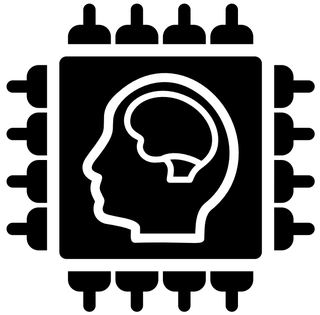
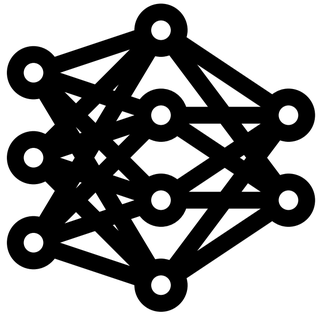
FlexDNN

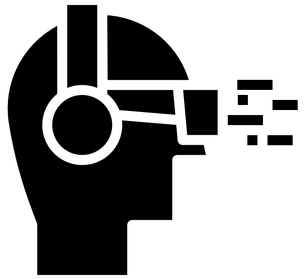
Input-Adaptive On-Device Deep Learning for Efficient Mobile Vision

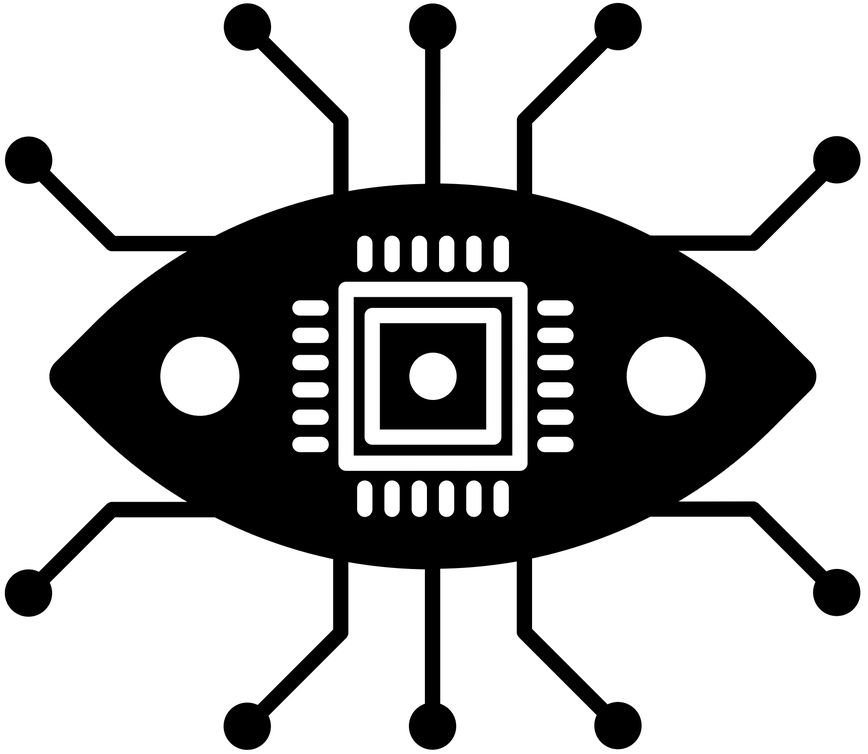


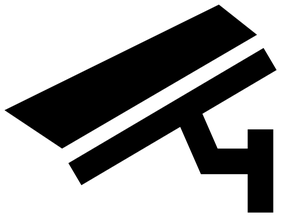


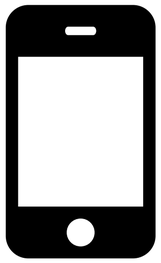
Biyi Fang, Xiao Zeng, Faen Zhang, Hui Xu, Mi Zhang

On Device Video Analytics

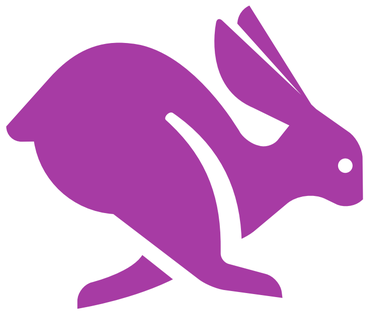


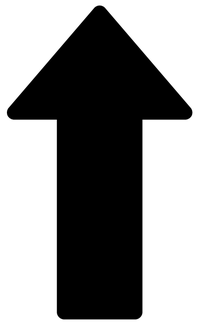


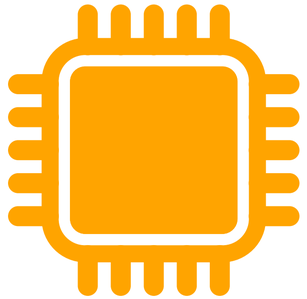


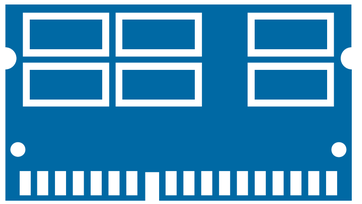


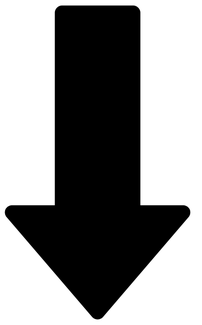
The Tradeoff











You Can’t Always Get What You Want

The Challenge

Fact 1: Edge devices have limited compute resources and battery capacity.

Fact 2: DNNs are computation-expensive with high energy consumption.

An Observation

Some video frames are easier to recognize than others.

