Task-oriented communication for edge AI

From "how to communicate" to "what to communicate"

Jun Zhang

MeditCom 2021 tutorial

Outline

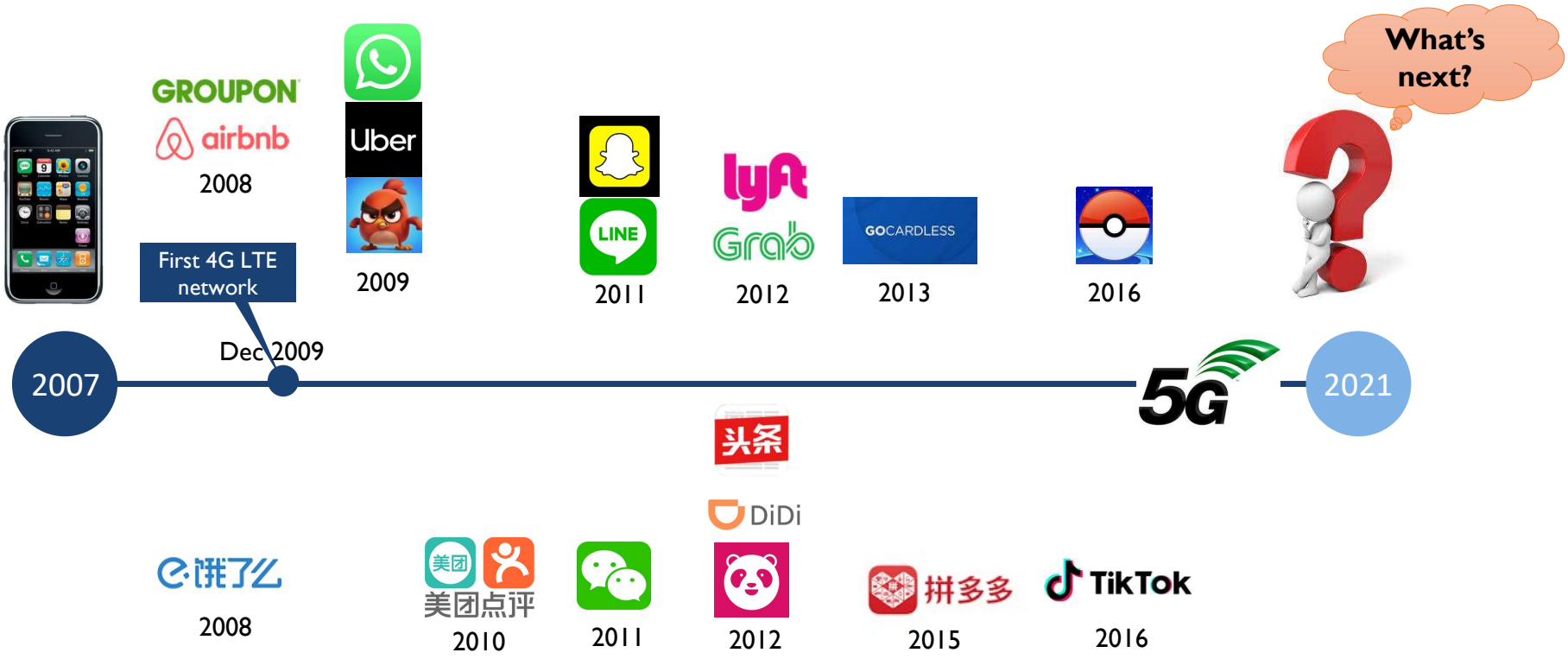
- Introduction
- Edge Computing and Edge AI
  - Mobile edge computing
  - Edge-assisted inference
- From data-oriented to task-oriented communication
- Task-oriented communication
  - End-to-end design
  - Feature encoding via information bottleneck
  - Communication-computation tradeoff
  - Distributed feature encoding for cooperative inference
- Conclusions

Introduction

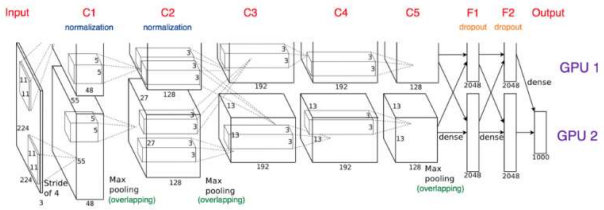
From mobile Internet of mobile intelligence

"When wireless is perfectly applied the whole earth will be converted into a huge brain, which in fact it is, all things being particles of a real and rhythmic whole. We shall be able to communicate with one another instantly, irrespective of distance. Not only this, but through television and telephony we shall see and hear one another as perfectly as though we were face to face, despite intervening distances of thousands of miles; and the instruments through which we shall be able to do this will [fit in a] vest pocket."

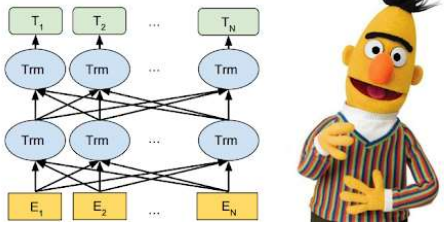
The era of Mobile Internet



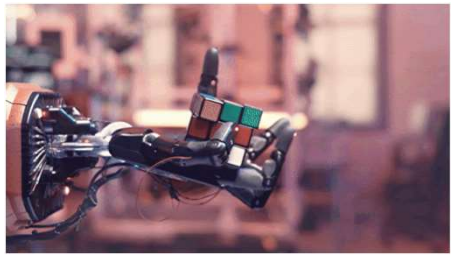
The rising of AI

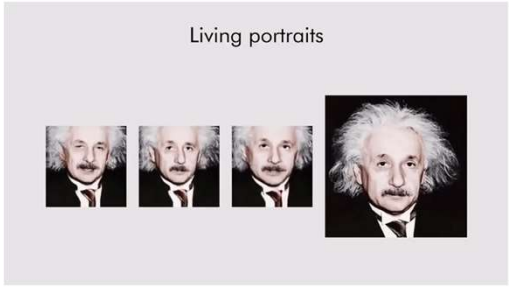


AlexNet (2012)



BERT (2018)

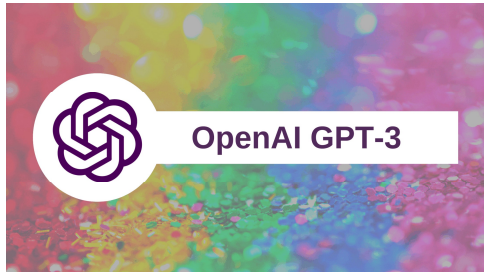




(2019)

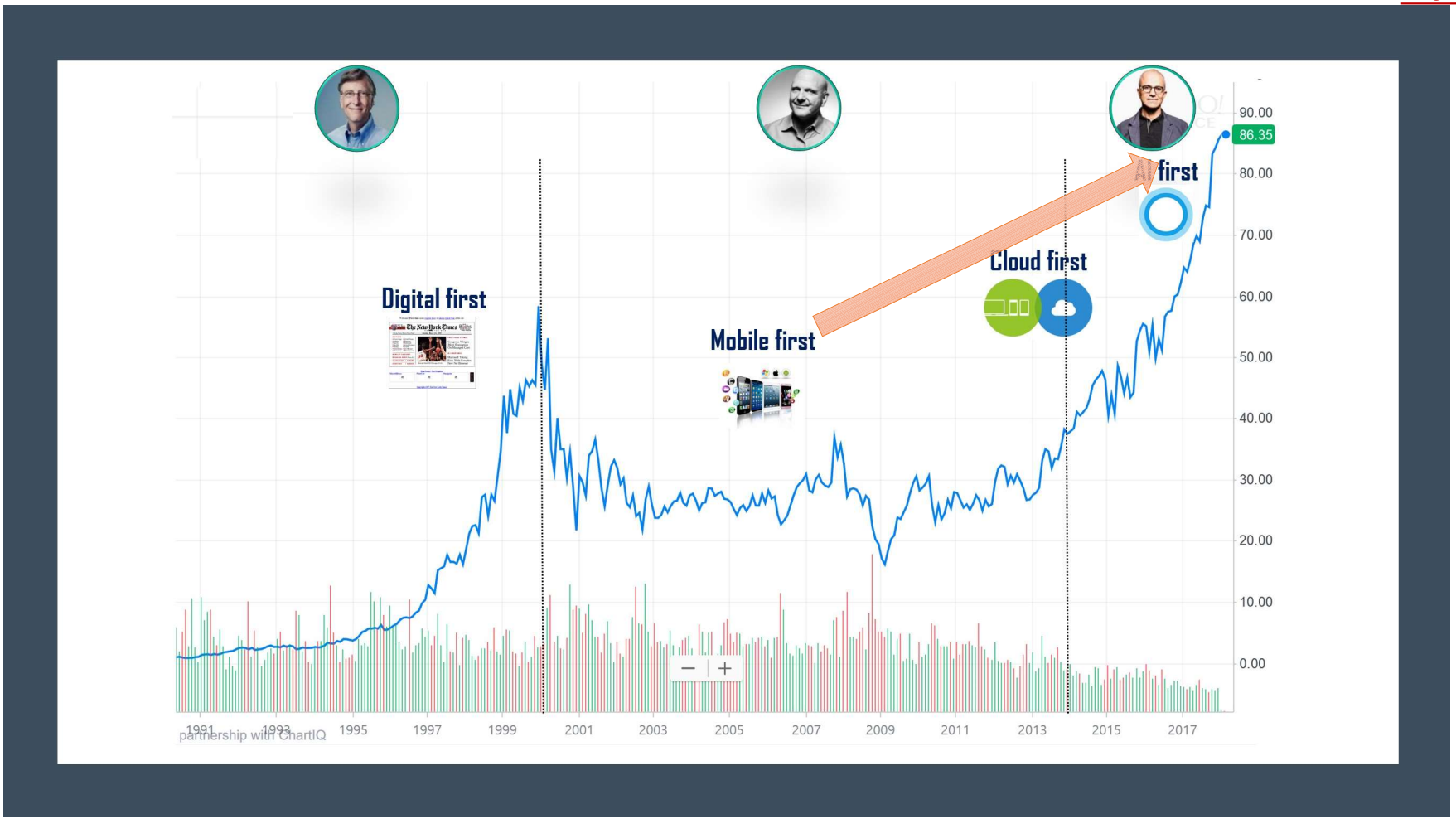
GAN (2014)

AlphaGo (2016)



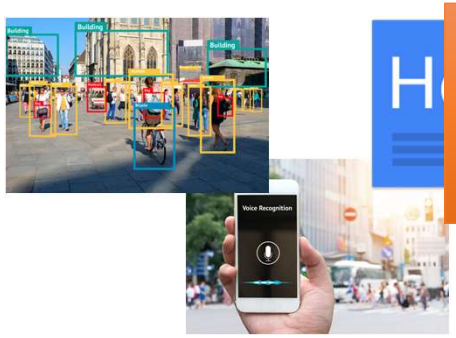
(2020)

2012 — 2021



https://marionoioso.com/2018/01/08/from-digital-first-to-ai-first/

Mobile Intelligence

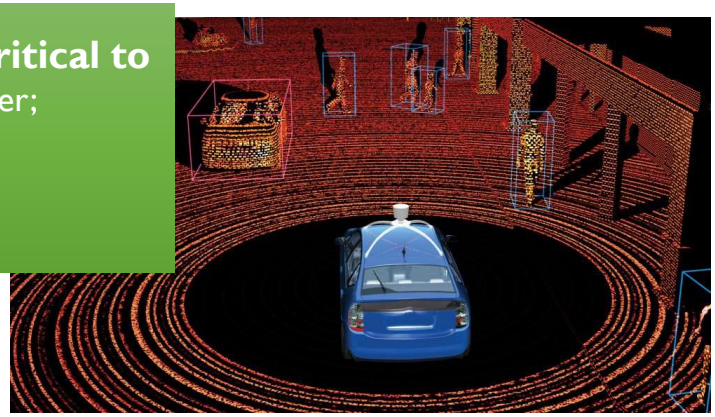


A single device is limited in

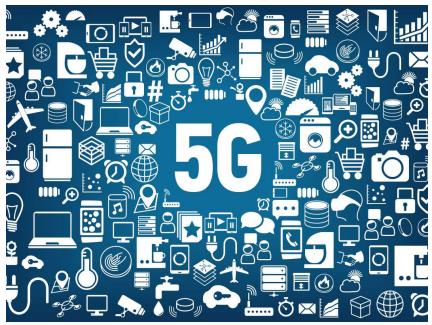
- onboard computing resources;
- limited perception capability;
- limited energy supply.

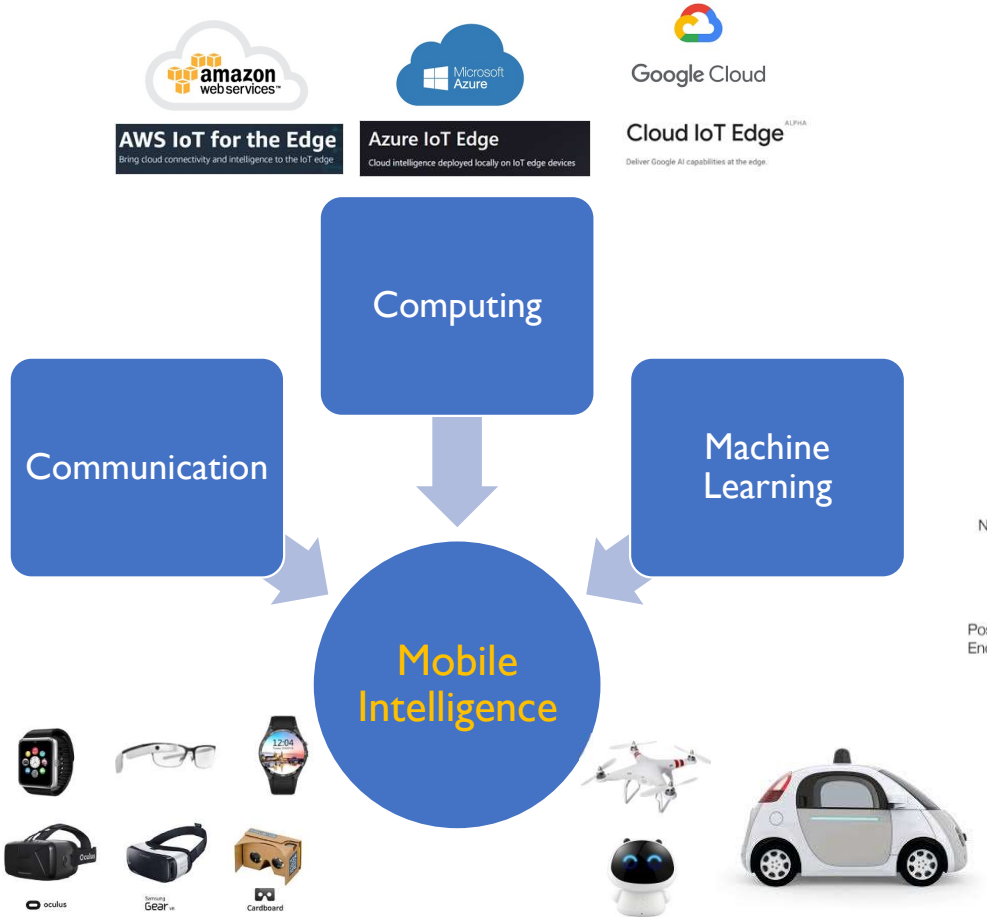
Effective communication is critical to

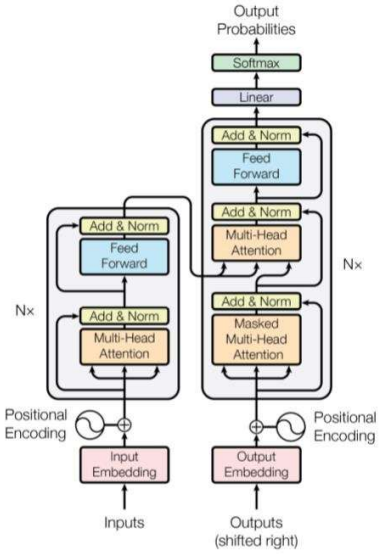
- access external computing power;
- improve perception capability;
- prolong battery time;
- overcome partial observation.



Mobile Intelligence







New communication challenges

- **Enormous volume of data**
  - For example, 4TB sensing data/day for autonomous vehicles
- **Communication for computing/inference**
  - Mobile edge computing, edge AI
- **Low-latency communication**
  - Millisecond-level latency for safety-critical applications
- **Resource-constrained devices**
  - Limited onboard computation and communication resources

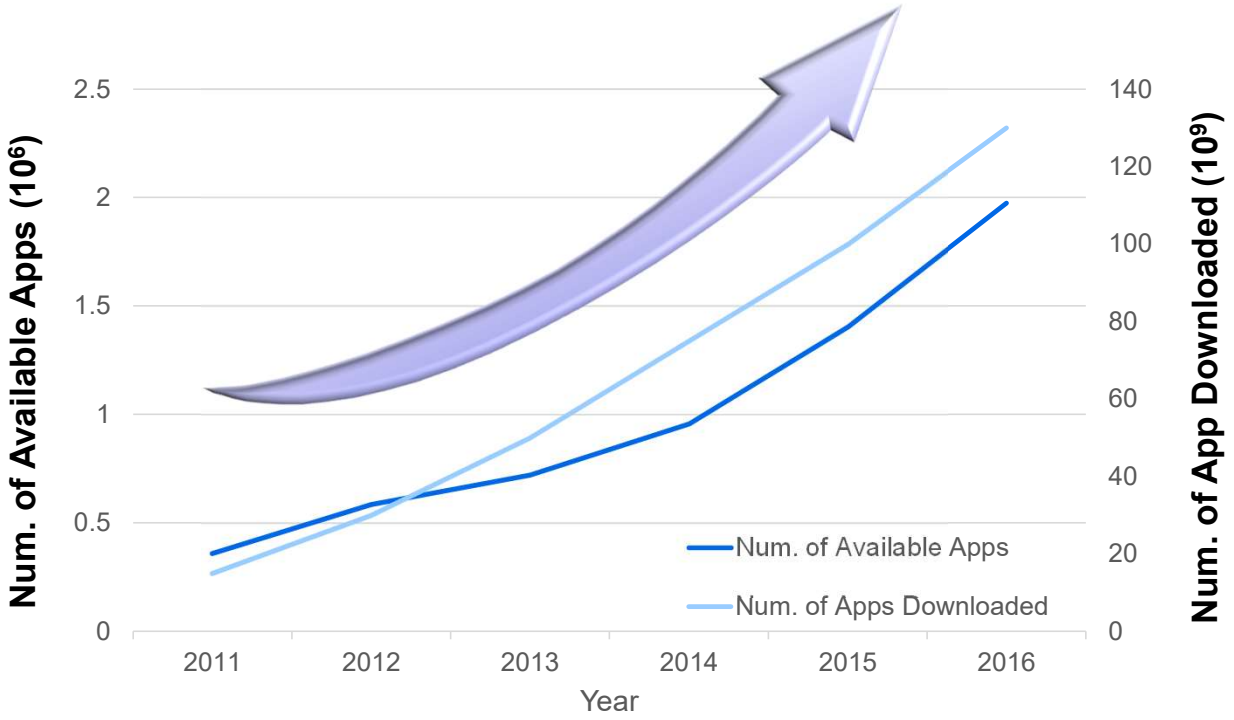
Edge Computing and Edge AI

From Cloud to Edge

The Tipping Point

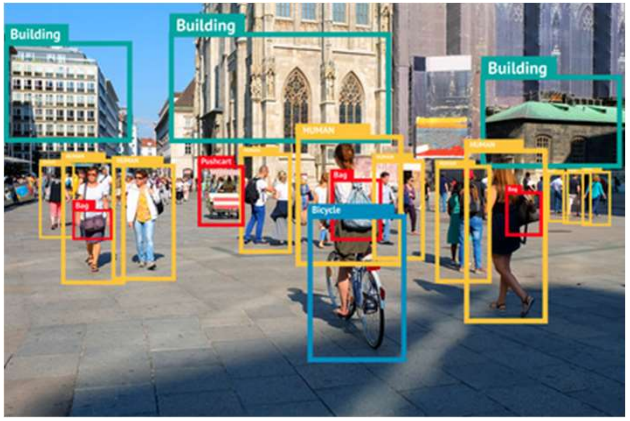
Before June 29, 2007 After

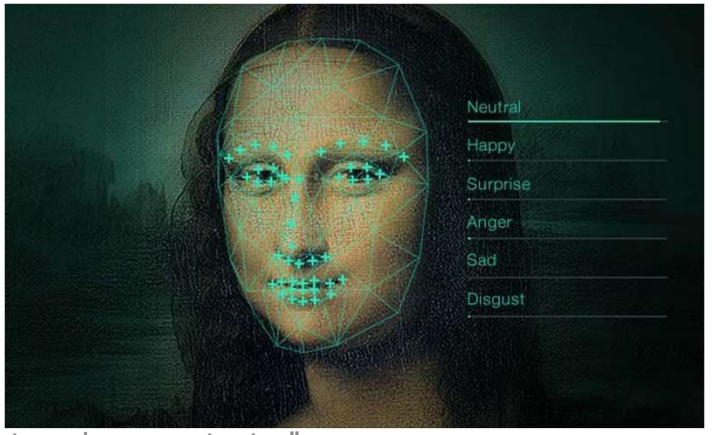
Growth of Mobile App Markets



[Source: Statista]

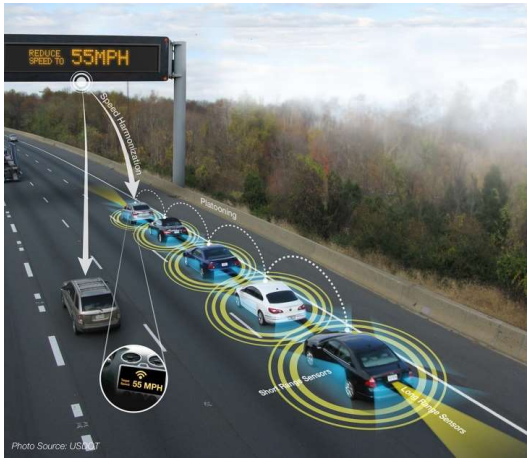
NEED *for intensive and low-latency computation*





NEED FOR SPEED

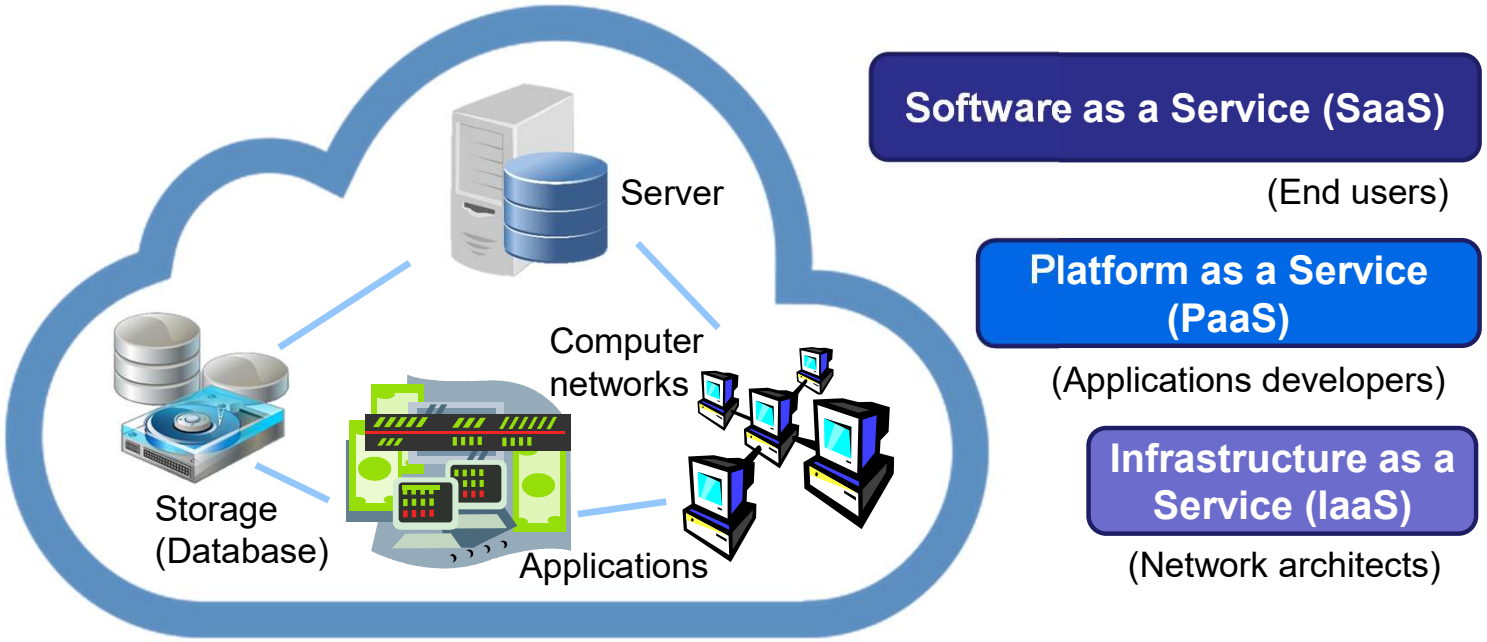
- VR/AR
  - Latency < 20 ms
  - Avoid cybersickness



- Autonomous Driving
  - For platooning control
  - Latency < 100 ms

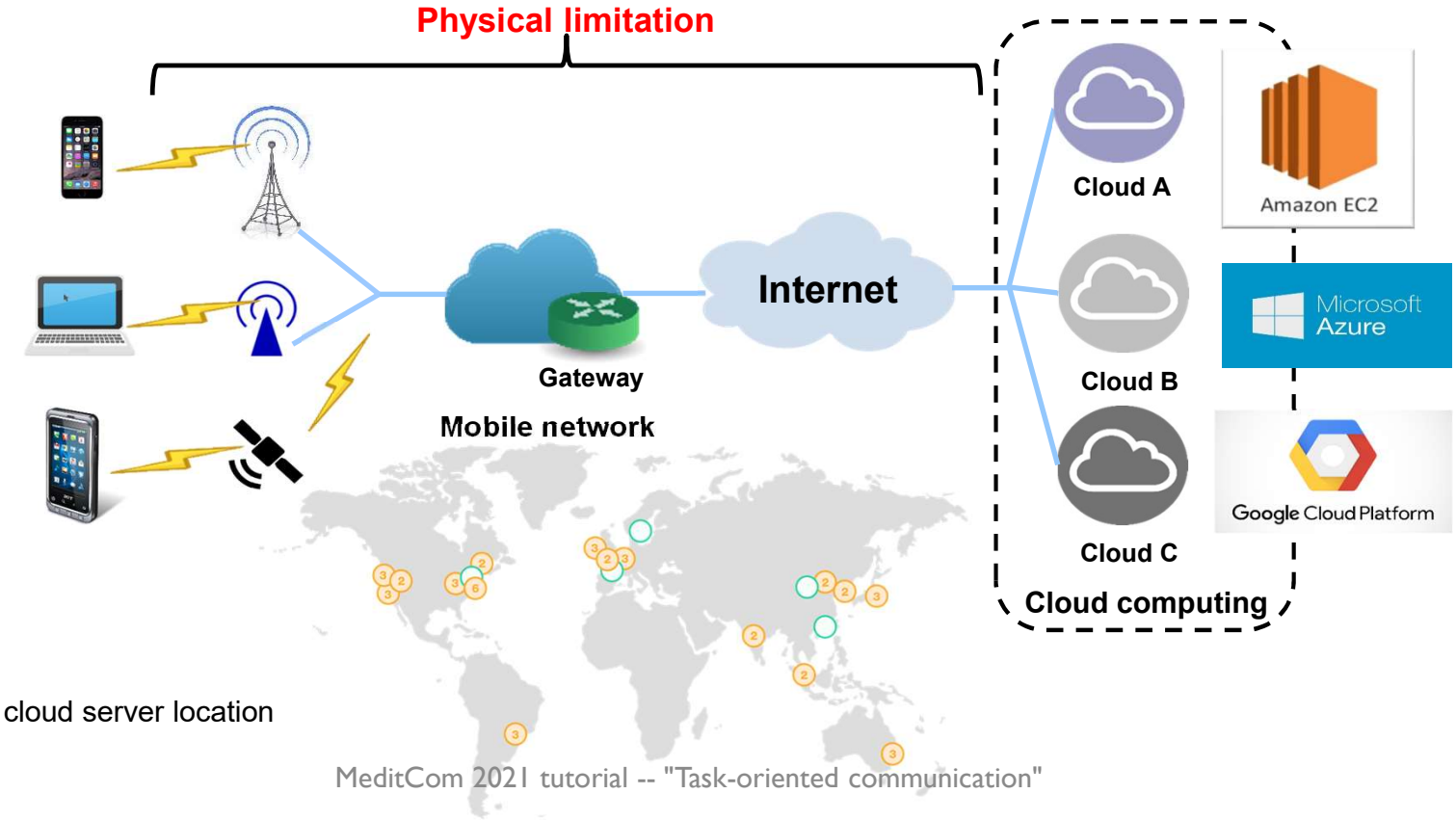
Old Paradigm for Mobile Computing (I)

