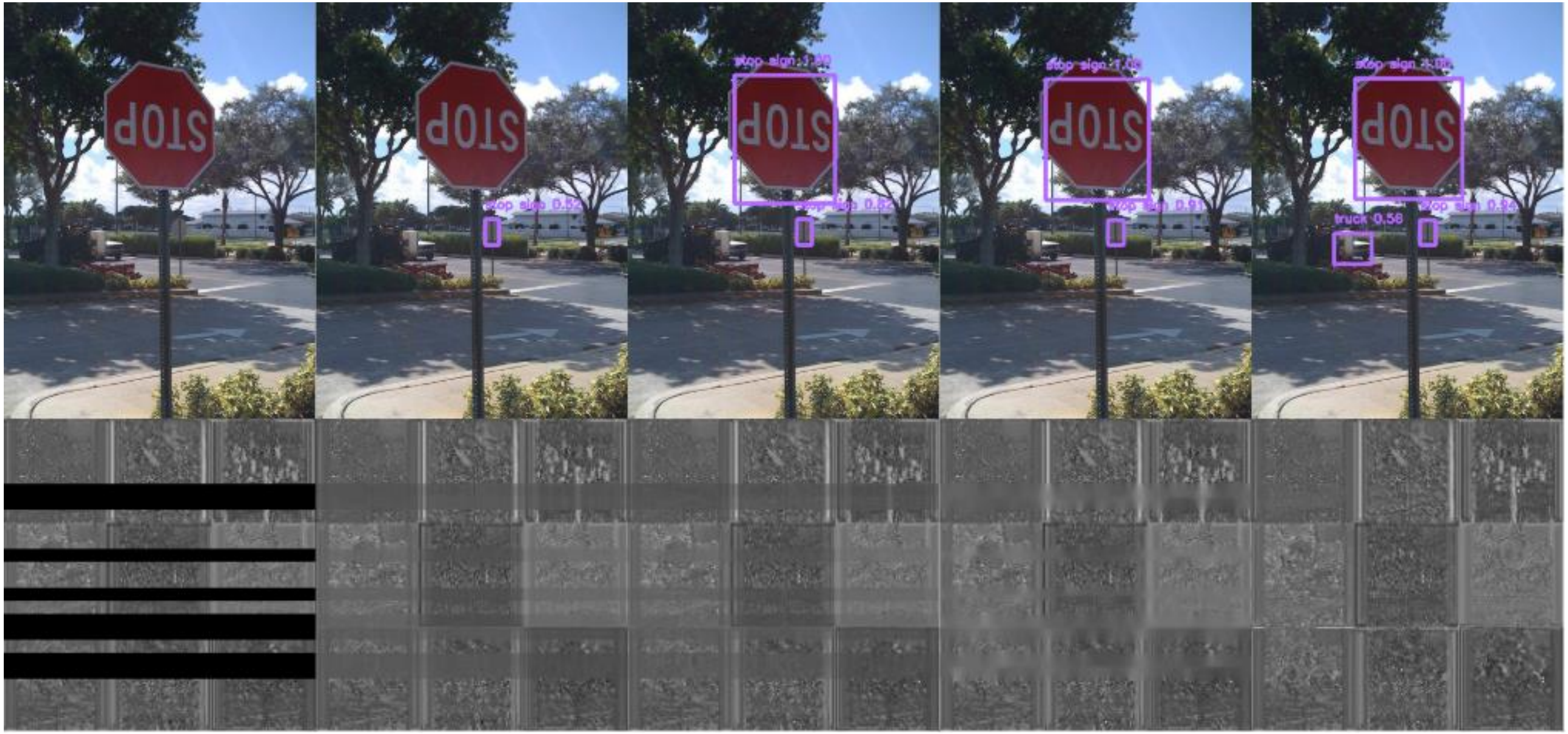


Method	Avg. mAP gain	Time per tensor (sec.)
SiLRTC-50	0.1028	17.0793
SiLRTC-250	0.3101	83.2044
Navier-Stokes	0.3823	0.1408
Telea	0.3837	0.1356

Avg. mAP gain (X) = [AUC (X) – AUC (No rec.)] / 0.3

I. V. Bajić, “Latent space inpainting for loss-resilient collaborative object detection,” Proc. IEEE ICC, Montreal, Canada, Jun. 2021.

TENSOR COMPLETION BY PDE-BASED INPAINTING



No recovery SiLRTC-50 SiLRTC-250 Telea No loss

I. V. Bajić, "Latent space inpainting for loss-resilient collaborative object detection," Proc. IEEE ICC, Montreal, Canada, Jun. 2021.

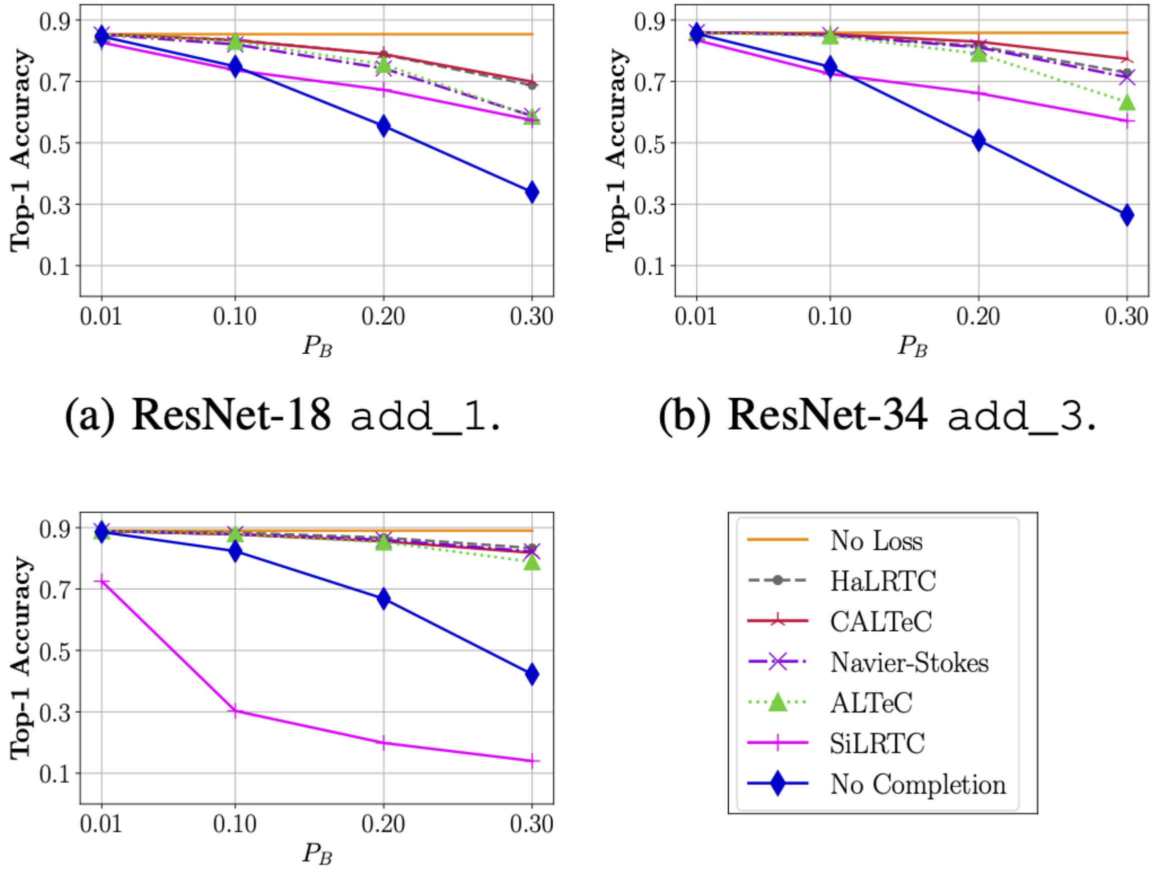
SUMMARY OF TENSOR COMPLETION FOR COLLABORATIVE INTELLIGENCE

Comparison of all methods on image classification

- CALTeC and PDE-based inpainting good across the board
- ALTeC also good, but requires pre-training
- HaLRTC good performance when allowed to run enough iterations, but extremely slow
- SiLRTC weakest and slow

If you want to experiment, Deep Feature Transmission Simulator (DFTS2) offers an easy-to-use environment:

`https://github.com/AshivDhondea/DFTS2`



(a) ResNet-18 `add_1`.

(b) ResNet-34 `add_3`.

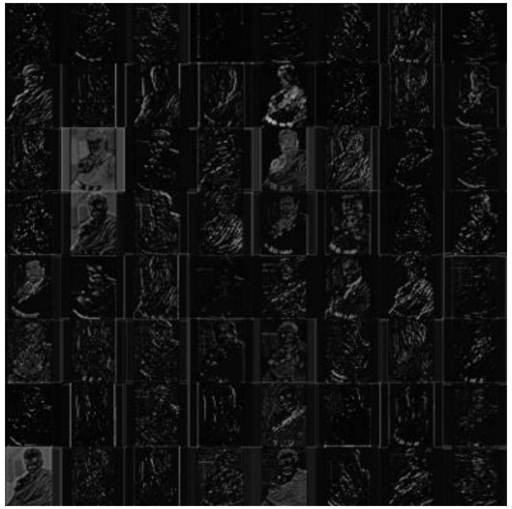
(c) DenseNet-121 `pool2_conv`.

A. Dhondea et al., “DFTS2: Deep feature transmission simulation for collaborative intelligence,” Proc. IEEE VCIP, Dec. 2021.

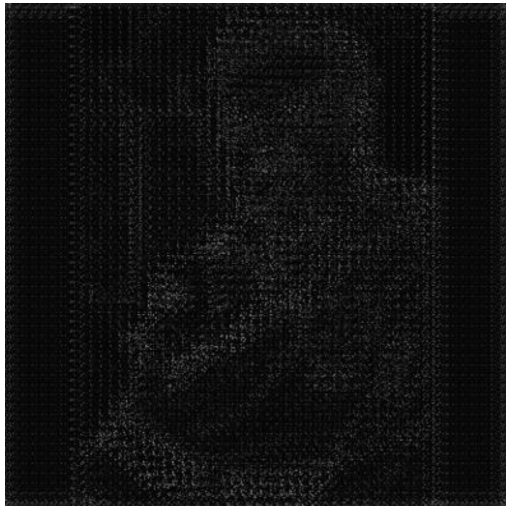
Questions?