An Observation

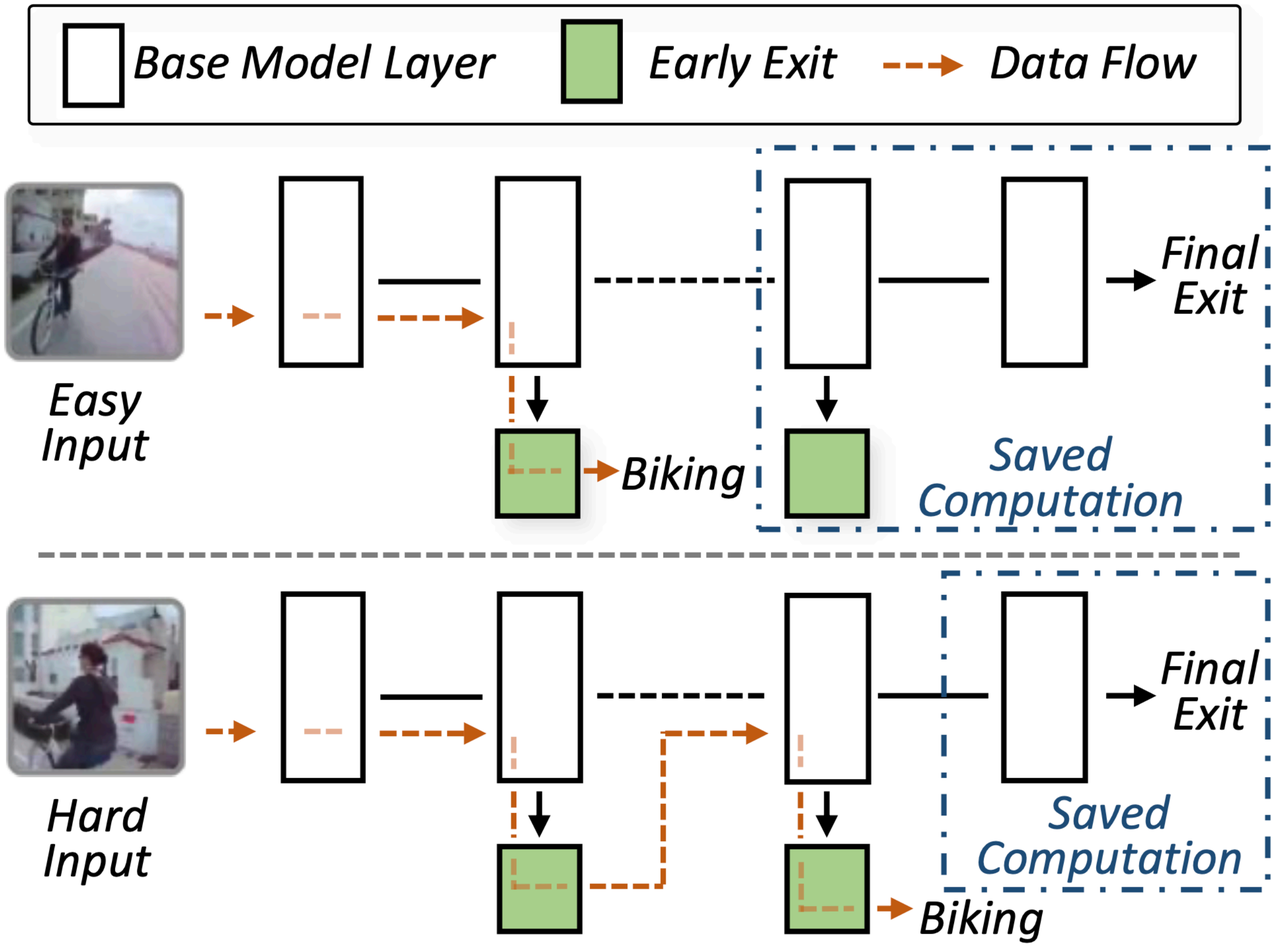
Some video frames are easier to recognize than others.

Easier

Harder

The Idea

Allow the DNN to adapt to the input by adding early exits.



Related Work

EdgeML (2021)
Chameleon (2018)
MCDNN (2016)
BranchyNet (2016)

Related Work

EdgeML (2021)

Chameleon (2018)

MCDNN (2016)

BranchyNet (2016)

[4] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems*, pages 742–751, 2017.

[5] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014.

[6] Charles Dubout and François Fleuret. Exact acceleration of linear object detectors. In *European Conference on Computer Vision*, pages 301–311. Springer, 2012.

[7] Biyi Fang, Xiao Zeng, Faen Zhang, Hui Xu, and Mi Zhang. FlexDNN: Input-Adaptive On-Device Deep Learning for Efficient Mobile Vision. In *ACM/IEEE Symposium on Edge Computing (SEC)*, 2020.

[8] Biyi Fang, Xiao Zeng, and Mi Zhang. Nestdnn: Resource-aware multi-tenant on-device deep learning for continuous mobile vision. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 115–127. ACM, 2018.

[9] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In *Advances in neural information processing systems*, pages 2962–2970, 2015.

[10] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International Conference on Machine Learning*, pages 1737–1746, 2015.