- Cloud computing
  - *"Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand"*



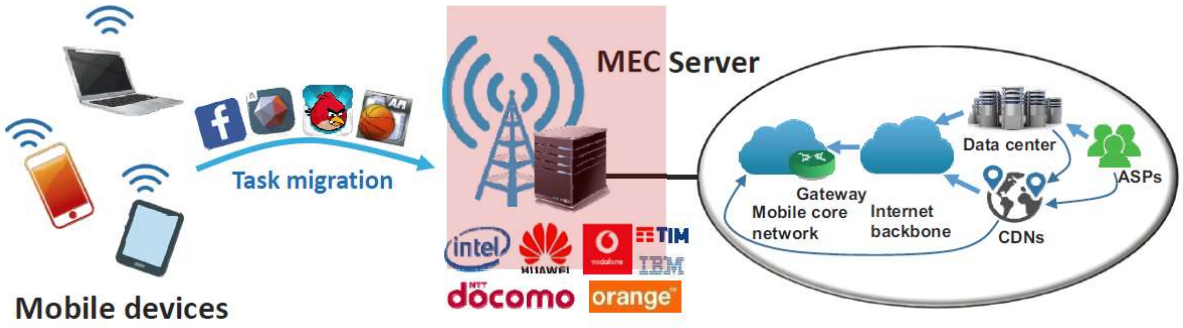
Old Paradigm for Mobile Computing (II)

- Mobile cloud computing (MCC)

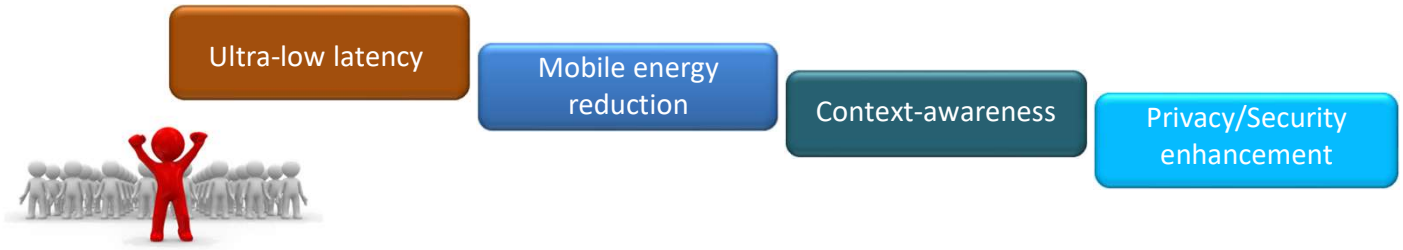


Amazon cloud server location

A new paradigm: Mobile Edge Computing (MEC)



- *European Telecommunications Standard Institute* (ETSI), 2014
  - MEC "*provides IT and cloud-computing capabilities within the Radio Access Network (RAN) in close proximity to mobile subscribers*"

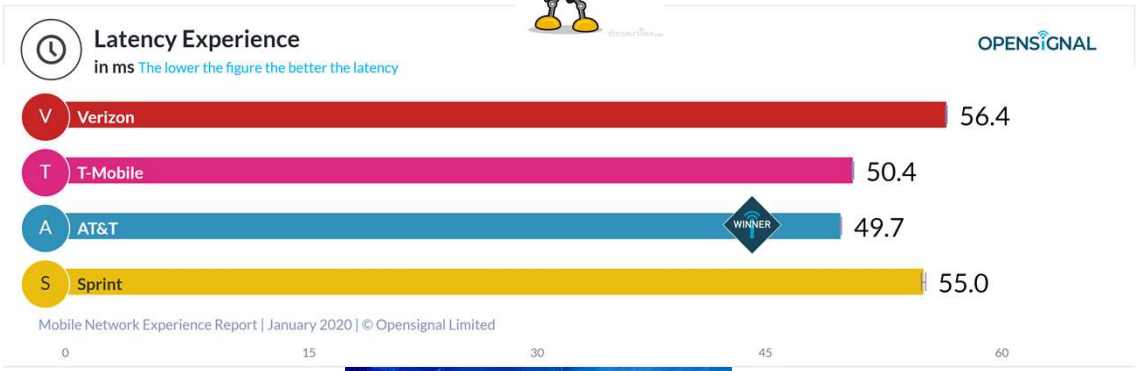


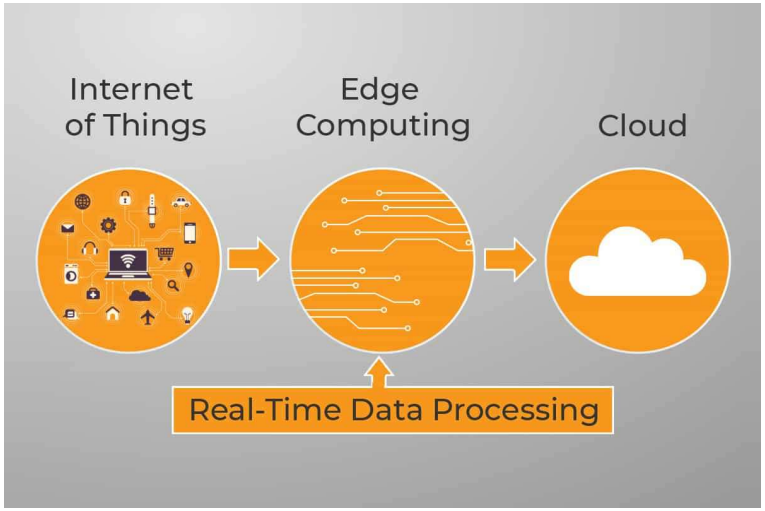
Communication Latency

Communication Latency = **Access Latency** + **Propagation Latency**

5G and beyond

Live on the edge

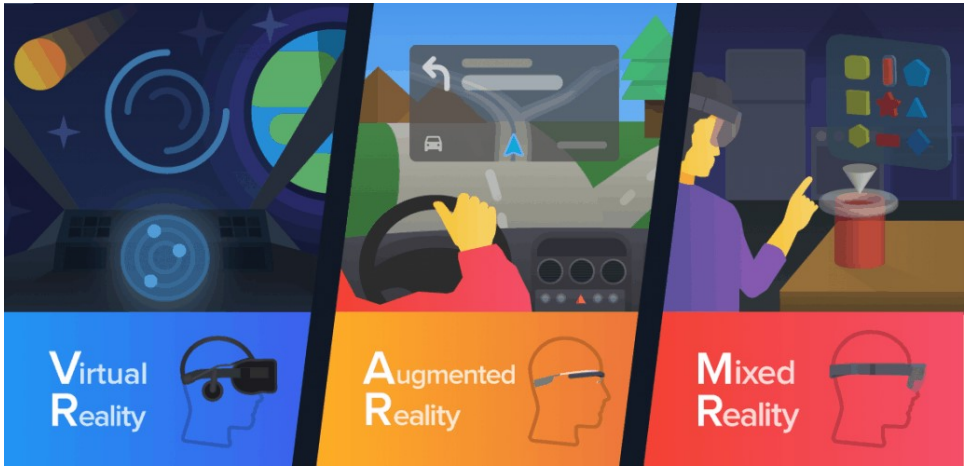




MEC vs. MCC

	Mobile Edge Computing	Mobile Cloud Computing
Hardware	Small-scale data centers	Large-scale data centers
Server location	Co-located with wireless gateways, WiFi routers and BSs	Installed at dedicated buildings
Deployment	Lightweight configuration and planning	Sophisticated configuration and planning
Backhaul Usage	Infrequency use, alleviate congestion	Frequent use, likely to cause congestion
Distance to Users	Tens to hundreds of meters	Across the country boarders

Application: XR



Edge-assisted VR

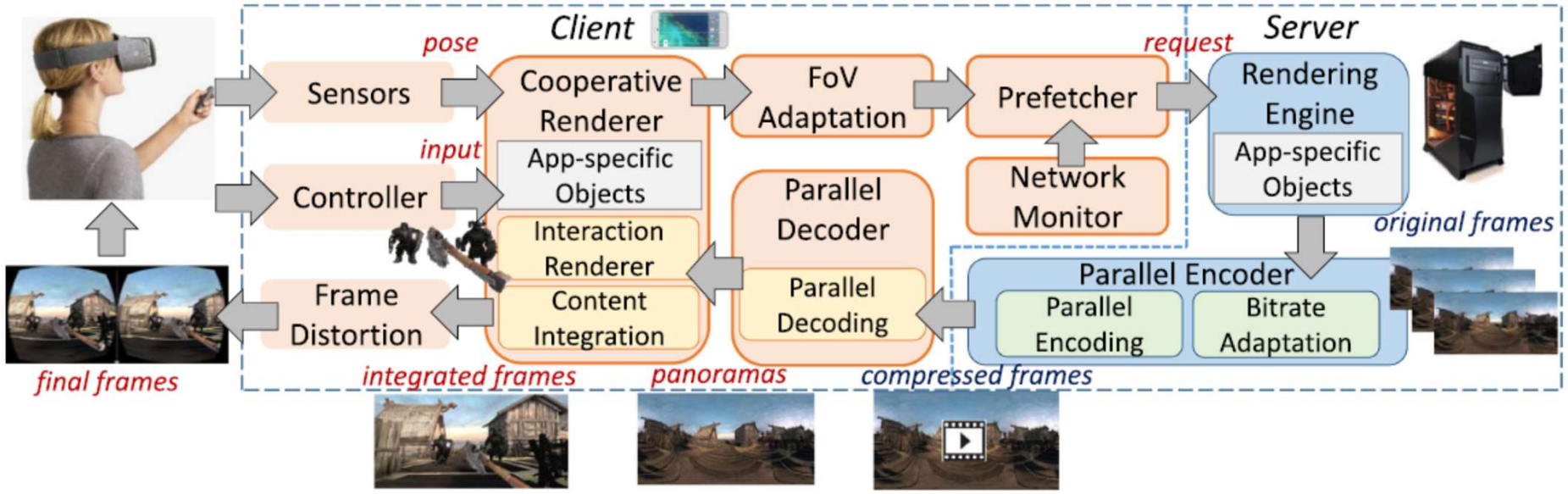
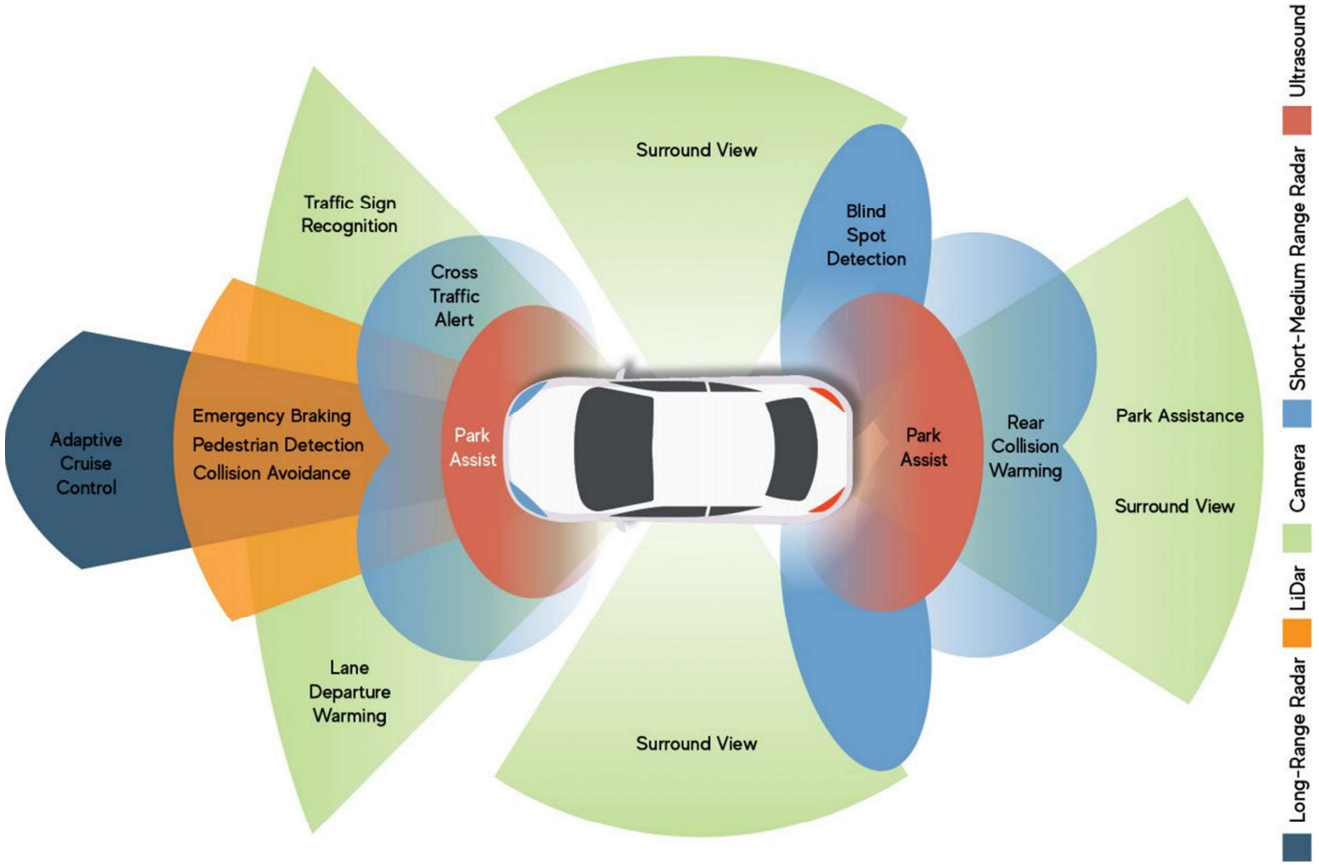


Fig. 7. FURION system architecture.

Z. Lai, Y. C. Hu, Y. Cui, L. Sun, N. Dai and H. Lee, "Furion: Engineering High-Quality Immersive Virtual Reality on Today's Mobile Devices," in *IEEE Transactions on Mobile Computing*, vol. 19, no. 7, pp. 1586-1602, 1 July 2020

Application: Autonomous driving



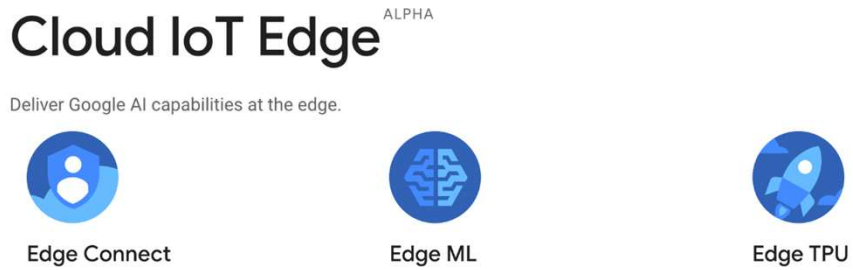
https://www.geospatialworld.net/article/hd-maps-autonomous-vehicles/

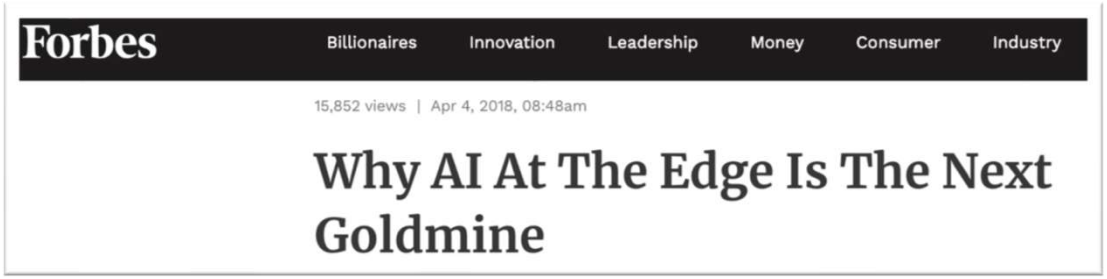
Edge-assisted autonomous driving

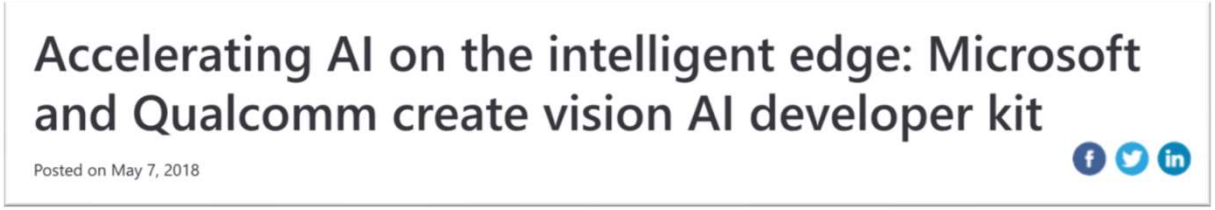
J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the Internet of vehicles," *Proc. IEEE*, Feb. 2020.

AI revolution

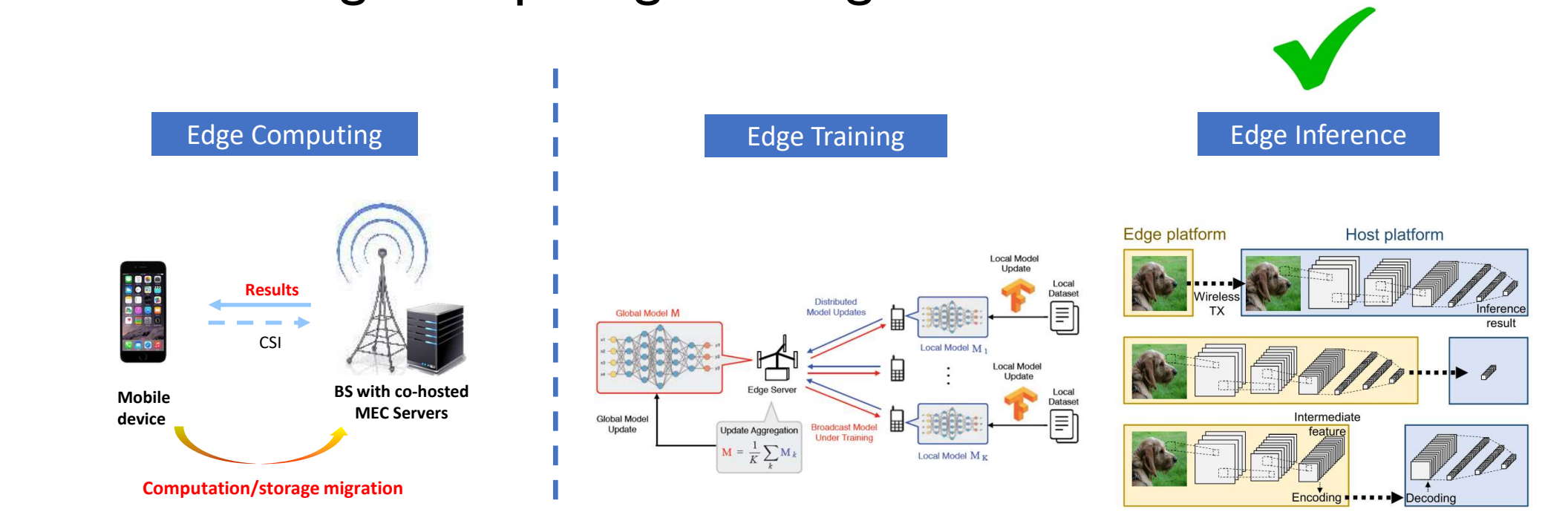
Edge AI is coming







From "edge computing" to "edge AI"



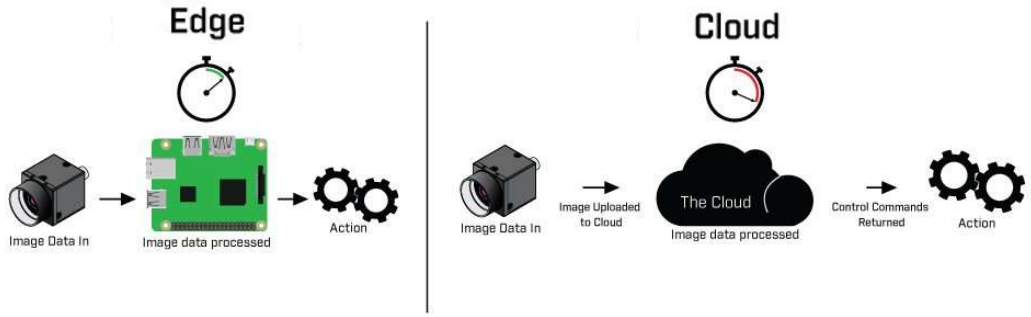
- Key problem: computing offloading
  - **Where to compute?**
  - **How to offload?**

- Key problem: model/feature transmission
  - **What to compute (locally and remotely)?**
  - **What to communicate?**

References on edge AI

- Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738-1762, Aug. 2019.
- X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan and X. Chen, "Convergence of Edge Computing and Deep Learning: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 869-904, Second quarter 2020
- G. Zhu, D. Liu, Y. Du, C. You, **J. Zhang**, and K. Huang, "Toward an intelligent edge: Wireless communication meets machine learning," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 19–25, Jan. 2020.
- Y. Shi, K. Yang, T. Jiang, **J. Zhang**, and K. B. Letaief, "Communication-efficient edge AI: Algorithms and systems," *IEEE Commun. Surveys Tuts.*, 4th Quart. 2020.
- **J. Zhang** and K. B. Letaief, "Mobile edge intelligence and computing for the Internet of vehicles," *Proc. IEEE*, vol. 108, no. 2, pp. 246–261, Feb. 2020.
- L. Wang, **J. Zhang**, J. Chuan, R. Ma, and A. Fei, "Edge intelligence for mission cognitive wireless emergency networks," *IEEE Wireless Commun. Mag.*, vol. 27, no. 4, pp. 103–109, Aug 2020.

Edge-assisted Inference



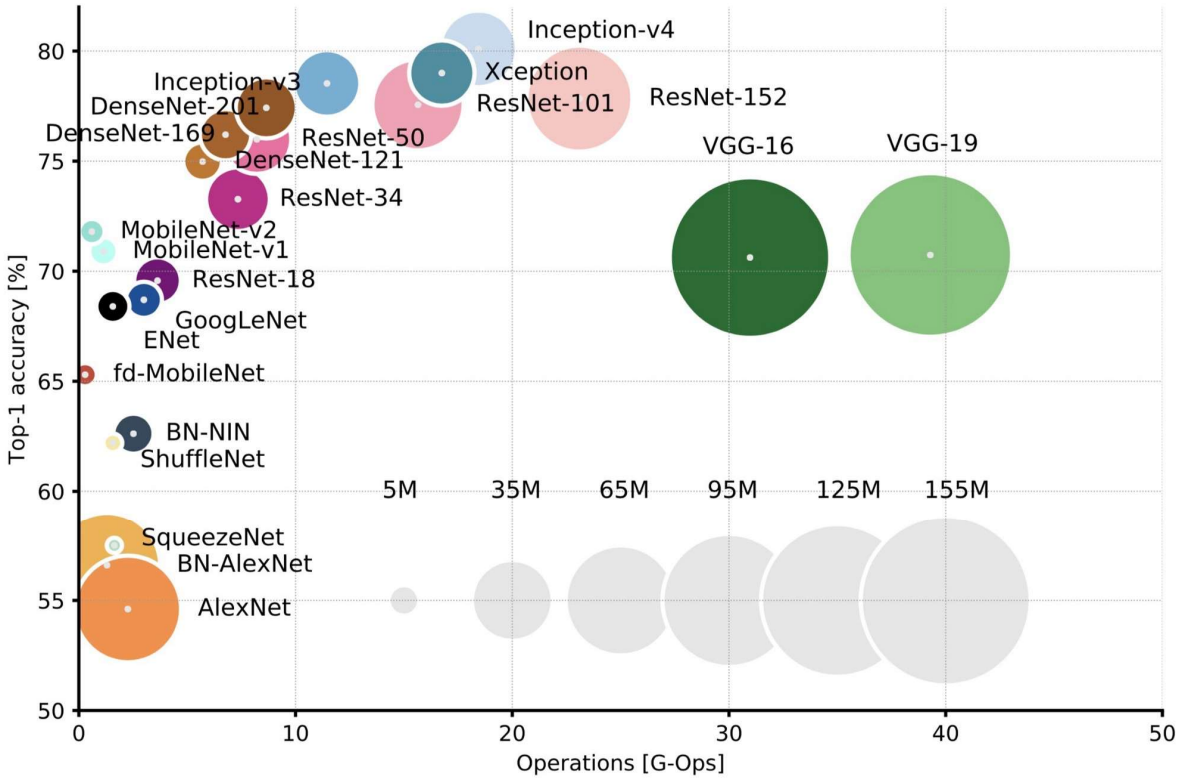
Edge Inference

- Perform low-latency AI inference at the wireless network edge.
- Problems: constrained on-device resources and limited wireless bandwidth.