FEATURE COMPRESSION

- How can we practically compress features obtained from a neural network?
- One idea:
  1. Reorganize the feature tensor into an image
     - Two possibilities – tiling and quilting (tiling works better)
  2. Quantize to 8 bits/tensor element
  3. Use an existing image codec (PNG, JPEG, JPEG2000, HEVC/BPG, VVC, …) to encode as a grayscale image



Tiling



Quilting

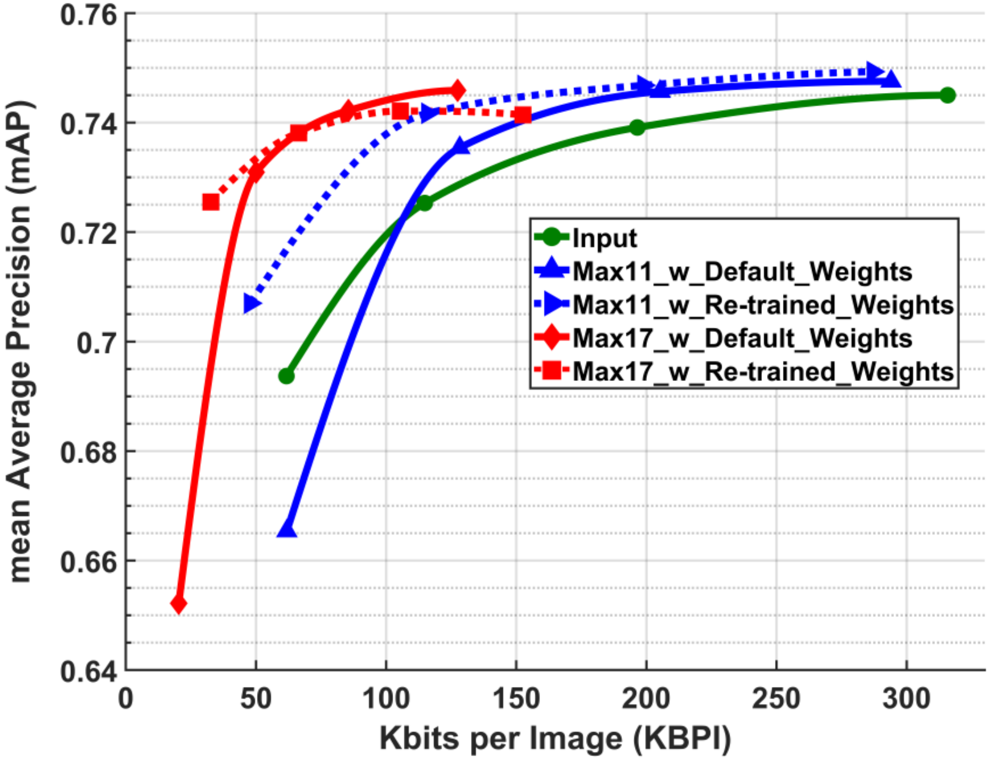
H. Choi and I. V. Bajić, “Deep feature compression for collaborative object detection,” Proc. IEEE ICIP, Oct. 2018.

FEATURE COMPRESSION

Results on YOLOv2 [1] object detector

- Features compressed by BPG (HEVC-Intra)
- Part of VOC2007 dataset for testing
- Images from VOC2007 and VOC2012 for re-training to account for quantization
- Savings of up to 60% bits at equivalent accuracy without re-training
- Savings of 70% bits with re-training

Split at	Default weights	Re-trained weights
max_11	-6.09%	$-\mathbf{45.23\%}$
max_17	-60.30%	$-\mathbf{70.30\%}$

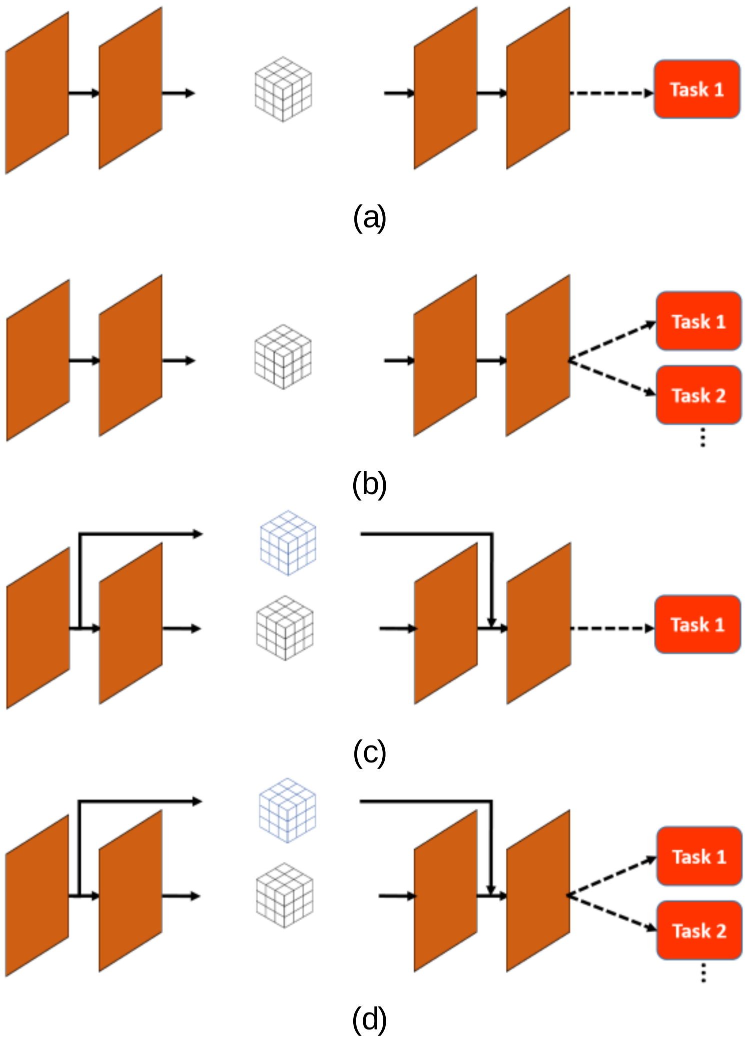


[1] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” Proc. IEEE CVPR, Jul. 2017.
[2] H. Choi and I. V. Bajić, “Deep feature compression for collaborative object detection,” Proc. IEEE ICIP, Oct. 2018.

MULTI-STREAM COLLABORATIVE INTELLIGENCE SYSTEMS

- Types of collaborative intelligence (CI) systems:
  a) Single-stream single task (1×1)
  b) Single-stream multi-task ($1 \times k$)
  c) Multi-stream single-task ($N \times 1$)
  d) Multi-stream multi-task ($N \times k$)

- In multi-stream CI systems, rates of individual streams need to be optimized
- In [Alvar and Bajić, TIP 2021]:
  - Tractable R-D model for CI systems proposed
  - Analytical bit allocation solution for $N \times 1$ systems
  - Pareto set characterization for $2 \times k$ systems
  - Bounds on Pareto set for 3×2 systems



S. R. Alvar and I. V. Bajić, “Pareto-optimal bit allocation for collaborative intelligence,” IEEE Trans. Image Processing, vol. 30, Feb. 2021.

TRACTABLE RD MODEL FOR MULTI-STREAM MULTI-TASK CI SYSTEMS

- Task distortion:

accuracy w/o compression

accuracy after compression

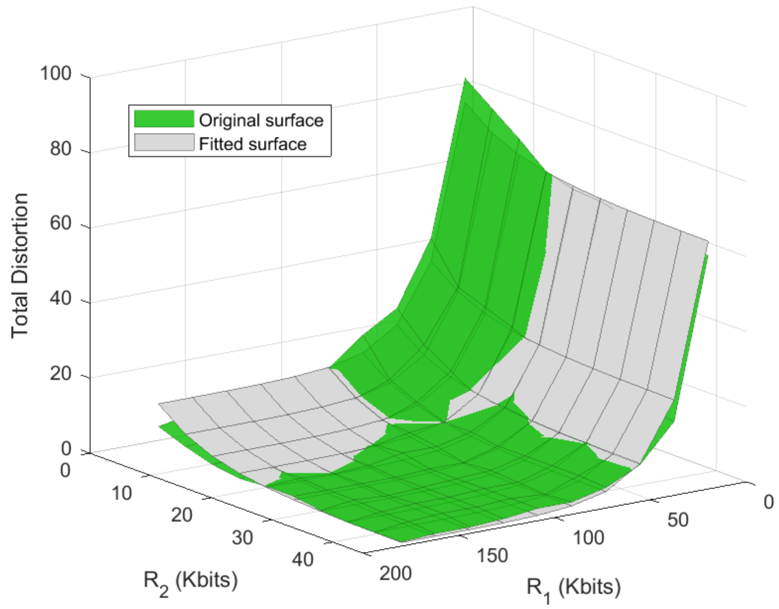
$$D_i = \frac{|\overline{A_i} - A_i|}{\overline{A_i}} \cdot 100$$

% change in task accuracy due to compression

- Rate-Distortion (RD) model

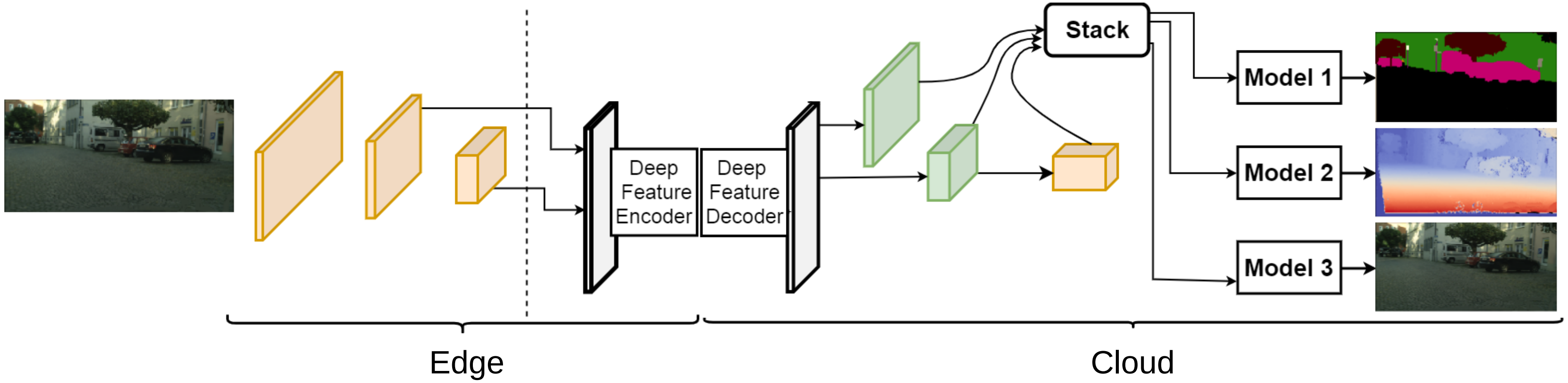
$$D_i(R_1, \ldots, R_N) \approx \gamma_i + \sum_{j=1}^{N} \alpha_{i,j} 2^{-\beta_{i,j} R_j}$$

- Benefits of this RD model:
  - “Makes sense” – distortion reduces exponentially with rates
  - Fits the data: $R^2 > 0.94$ in all our tests
  - Tractable – distortion is convex and monotonically decreasing with rate

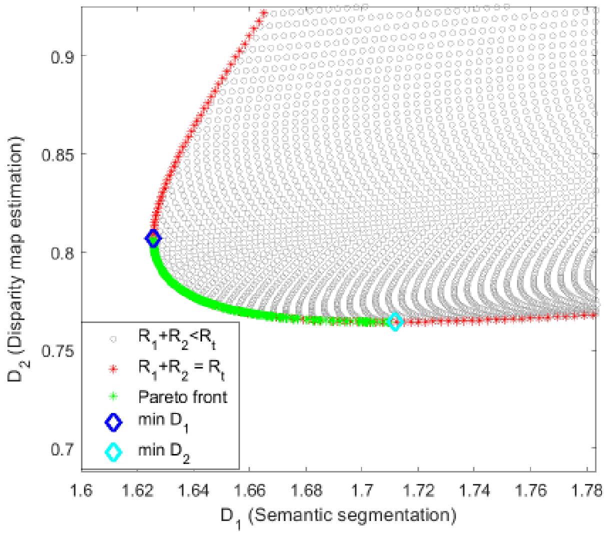


S. R. Alvar and I. V. Bajić, “Pareto-optimal bit allocation for collaborative intelligence,” IEEE Trans. Image Processing, vol. 30, Feb. 2021.