Related Work

EdgeML (2021)

Chameleon (2018)

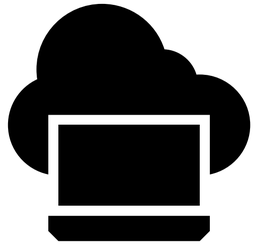
MCDNN (2016)

BranchyNet (2016)

- **Multiple Independent Model Variants**

- **Large Memory Footprint**

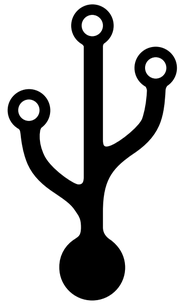
Related Work



EdgeML (2021)
Chameleon (2018)
MCDNN (2016)
BranchyNet (2016)

- **Cloud Assisted Processing**
- **Model Compression**
- **Catalog of Models**

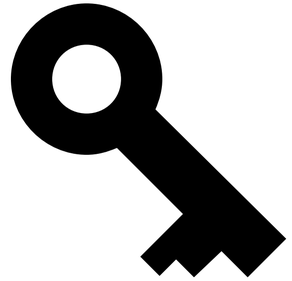
Related Work



EdgeML (2021)
Chameleon (2018)
MCDNN (2016)
BranchyNet (2016)

- **Single Model with Early Exits**
- **Execute on the device**

Key Contribution



FlexDNN finds an optimized answer to the questions:

- **How much compute should I spend checking early exits?**
- **When and where in the neural network should I check?**