Self-Driving Car

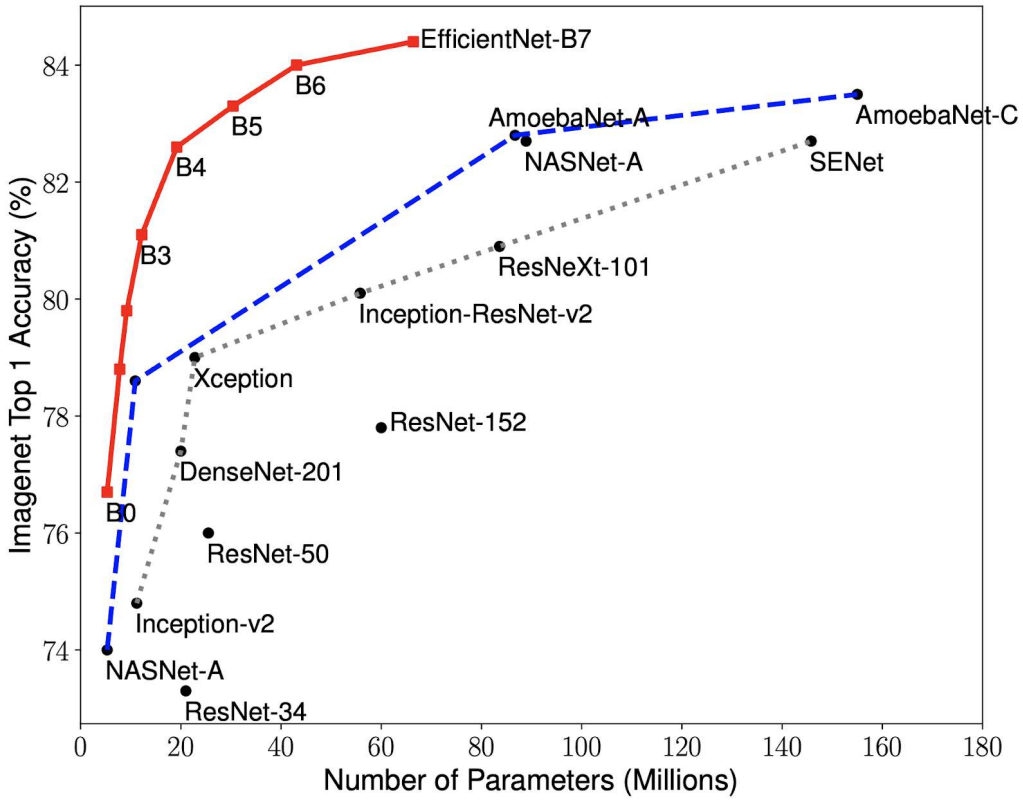
Aerial Tracking

Challenge of edge inference – LARGE size of DNN models



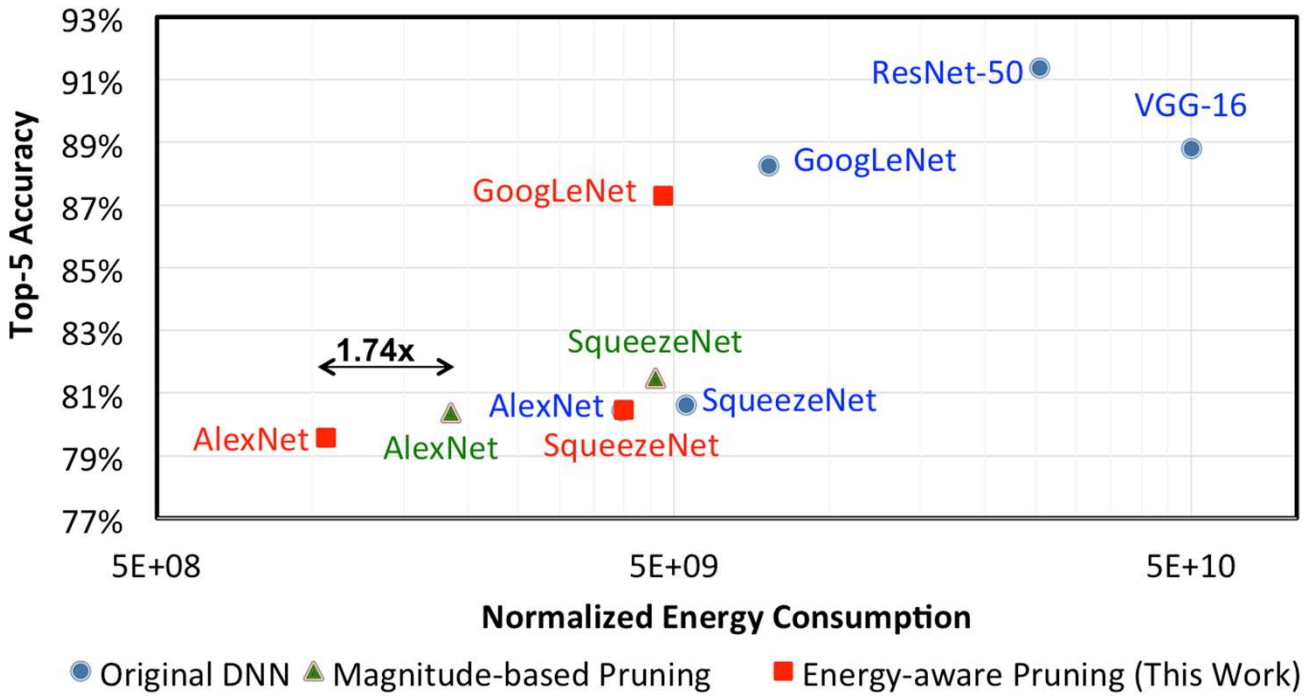
https://www.topbots.com/a-brief-history-of-neural-network-architectures/

Challenge of edge inference – LARGE size of DNN models



https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html, May 2019

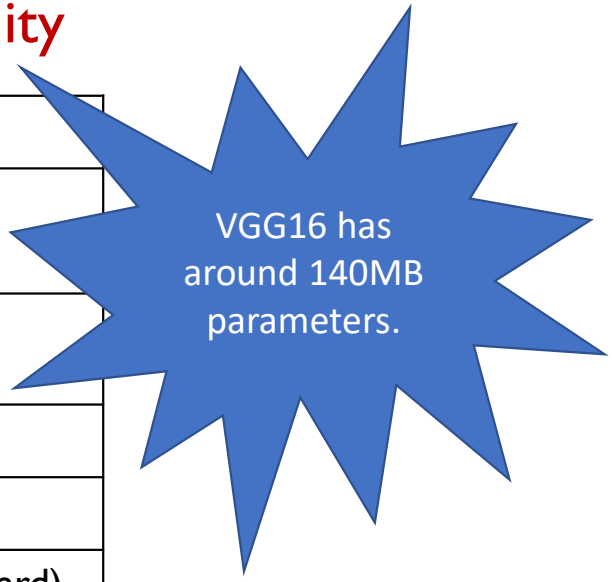
Challenge of edge inference – High energy of DNN models



Challenge of Edge Inference – Limited On-device Resources

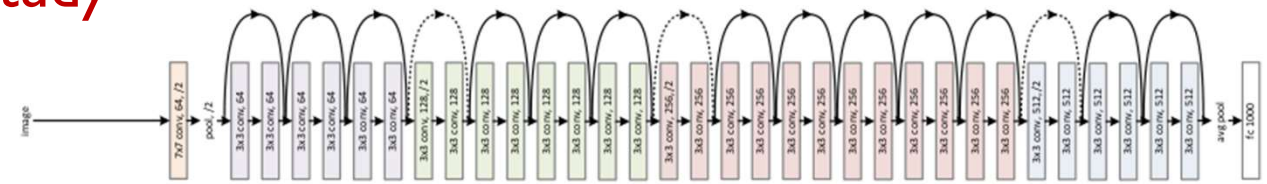
- Memory, energy, and computation ability

Devices	RAM	ROM
Arduino ATmega328	2KB	1KB
Arduino ATmega2560	8KB	4KB
MSP430 9	2KB	60KB
MSP430 19	4KB	120KB
Raspberry Pi 0	512MB	16GB (SD card)
Raspberry Pi 3	1GB	16GB (SD card)
Dell Workstation R3930 (Server)	8GB	1TB



Challenge of Edge Inference – Limited On-device Resources

- A case study



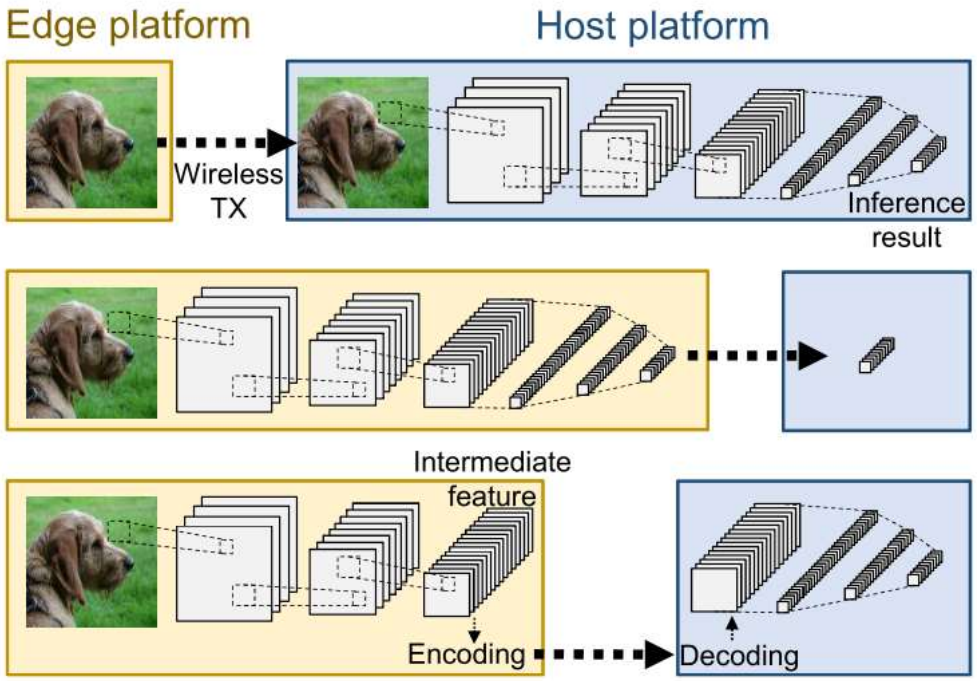
- ResNet-152: 30 GFLOP/image
- Streaming video
  - 15-30 frame/sec x 2-5 image/frame = 75 image/sec

DGX-1: 85 TFLOPS → 2833 image/sec
DGX-1 can handle 38 video streams

RPi3: 0.5 GFLOPS → 1 image/minute
RPi3 is 4500x too slow to handle one stream

Challenge:
How to run powerful deep learning models at resource-constrained devices?

Three different approaches of edge inference



Server-based method

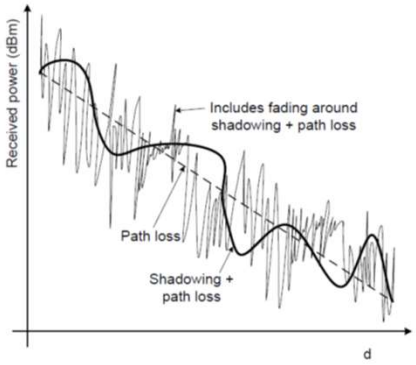
On-device processing

Device-edge co-inference

Solution I: Offloading to an edge server

- High communication overhead
  - 256×256 camera with 24 FPS creates around 1.5 MB data per second.
  - 4k camera with 30 FPS creates around 230MB data per second.

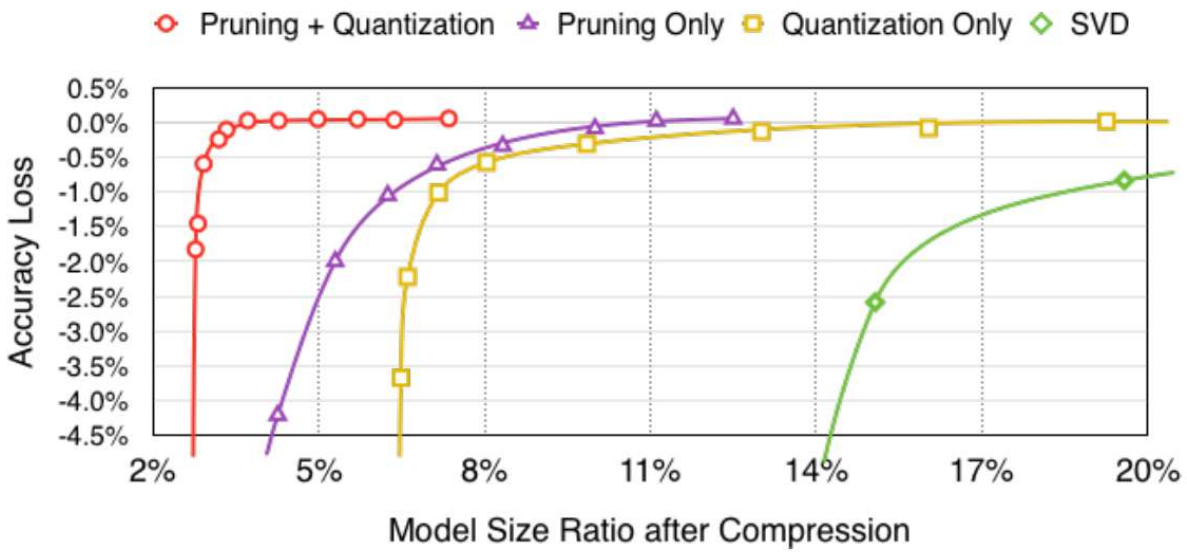
- Unstable communication link
  - Wi-Fi: 18 Mbps
  - 4G: 5.85 Mbps
  - 3G: 1.1 Mbps



- Data privacy

Solution II: On-device inference with model compression

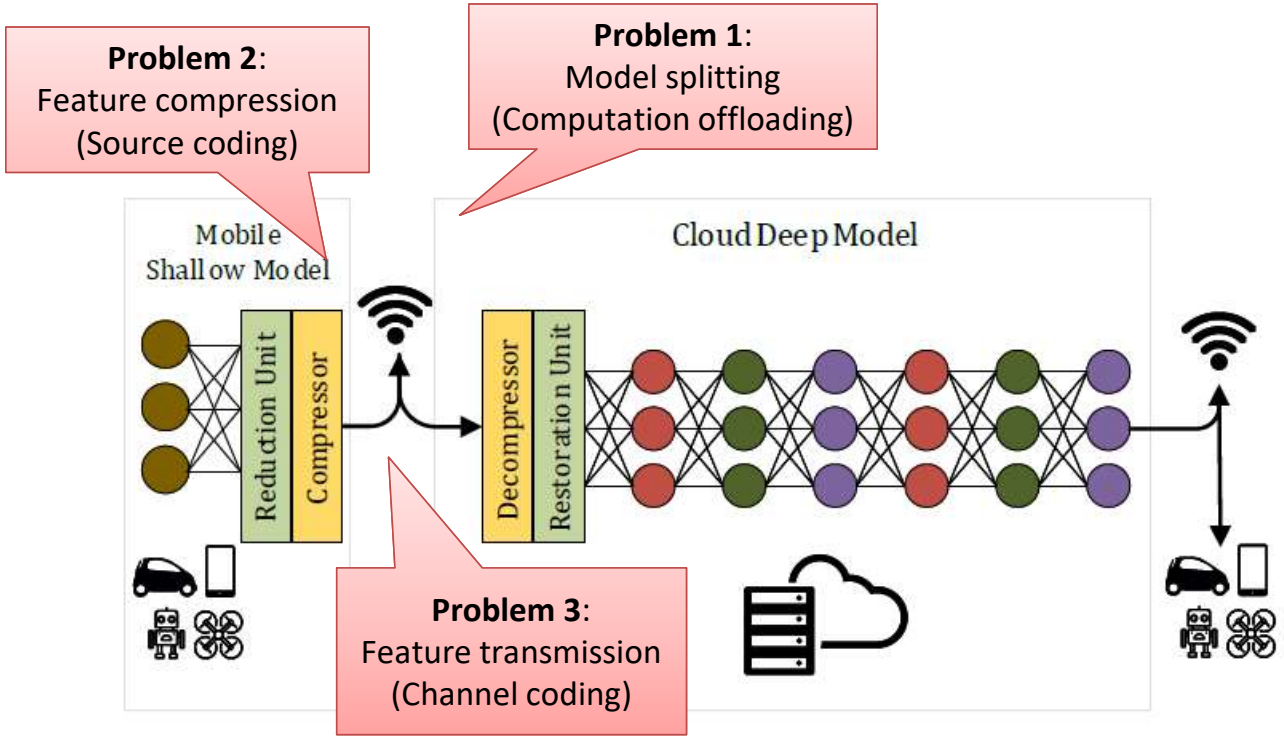
- High compression ratio → large performance loss



Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." arXiv preprint arXiv:1510.00149 (2015).

Solution III: Device-edge co-inference

- Provide better communication vs. on-device computation tradeoff



Model splitting: An example

- Neurosurgeon

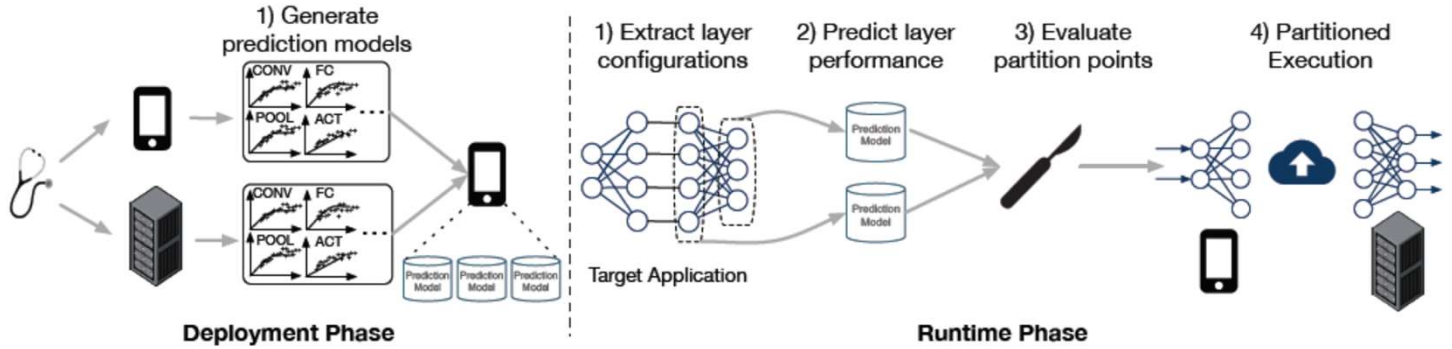
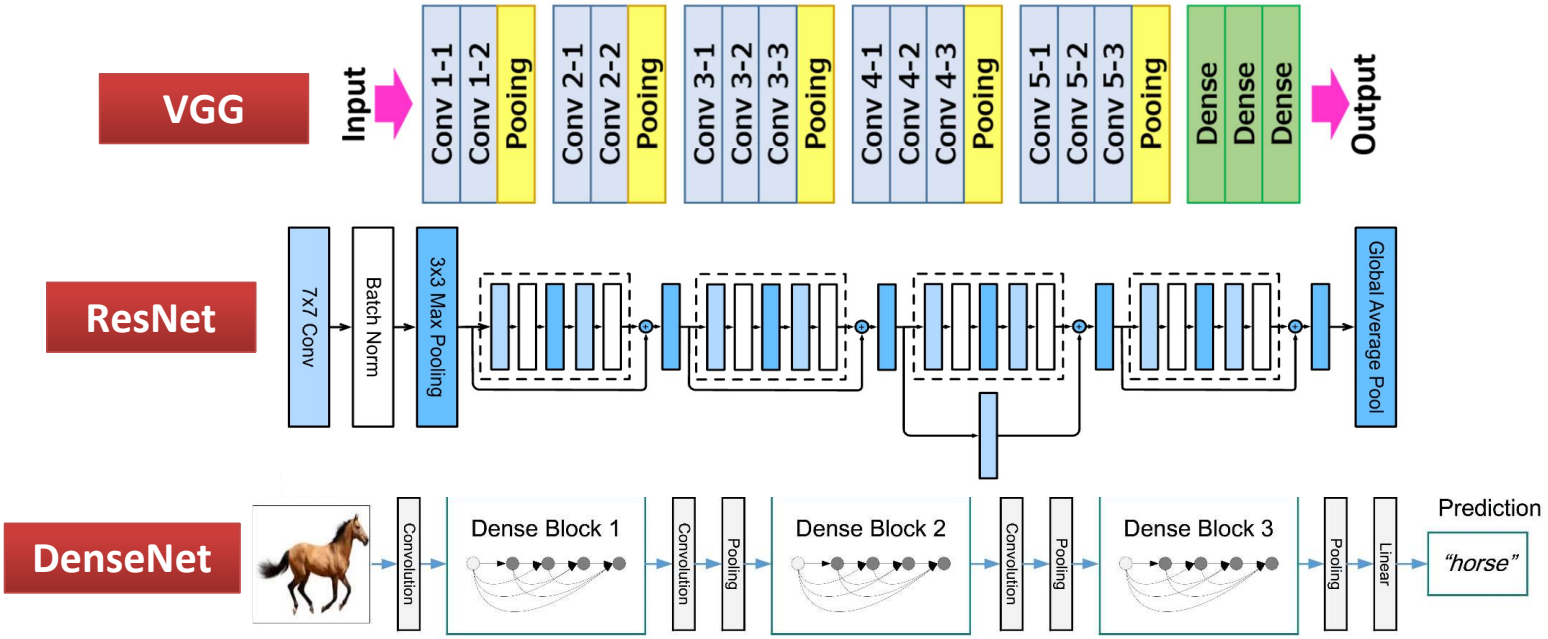


Figure 10: Overview of Neurosurgeon. At deployment, Neurosurgeon generates prediction models for each layer type. During runtime, Neurosurgeon predicts each layer's latency/energy cost based on the layer's type and configuration, and selects the best partition point based on various dynamic factors.

Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang. "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," *ACM SIGPLAN Notices*, 52(4):615–629, 2017.

Model splitting: Where to split?

- Not all layer are suitable to split
  - Depending on the neural network architecture



Model splitting: Data amplification

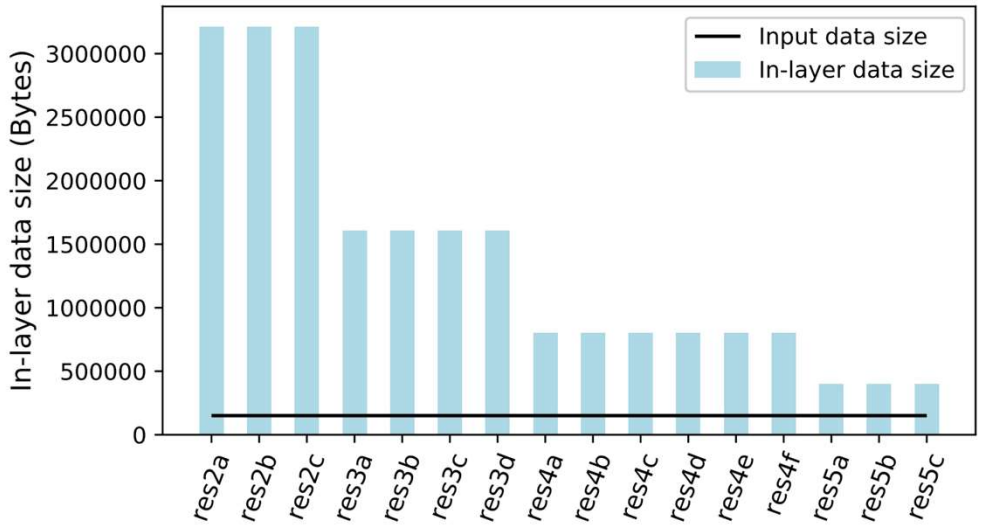


Fig. 2: Data size at each decoupling point in ResNet, showing that the input data is amplified in in-layer feature maps.

Li H, Hu C, Jiang J, et al. "Jalad: Joint accuracy-and latency-aware deep structure decoupling for edge-cloud execution", 2018 IEEE 24th International Conference on Parallel and Distributed Systems. IEEE, 2018: 671-678.

Model splitting as a computation offloading problem

- Model splitting can be regarded as a computation offloading problem with dependency among sub-tasks.

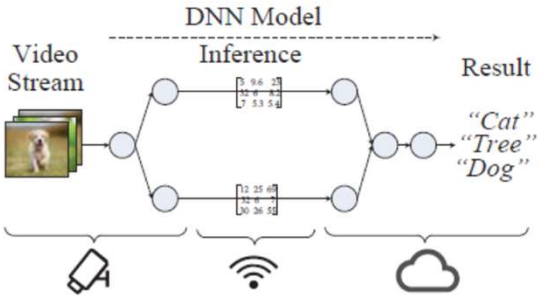


Fig. 4: A 7-layer DNN model classifies frames of video.

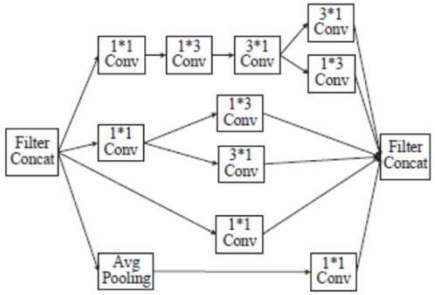


Fig. 5: The inception v4 network represented in layer form.

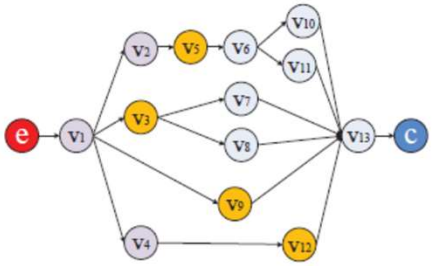
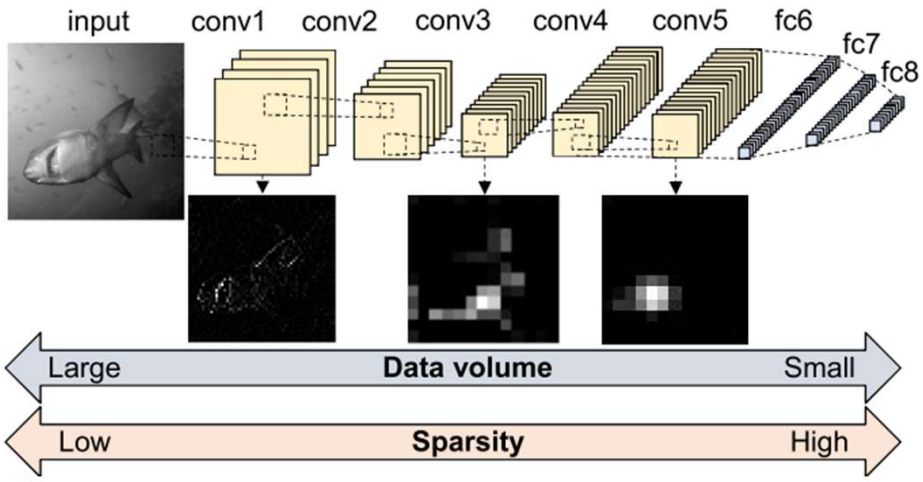


Fig. 6: Graph representation of inception v4 network.

C. Hu, W. Bao, D. Wang, and F. Liu, "Dynamic Adaptive DNN Surgery for Inference Acceleration on the Edge," in *Proc. IEEE INFOCOM 2019*, 2019, pp. 1423-1431.

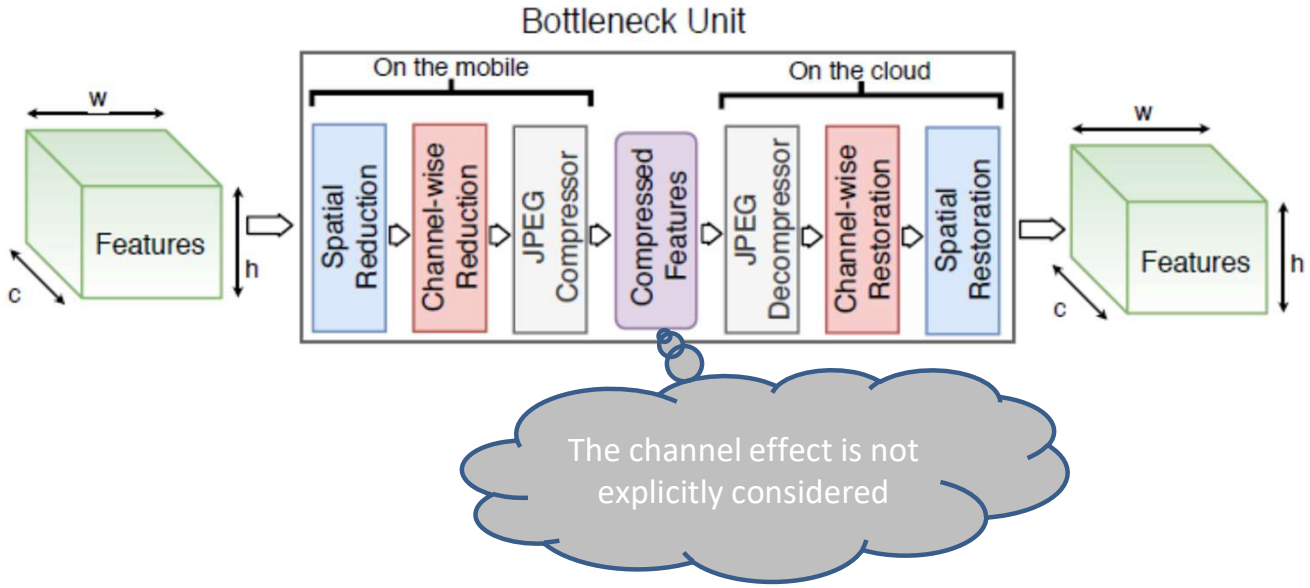
Feature compression

- The intermediate feature is sparse
  - Lossless compression (ZIP, PNG, GIF)
- The neural network (NN) has fault-tolerant property
  - Lossy compression (JPEG, DCT, Quantification)



Feature compression: An example

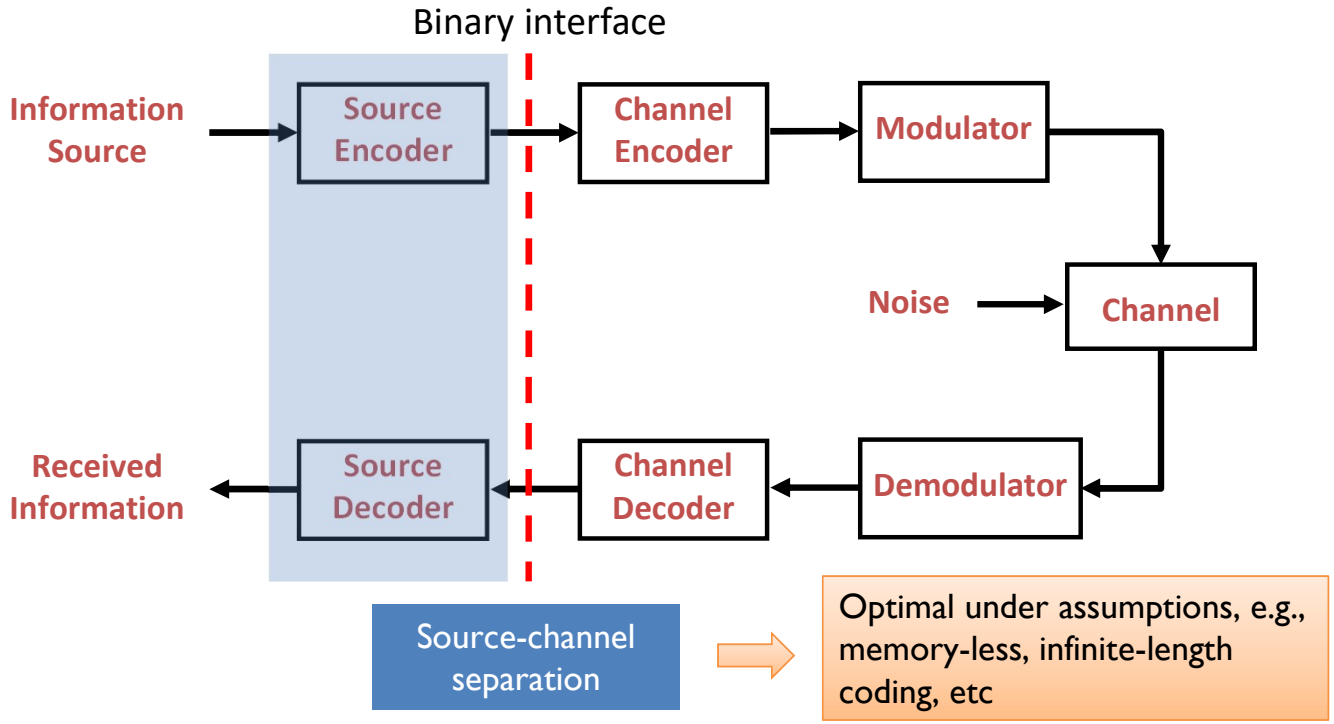
- BottleNet



A. E. Eshratifar, A. Esmaili, and M. Pedram, "Bottlenet: A deep learning architecture for intelligent mobile cloud computing services," arXiv preprint arXiv:1902.01000, 2019.