EXAMPLE OF BIT ALLOCATION IN MULTI-STREAM MULTI-TASK MODELS

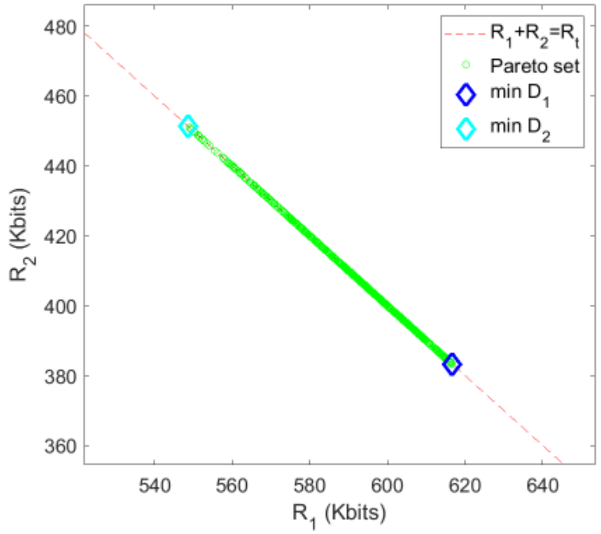


Pareto front



Pareto set

(rates that achieve the Pareto front)



S. R. Alvar and I. V. Bajić, “Pareto-optimal bit allocation for collaborative intelligence,” IEEE Trans. Image Processing, vol. 30,Feb. 2021.

Questions?

LATENT SPACE SCALABILITY

- In multi-task systems we looked at so far, all features supported all tasks; but a better design is possible

$$\begin{tikzcd}
 & & \hat{X} \arrow[r, dashed] & T \\
X \arrow[r] & \mathcal{Y} \arrow[ur, bend left] \arrow[dr, bend right] & & \\
 & & T &
\end{tikzcd}$$

Reconstructed image

CV task

Input image

Latent representation

CV task

- The tasks often include input image reconstruction ($\hat{X}$) and/or some computer vision (CV) inference tasks T
- But CV inference can also be obtained from $\hat{X}$ (common in practice)
- Data processing inequality (DPI) applied to $\mathcal{Y} \rightarrow \hat{X} \rightarrow T$:

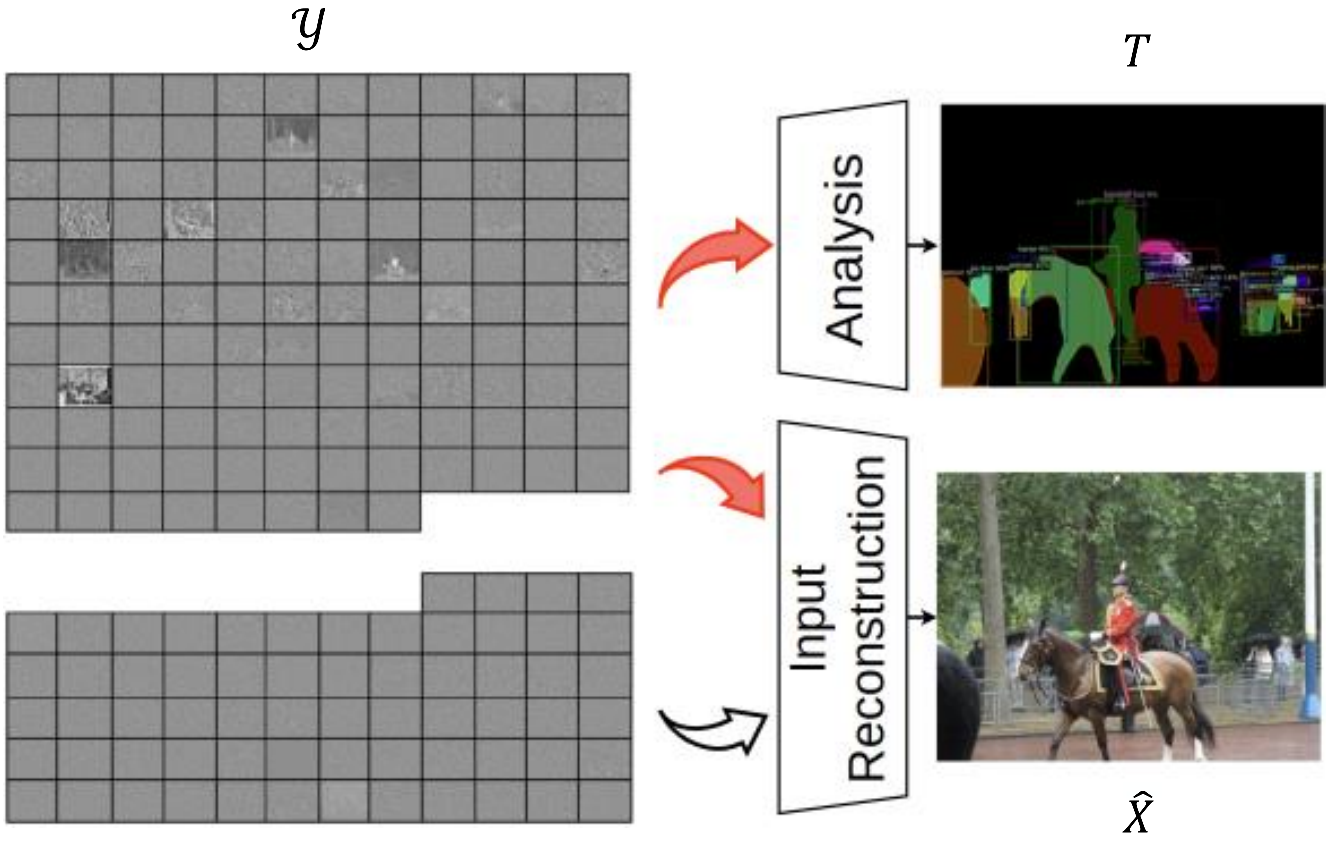
$$I(\mathcal{Y};\hat{X}) \geq I(\mathcal{Y};T)$$

H. Choi and I. V. Bajić, “Scalable image coding for humans and machines,” IEEE TIP, 2022.

LATENT SPACE SCALABILITY

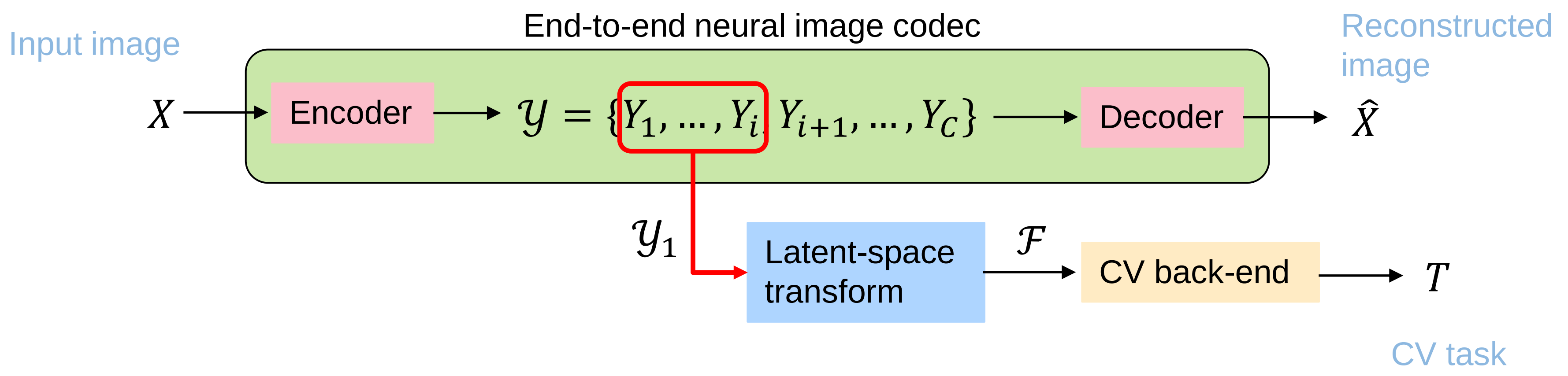
$$I(\mathcal{Y};\hat{X}) \geq I(\mathcal{Y};T)$$

- Latent space $\mathcal{Y}$ contains less information about CV task T than about input reconstruction $\hat{X}$
- Dedicate a subset of $\mathcal{Y}$ to T, all of it to $\hat{X}$
- When only T is needed, decode only a subset of $\mathcal{Y}$



H. Choi and I. V. Bajić, “Scalable image coding for humans and machines,” IEEE TIP, 2022.

LATENT SPACE SCALABILITY

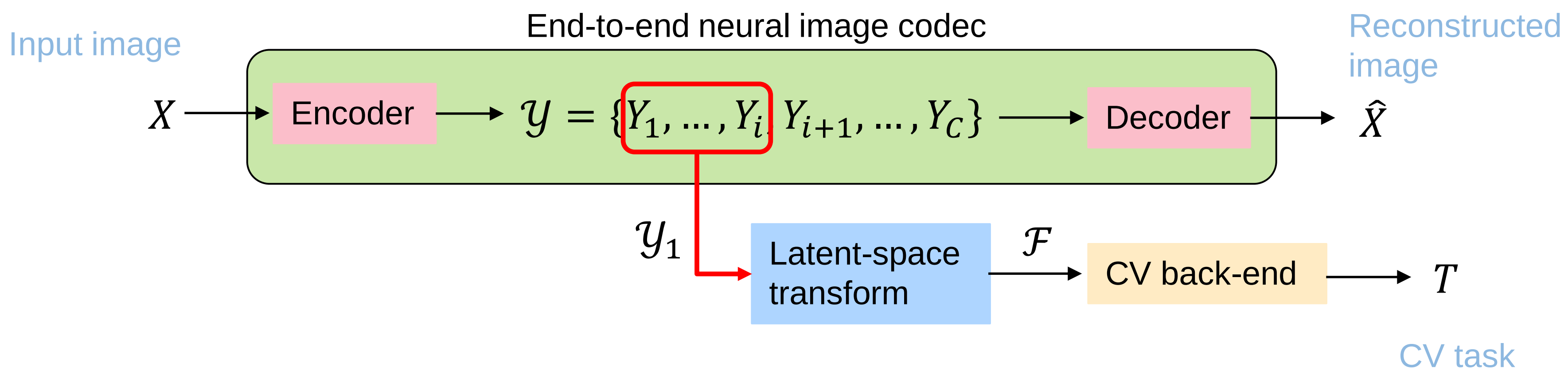


Example 2-layer scalable system:

- End-to-end image codec backbone [2]
- Subset of latent space ($\mathcal{Y}_1$) needs to be transformed into the latent space $\mathcal{F}$ of the CV back-end
  - Need latent-space transform (another neural network)
- CV back-end (for object detection) is YOLOv3 [3] starting at layer 13

[1] H. Choi and I. V. Bajić, "Scalable image coding for humans and machines," IEEE TIP, 2022.
[2] Z. Cheng et al., "Learned image compression with discretized gaussian mixture likelihoods and attention modules," Proc. IEEE CVPR, 2020.
[3] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv:1804.02767, Apr. 2018.