Agenda

1. Introduction
2. Background and Motivation
3. FlexDNN Design
4. Evaluation
5. Related Work
6. Conclusion

Agenda

1. Introduction
2. **Background and Motivation**
3. **FlexDNN Design**
4. **Evaluation**
5. Related Work
6. Conclusion

Agenda

1. Introduction
2. **Background and Motivation**
3. FlexDNN Design
4. Evaluation
5. Related Work
6. Conclusion

Easier

Optimal Model is Smaller

Harder

Optimal Model is Larger

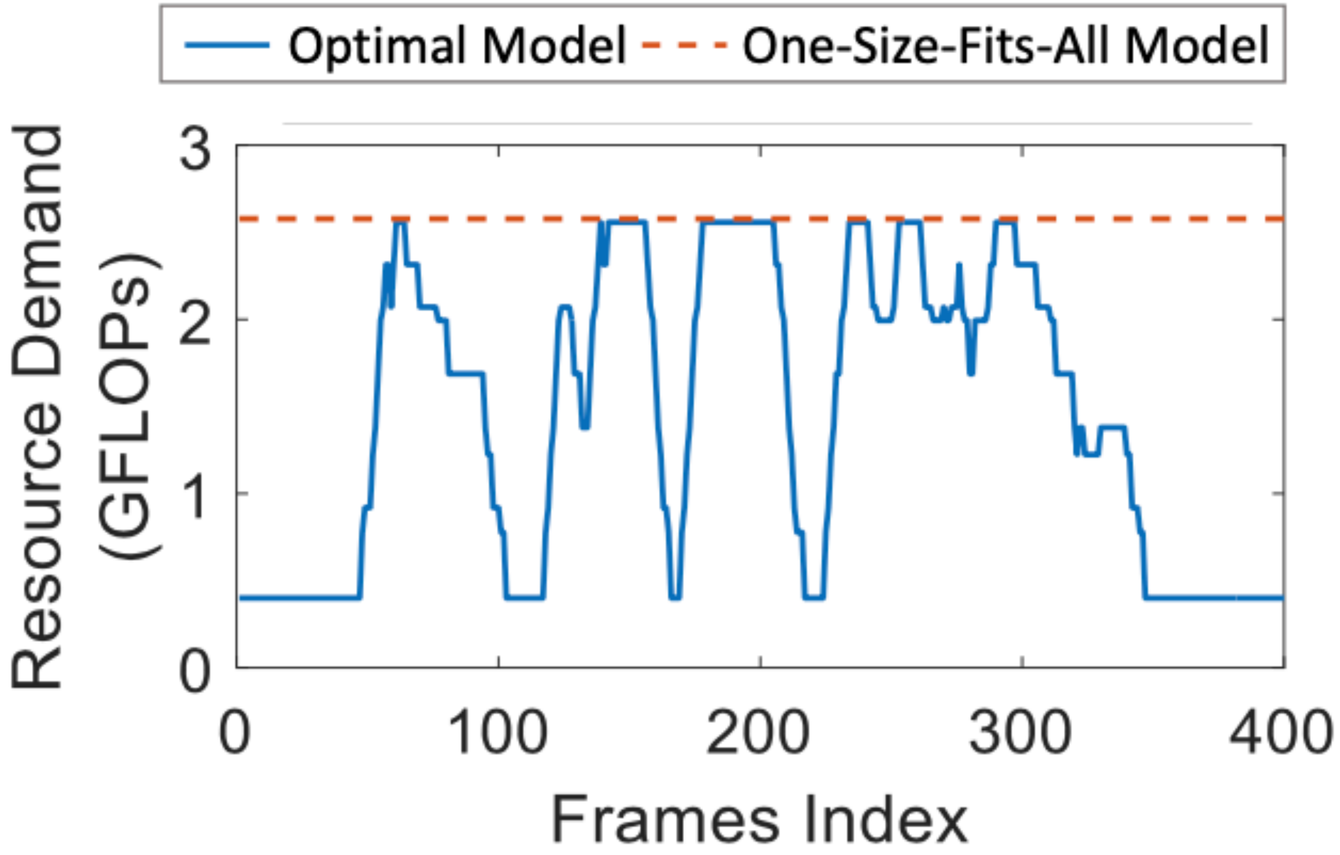
Harder

Optimal Model is Larger

Easier

Optimal Model is Smaller

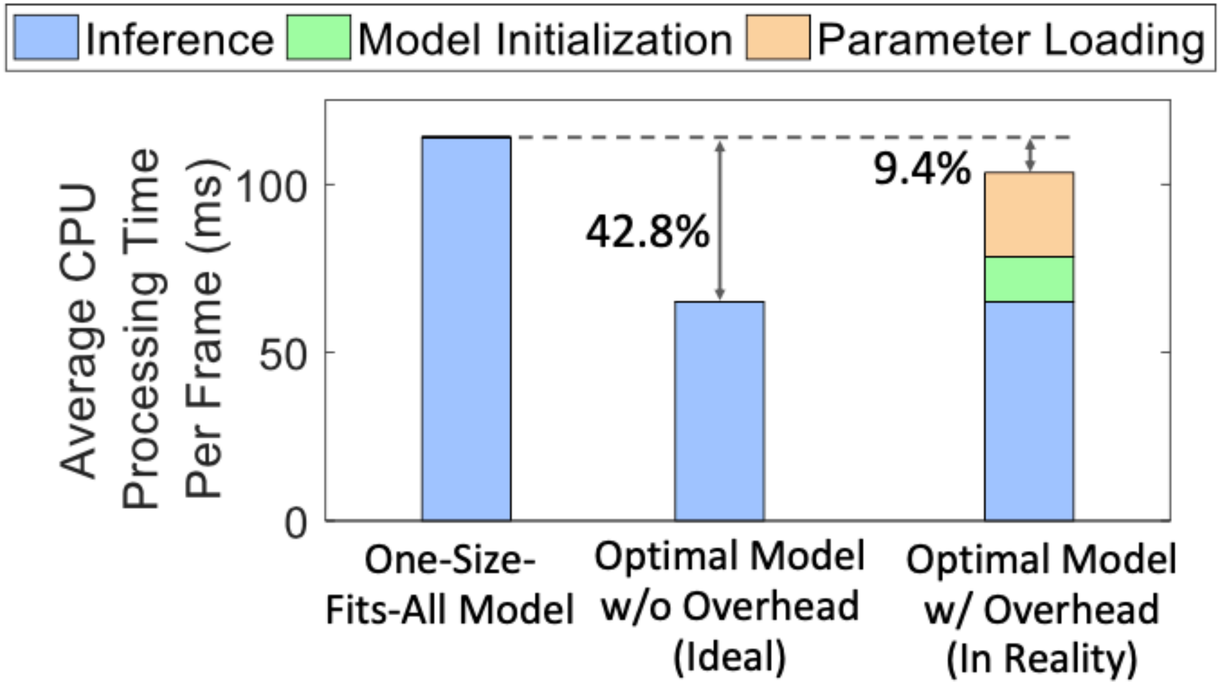
The Inefficiency



Bag-of-Models

Large Memory Footprint

Large Overhead from Parameter Loading and Model initialization