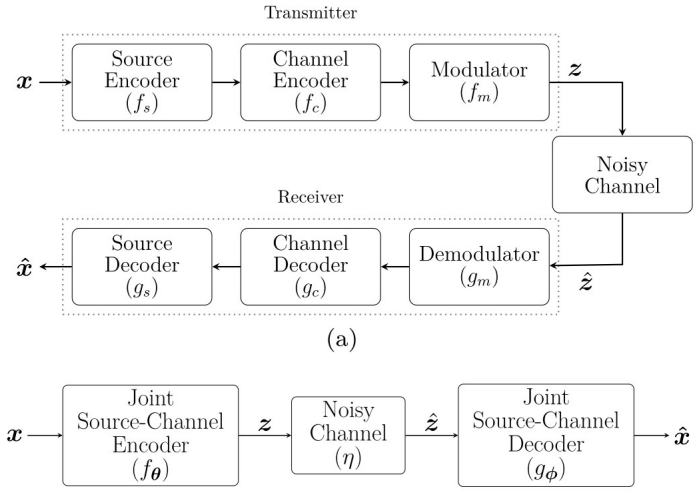
Feature transmission: Traditional communication

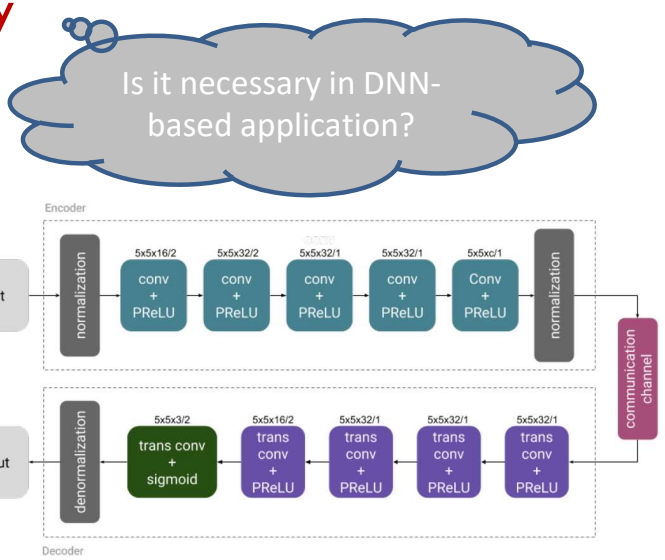
- Separate design of source and channel coding



Feature transmission: Joint source-channel coding

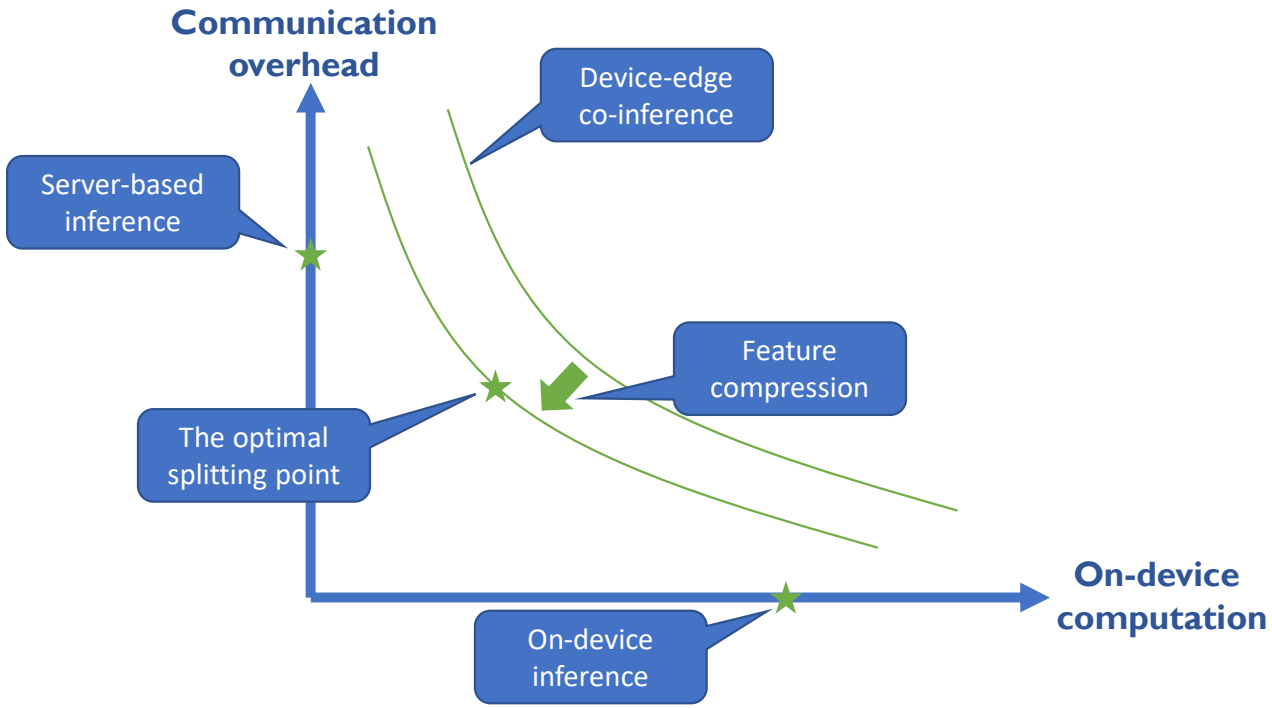
- Joint source-channel coding
  - Use learning-based method for effective design
  - The object is to transmit the data reliably





E. Bourtsoulatze, D. B. Kurka, and D. Gndz, "Deep joint source-channel coding for wireless image transmission," in ICASSP 2019, May 2019, pp. 4774-4778.

The Big Picture



Wishful thinking

- Pseudo formulation

Minimize Inference Latency (= computation + communication)

Subject to On-device computation cost < c
Inference error < t

How to make it more tractable?

From "edge computing" to "edge AI"

- **Essential idea**:
  - To provide DNN-based high-performance inference (target) on resource-constrained devices (constraint) by jointly designing local computation and offloading/communication (methodology)

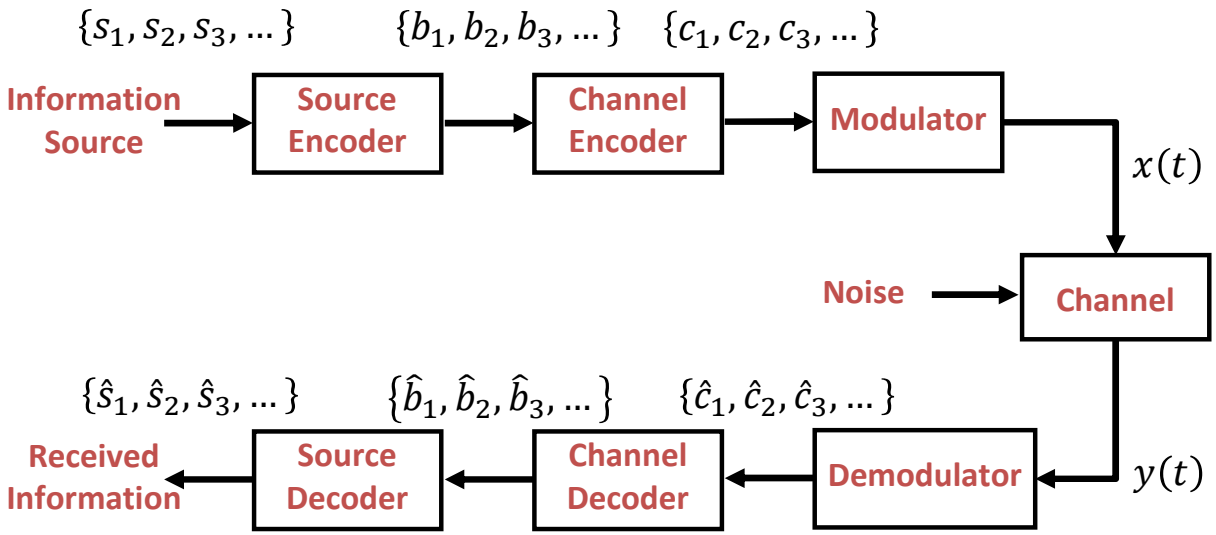
- Key considerations
  - **Communication**: informative and concise representation
  - **Computation**: low-complexity feature extractor and encoder

Interplay between communication, computation, and machine learning

A paradigm shift

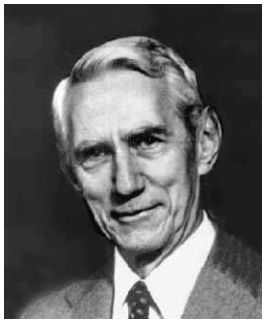
From data-oriented to task-oriented communication

Digital communications 101



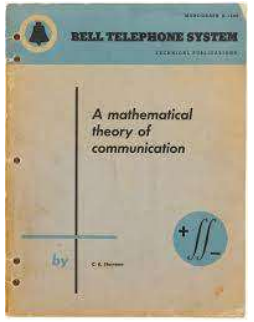
Digital communications 101

- Shannon's wisdom



"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point."

The "sematic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one selected from a set of possible messages."



C. E. Shannon. *A mathematical theory of communication*. Bell System Technical Journal, 27:379–423, 623–656, 1948.

Task-oriented communication

A broader view: Three levels of communications

Shannon's information theory

Level A
The technical problem

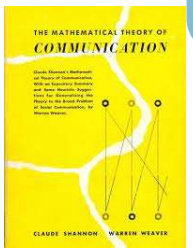
- How *accurately* can the symbols of communication be transmitted?

Level B
The semantic problem

- How *precisely* do the transmitted symbols convey the desired meaning?

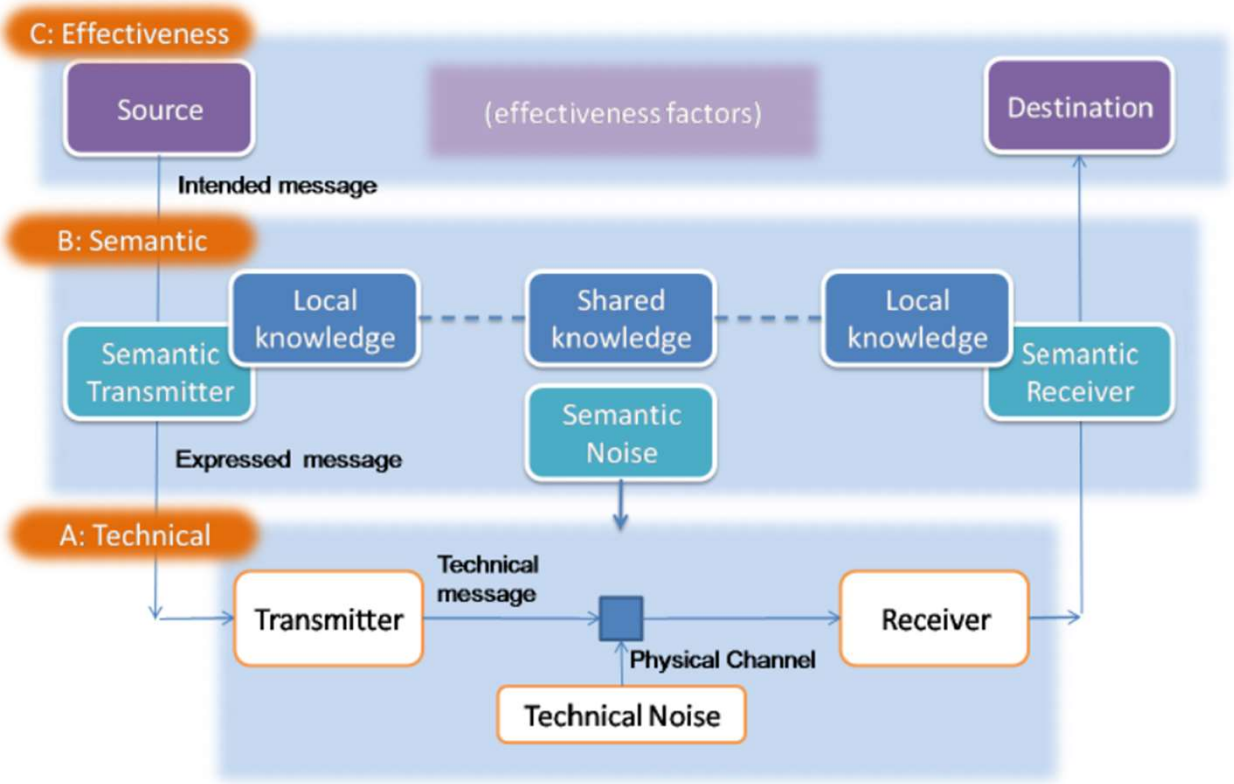
Level C
The effectiveness problem

- How *effectively* does the received meaning affect conduct in the desired way?



W. Weaver. Recent contributions to the mathematical theory of communication. In C. E. Shannon and W. Weaver, editors, *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, 1949.

Three levels of communications



A Proposal for the

DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE

June 17 - Aug. 16

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

The following are some aspects of the artificial intelligence problem:

1) Automatic Computers

If a machine can do a job, then an automatic calculator can be programmed to simulate the machine. The speeds and memory capacities of present computers may be insufficient to simulate many of the higher functions of the human brain, but the major obstacle is not lack of machine capacity, but our inability to write programs taking full advantage of what we have.

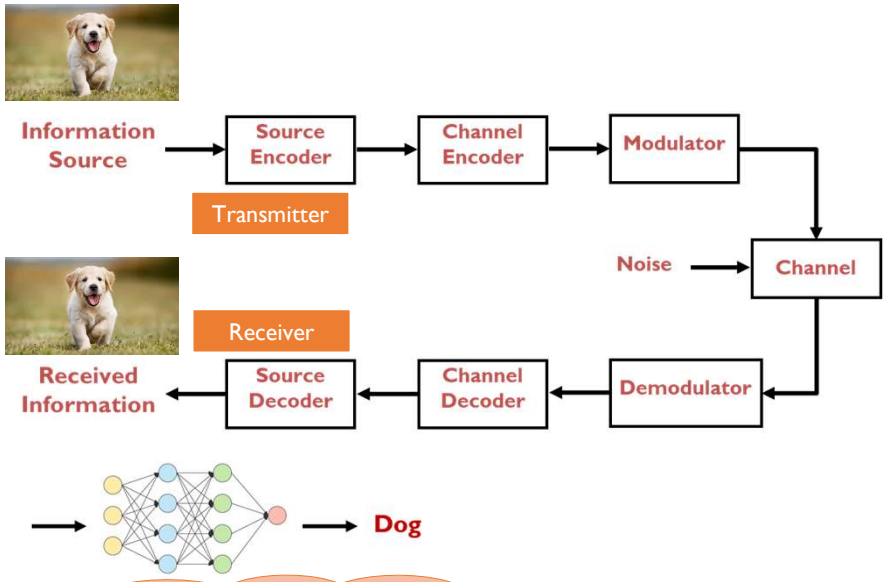
2) How Can a Computer be Programmed to Use a Language

It may be speculated that a large part of human thought consists of manipulating words according to rules of reasoning

1956 Dartmouth Conference

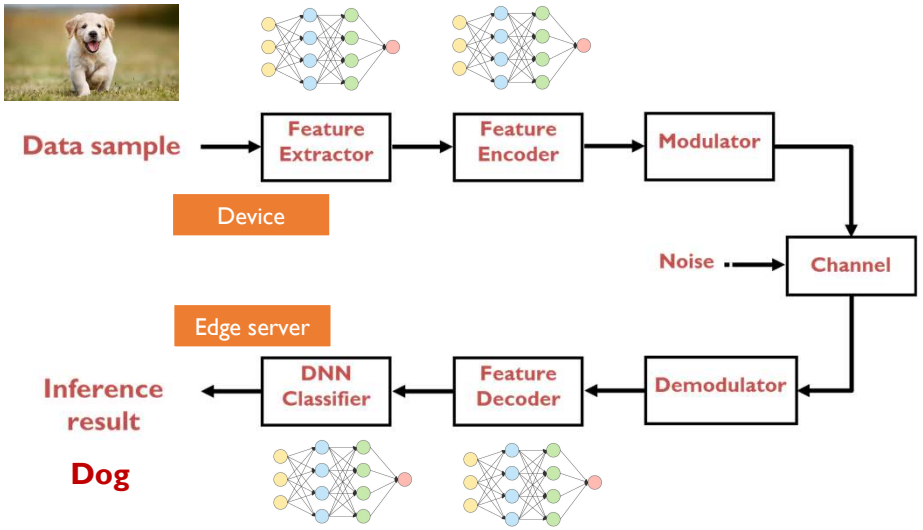
Data-oriented vs. task-oriented communication

Data-oriented Communication



This is how current security camera systems and virtual assistants (e.g., Siri) work!

Task-oriented Communication



Challenges and potential solutions of task-oriented communication

- Challenges

High-dimensional data | **Limited onboard resources** | **Unknown model**

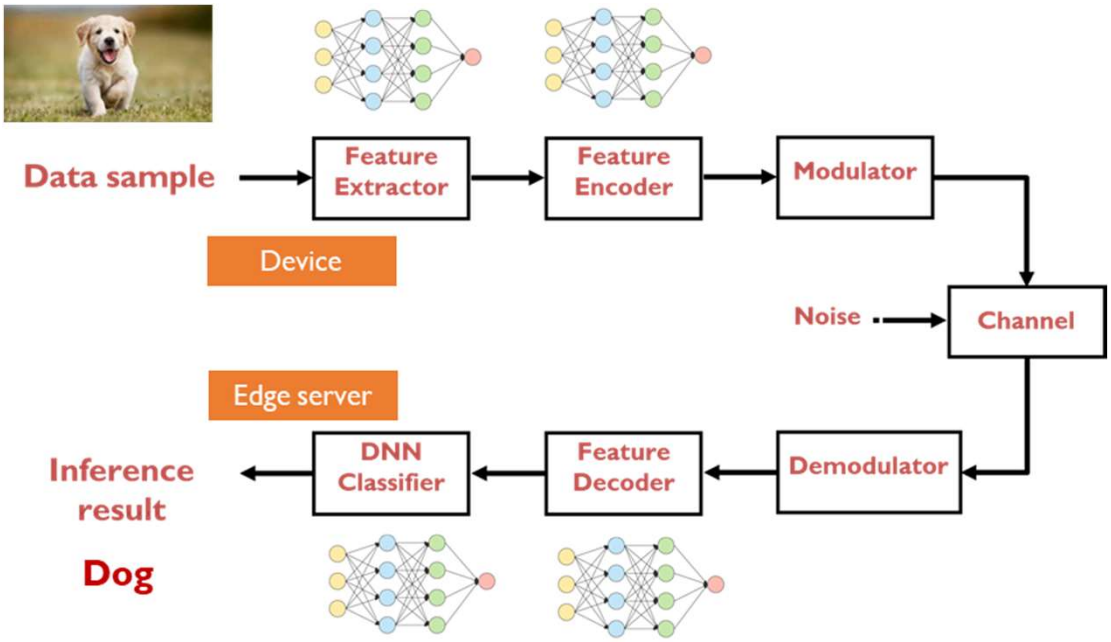
- Key requirements

Small size of intermediate feature | **Limited on-device computation** | **Task-oriented, NOT data-oriented**

- Solutions

Information bottleneck | **Neural architecture optimization** | **End-to-end design via deep learning**

Task-oriented communication



Key design tools
(available only recently)

- End-to-end design via deep learning
- Feature encoding via information bottleneck
- Neural architecture optimization for on-device network

Related endeavors in rethinking communication problem

Deep Learning Enabled Semantic Communication Systems

Huiqiang Xie, Zhijin Qin, *Member, IEEE,* Geoffrey Ye Li, *Fellow, IEEE,* and Biing-Hwang Juang *Life Fellow, IEEE*

Semantic Communications in Networked Systems

Elif Uysal, Onur Kaya, Anthony Ephremides, James Gross, Marian Codreanu, Petar Popovski, Mohamad Assaad, Gianluigi Liva, Andrea Munari, Touraj Soleymani, Beatriz Soret and Karl Henrik Johansson*

6G Networks: Beyond Shannon Towards Semantic and Goal-Oriented Communications

Emilio Calvanese Strinati[a,1,*], Sergio Barbarossa[b]

Task-oriented Communication System Design in Cyber-Physical Systems: A Survey on Theory and Applications

Arsham Mostaani, *Student Member, IEEE*, Thang X. Vu, *Member, IEEE*, and Symeon Chatzinotas, *Senior Member, IEEE*

End-to-end design: BottleNet++, Branchy-GNN

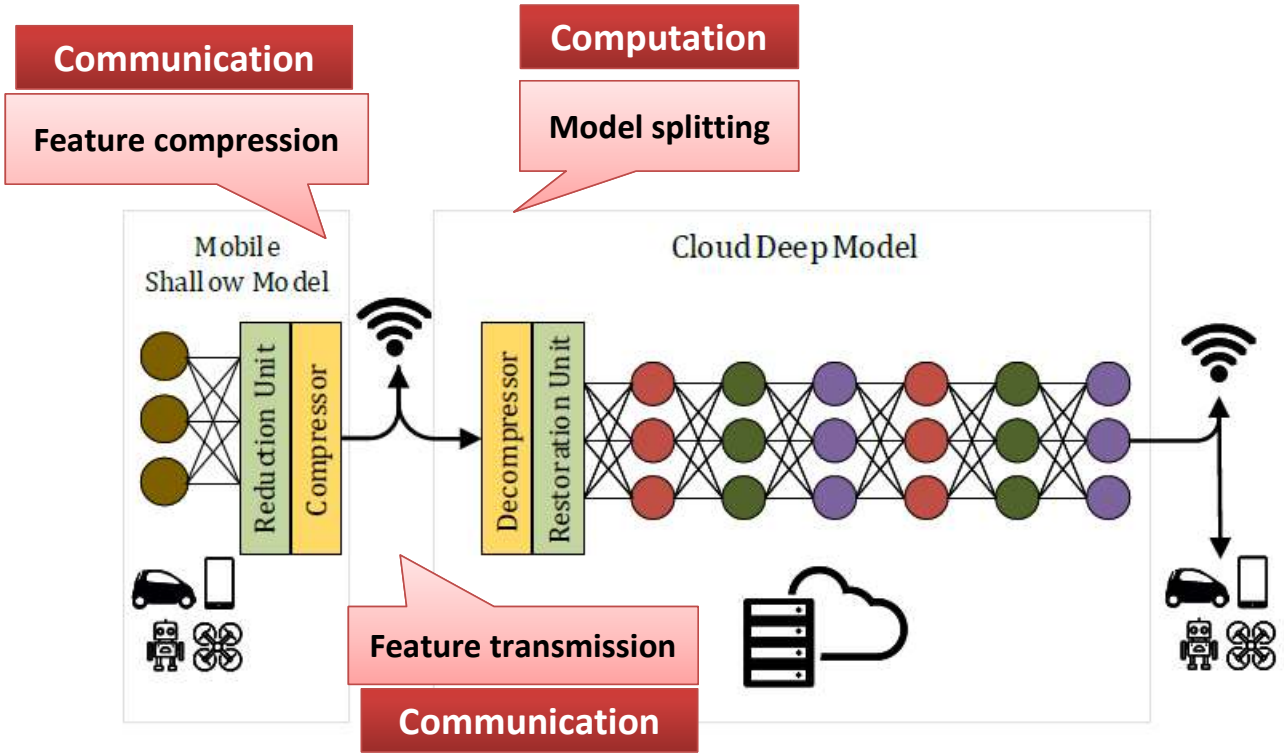
J. Shao, **J. Zhang**, "BottleNet++: An end-to-end approach for feature compression in device-edge co-inference systems," *IEEE Int. Conf. Commun. (ICC) Workshop* 2020.

J. Shao, H. Zhang, Y, Mao, and **J. Zhang**, "Branchy-GNN: a device-edge co-inference framework for efficient point cloud processing," *ICASSP 2021*.