LATENT SPACE SCALABILITY



- Loss function:

$$\mathcal{L} = R + \lambda \cdot \underbrace{\left[\mathrm{MSE}\left(X,\hat{X}\right) + \gamma \cdot \mathrm{MSE}\left(\mathcal{F},\hat{\mathcal{F}}\right)\right]}_{D}$$

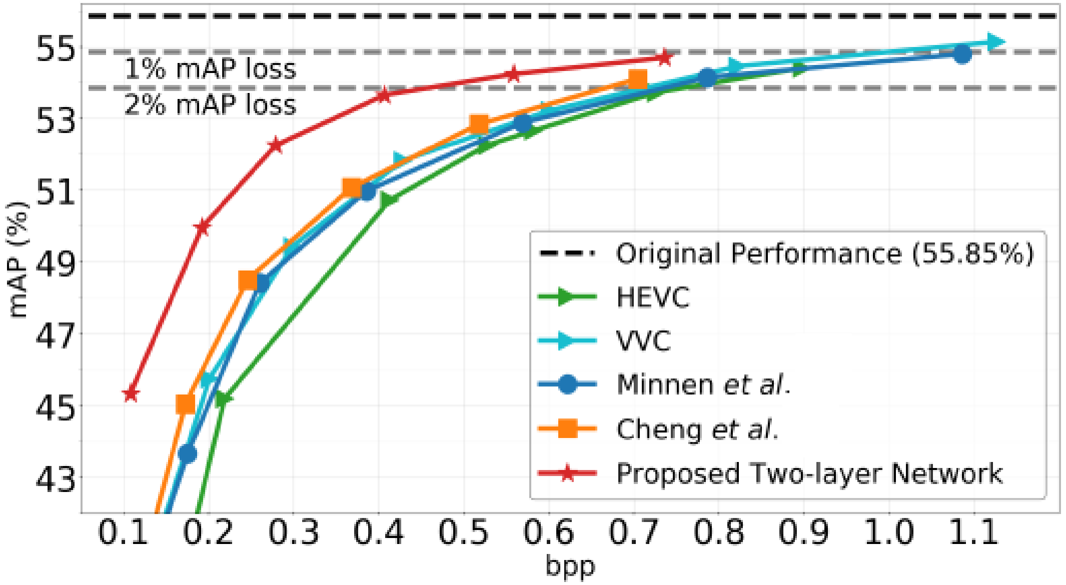
- R is the rate estimate [2]
- Distortion D composed of input reconstruction $\mathrm{MSE}\left(X,\hat{X}\right)$ and CV feature reconstruction $\mathrm{MSE}\left(\mathcal{F},\hat{\mathcal{F}}\right)$
- Since $\mathrm{MSE}\left(\mathcal{F},\hat{\mathcal{F}}\right)$ depends only on $\mathcal{Y}_1$ (and not on $\mathcal{Y}\backslash\mathcal{Y}_1$), CV-relevant information is steered to $\mathcal{Y}_1$

[1]. H. Choi and I. V. Bajić, “Scalable image coding for humans and machines,” IEEE TIP, 2022.
[2]. D. Minnen, J. Balle, and G. D. Toderici, “Joint autoregressive and hierarchical priors for learned image compression,” NeurIPS, 2018.

LATENT SPACE SCALABILITY

- Object detection experiments on the COCO dataset
- Performance much better than compressing input directly:
  - 37 – 48% bit savings compared to state-of-the-art image codecs
  - 2.8 – 4.5% more accurate detection at the same bit rate
  - Reason: not all pixel details are needed for object detection

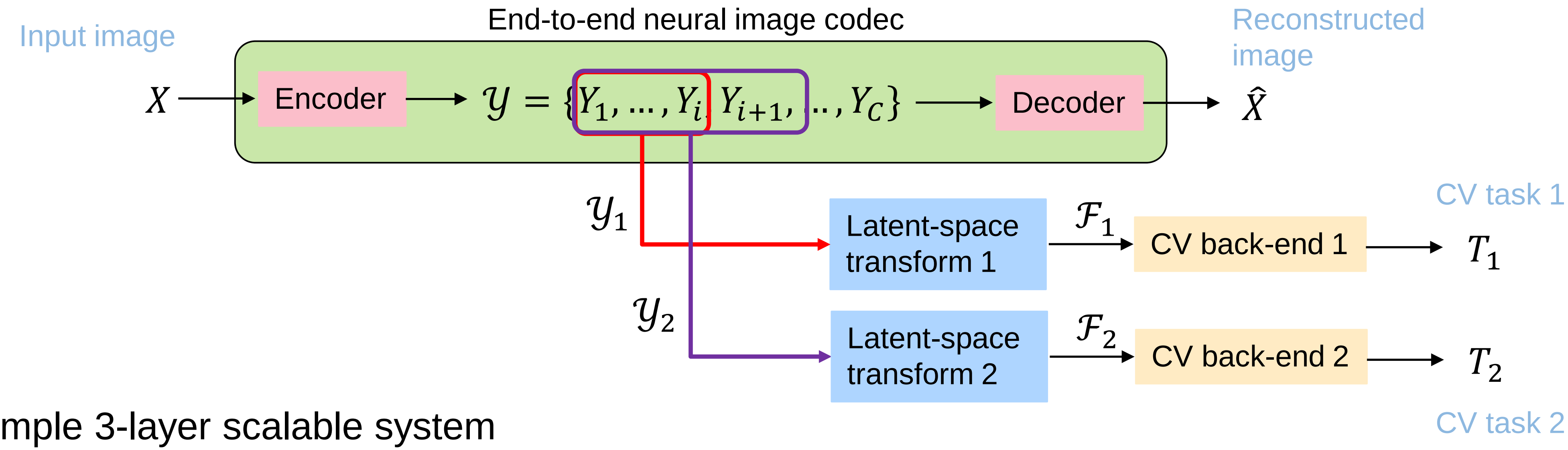
2-layer system: object detection + input reconstruction



	Two-layer Network	
Benchmarks	BD-Bitrate	BD-mAP
VVC	–39.8	2.79
HEVC	–47.9	4.55
Minnen *et al.*	–41.3	3.26
Cheng *et al.*	–37.4	2.89

[1] H. Choi and I. V. Bajić, “Scalable image coding for humans and machines,” IEEE TIP, 2022.
[2] Z. Cheng et al., “Learned image compression with discretized gaussian mixture likelihoods and attention modules,” Proc. IEEE CVPR, 2020.
[3] D. Minnen, J. Balle, and G. D. Toderici, “Joint autoregressive and hierarchical priors for learned image compression,” NeurIPS, 2018.

LATENT SPACE SCALABILITY



Example 3-layer scalable system

- End-to-end image codec backbone [2]
- CV task 1: object detection using Detectron [3] back-end
- CV task 2: semantic segmentation using Detectron [3] back-end
  - Object detection ⊂ semantic segmentation $\Rightarrow$ $\mathcal{Y}_1 \subset \mathcal{Y}_2$

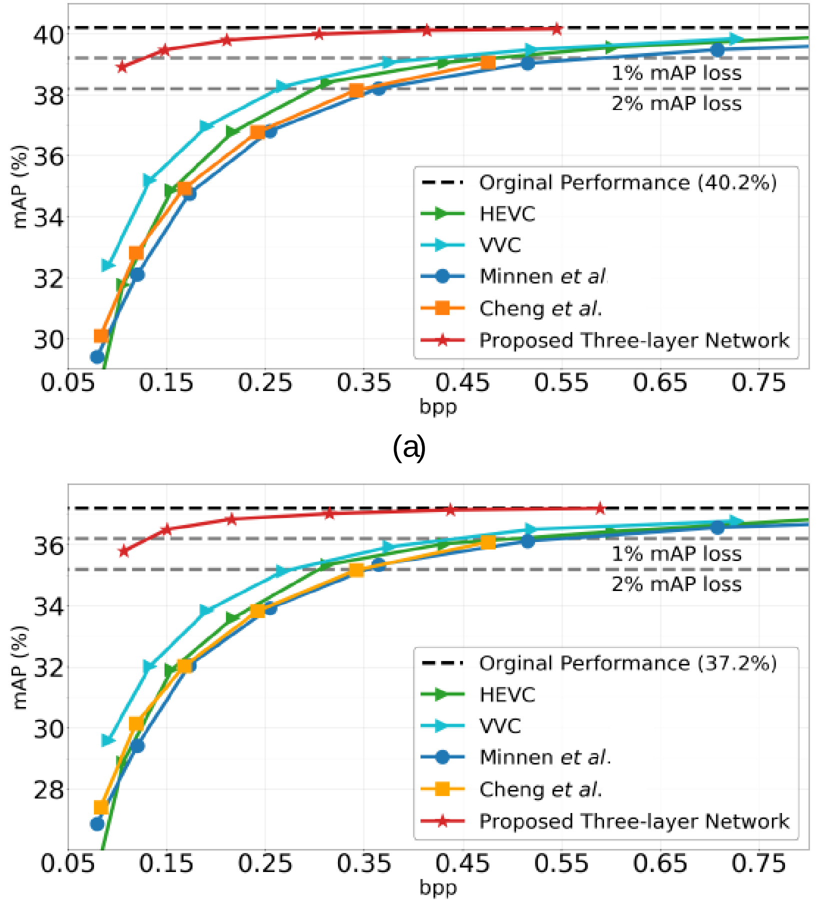
[1] H. Choi and I. V. Bajić, “Scalable image coding for humans and machines,” IEEE TIP, 2022.
[2] Z. Cheng et al., “Learned image compression with discretized gaussian mixture likelihoods and attention modules,” Proc. IEEE CVPR, 2020.
[3] R. Girshick et al., “Detectron,” https://github.com/facebookresearch/detectron, 2018.