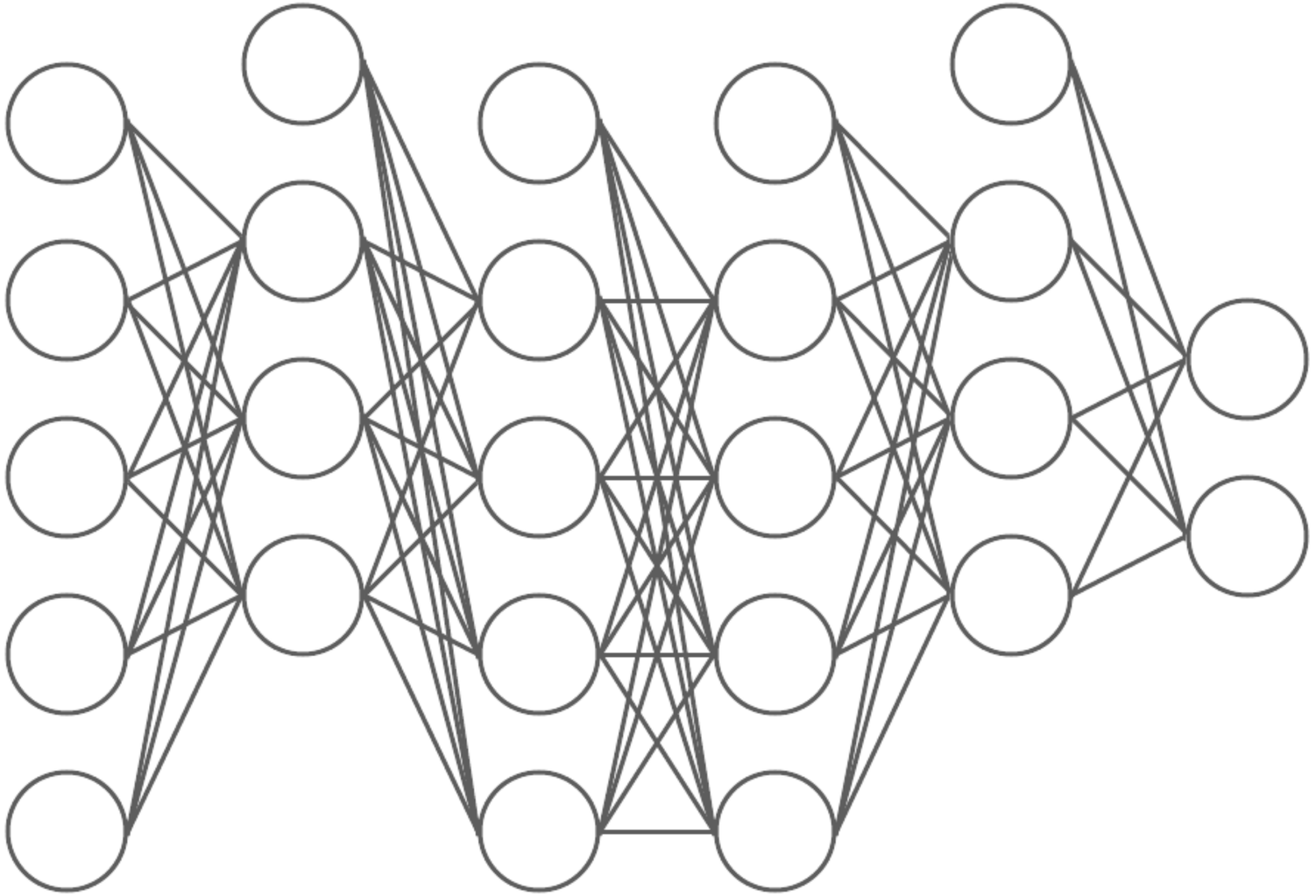
Agenda

1. Introduction
2. **Background and Motivation**
3. FlexDNN Design
4. Evaluation
5. Related Work
6. Conclusion

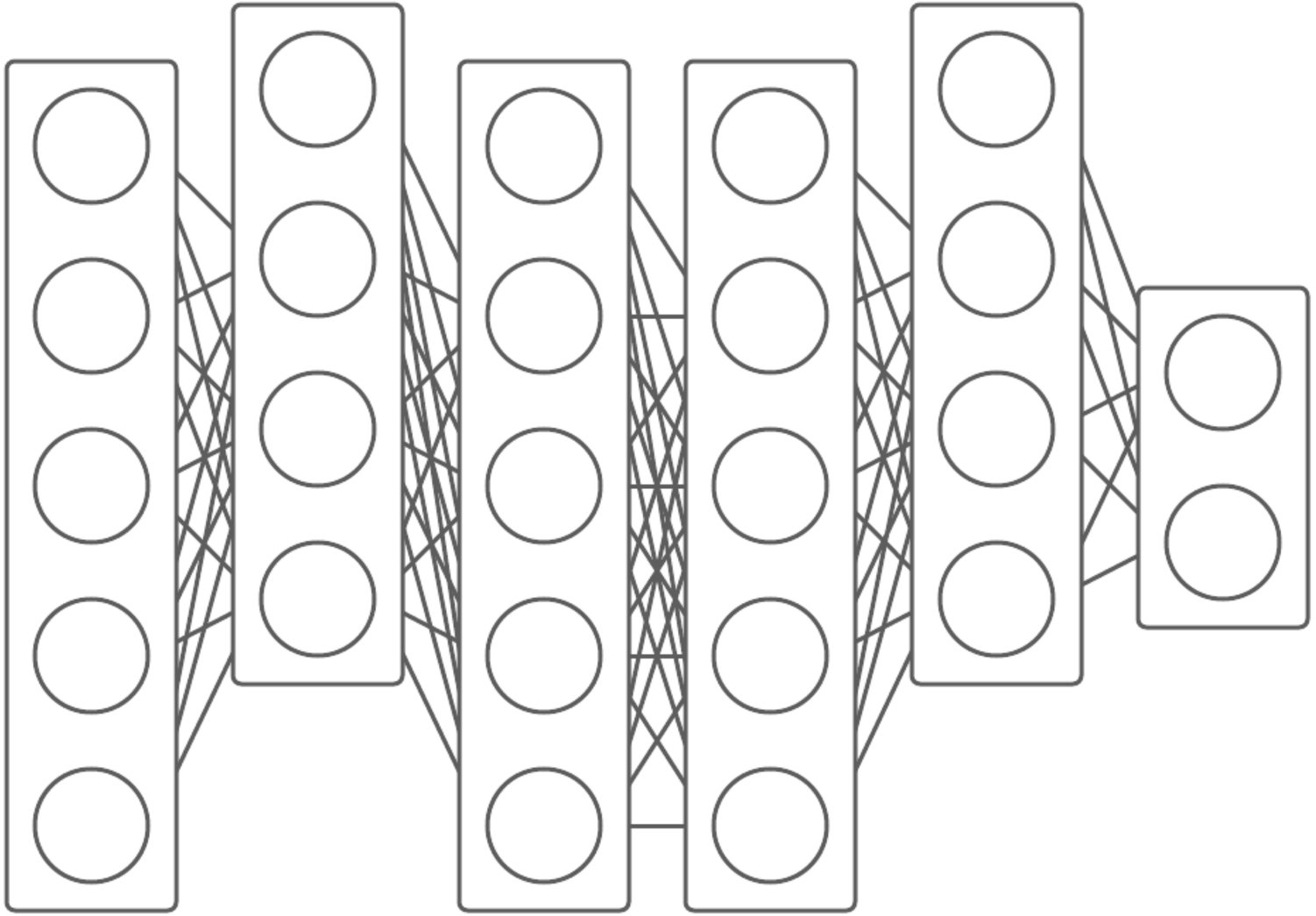
Agenda

1. Introduction
2. Background and Motivation
3. **FlexDNN Design**
4. Evaluation
5. Related Work
6. Conclusion