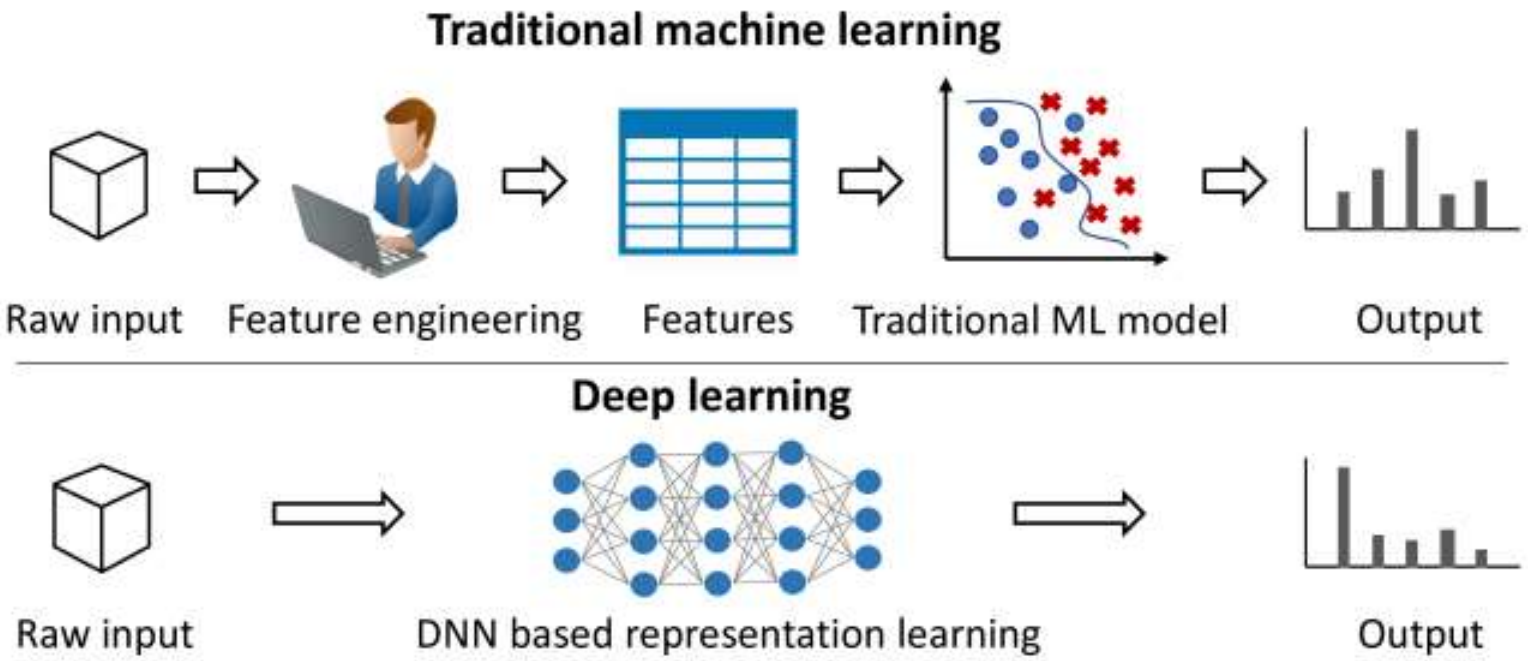
End-to-end design

Existing studies

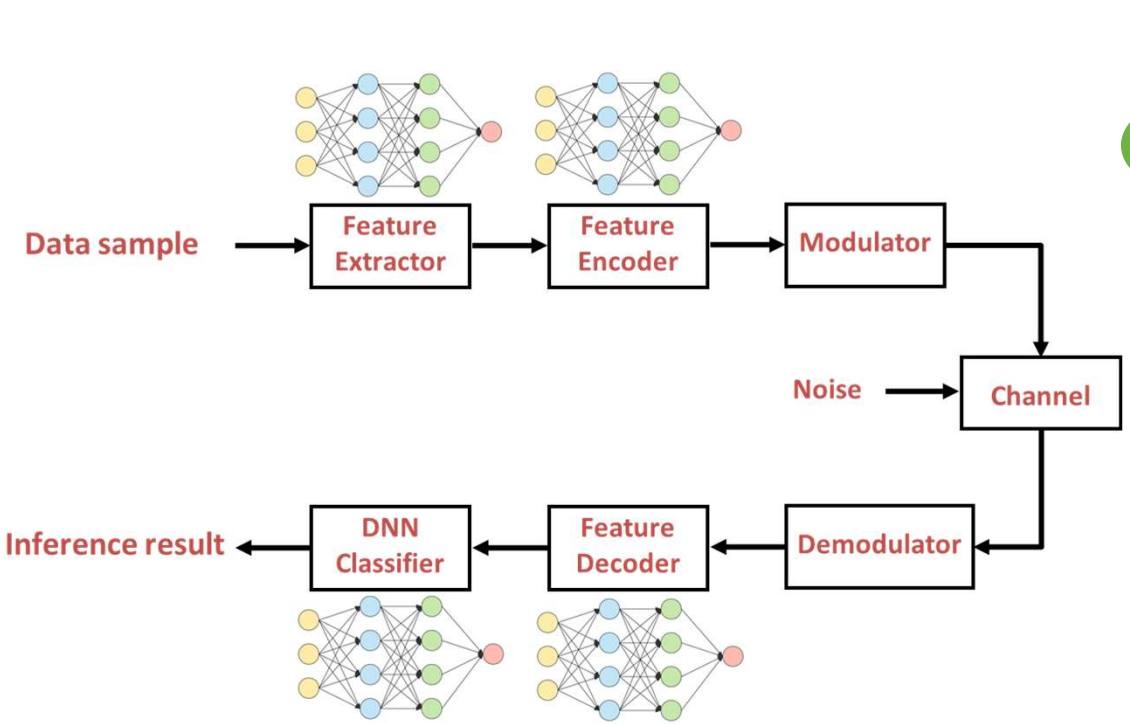
- Separately design model splitting, feature compression, and transmission.



End-to-end design via deep learning



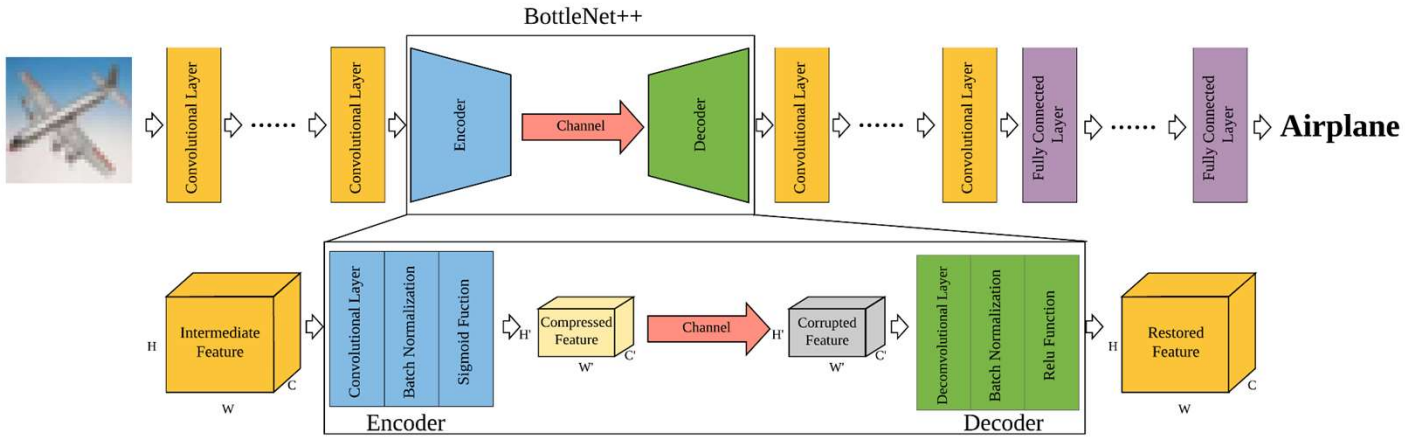
End-to-end design via deep learning



Stochastic gradient descent (SGD)
or
Grad student descent (GSD)

Case study 1: BottleNet++ for image classification

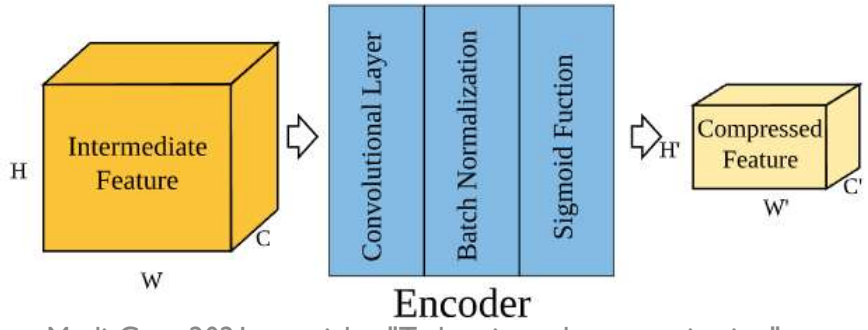
- **BottleNet++**
  - **End-to-End** architecture.
  - The encoder and decoder are a pair of complementary CNNs.
  - The wireless channel is modeled as a non-trainable layer (a transfer function).
  - **Key insight**: Accurate data recovery after transmission is not needed.



Joint source-channel coding

Proposed Architecture

- **Encoder**
  - Plays the role of feature compression and transmission (joint source-channel coding).
  - Key components:
    - A convolutional layer: to reduce the dimension of intermediate feature tensor.
    - A batch normalization layer and a Sigmoid function: to introduce non-linearity, which improve the NN compression capability.
  - The channel condition will be an additional input, e.g., variance of the noise or bit erasure rate, for encoder.

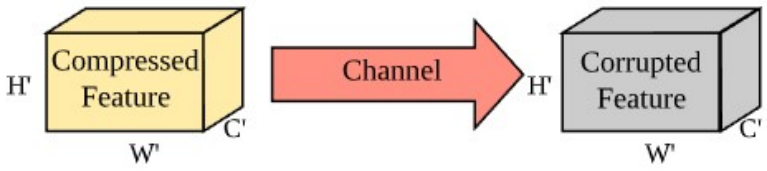


Proposed Architecture

- **Channel**
  - The wireless channel is modeled as a non-trainable layer.
  - AWGN channel: $f(x) = x + n \quad n \sim \mathcal{N}(0, \sigma^2)$
  - Binary Erasure Channel (BEC):

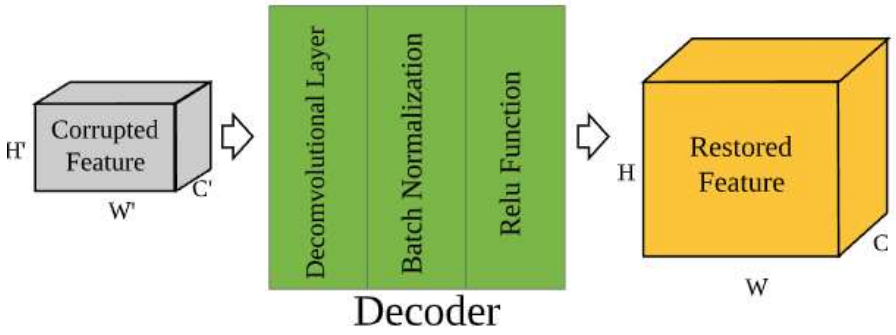
$$x \overset{Quantify}{\longrightarrow} \hat{x} \overset{BEC}{\longrightarrow} \tilde{x} \overset{Recovery}{\longrightarrow} f(x)$$

$$13.02 \overset{Quantify}{\longrightarrow} 1101 \overset{BEC}{\longrightarrow} 110? \overset{Recovery}{\longrightarrow} \text{Average}(1100, 1101) \rightarrow 12.5$$
$$13.02 \overset{Quantify}{\longrightarrow} 1101 \overset{BEC}{\longrightarrow} 11?1 \overset{Recovery}{\longrightarrow} \text{Average}(1101, 1111) \rightarrow 14$$
$$13.02 \overset{Quantify}{\longrightarrow} 1101 \overset{BEC}{\longrightarrow} 11?? \overset{Recovery}{\longrightarrow} \text{Average}(1100, 1101, 1110, 1111) \rightarrow 13.5$$



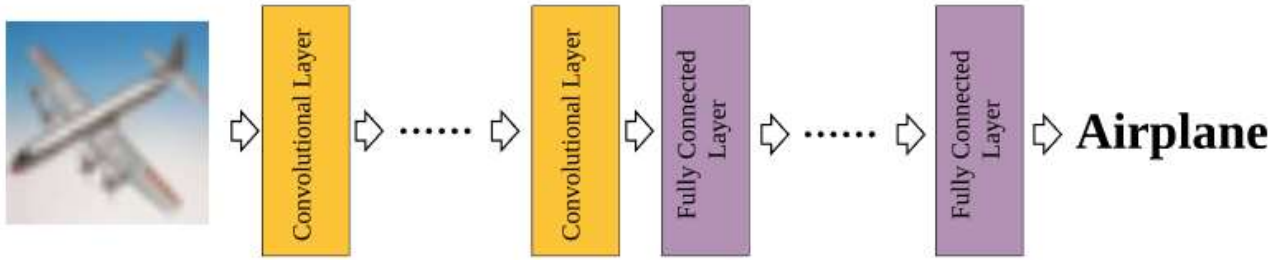
Proposed Architecture

- **Decoder**
  - Plays the role of joint source-channel decoding.
  - Key components:
    - A convolutional layer: to restore the corrupted feature.
    - A batch normalization layer and a Sigmoid function: to introduce non-linearity.



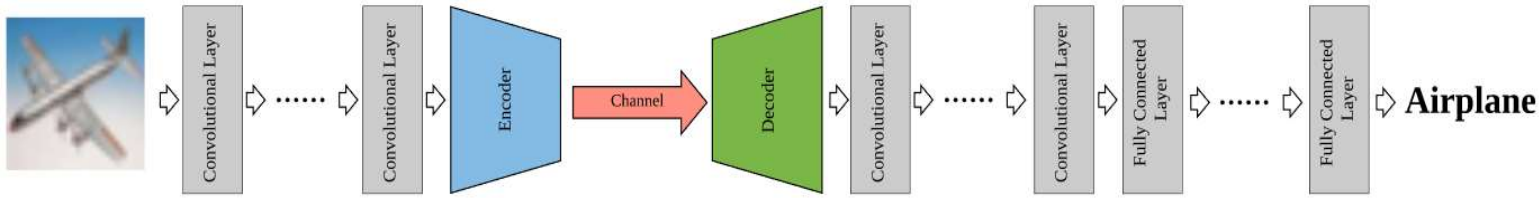
Proposed Architecture

- **Three-step Training**
  - First Step: train the DNN to reach the desired accuracy, find the model splitting point.



Proposed Architecture

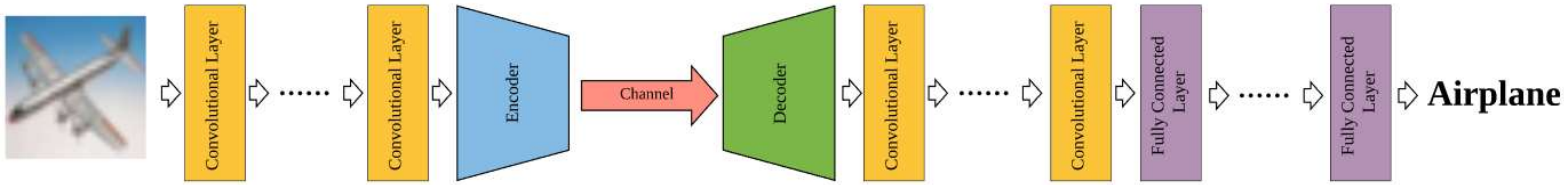
- **Three-step Training**
  - Second Step: train the encoder and decoder, fixing other layers.



End-to-end design

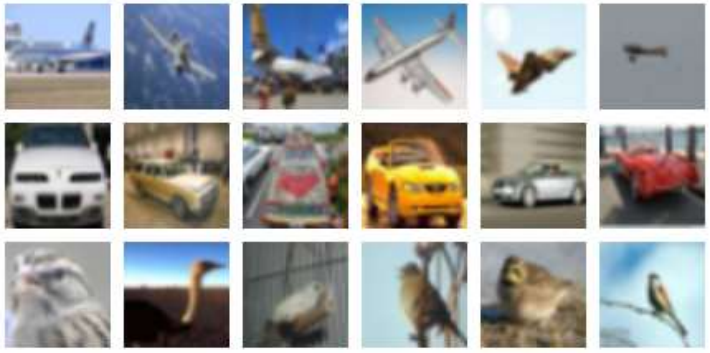
Proposed Architecture

- **Three-step Training**
  - Third Step: fine-tune all layers.

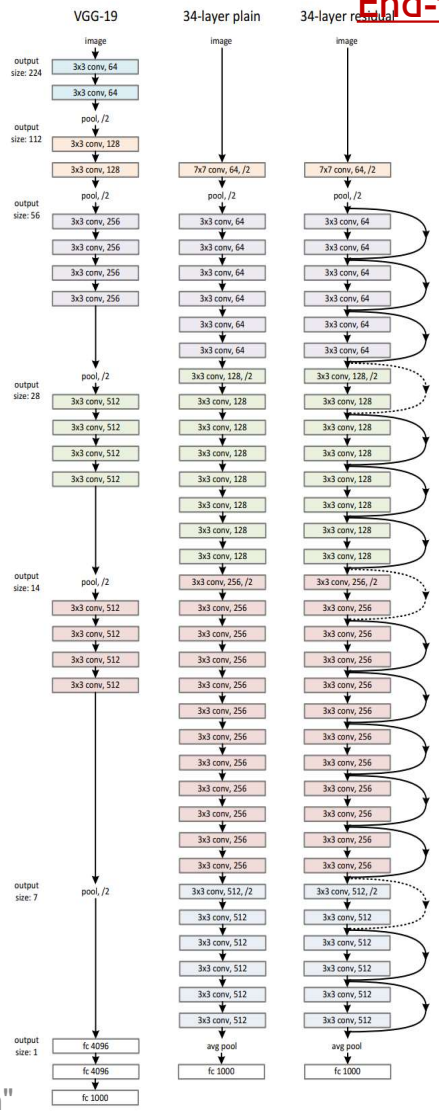


Experiment

- Experiment setup
  - DNN models: VGG16 and ResNet50.
  - Wireless Channel: the AWGN channel and Binary Erasure Channel (BEC).
  - Dataset: CIFAR-100.
  - Task: Image classification.



End-to-end design

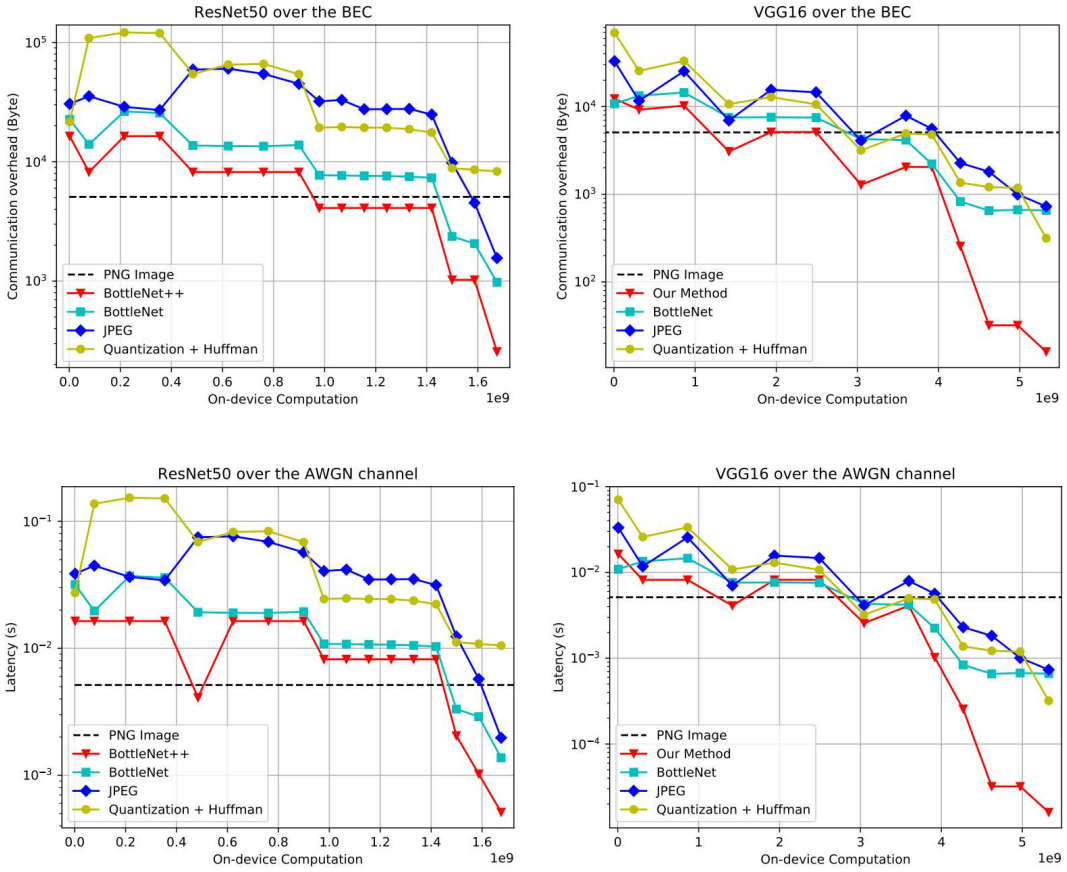


Experiment

- Compression capability comparison
  - Other intermediate feature compression methods.
    1. Quantize the intermediate feature and the use the Huffman code to compress it. (Quantification + Huffman)
    2. Use JPEG algorithm to compress the intermediate feature. (JPEG)
    3. Compress the feature using learning based method and JPEG algorithm (BottleNet)
  - Evaluation criterion
    - Communication overhead (transmitted data size/transmitted latency)
    - On-device computation (approximated by float-point multiplication in convolutional layer)
    - The accuracy degradation < 2%.
  - Definition of peak signal-to-noise ratio (PSNR) in AWGN channel.

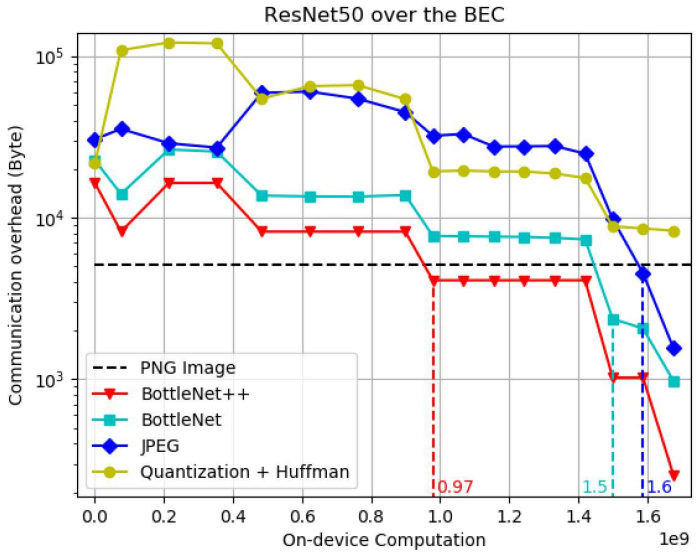
$$\mathrm{PSNR} = 10 \log_{10} \frac{1}{\sigma^2} \quad (\mathrm{dB})$$

Experiment Result



Experiment

- Split the network when the communication overhead less than transmitting PNG image.

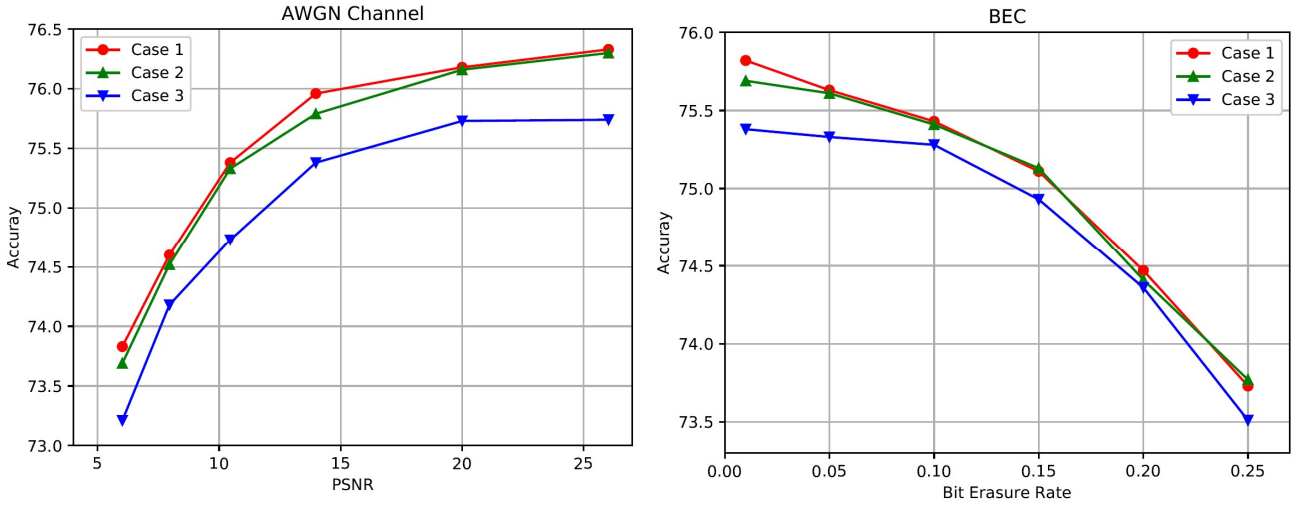


On-device Computation	BottleNet++ (ours)	BottleNet	JPEG	Quan.+Huffman
ResNet50 (BEC)	$\mathbf{9.7 \times 10^8}$	1.5×10^9	1.6×10^9	——
ResNet50 (AWGN)	$\mathbf{4.8 \times 10^8}$	1.4×10^9	1.6×10^9	——
VGG16 (BEC)	$\mathbf{1.4 \times 10^9}$	3.0×10^9	3.0×10^9	3.0×10^9
VGG16 (AWGN)	$\mathbf{1.4 \times 10^9}$	3.0×10^9	3.0×10^9	3.0×10^9

2x~3x

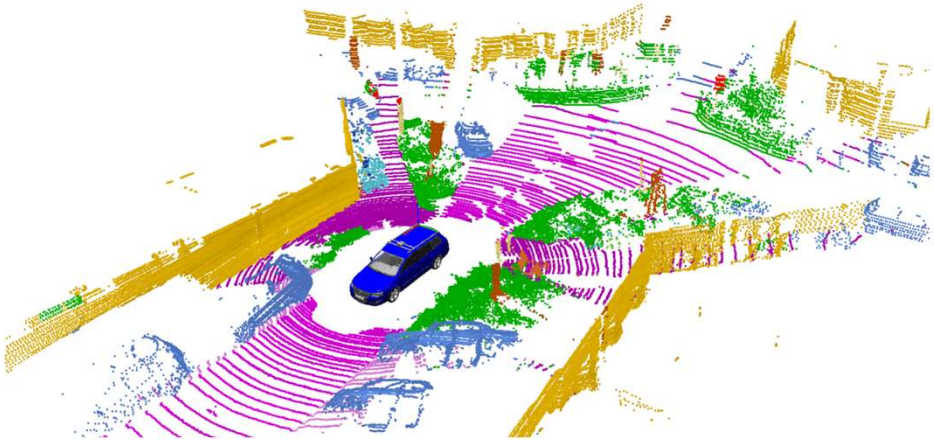
Experiment Result

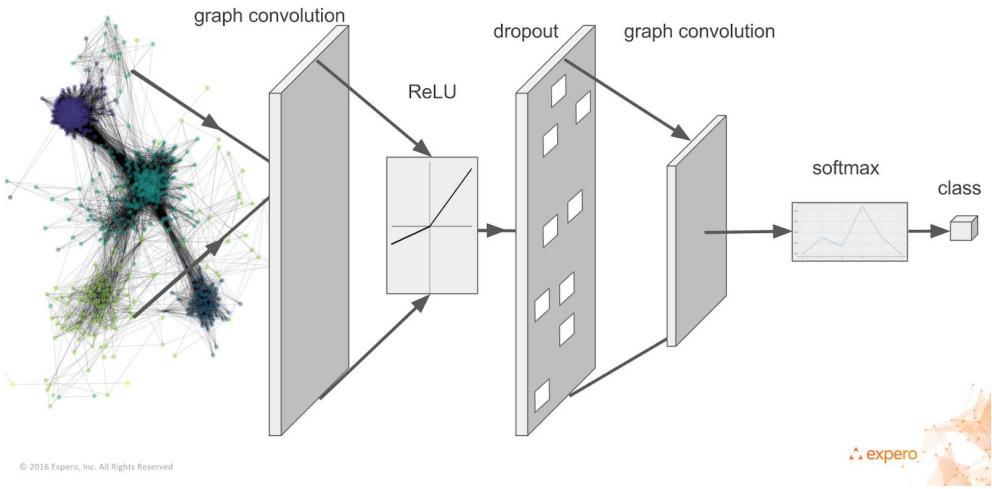
- Generalization Ability and Robustness Analysis
  - **Case 1**:The encoder knows the channel condition in both the training and testing processes.
  - **Case 2**:The encoder only knows the channel condition in the training process. In the testing process, the encoder assumes the channel condition to be 15dB in the AWGN channel or 0.125 in the BEC.
  - **Case 3:** The encoder does not know the channel condition in neither the training nor the testing process.



Case study 2: Branchy-GNN for point cloud processing

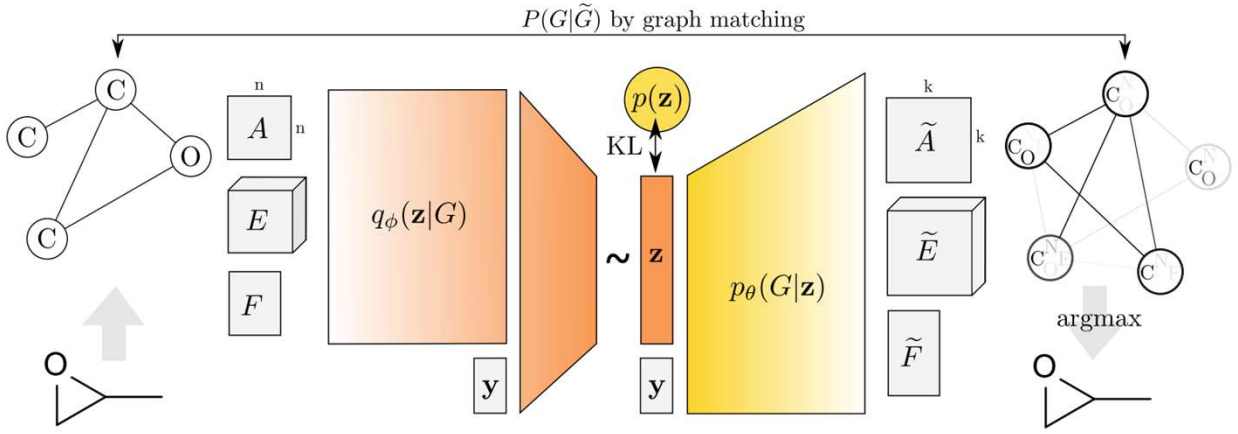
- Graph neural networks (GNNs) for point cloud processing.
- Generate edge features for the point cloud to build a graph representation.