LATENT SPACE SCALABILITY

- Detection and segmentation experiments on COCO
- Again, Performance much better than compressing input directly:
  - 71 – 78% bit savings compared to state-of-the-art image codecs
  - 2.3 – 3.5% more accurate detection at the same bit rate

	Three-layer Network			
	Object Detection		Segmentation	
Benchmarks	BD-Bitrate	BD-mAP	BD-Bitrate	BD-mAP
VVC	–73.2	2.33	–71.2	2.34
HEVC	–73.2	3.05	–74.7	2.96
Minnen *et al.*	–78.7	3.73	–77.2	3.38
Cheng *et al.*	–76.6	3.62	–75.4	3.49

3-layer system: (a) object detection, (b) segmentation



(a)

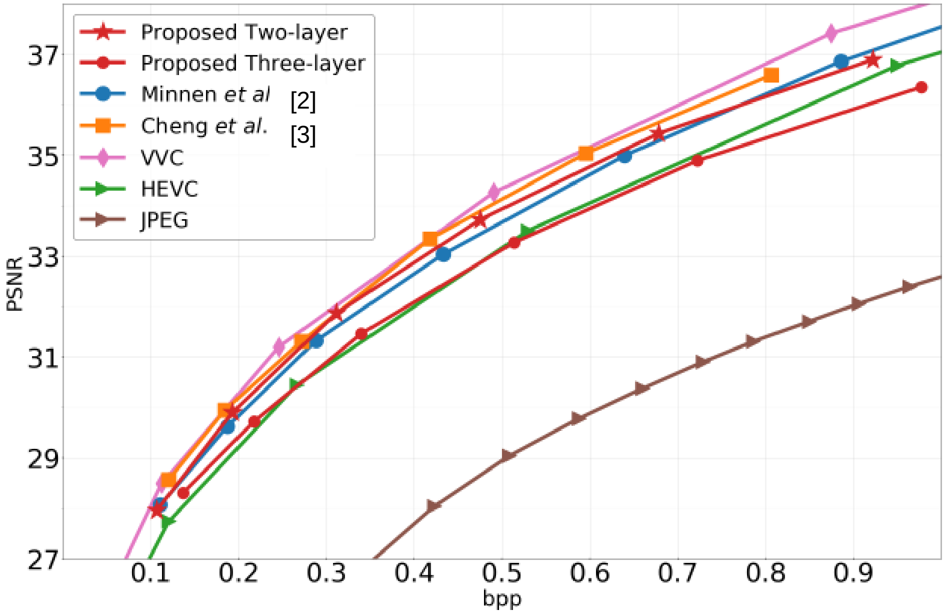
(b)

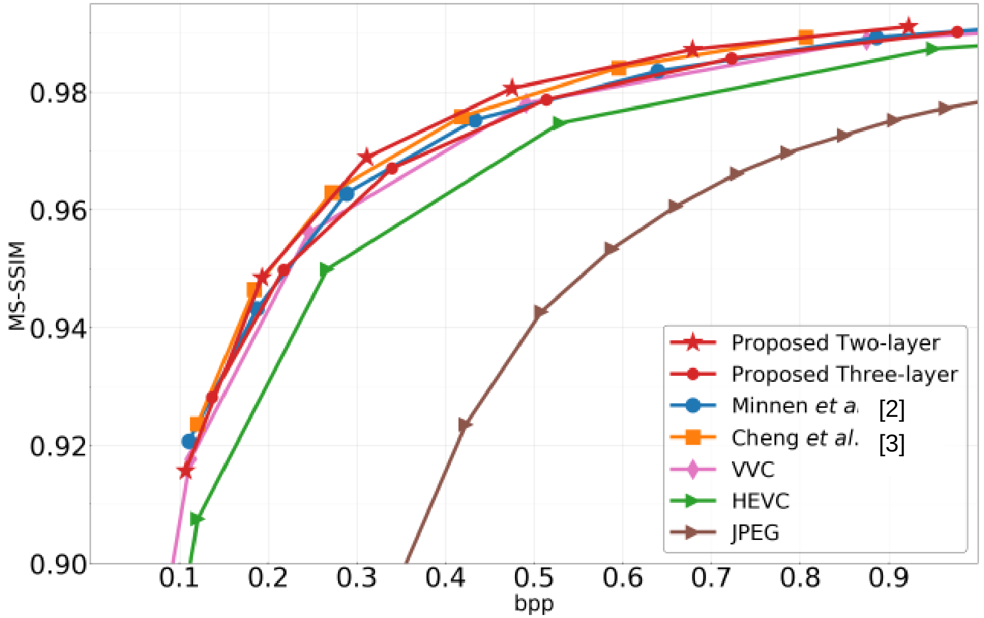
[1] H. Choi and I. V. Bajić, "Scalable image coding for humans and machines," IEEE TIP, 2022

[2] Z. Cheng et al., "Learned image compression with discretized gaussian mixture likelihoods and attention modules," Proc. IEEE CVPR, 2020.

[3] D. Minnen, J. Balle, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," NeurIPS, 2018.

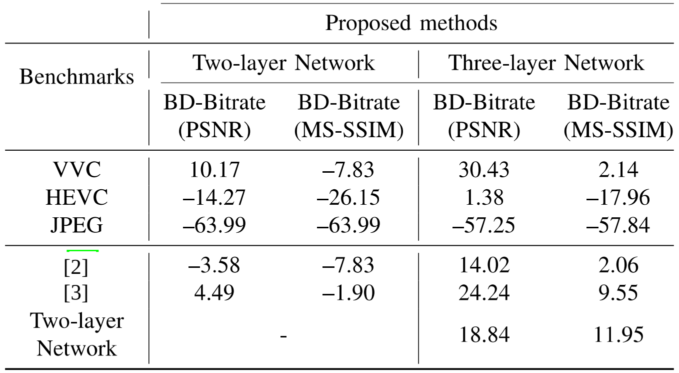
LATENT SPACE SCALABILITY





Results on the Kodak dataset

- Proposed scalable codec comparable to state-of-the-art on input reconstruction
- 10 – 20% degradation by adding a scalability layer (2 → 3), in line with earlier work on scalable video coding



Benchmarks	Proposed methods			
	Two-layer Network		Three-layer Network	
	BD-Bitrate (PSNR)	BD-Bitrate (MS-SSIM)	BD-Bitrate (PSNR)	BD-Bitrate (MS-SSIM)
VVC	10.17	–7.83	30.43	2.14
HEVC	–14.27	–26.15	1.38	–17.96
JPEG	–63.99	–63.99	–57.25	–57.84
[2]	–3.58	–7.83	14.02	2.06
[3]	4.49	–1.90	24.24	9.55
Two-layer Network	-		18.84	11.95

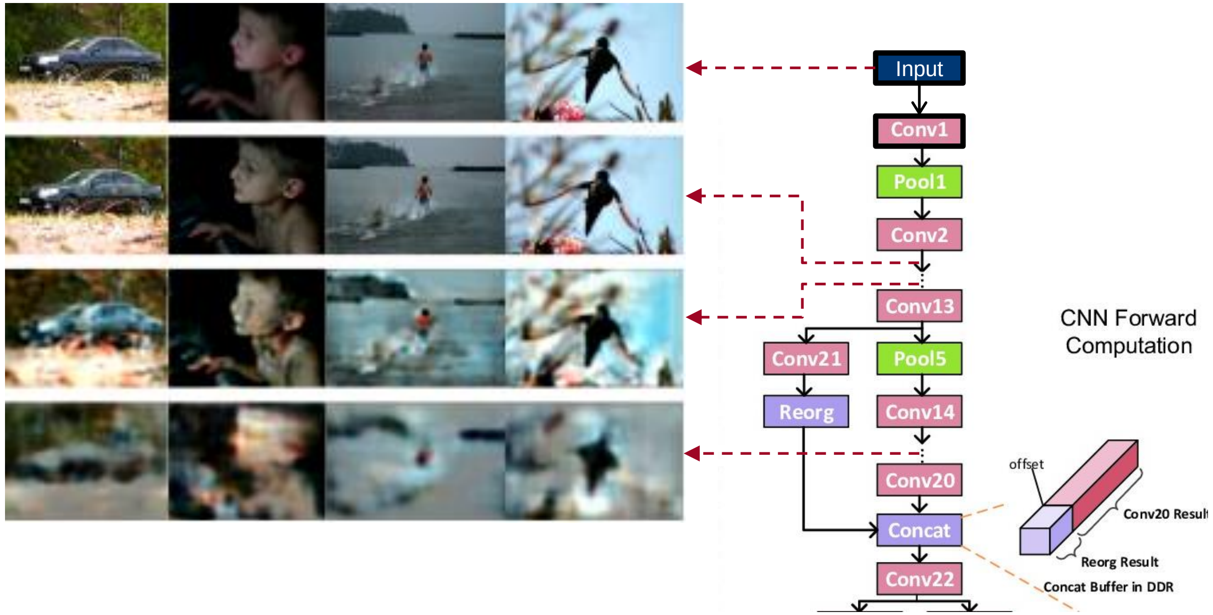
[1] H. Choi and I. V. Bajić, “Scalable image coding for humans and machines,” IEEE TIP, 2022.
[2] D. Minnen, J. Balle, and G. D. Toderici, “Joint autoregressive and hierarchical priors for learned image compression,” NeurIPS, 2018.
[3] Z. Cheng et al., “Learned image compression with discretized gaussian mixture likelihoods and attention modules,” Proc. IEEE CVPR, 2020.

Questions?

PRIVACY

- Features, not input, sent from edge to cloud → potential for privacy
- Are features privacy-preserving?
- Need precise definition of privacy
- Strategies for privacy
    - Resilience to model inversion
    - Adding noise to features
    - Information-theoretic privacy

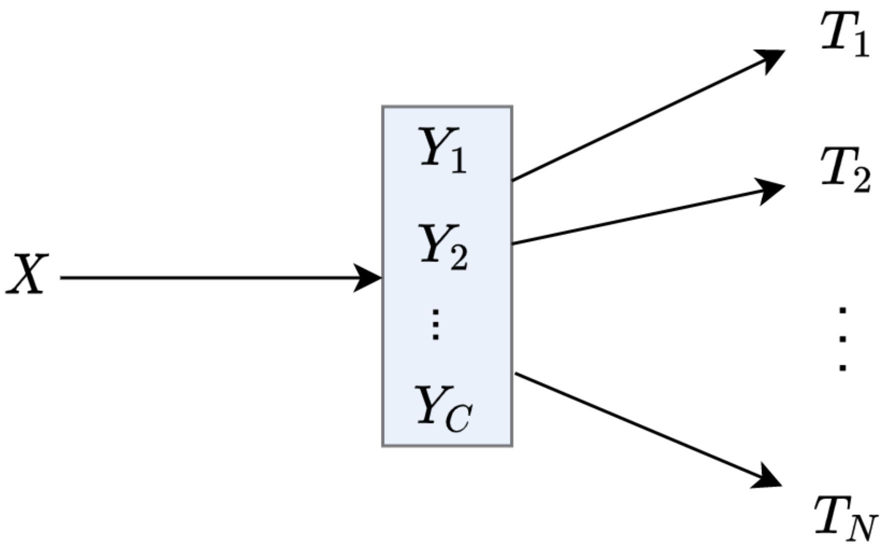
Input reconstruction from YOLOv2



H. Choi and I. V. Bajić, "Near-lossless deep feature compression for collaborative intelligence," Proc. IEEE MMSP, Aug. 2018.