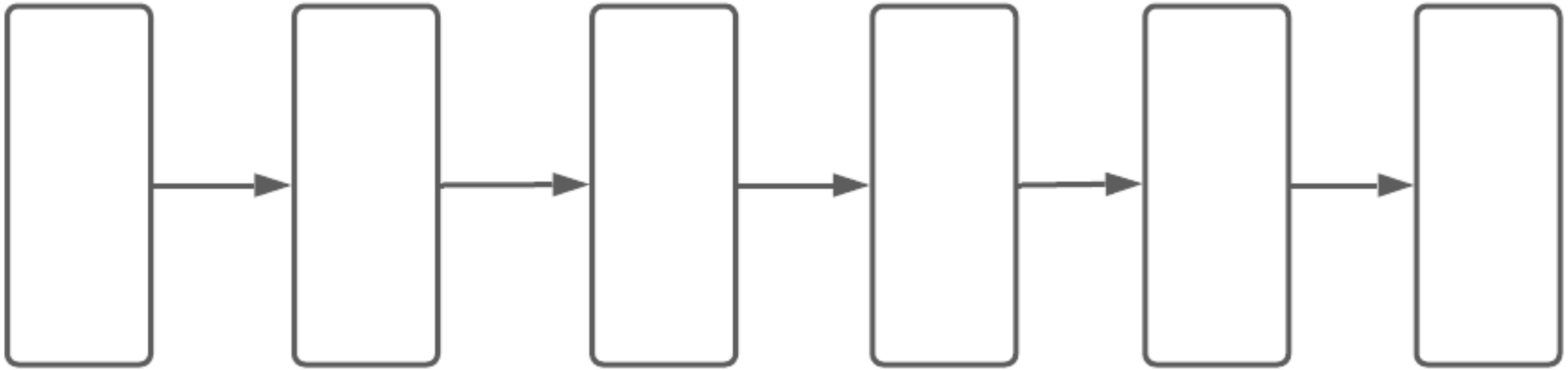
Architecture



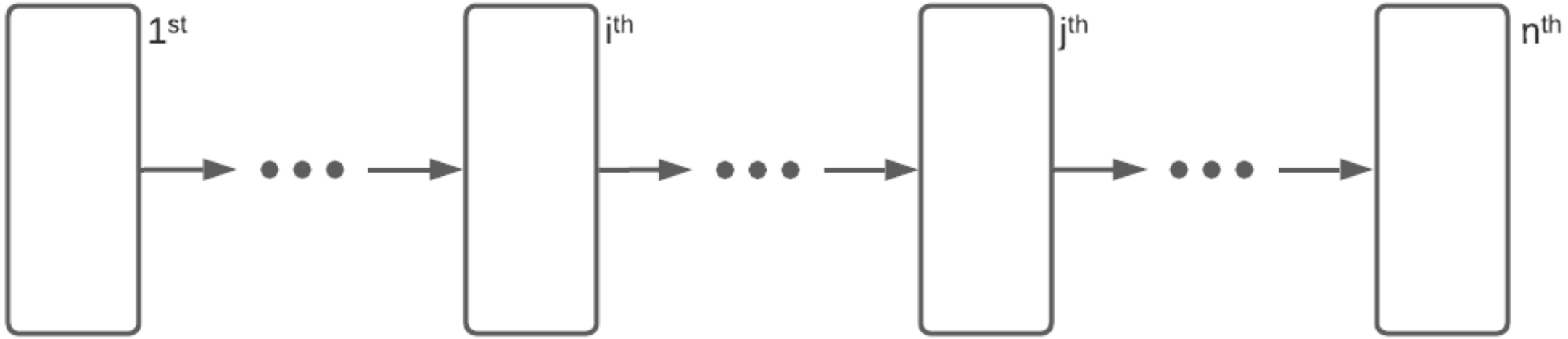
Architecture



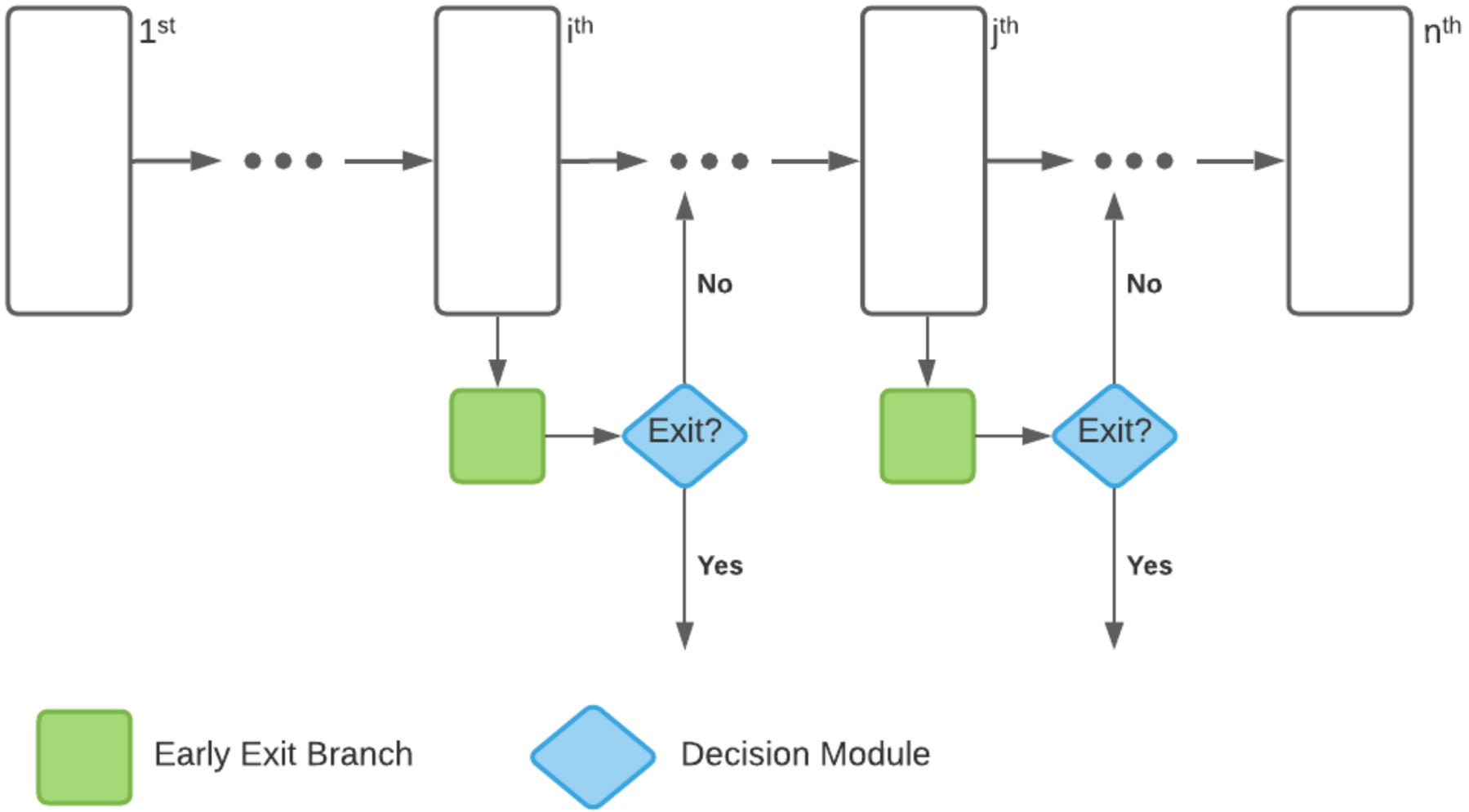
Architecture



Architecture

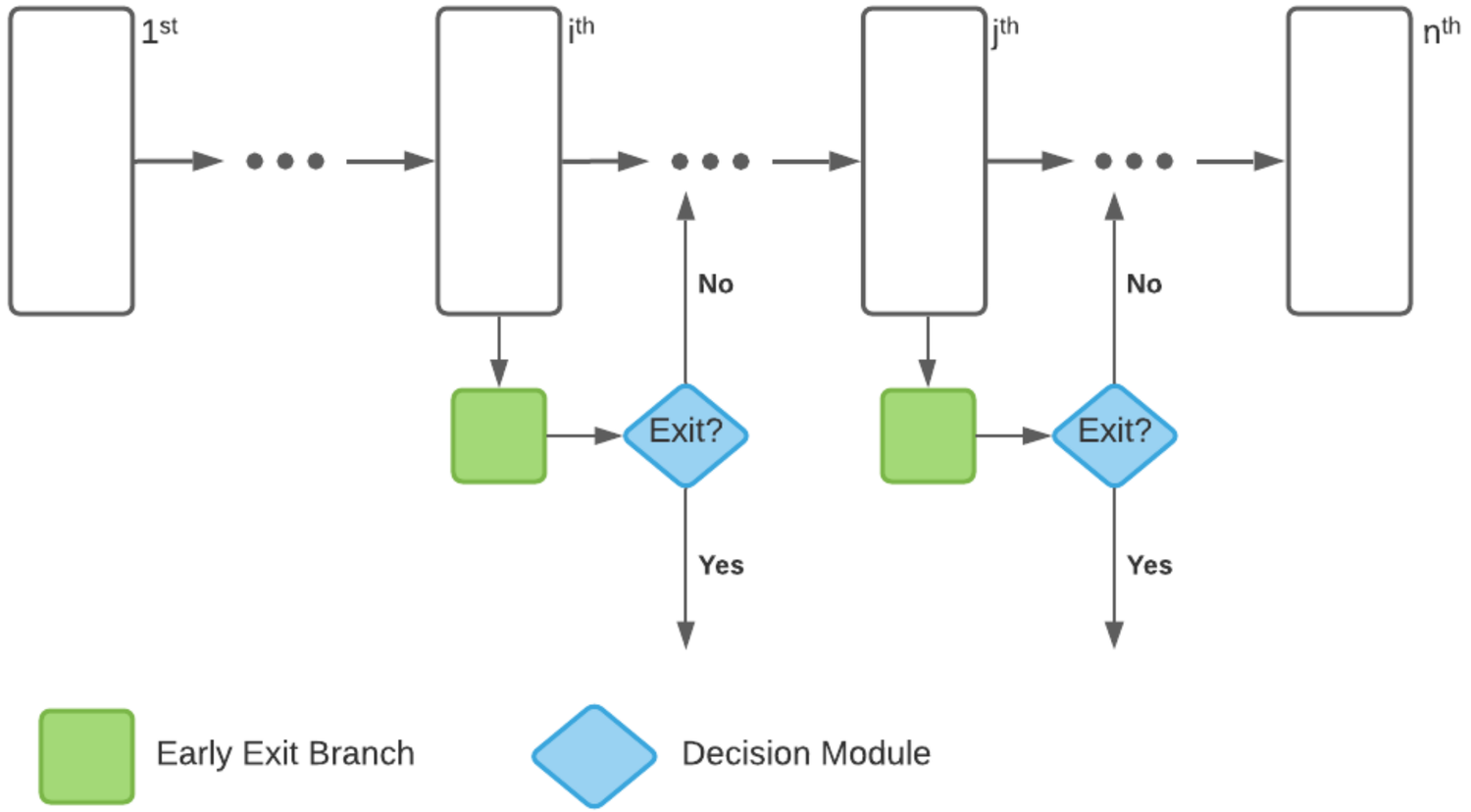


Architecture



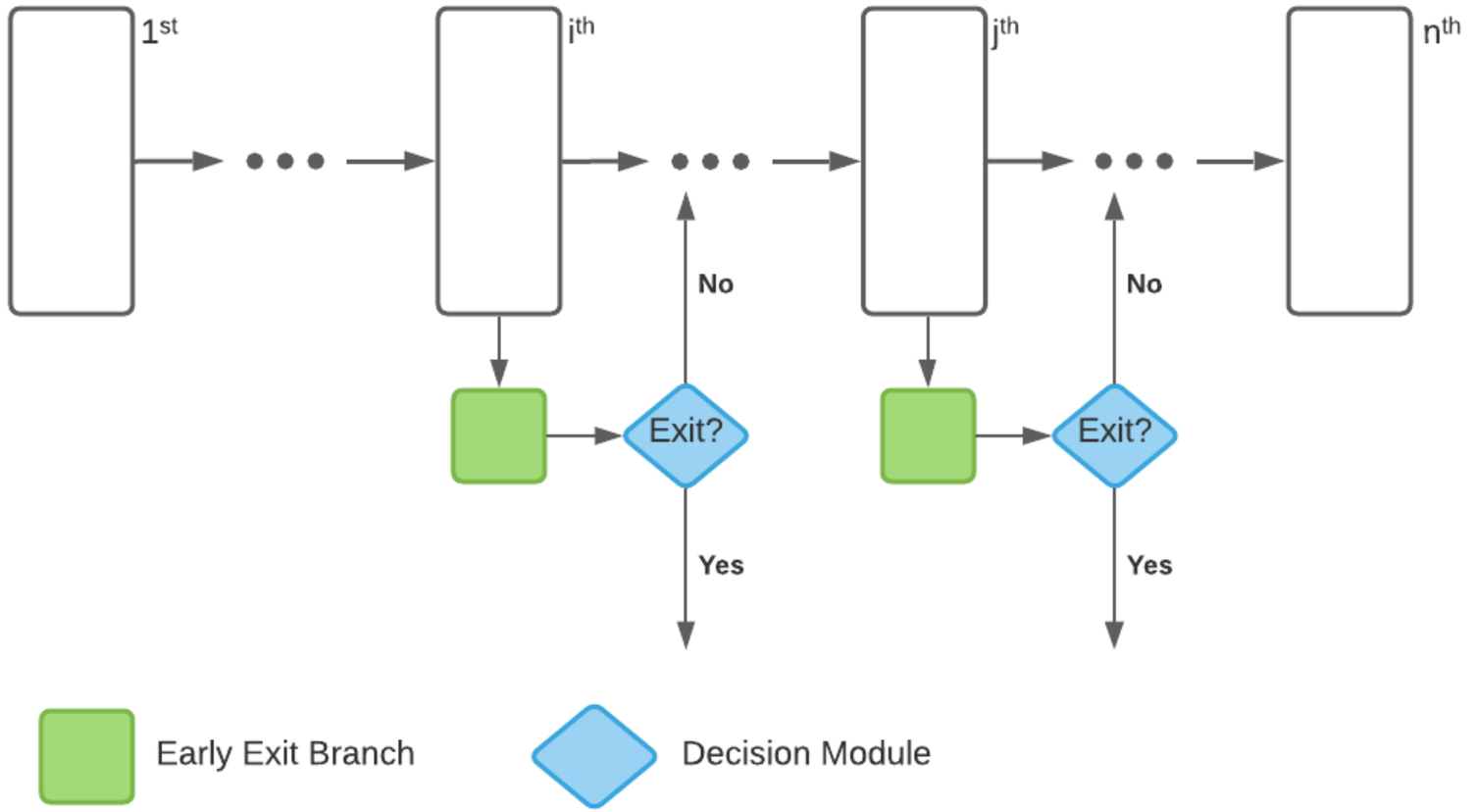
Architecture

Easier Image



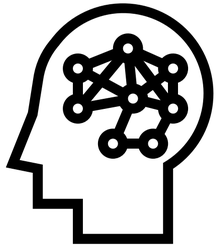