Case study 2: Branchy-GNN for point cloud processing

- New challenges
  - More serious data amplification (high-dimensional node feature)
  - Hard-to-compress graph data (complex dependence between adjacent matrix and node/edge features)

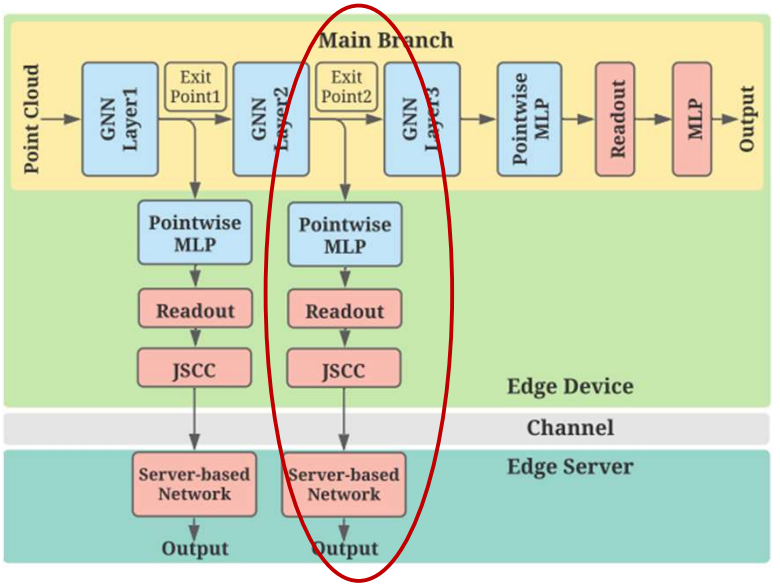


S., Martin, and N. Komodakis. "GraphVAE: Towards generation of small graphs using variational autoencoders." In International Conference on Artificial Neural Networks, pp. 412-422. Springer, Cham, 2018.

End-to-end design

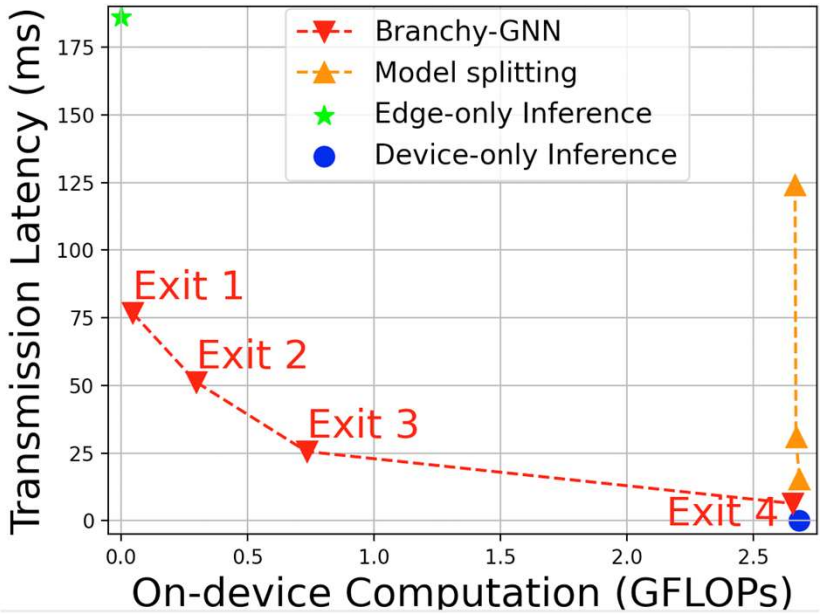
Solution: Branch structure

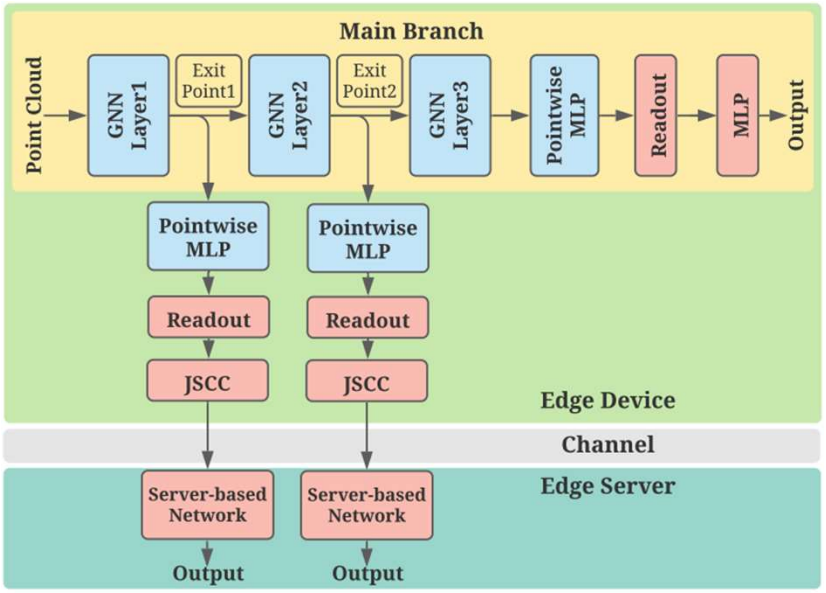
- Early exiting from the main branch
- Intermediate feature compression
  - Readout layer:
    - maps the graph representation to a fixed-size vector
  - Joint Source-Channel Coding (JSCC)
    - Directly map the input feature to the channel symbols.
    - Train the coding scheme with channel noise in an end-to-end manner.
- Server-based network



Branchy-GNN framework

- Achieve a better communication-computation tradeoff
  - Flexibly select the exit point according to the edge environments.



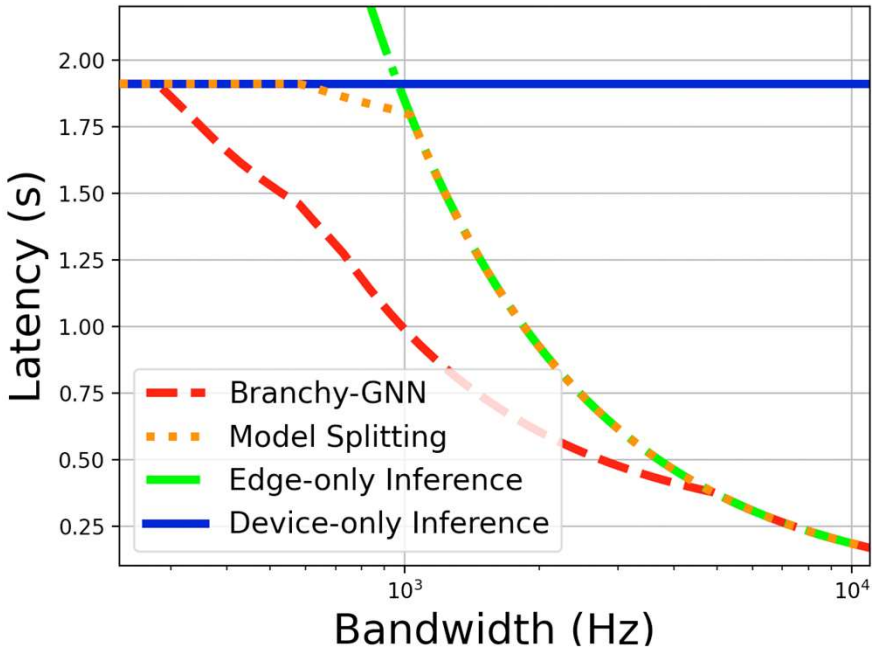


Experiment

- Setup
  - Point cloud classification based on ModelNet40 dataset
  - 4 exit points in Branchy-GNN
- Baselines:
  - Model splitting
  - Server-only inference
  - Edge-only inference

Experiment

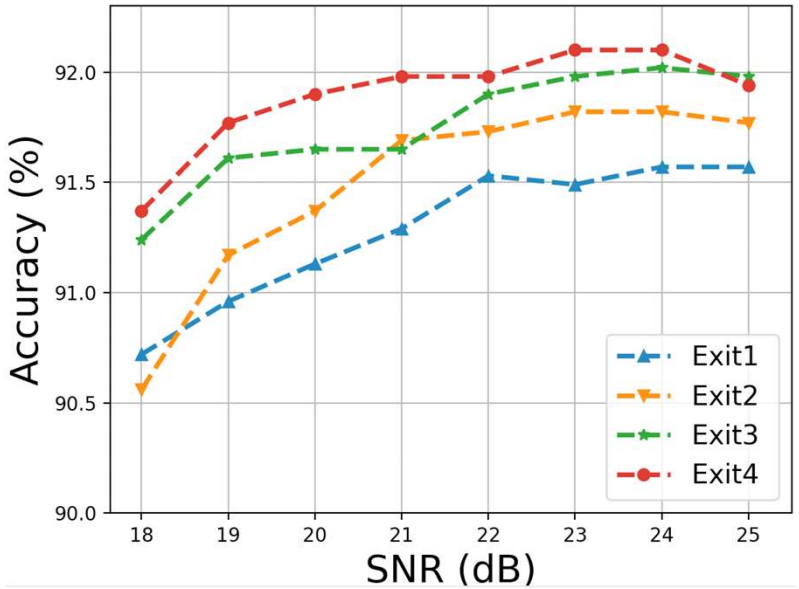
Edge inference speedup



Experiment

Robustness of the proposed method

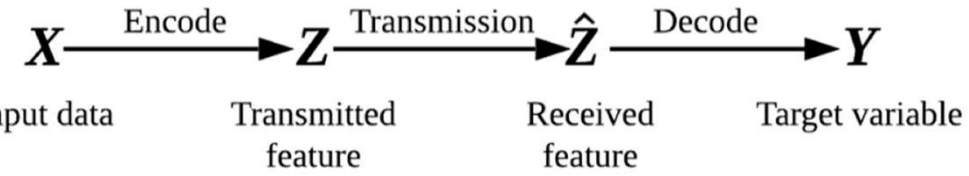
- Train the network at the SNR of 20dB and test it with SNR ranging from 18dB to 25dB.



Task-oriented Communication via Information Bottleneck

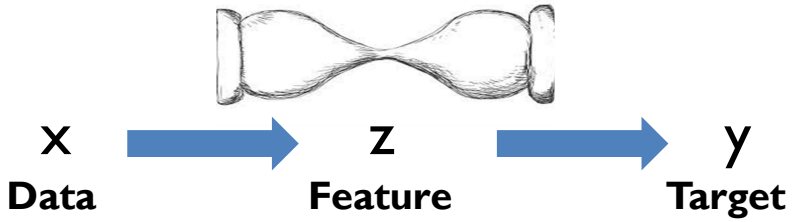
J. Shao, Y. Mao, and **J. Zhang**, "Learning task-oriented communication for edge inference: An information bottleneck approach," JSAC, to appear. (https://arxiv.org/abs/2102.04170)

Feature encoding via information bottleneck



Dog

Information Bottleneck (IB)



IB strive for *minimality* and *sufficiency* of the latent Z

- **Minimality**: minimizing amount of information necessary of X for the task;
- **Sufficiency**: preserving the information to solve the task.

N. Tishby, F.C. Pereira, and W. Biale. The information bottleneck method. In *The 37th annual Allerton Conf. on Communication, Control, and Computing*, 1999.

Information Bottleneck

$$\min \underbrace{-I(\hat{Z}, Y)}_{\text{Distortion}} + \beta \cdot \underbrace{I(\hat{Z}, X)}_{\text{Rate}}$$

How well Z predicts Y

To promote accuracy

How much Z compresses X

To promote generalization

Tradeoff

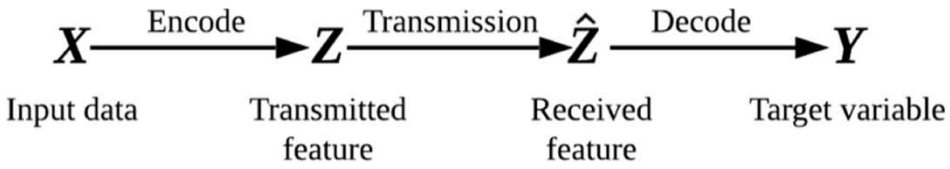
preserving "relevant" information vs. finding "compact" representation

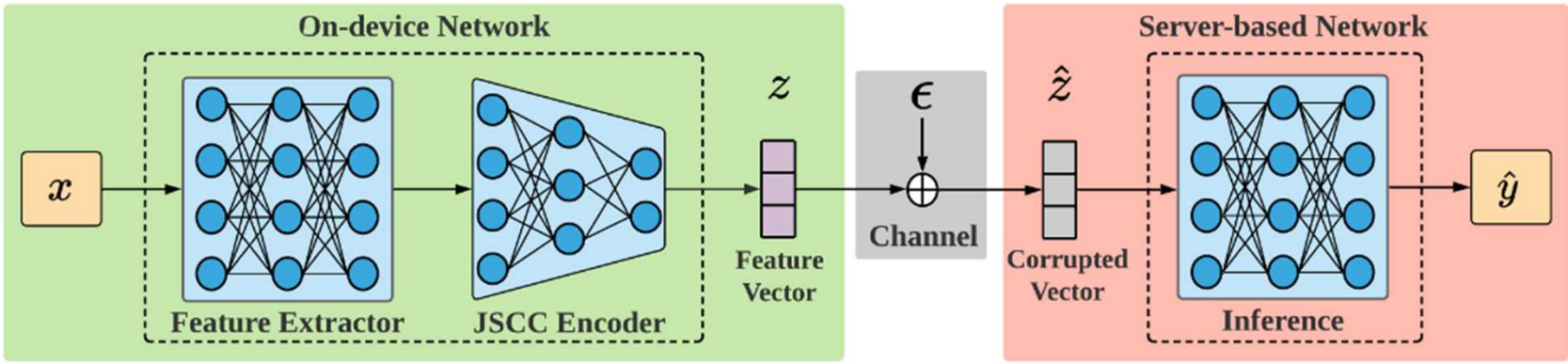
- Applications of information bottleneck
  - Understand deep learning
  - Improve generalization performance and robustness to adversarial attack

A natural fit for edge inference

Information bottleneck

Variational Feature Encoding (VFE)





Information bottleneck

Variational Feature Encoding (VFE)

$$-I(Y,\hat{Z})+\beta I(\hat{Z},X) = -\int p(\boldsymbol{y}|\hat{\boldsymbol{z}})p(\hat{\boldsymbol{z}})\log p(\boldsymbol{y}|\hat{\boldsymbol{z}})d\boldsymbol{y}d\hat{\boldsymbol{z}} + \beta\int p_{\boldsymbol{\phi}}(\hat{\boldsymbol{z}}|\boldsymbol{x})p(\boldsymbol{x})\log\frac{p_{\boldsymbol{\phi}}(\hat{\boldsymbol{z}}|\boldsymbol{x})}{p(\hat{\boldsymbol{z}})}d\boldsymbol{x}d\hat{\boldsymbol{z}} + H(Y)$$

$$\leq -\int p(\boldsymbol{y}|\hat{\boldsymbol{z}})p(\hat{\boldsymbol{z}})\log q_{\theta}(\boldsymbol{y}|\hat{\boldsymbol{z}})d\boldsymbol{y}d\hat{\boldsymbol{z}} + \beta\int p_{\boldsymbol{\phi}}(\hat{\boldsymbol{z}}|\boldsymbol{x})p(\boldsymbol{x})\log\frac{p_{\boldsymbol{\phi}}(\hat{\boldsymbol{z}}|\boldsymbol{x})}{q(\hat{\boldsymbol{z}})}d\boldsymbol{x}d\hat{\boldsymbol{z}} + \underbrace{H(Y)}_{\text{constant}}$$

Highly challenging to estimate!

Cross-entropy **Variational inference**

(Alemi et. al., 2016)

$p_{\phi}(\hat{z}|x)$ is defined by the neural network (encoder)

$p(\hat{\boldsymbol{z}})$ is an unknown marginal distribution

$q(\hat{\boldsymbol{z}})$ is a **log-uniform distribution** to approximate the unknown marginal distribution

Some details: Approximated closed-form solution

$p_\phi(\hat{z}|x)$ is a factorized Gaussian distribution

$q(\hat{\boldsymbol{z}})$ is a log-uniform distribution

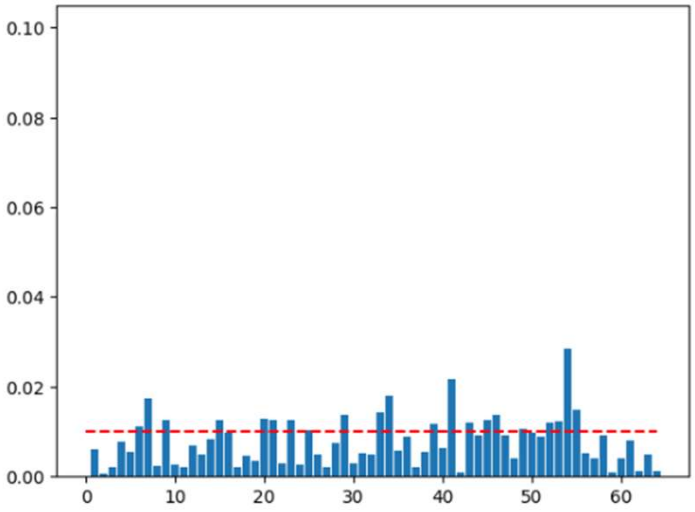
$$D_{KL}(p_\phi(\hat{z}|x)\|q(x)) = \sum_{i=1}^{n} D_{KL}(p_\phi(\hat{z}_i|x)\|q(\hat{z}_i))$$

$$\begin{aligned} -D_{KL}(p_\phi(\hat{z}_i \mid \boldsymbol{x})\|q(\hat{z}_i)) &= \frac{1}{2}\log\alpha_i - \mathbb{E}_{\epsilon\sim\mathcal{N}(1,\alpha_i)}\log|\epsilon| + \mathrm{C} \\ &\approx k_1 S(k_2 + k_3\log\alpha_i) - 0.5\log\left(1+\alpha_i^{-1}\right) + \mathrm{C} \end{aligned}$$

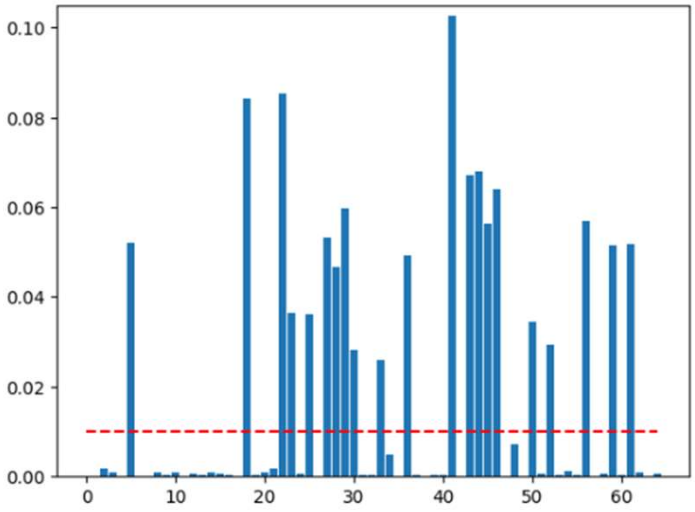
(Molchanov et. al., 2017)

Empirical results

Soft gate



Gaussian prior

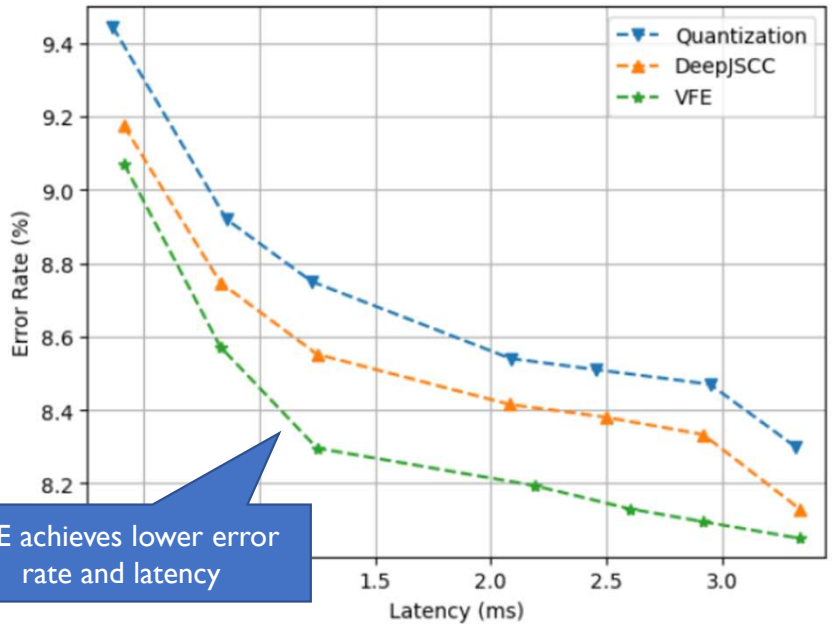


Log-uniform distribution

Experiment

- **Baselines** (data-oriented communication):
  - DeepJSCC (Joint Source-Channel Coding)
  - Learning-based quantization (w/ ideal channel coding)

Rate-distortion on CIFAR-10 dataset

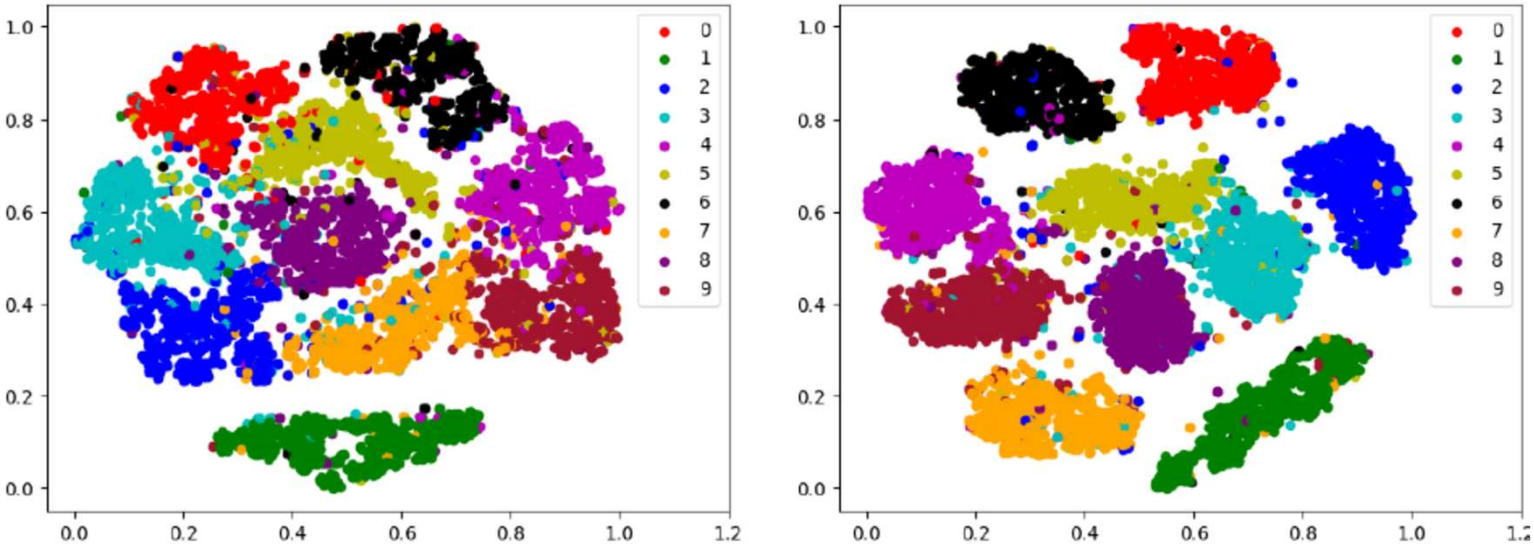


VFE achieves lower error rate and latency

- **Task-oriented communication reduces transmitted data size**
  - With simple encoder/decoder
- **Key enablers**
  - End-to-end design
  - Information bottleneck
  - Joint source-channel coding

Experiment

- VFE method can better distinguish the data from different classes compared with DeepJSCC.

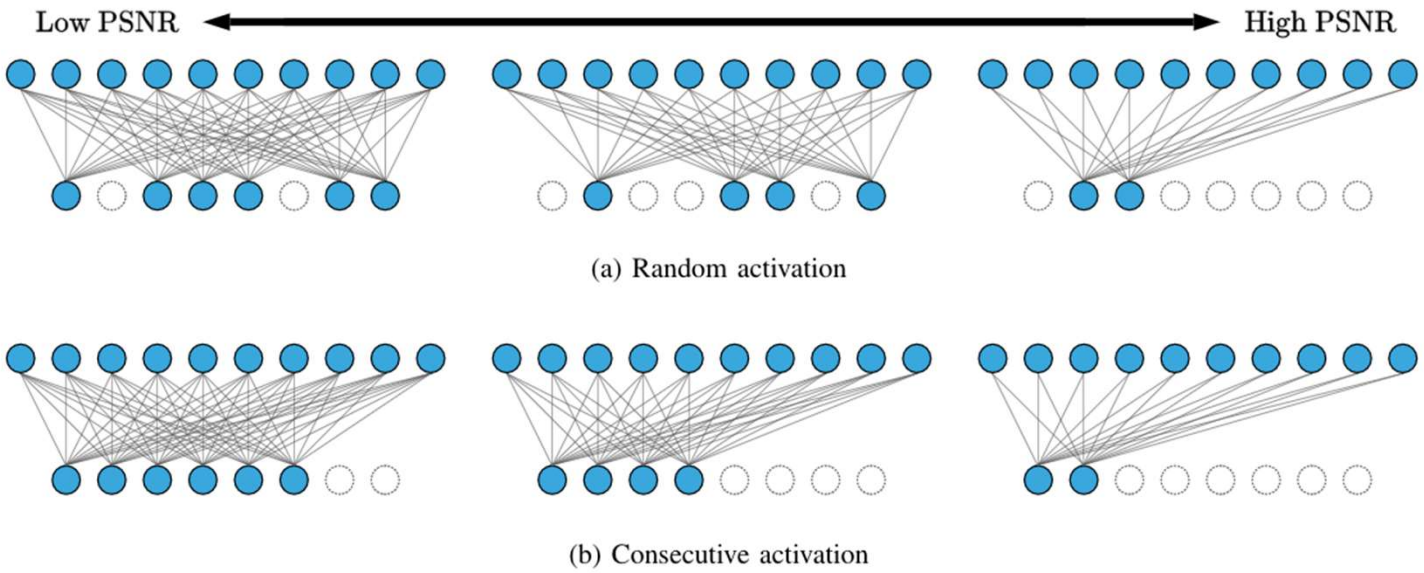


(a) DeepJSCC: Accuracy = 96.77%, dimension n = 24. (b) Proposed VFE: Accuracy = 97.39%, dimension n = 24.

2-dimensional t-SNE embedding of the received feature in the MNIST classification task with PSNR = 20 dB.