PRIVACY FAN

- “Privacy fan” – an information-theoretic privacy model for collaborative intelligence and multi-task compression
- $Y_1, \ldots, Y_C$ - features
- $T_1, \ldots, T_N$ - tasks
- Some task outputs reveal private information (e.g. input reconstruction), some not
- Let $\mathcal{P}$ be the set of “private” tasks
- Goal: identify a set of features $\mathcal{B}$ that carry minimum information about private tasks, while providing sufficient information about non-private ones



S. R. Alvar and I. V. Bajić, “Scalable privacy in multi-task image compression,” Proc. IEEE VCIP, Dec. 2021.

PRIVACY FAN

- Privacy fan formulation

$$\min_{\mathcal{B}} \sum_{i\in\mathcal{B}} \sum_{j\in\mathcal{P}} I(Y_i;T_j)\,, \qquad \text{such that} \sum_{i\in\mathcal{B}} \sum_{j\notin\mathcal{P}} I(Y_i;T_j) \geq R$$

- Solution: define a Lagrangian $\mathcal{L}_i$ for each feature Y_i:

$$\mathcal{L}_i = \sum_{j\in\mathcal{P}} I(Y_i;T_j) - \beta \cdot \sum_{j\notin\mathcal{P}} I(Y_i;T_j)$$

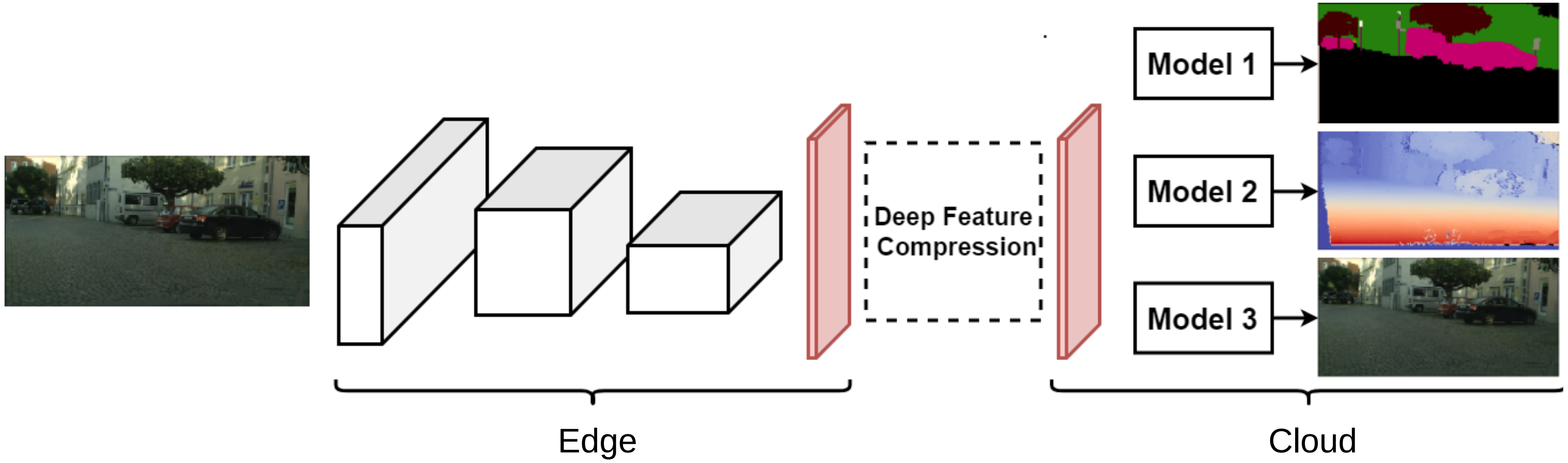
where $\beta > 0$ is the Lagrange multiplier controlling the privacy-accuracy trade-off

  - ○ $\mathcal{B} = \{Y_i \;:\; \mathcal{L}_i < 0\}$

- Special case, practically important: set $\mathcal{B}$ is limited to C' features: $|\mathcal{B}| \leq C'$

S. R. Alvar and I. V. Bajić, “Scalable privacy in multi-task image compression,” Proc. IEEE VCIP, Dec. 2021.

SCALABLE PRIVACY



- 3-task model: $$\mathcal{L}_i = \underbrace{I(Y_i;T_3)}_{\text{Input reconstruction (private)}} - \beta \cdot [\underbrace{I(Y_i;T_1) + I(Y_i;T_2)}_{\text{Segmentation and depth est. (non-private)}}]$$

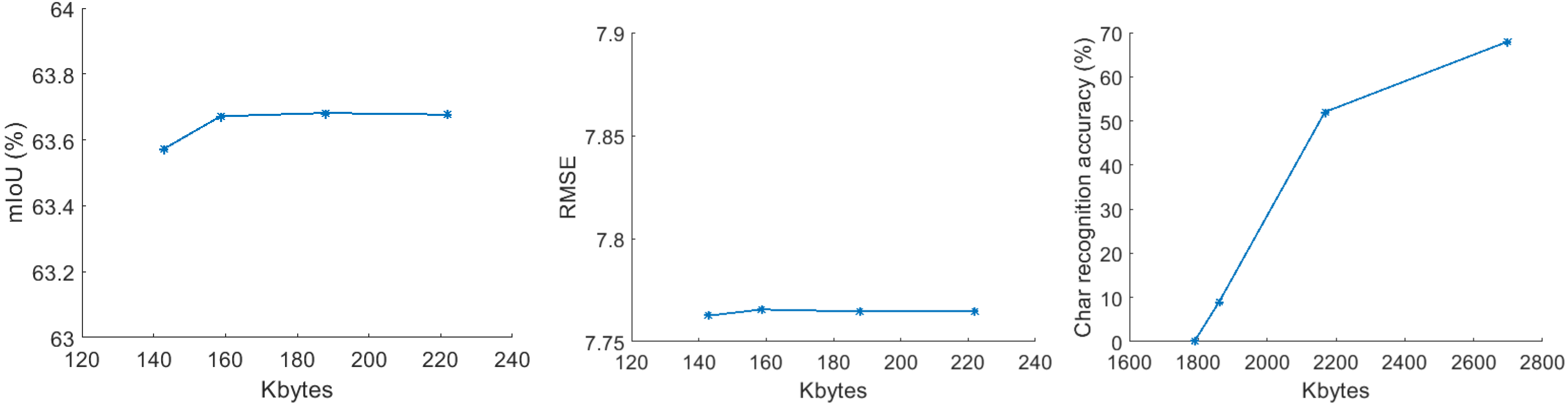
- Obtain set $\mathcal{B}$ by solving the privacy fan – call these “base” features

- Encode “base” features at high quality, other (“enhancement”) features at appropriate quality, depending on the application

S. R. Alvar and I. V. Bajić, “Scalable privacy in multi-task image compression,” Proc. IEEE VCIP, Dec. 2021.

SCALABLE PRIVACY

Varying the rate of enhancement layer



Semantic segmentation

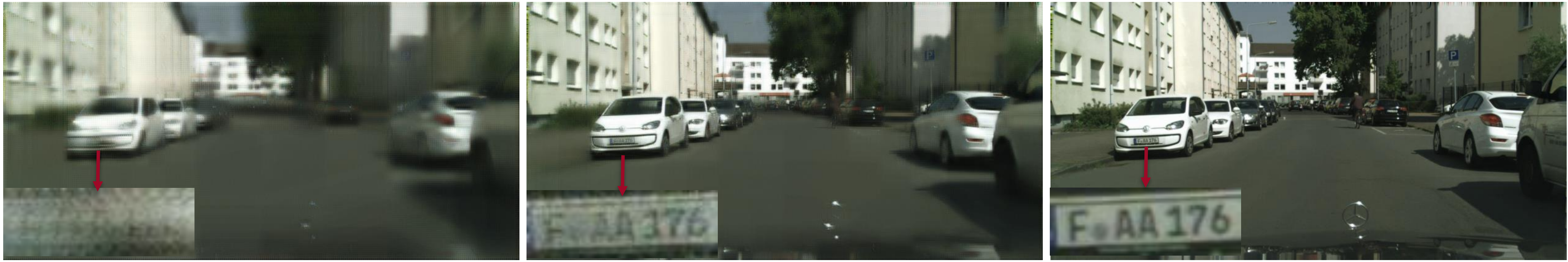
Depth estimation

Character recognition

S. R. Alvar and I. V. Bajić, “Scalable privacy in multi-task image compression,” Proc. IEEE VCIP, Dec. 2021.
S. R. Alvar, K. Uyanik, and I. V. Bajić, “License plate privacy in visual analysis of traffic scenes,” to be presented at IEEE MIPR, Aug. 2022.

SCALABLE PRIVACY

Varying the rate of enhancement layer



S. R. Alvar and I. V. Bajić, “Scalable privacy in multi-task image compression,” Proc. IEEE VCIP, Dec. 2021.
S. R. Alvar, K. Uyanik, and I. V. Bajić, “License plate privacy in visual analysis of traffic scenes,” to be presented at IEEE MIPR, Aug. 2022.

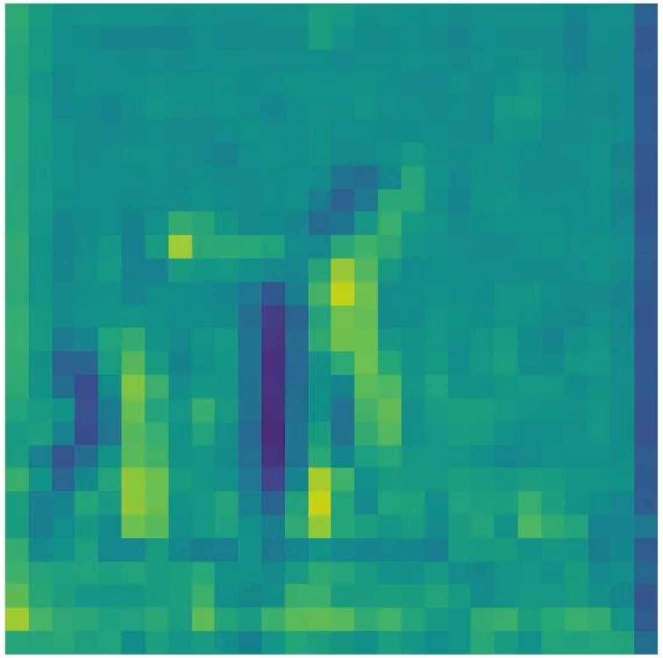
Questions?

LATENT-SPACE MOTION

What is shown in the image?

Observation:

- Input motion seems to be preserved in the latent space
- Why?

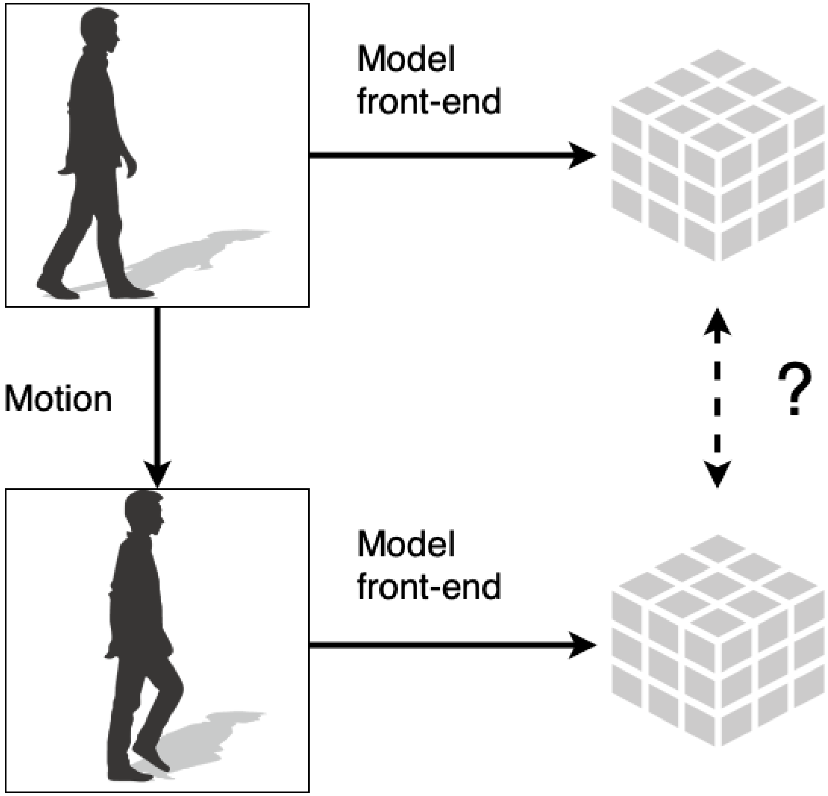


One feature tensor channel from `add_3` layer of ResNet-34

LATENT-SPACE MOTION

Understanding latent-space motion

- Consider motion in the input space between two consecutive frames
- Map each frame to the latent space via the model front—end
- What is the relationship between the corresponding feature tensors?



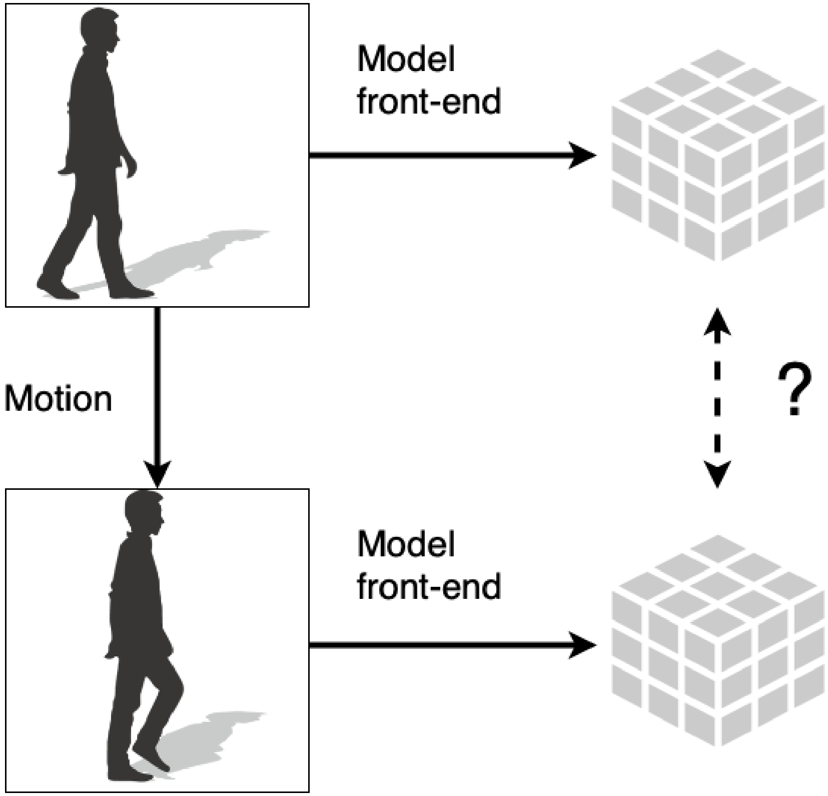
M. Ulhaq and I. V. Bajić, "Latent space motion analysis for collaborative intelligence," Proc. IEEE ICASSP, pp. 8498-8502, Jun. 2021.

LATENT-SPACE MOTION

- A popular motion model in computer vision is “optical flow”:

$$\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} = 0$$

  - I – image intensity; t – time
  - (v_x, v_y) – optical flow



- If this model describes motion in the input space, what it its equivalent in the latent space?
- Note: the same equation was used to describe “surface flow” in PDE-based inpainting
  - Can reuse that analysis, but interpretation slightly different

M. Ulhaq and I. V. Bajić, ”Latent space motion analysis for collaborative intelligence," Proc. IEEE ICASSP, pp. 8498-8502, Jun. 2021.

LATENT-SPACE MOTION

- When input image I is convolved with kernel f, the resulting flow equation is

$$\frac{\partial}{\partial x}(f * I)u_x + \frac{\partial}{\partial y}(f * I)u_y + \frac{\partial}{\partial t}(f * I) = 0$$

where (u_x, u_y) is the flow field after convolution

- Convolution and differentiation commute:

$$f * \underbrace{\left(\frac{\partial I}{\partial x}u_x + \frac{\partial I}{\partial y}u_y + \frac{\partial I}{\partial t}\right)}_{\text{same flow equation as in input space} \implies \text{solution to input flow is one solution to output flow}} = 0$$

M. Ulhaq and I. V. Bajić, ”Latent space motion analysis for collaborative intelligence," Proc. IEEE ICASSP, pp. 8498-8502, Jun. 2021.

LATENT-SPACE MOTION

- When input image I passes through nonlinear activation $\sigma(\cdot)$, the resulting flow equation is

$$\frac{\partial\, \sigma(I)}{\partial x} u_x + \frac{\partial\, \sigma(I)}{\partial y} u_y + \frac{\partial\, \sigma(I)}{\partial t} = 0$$

where (u_x, u_y) is the flow field after nonlinear activation

- Using the chain rule of differentiation:

$$\sigma'(I) \cdot \underbrace{\left(\frac{\partial I}{\partial x} u_x + \frac{\partial I}{\partial y} u_y + \frac{\partial I}{\partial t} \right)} = 0$$

same flow equation as in input space $\Longrightarrow$ solution to input flow is one solution to output flow

M. Ulhaq and I. V. Bajić, ”Latent space motion analysis for collaborative intelligence," Proc. IEEE ICASSP, pp. 8498-8502, Jun. 2021.

LATENT-SPACE MOTION

- Following pooling operations are commonly found in deep neural networks:
  - Max pooling
  - Mean pooling
  - Learnt pooling (strided convolution)
- Each of these pooling operations can be represented as

  spatial operation followed by *spatial scale change*

  where *spatial operation* is
  - Local maximum in case of max pooling
  - Local average in case of mean pooling
  - Weighted local average in case of learnt pooling
- Note: (weighted) local average is a special case of of convolution, and it was already shown that optical flow before convolution is one solution to the optical flow after the convolution

M. Ulhaq and I. V. Bajić, "Latent space motion analysis for collaborative intelligence," Proc. IEEE ICASSP, pp. 8498-8502, Jun. 2021.

LATENT-SPACE MOTION

- Using a linear approximation to the local maximum, it can also be shown [1, 2] that optical flow prior to local max is an approximate solution to the optical flow after the local max
- For the spatial scale change, define $I_s(x, y, t) = I(s \cdot x, s \cdot y, t)$ to be the signal after scale change in x- and y-directions by a factor of s, and consider optical flow in I_s:

$$\frac{\partial I_s}{\partial x} u_x + \frac{\partial I_s}{\partial y} u_y + \frac{\partial I_s}{\partial t} = 0$$

where (u_x, u_y) is the flow field after spatial scale change

- Since $\frac{\partial I_s}{\partial x} = s \cdot \frac{\partial I}{\partial x}$, $\frac{\partial I_s}{\partial y} = s \cdot \frac{\partial I}{\partial y}$, and $\frac{\partial I_s}{\partial t} = \frac{\partial I}{\partial t}$, the new field is

$$(u_x, u_y) = \left(\frac{v_x}{s}, \frac{v_y}{s}\right)$$

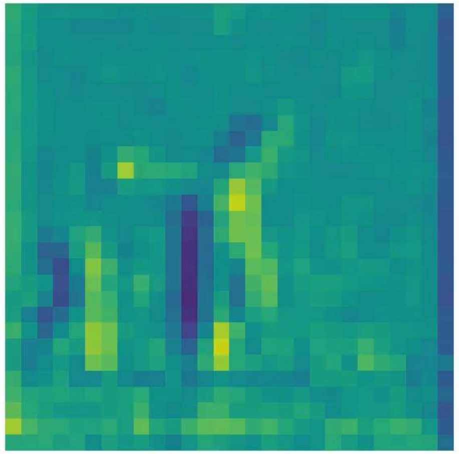
- Hence, after spatial scale change, flow field scales accordingly

[1] M. Ulhaq and I. V. Bajić, ”Latent space motion analysis for collaborative intelligence," Proc. IEEE ICASSP, pp. 8498-8502, Jun. 2021.
[2] I. V. Bajić, “Latent space inpainting for loss-resilient collaborative object detection,” Proc. IEEE ICC, Montreal, Canada, Jun. 2021.

LATENT-SPACE MOTION

Summary

- Optical flow of the input remains one (approximate) solution to the optical flow after common operations (convolution, nonlinear activation, pooling, etc.)
- Pooling with a spatial scale change causes a corresponding scale change in the optical flow
  - For example, 2 × 2 pooling scales the flow field by a factor of ½
- This is why input motion is approximately preserved in the latent space
- Motion compensation from video coding may be a good strategy for compression of sequences of feature tensors derived from input video



M. Ulhaq and I. V. Bajić, "Latent space motion analysis for collaborative intelligence," Proc. IEEE ICASSP, pp. 8498-8502, Jun. 2021.

Questions?

Part 3

Standardization

STANDARDIZATION

- Standards are important
  - Ensure interoperability
  - Give device developers confidence that their products will have a large market
- There are a number of standardization activities related to collaborative intelligence and, more broadly, IoT
- We will briefly describe those related to compression:
  - JPEG AI (Joint Photographic Experts Group – Artificial Intelligence)
  - MPEG-VCM (Motion Pictures Experts Group – Video Coding for Machines)

I. V. Bajić, W. Lin, and Y. Tian, ”Collaborative intelligence: Challenges and opportunities," Proc. IEEE ICASSP, pp. 8493-8497, Jun. 2021.
W. Gao et al., “Recent standard development activities on Video Coding for Machines,” arXiv:2105.12653, May 2021.

JPEG AI

- Scope

  “*The scope of the JPEG AI is the creation of a learning-based image coding standard offering a **single-stream, compact** compressed domain representation, targeting both **human visualization**, with significant compression efficiency improvement over image coding standards in common use at equivalent subjective quality, and effective performance for **image processing and computer vision tasks**, with the goal of supporting a **royalty-free baseline**.*” [JPEG AI White Paper, 2021]

- Difference from earlier image coding standards
  - Learning-based
  - Support for image processing and computer vision tasks (besides human vision)

https://jpeg.org/jpegai/
ISO/IEC JTC 1/SC29/WG1 N90049, "White Paper on JPEG AI Scope and Framework v1.0," 2021.