Variable-length Variational Feature Encoding (VL-VFE)

- To adapt to channel states: variable-length coding
- To reduce signaling overhead, the coding scheme should be consecutive and monotonic

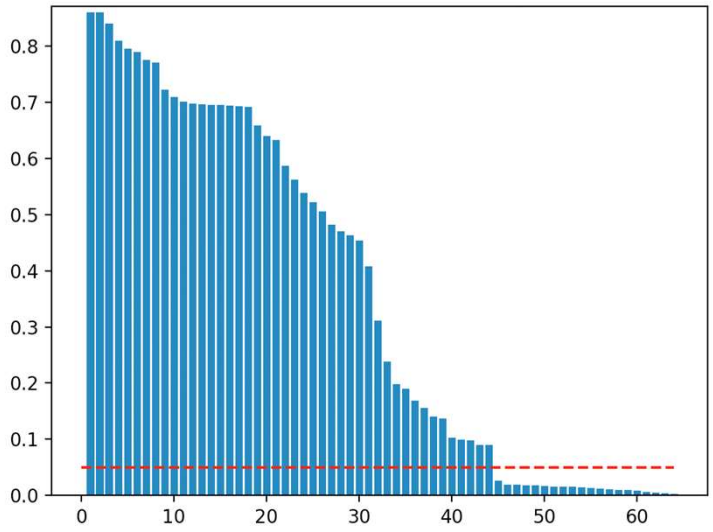


(a) Random activation

(b) Consecutive activation

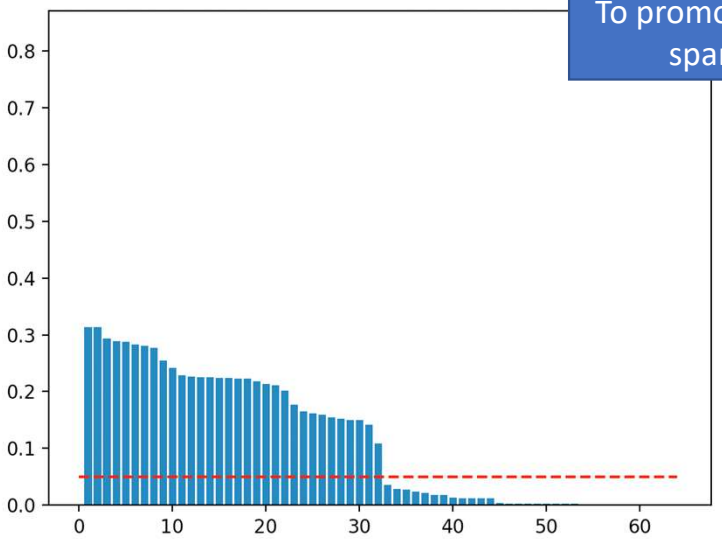
Variable-length Variational Feature Encoding (VL-VFE)

Dimension importance $\gamma_i(\sigma^2) = \sum_{j=i}^{n} |g_j(\sigma^2)|$ **Soft gate function**



PSNR = 10dB

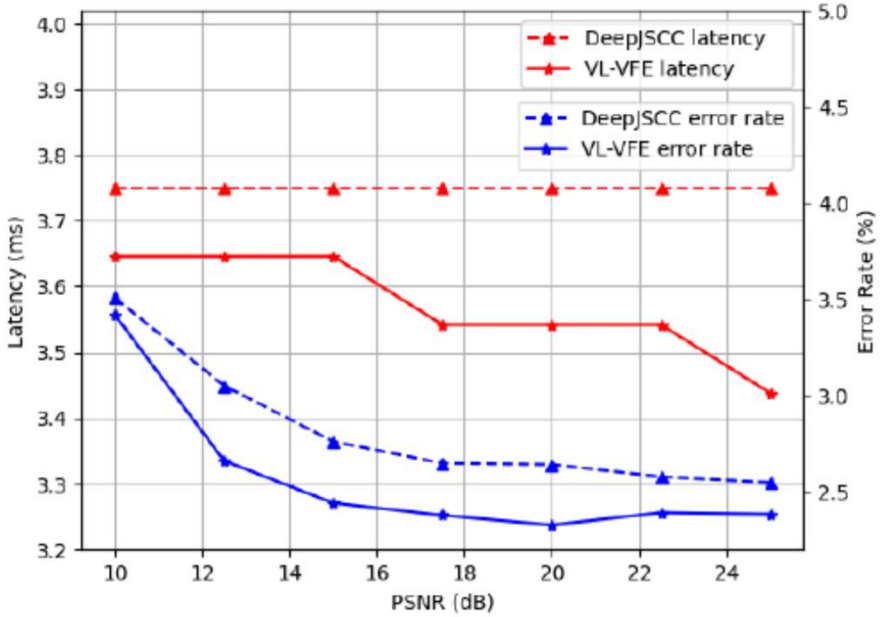
To promote group sparsity



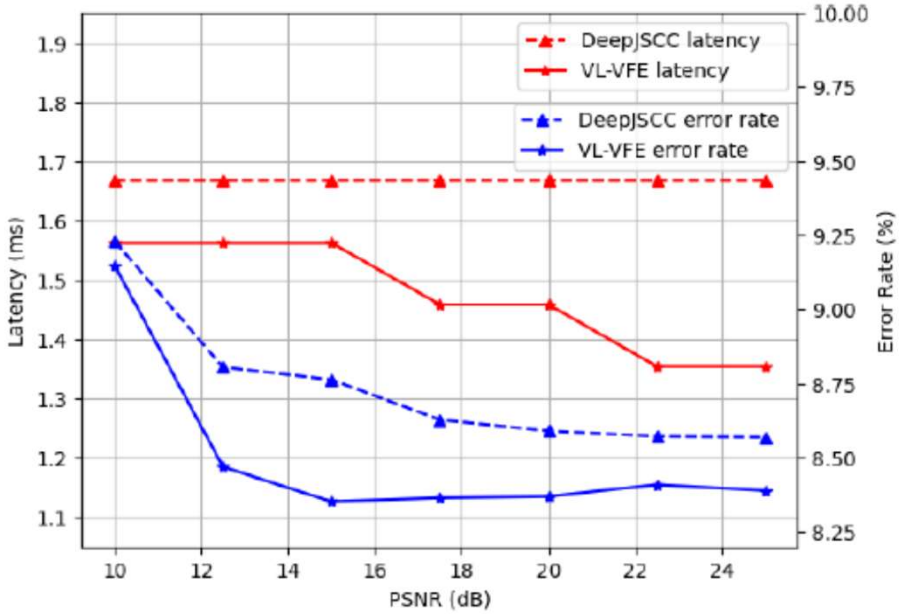
PSNR = 20dB

Information bottleneck

Variable-length Variational Feature Encoding (VL-VFE)



(a) The MNIST classification task



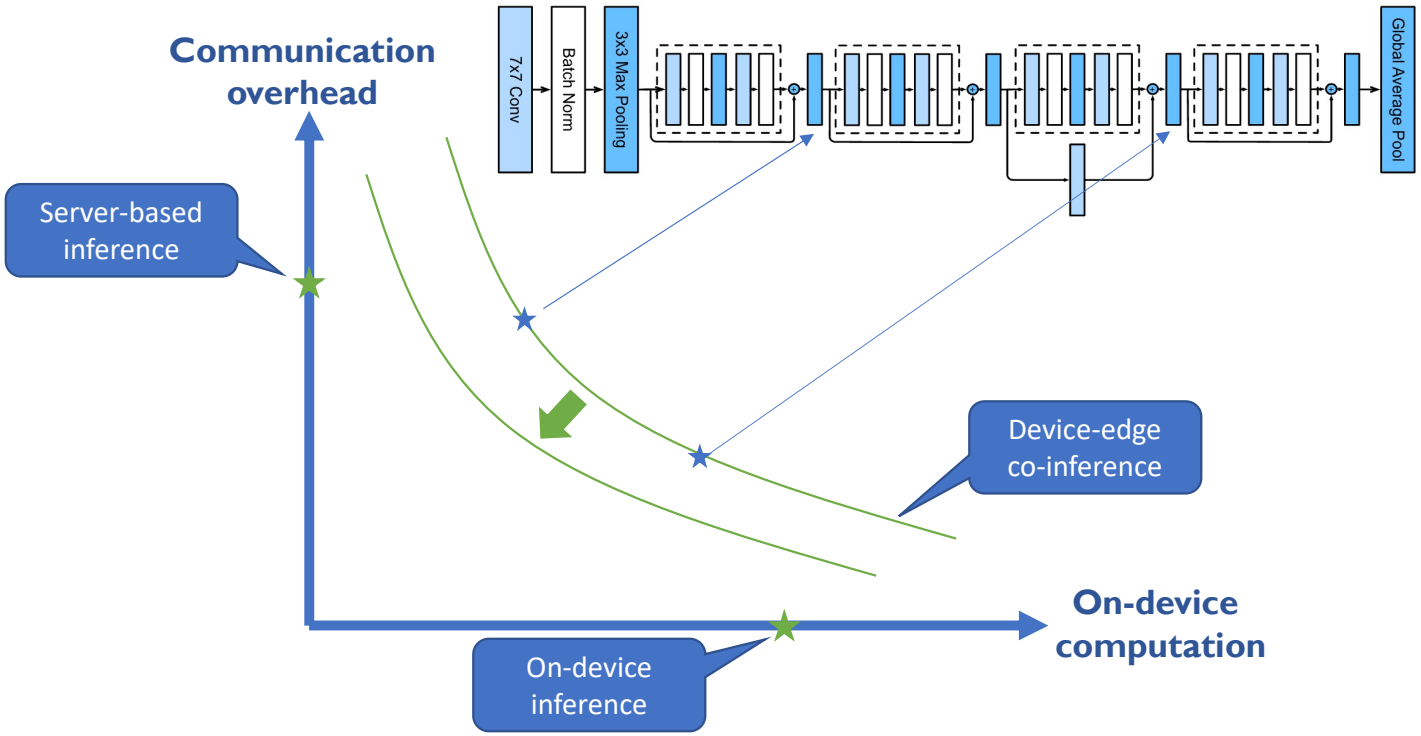
(b) The CIFAR-10 classification task

Improve communication-computation tradeoff

J. Shao, **J. Zhang**, "Communication-computation trade-off in resource-constrained edge inference," *IEEE Commun. Mag.*, Dec 2020.

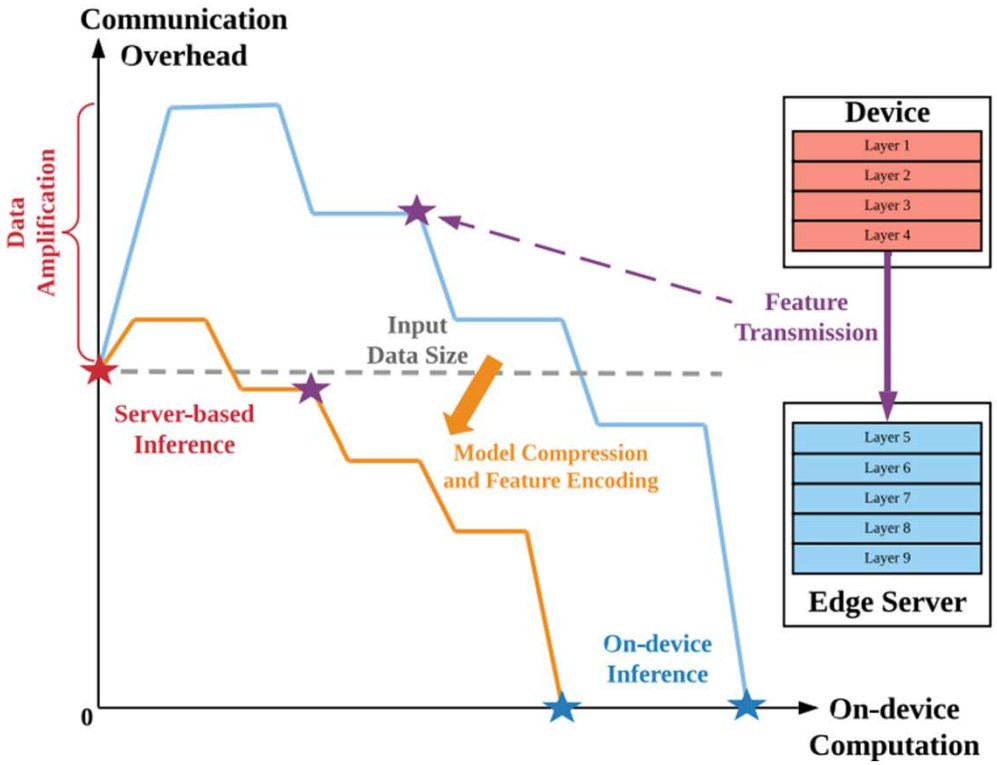
X. Zhang, J. Shao, Y. Mao, and **J. Zhang**, "Communication-computation efficient device-edge co-inference via AutoML," *IEEE GLOBECOM 2021*.

Communication-computation tradeoff



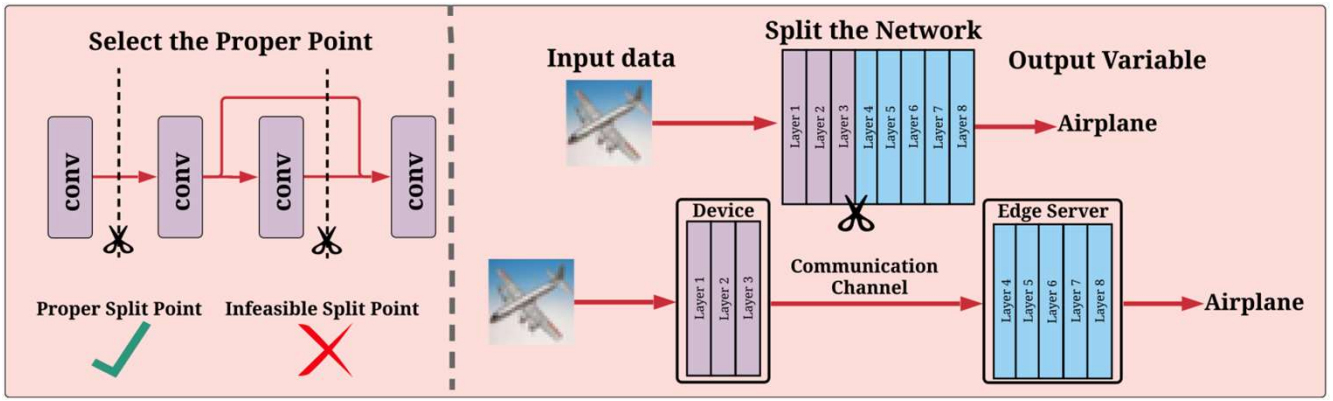
A three-step framework

1. Split the network
2. Compress the on-device network
3. Encode the intermediate feature



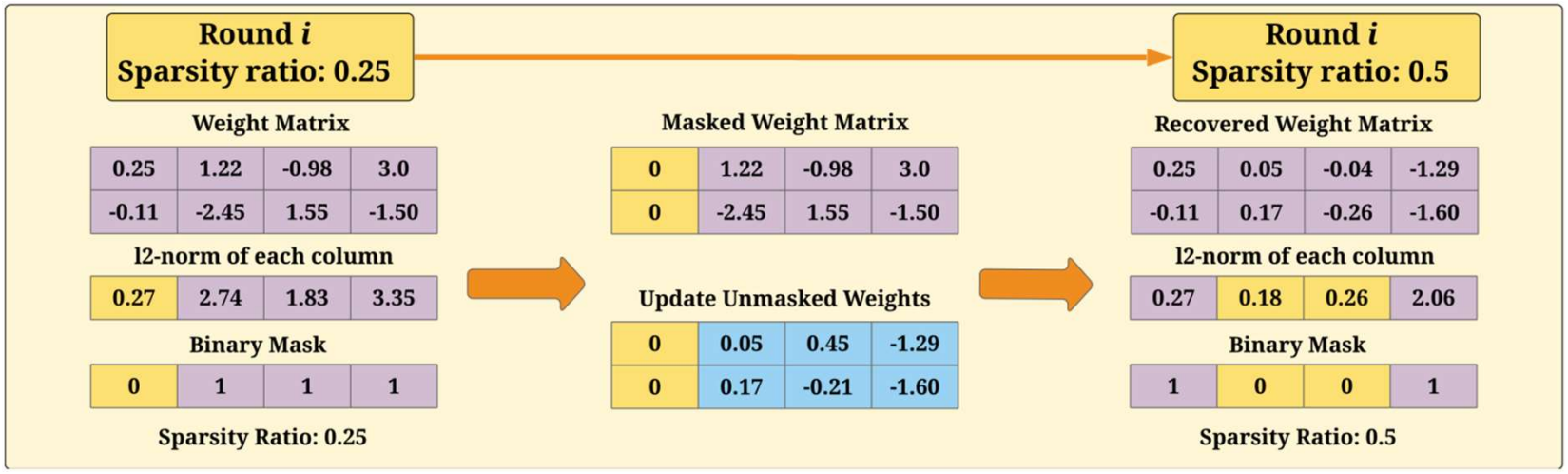
Step 1: Split the network

- Avoid letting the split points transmit multiple data flows.
- Balance the on-device computation and communication overhead.



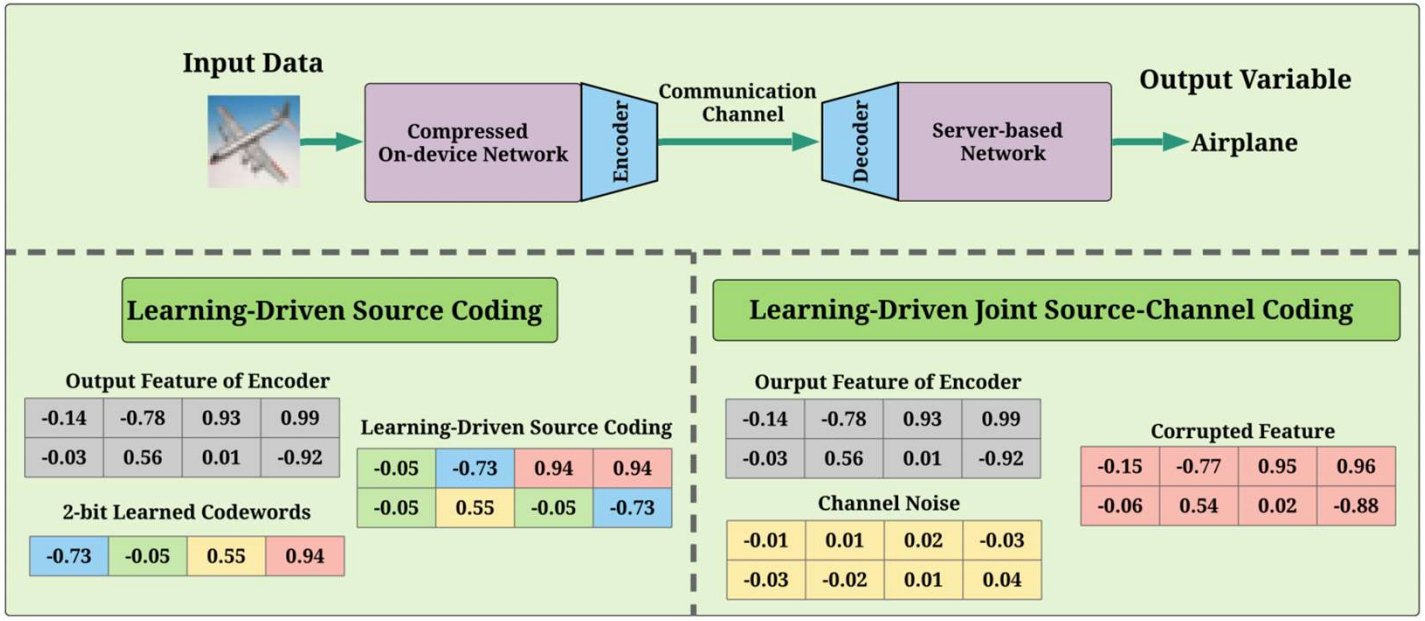
Step 2: Compress the on-device network

- Incremental network pruning
  - Pruning the weights based on the l-2 norm.
  - Incrementally increase the sparsity ratio: robust pruning process.
  - Structural network pruning: easy for hardware acceleration.



Step 3: Encode the intermediate feature

- Learning-based coding scheme:
  - Source coding: map the input to the learned codewords.
  - Joint Source-channel coding: map the input to the channel symbols.

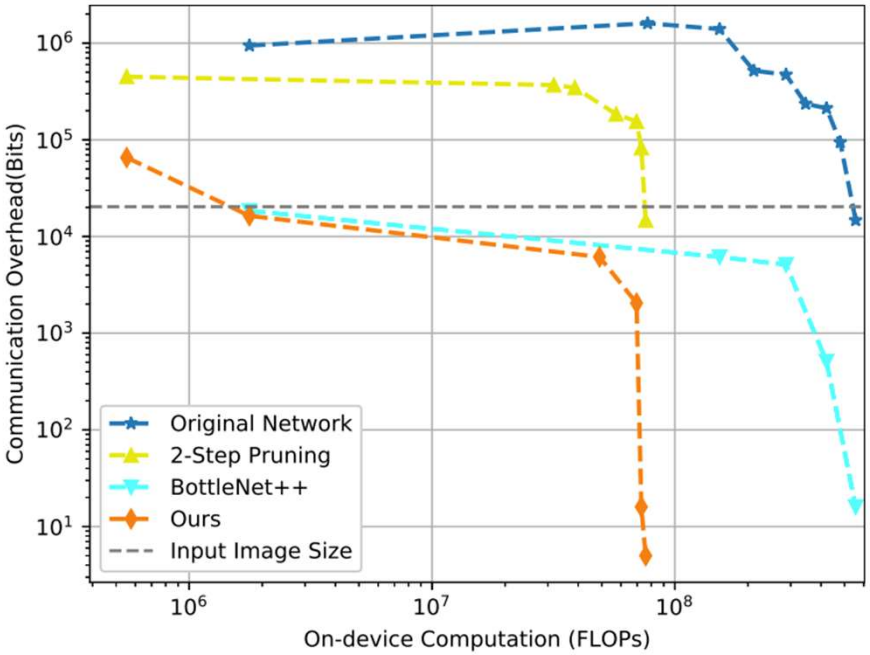


Experiment

- Setup
  - ResNet18 for CIFAR10 classification
- Baselines:
  - BottleNet++ (feature compression)
  - 2-step pruning (model compression)
  - Original network

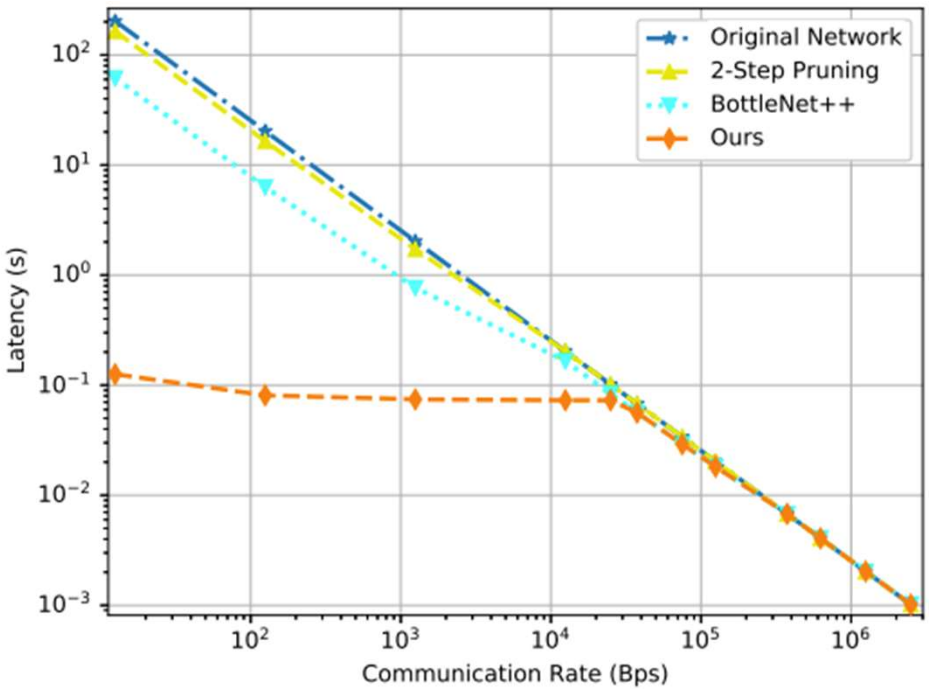
Experiment

- Communication-Computation Tradeoff



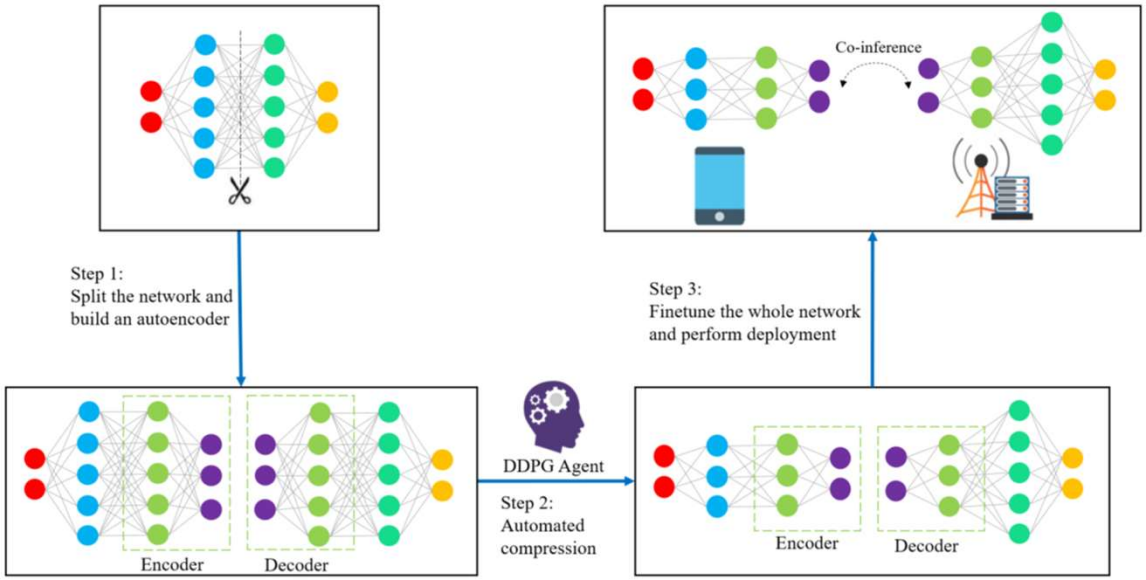
Experiment

- Real-word test
  - Raspberry Pi 3 as an edge device
  - PC with RTX 2080 Ti as a server

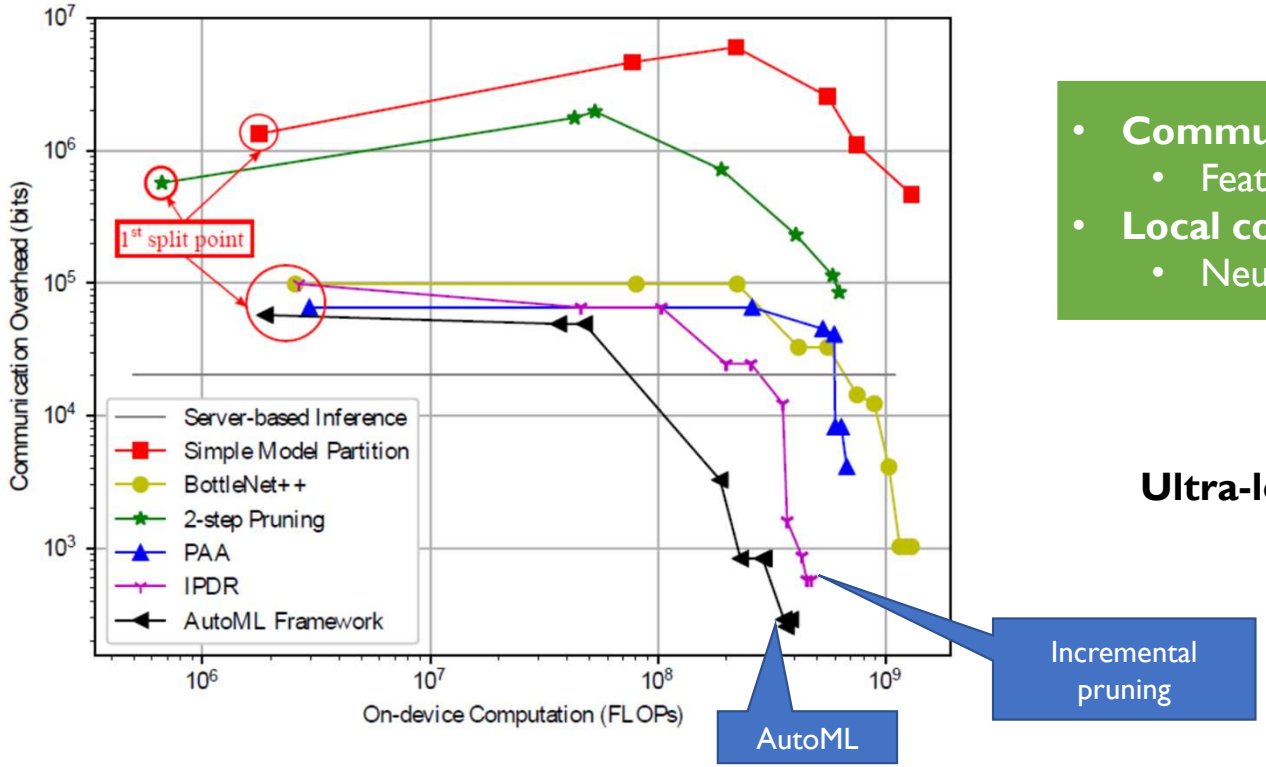


Neural architecture optimization: AutoML

- An automated machine learning (AutoML) framework
  - It determines the sparsity level for each on-device DNN layer and the compression ratio for the intermediate feature vector.
  - A deep deterministic policy gradient (DDPG) algorithm is adopted to choose actions.



Communication-computation tradeoff



- **Communication overhead reduction**
  - Feature compression
- **Local computation load reduction**
  - Neural architecture design

Ultra-low latency/power inference

Task-oriented communication for cooperative inference

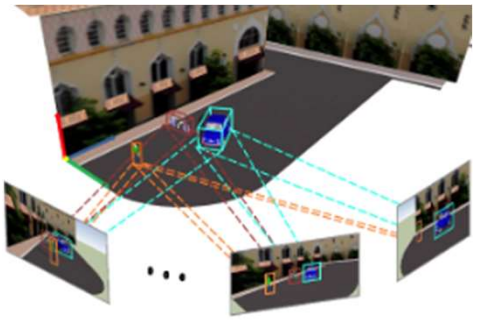
J. Shao, Y. Mao, and **J. Zhang**, "Task-oriented communication for multi-device cooperative edge inference," submitted to IEEE Transactions on Wireless Communications.
https://arxiv.org/abs/2109.00172

New Applications: Cooperative Inference

- Multi-device system.
  - Cooperation among multiple devices with distinct views improves sensing capability.

Vehicle Re-identification

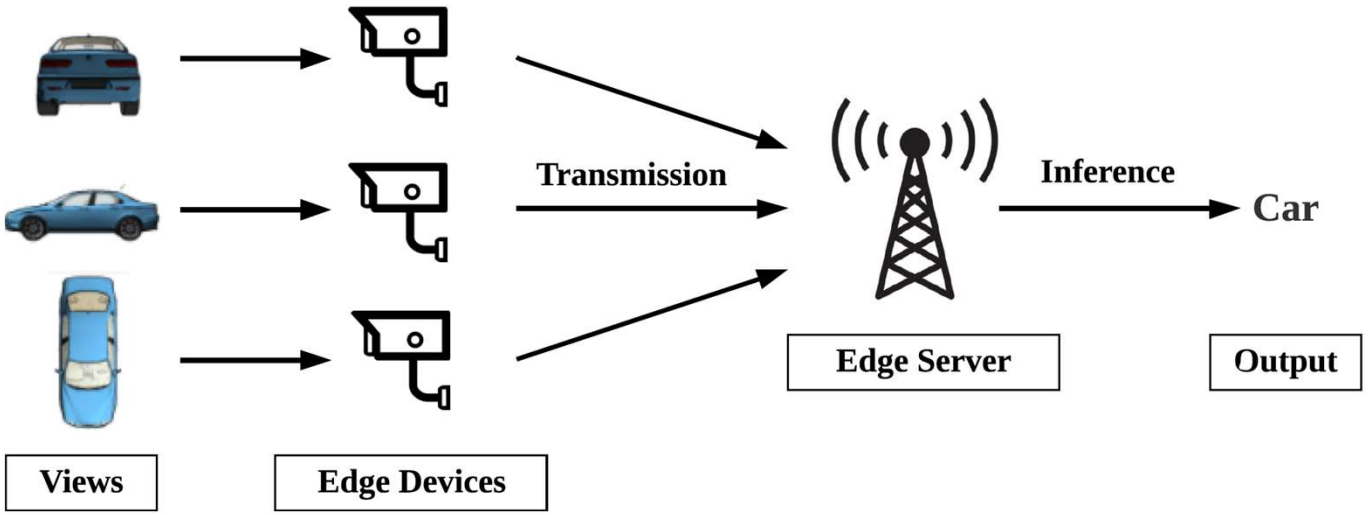
Pose Estimation



3D Localization

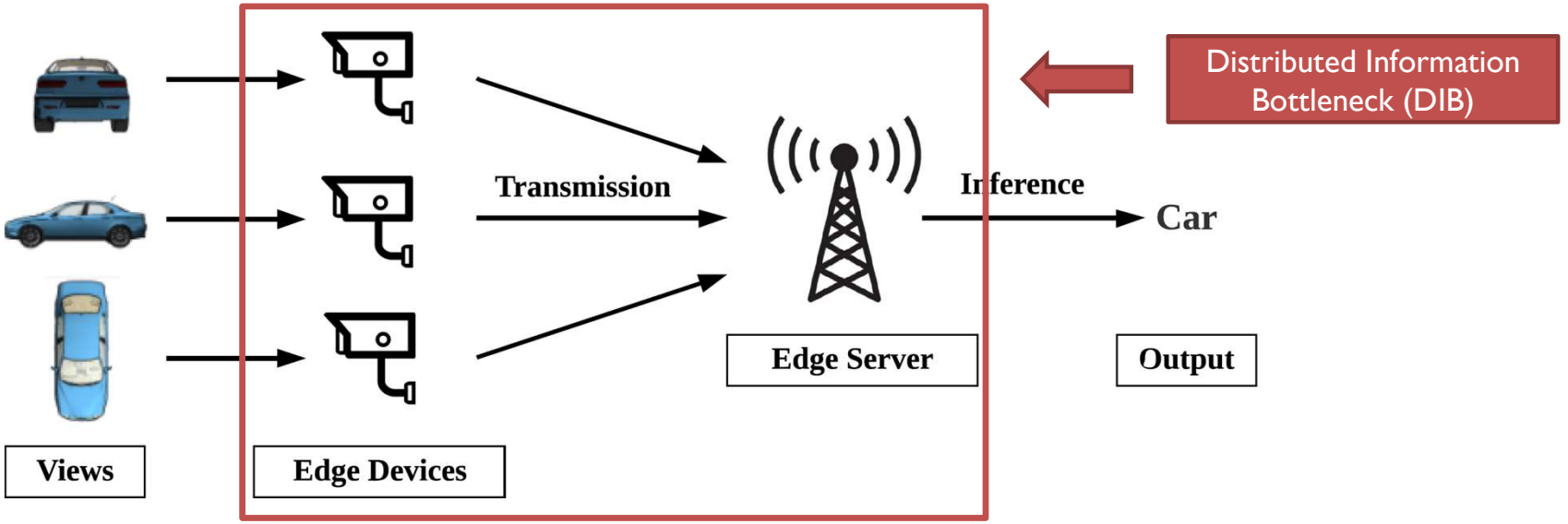
Multi-Device Cooperative Edge Inference

- Device-Edge Co-Inference in multi-device systems.
  - Feature extraction (e.g., the VFE).
  - Distributed feature encoding.



Multi-Device Cooperative Edge Inference

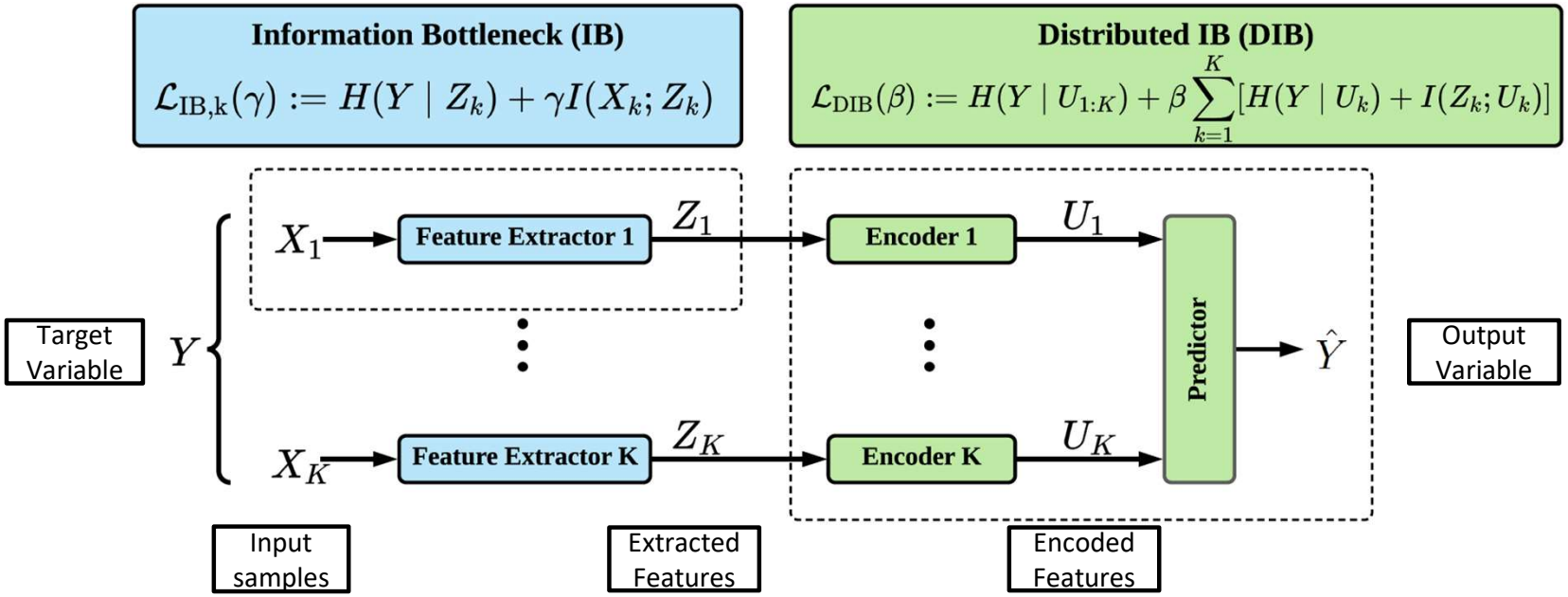
- Design an efficient method that can fully exploit the correlation among multiple features in distributed feature encoding.



Aguerri, Inaki Estella, and Abdellatif Zaidi. "Distributed variational representation learning." *IEEE Trans. Pattern Anal. Machine Intell.* 120-138, 2019

Multi-Device Cooperative Edge Inference

- Probabilistic modeling with *K* devices
- Loss functions



Distributed Deterministic Information Bottleneck (DDIB)

- DIB objective

$$\mathcal{L}_{\mathrm{DIB}}(\beta) := H(Y \mid U_{1:K}) + \beta \sum_{k=1}^{K} [H(Y \mid U_k) + \underbrace{I(Z_k; U_k)}_{\text{Rate}}]$$

The minimality is only satisfied in the asymptotic limit

- DDIB objective

$$\mathcal{L}_{\mathrm{DDIB}}(\beta) := H(Y \mid U_{1:K}) + \beta \sum_{k=1}^{K} [H(Y \mid U_k) + R_{\mathrm{bit}}(U_k)]$$

Proposed method: Variational DDIB (VDDIB)

- Using variational inference to estimate the intractable (entropy) terms.

$$\mathcal{L}_{\mathrm{DDIB}}(\beta) := H(Y \mid U_{1:K}) + \beta \sum_{k=1}^{K} [H(Y \mid U_k) + R_{\mathrm{bit}}(U_k)]$$

$$\mathcal{L}_{\mathrm{VDDIB}}(\beta; \boldsymbol{\phi}, \boldsymbol{\psi}) := \mathbf{E}_{p_\theta(\boldsymbol{z}_{1:K}, \boldsymbol{y})} \{ -\log p_{\psi_0}(\boldsymbol{y} \mid \boldsymbol{u}_{1:K}) + \beta \left\{ \sum_{k=1}^{K} -\log p_{\psi_k}(\boldsymbol{y} \mid \boldsymbol{u}_k) + \sum_{k=1}^{K} R_{\mathrm{bit}}(\boldsymbol{u}_k) \right\} \}$$

Minimizing the VDDIB objective may not result in the optimal rate-relevance tradeoff due to the approximations

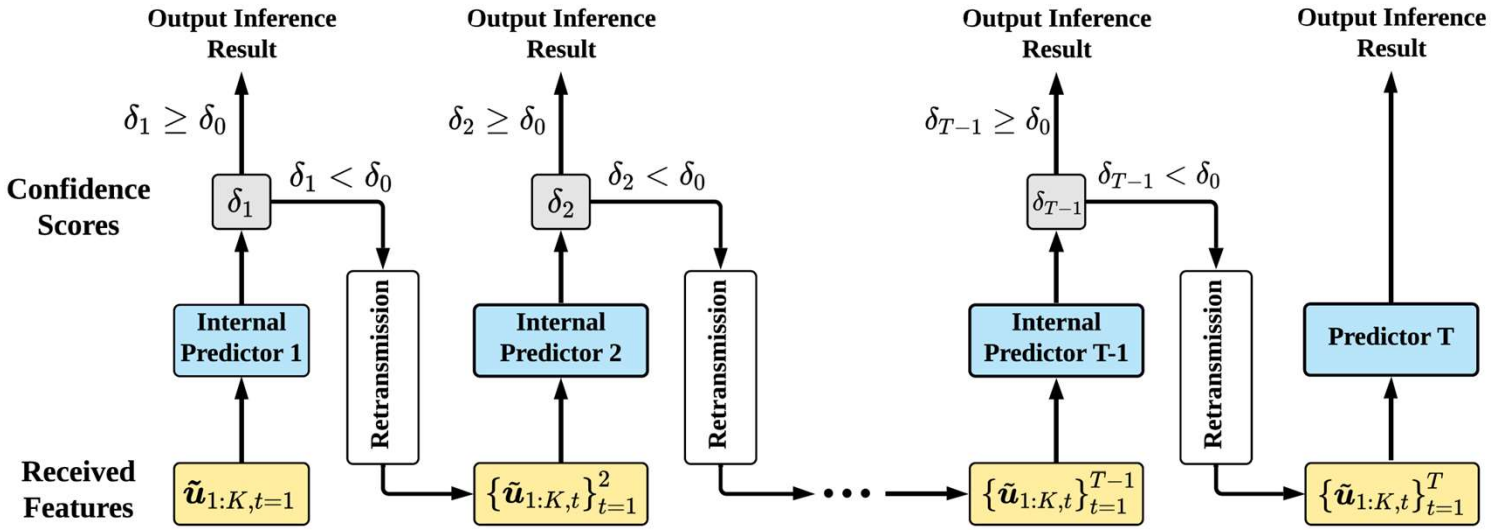
Introduce a selective retransmission (SR) mechanism to further reduce the communication overhead caused by the redundancy among the extracted features.

Selective Retransmission Mechanism

- The edge server selectively activates the edge devices to retransmit their encoded features based on the informativeness of the received features.
- The mechanism consists of a stopping policy and an attention module.

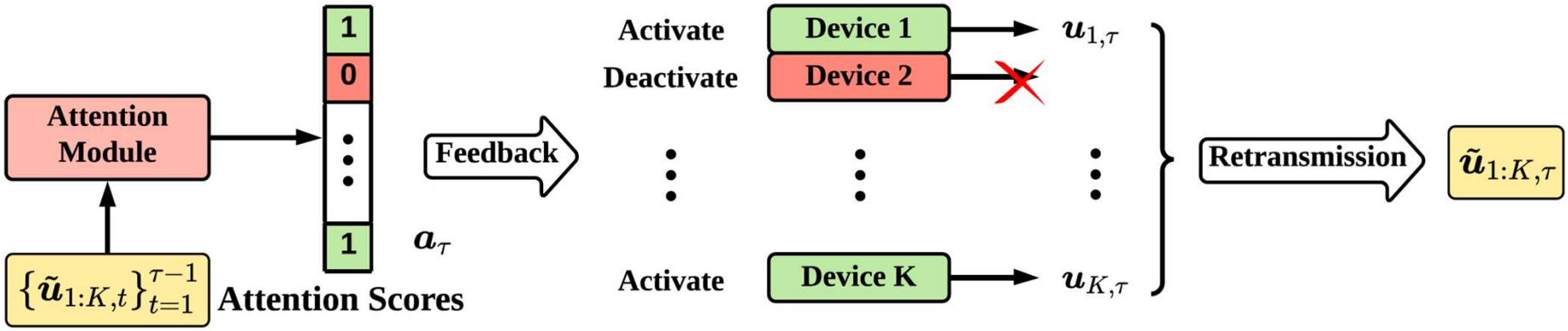
Selective Retransmission Mechanism

- Stopping policy
  - Each edge device is allowed to transmit the encoded feature with a maximum number of *T* attempts.
  - Once the received features are sufficient to output a confident result, the remaining retransmission attempts can be saved.



Selective Retransmission Mechanism

- Attention Module
  - Select the most informative features to retransmit based on the attention scores.



VDDIB with Selective Retransmission Mechanism (VDDIB-SR)

- VDDIB-SR loss function

$$\mathcal{L}_{\mathrm{VDDIB}}(\beta;\boldsymbol{\phi},\boldsymbol{\psi}) := \mathbf{E}_{p_\theta(\boldsymbol{z}_{1:K},\boldsymbol{y})}\Big\{-\log p_{\psi_0}(\boldsymbol{y}\mid \boldsymbol{u}_{1:K}) + \beta\Big\{\sum_{k=1}^{K} -\log p_{\psi_k}(\boldsymbol{y}\mid \boldsymbol{u}_k) + \sum_{k=1}^{K} R_{\mathrm{bit}}(\boldsymbol{u}_k)\Big\}\Big\}$$

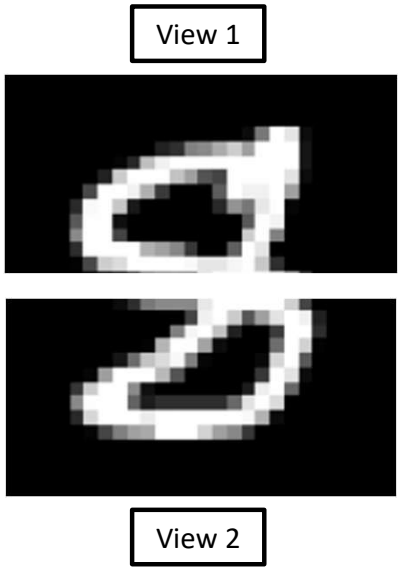
Account for T predictors

Communication cost by the SR mechanism

$$\mathcal{L}_{\mathrm{VDDIB-SR}}\left(\beta,T;\tilde{\boldsymbol{\phi}},\tilde{\boldsymbol{\psi}},\{\boldsymbol{\psi}_k\}_{k=1}^{K}\right) := \mathbf{E}_{p_\theta(\boldsymbol{z}_{1:K},\boldsymbol{y})}\Big\{\frac{1}{T}\sum_{\tau=1}^{T} -\log p_{\tilde{\psi}_\tau}\left(\boldsymbol{y}\mid \{\tilde{\boldsymbol{u}}_{1:K,t}\}_{t=1}^{\tau}\right) + \beta\Big\{\sum_{k=1}^{K} -\log p_{\psi_k}(\boldsymbol{y}\mid \boldsymbol{u}_k) + \sum_{k=1}^{K}\sum_{t=1}^{T} R_{\mathrm{bit}}(\tilde{\boldsymbol{u}}_{k,t})\Big\}\Big\}$$

Performance Evaluation

- Cooperative inference tasks



Two-view MNIST classification

Twelve-view Shape Recognition on ModelNet40 dataset

Performance Evaluation

- The accuracy of the cooperative tasks under different bit constraints.
- Data-oriented communication leads to
  - 1.3 kbits overhead with 98.6% accuracy in the MNIST classification task.
  - 120 KB overhead with 92% accuracy in the shape recognition task.

MNIST classification

	R_{sum}		
	6 bits	10 bits	14 bits
NN-REG	95.93%	97.49%	97.78%
NN-GBI	96.62%	97.79%	98.02%
eSAFS	96.97%	97.87%	98.05%
CAFS	94.14%	97.43%	97.42%
VDDIB (ours)	97.08%	97.82%	98.06%
VDDIB-SR (T=2) (ours)	**97.13%**	**98.13%**	**98.22%**

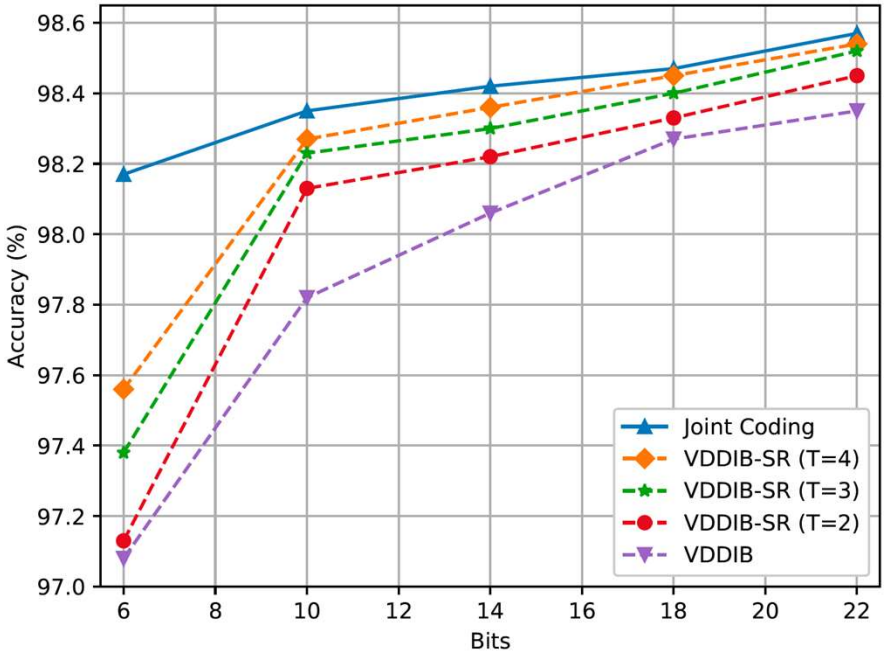
Shape Recognition

	R_{sum}		
	120 bits	240 bits	360 bits
NN-REG	87.50%	88.25%	89.03%
NN-GBI*	88.82%	—	—
eSAFS	85.88%	87.87%	89.50%
CAFS	86.75%	89.56%	90.67%
VDDIB (ours)	89.25%	90.03%	90.75%
VDDIB-SR (T=2) (ours)	**90.25%**	**91.31%**	**91.62%**

* The GBI quantization algorithm is computationally prohibitive when the number of bits is too large.

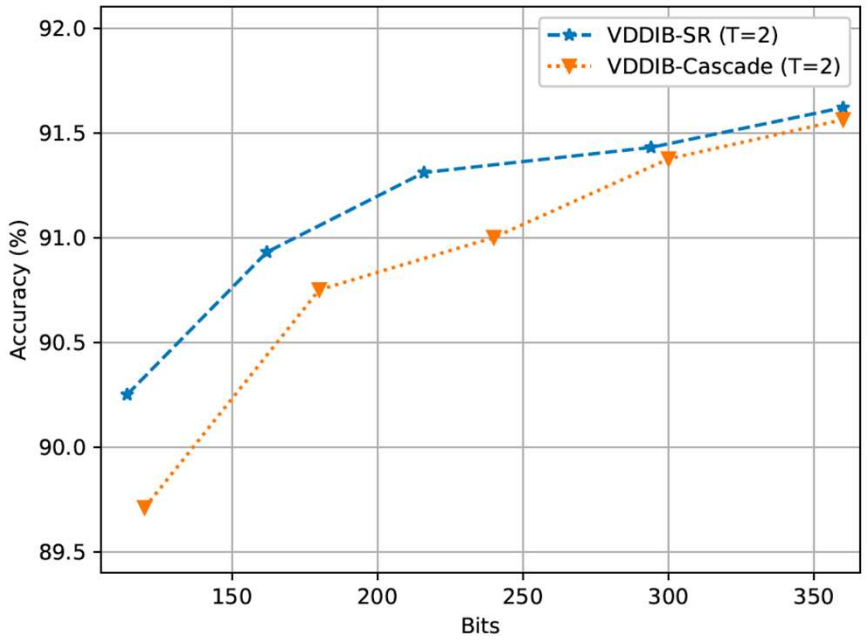
Ablation Study

- Impact of the maximum transmission attempts *T*.
  - the performance of the VDDIB-SR method improves *with T*.



Ablation Study

- Impact of the attention module
  - We propose a baseline method that removes the attention module denoted as VDDIB-Cascade for comparison.

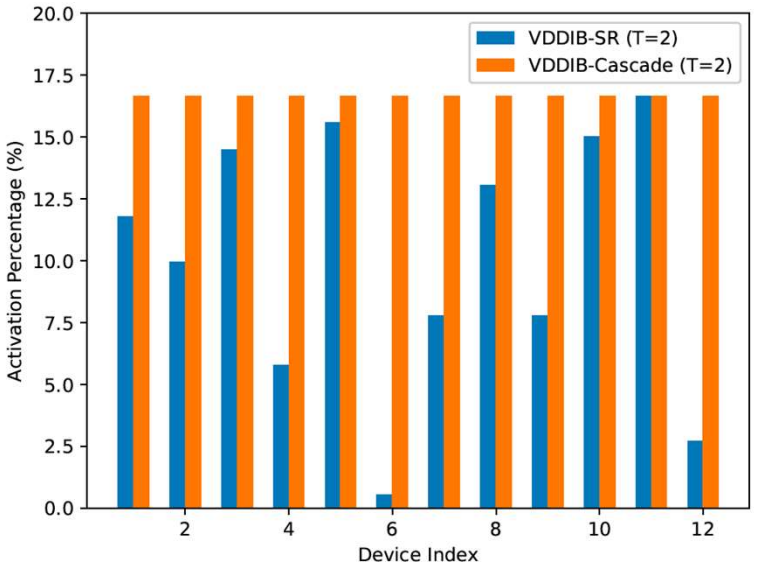


Ablation Study

- Impact of the attention module
  - We propose a baseline method that removes the attention module denoted as VDDIB-Cascade for comparison.
  - The attention module can schedule the most important features to retransmit.

	accuracy	Bits
VDDIB-SR	91.25%	216
VDDIB-Cascade	90.88%	240

The activation percentages of the edge devices in the retransmission



Conclusions

Conclusions

- A paradigm shift in communication system design
  - From data-oriented communication to task-oriented communication
  - From "how to communicate" to "what to communicate"

- Enabling techniques
  - End-to-end design via deep learning
  - Information bottleneck principle (with variational approximation)
  - Neural architecture design

- New tradeoffs
  - Communication-accuracy tradeoff
  - Communication-computation tradeoff

An invitation to this exciting topic

- Related research areas
  - Deep learning
  - Wireless communications
  - Information theory
  - Distributed computing systems
- Application scenarios
  - Video analytics
  - AR/VR
  - Autonomous driving
  - Smart drones
  - …

References

- J. Shao, **J. Zhang**, "BottleNet++: An end-to-end approach for feature compression in device-edge co-inference systems," *IEEE Int. Conf. Commun. (ICC) Workshop on Edge Machine Learning for 5G Mobile Networks and Beyond*, Jun. 2020.
- J. Shao, **J. Zhang**, "Communication-computation trade-off in resource-constrained edge inference," *IEEE Commun. Mag.*, Dec 2020.
- J. Shao, H. Zhang, Y, Mao, and **J. Zhang**, "Branchy-GNN: a device-edge co-inference framework for efficient point cloud processing," *ICASSP 2021*.
- X. Zhang, J. Shao, Y. Mao, and **J. Zhang**, "Communication-computation efficient device-edge co-inference via AutoML," *IEEE GLOBECOM 2021*.
- J. Shao, Y. Mao, **J. Zhang**, "Learning task-oriented communication for edge inference: An information bottleneck approach," *JSAC*, to appear. (https://arxiv.org/abs/2102.04170)
- J. Shao, Y. Mao, and **J. Zhang**, "Task-oriented communication for multi-device cooperative edge inference," submitted to *IEEE Trans. Wireless Communications*. (https://arxiv.org/abs/2109.00172)

Thank you!

- For more details

  http://www.eie.polyu.edu.hk/~jeiezhang/

  eejzhang@ust.hk