JPEG AI

- Use cases
  - Cloud storage
  - Visual surveillance
  - Autonomous vehicles and devices
  - Image collection storage and management
  - Live monitoring of visual data
  - Media distribution
  - Television broadcast distribution and editing

ISO/IEC JTC 1/SC29/WG1 N92014, REQ "JPEG AI Second Draft Call for Proposals," 92nd Meeting, July 2021.

JPEG AI

- Examples of image processing tasks
  - Super-resolution
  - Denoising
  - Low-light enhancement, exposure compensation, color correction
  - Inpainting
- Examples of computer vision tasks
  - Image classification
  - Object/face detection, recognition, identification
  - Semantic segmentation
  - Event detection, action recognition

ISO/IEC JTC 1/SC29/WG1 N92014, REQ "JPEG AI Second Draft Call for Proposals," 92nd Meeting, July 2021.
ISO/IEC JTC 1/SC29/WG1 N100190, REQ " Submission Instructions for the JPEG AI Call for Proposals," 95th Meeting, April 2022.

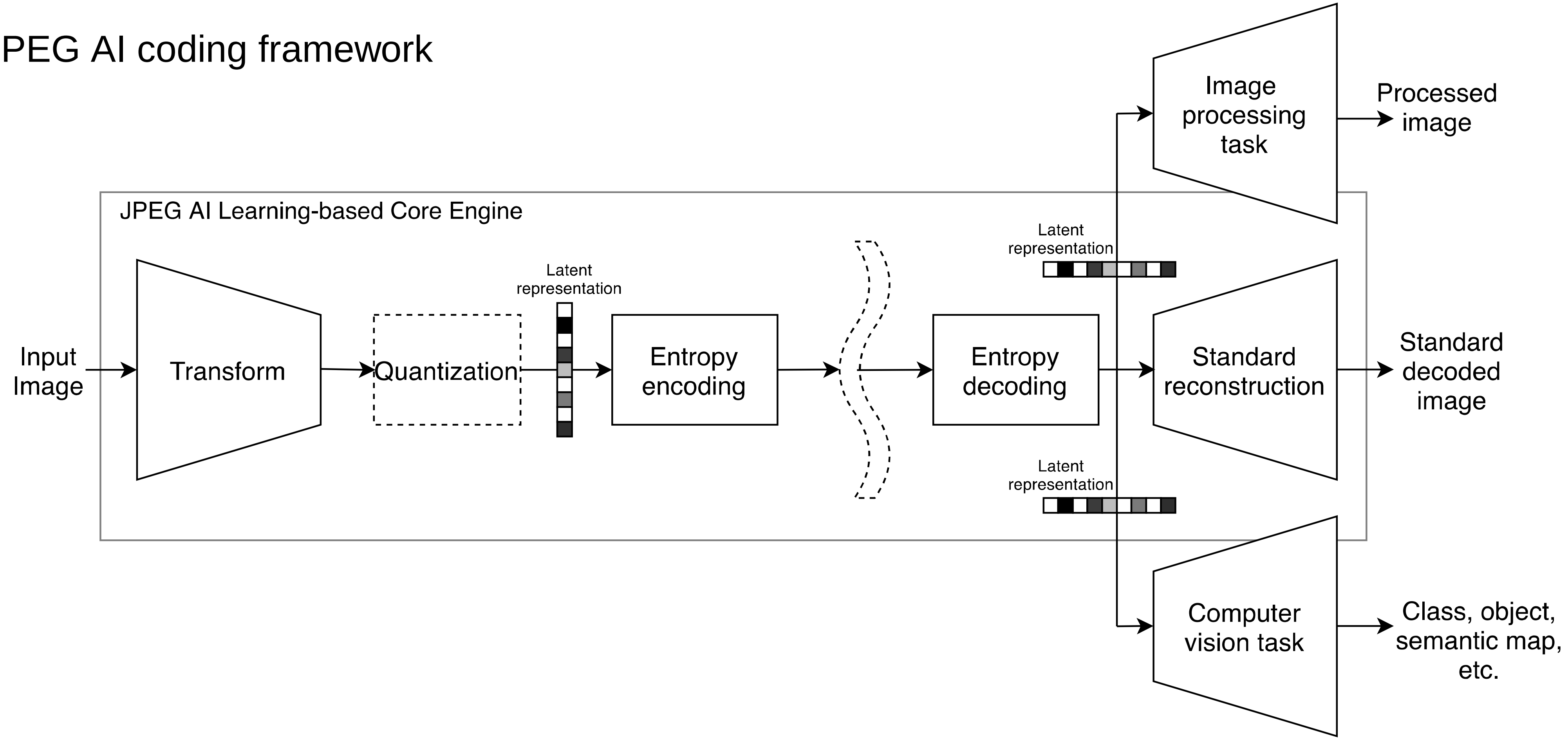
JPEG AI

- Timeline
  - January 2022 – Final Call for Proposals
  - February 2022 – Proposal registration
  - April 2022 – Proposal submission
  - ...
  - October 2023 – Draft standard
  - April 2024 – Final standard

ISO/IEC JTC 1/SC29/WG1 N92014, REQ "JPEG AI Second Draft Call for Proposals," 92nd Meeting, July 2021.

JPEG AI

JPEG AI coding framework



ISO/IEC JTC 1/SC29/WG1 N92014, REQ "JPEG AI Second Draft Call for Proposals," 92nd Meeting, July 2021.

MPEG-VCM

- Scope

  “*MPEG-VCM aims to define a bitstream from* ***compressing video or feature extracted from video*** *that is efficient in terms of bitrate/size and can be* ***used by a network of machines after decompression*** *to perform multiple tasks without significantly degrading task performance. The decoded video or feature can be used for* ***machine consumption or hybrid machine and human consumption****.*

  The differences between VCM and video coding with deep learning are:

  1. *VCM is used for machine consumption or hybrid machine and human consumption, while current video coding aims for human consumption;*
  2. *VCM technologies could be but is not required to be based on deep learning*
  3. *VCM can achieve analysis efficiency, computational offloading and privacy protection as well as compression efficiency, while traditional video coding pursues mainly on compression efficiency.* ” [VCM m57648 , 2021]

Y. Zhang et al., "[VCM] Updates to use cases and requirements for video coding for machines", m57648, July 2021.

MPEG-VCM

- Use cases
  - Surveillance
  - Intelligent transportation
  - Smart city
  - Intelligent industry
  - Intelligent content
  - Consumer electronics
  - Smart retail
  - Smart agriculture
  - Autonomous vehicles / UAV

Y. Zhang et al., "[VCM] Updates to use cases and requirements for video coding for machines", m57648, July 2021.

MPEG-VCM

- Examples of image processing tasks
  - Image/video enhancement
  - Stereo/Multiview processing
- Examples of computer vision tasks
  - Object detection, segmentation, masking, tracking, measurement
  - Event search, detection, prediction
  - Anomaly detection
  - Crowd density estimation
  - Pose estimation and tracking

Y. Zhang et al., "[VCM] Updates to use cases and requirements for video coding for machines", m57648, July 2021.
ISO/IEC JTC 1/SC 29/WG 2, “Evaluation Framework for Video Coding for Machines ,” N0193, Apr. 2022.

MPEG-VCM

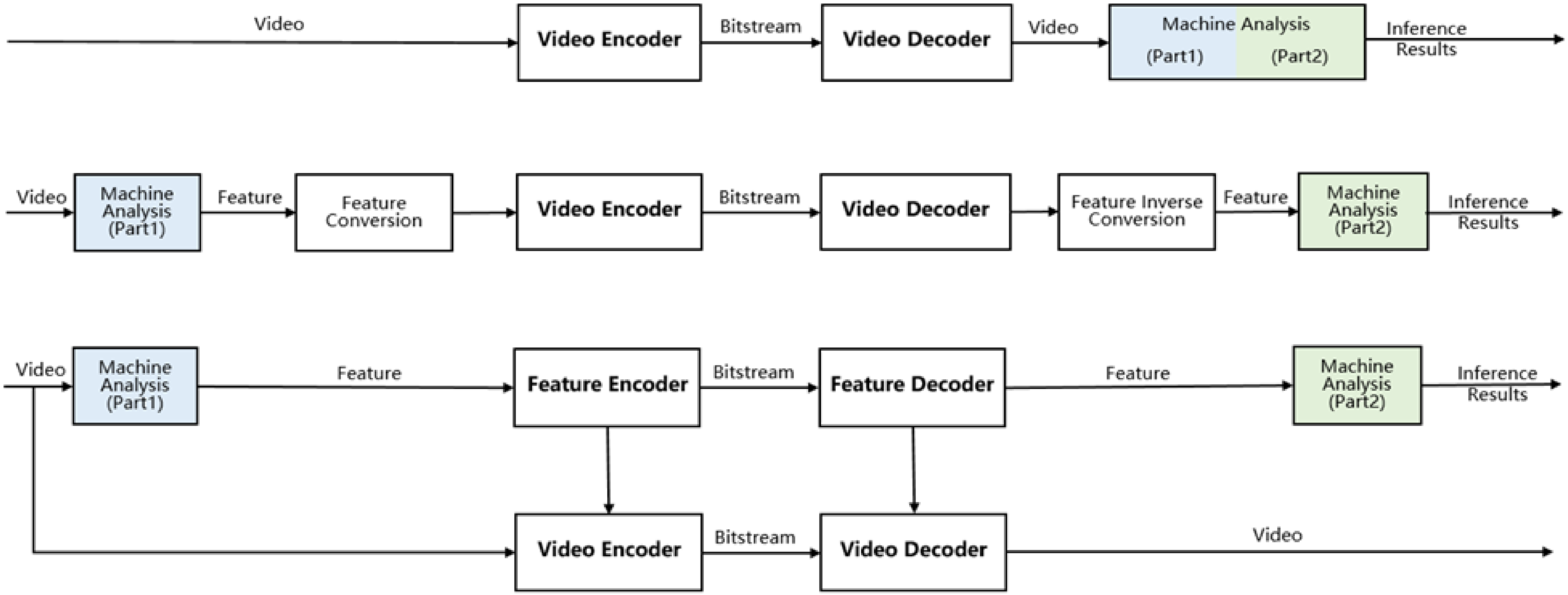
- Track 1 – Feature extraction and compression
  - Focus on machine vision
  - Call for Evidence: July 2022

- Track 2 – Image and video compression
  - Both human and machine vision
  - Call for Proposals: April 2022

M. Rafie et al., "AhG on report on video coding for machines," m59226, April 2022.

MPEG-VCM

Coding pipelines under consideration



ISO/IEC JTC 1/SC29/WG2 N78, ”Evaluation Framework for Video Coding for Machines," April 2021.

Questions?

SUMMARY

- Collaborative intelligence – a perfect fit for IoT
  - Enables lower latency and better energy efficiency at the edge
  - Still take advantage of computing resources in the cloud
- What we have learned:
  - Features produced by neural networks are more compressible than the input
  - They have their own structure, which allows recovering missing data
  - Approximate invariance to flow PDE – enables data recovery and explains why motion is preserved in the latent space
  - Privacy fan model for privacy protection in collaborative intelligence
  - Various methods for single- and multi-stream feature compression – more to come in the near future
  - Related standardization activities: JPEG AI and MPEG-VCM

Thank you!