Architecture

Harder Image



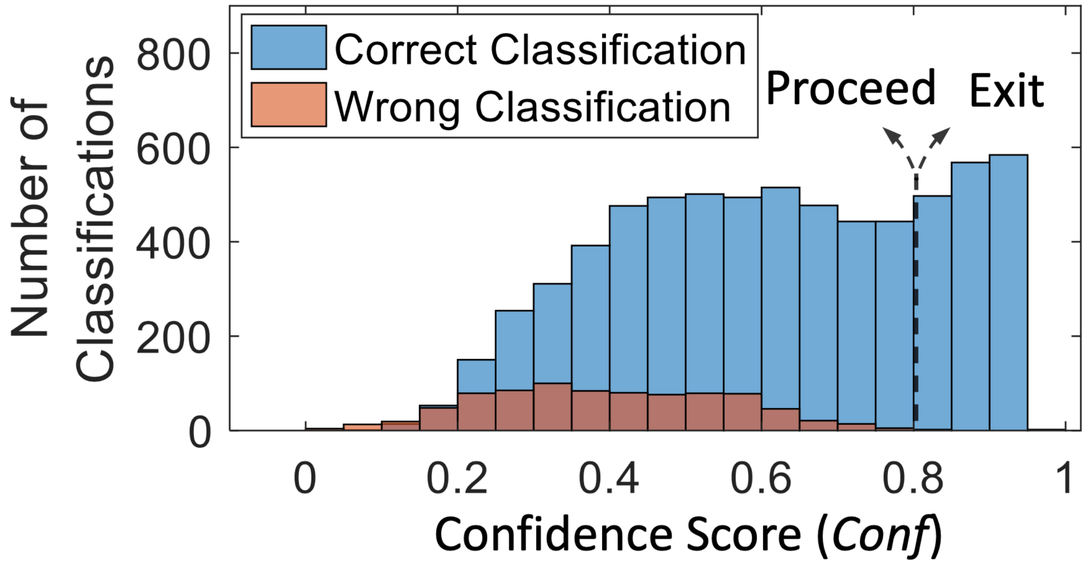
Early Exit Branch

Small Neural Network



Decision Module

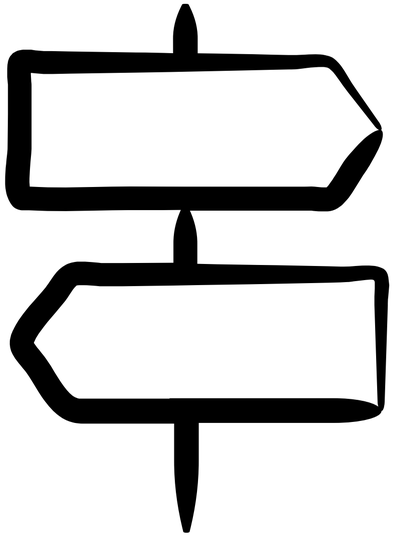
Confidence Threshold



A Dilemma

How big should the Early Exit Neural Network be?

Too **BIG** and we add significant overhead for relatively hard images.



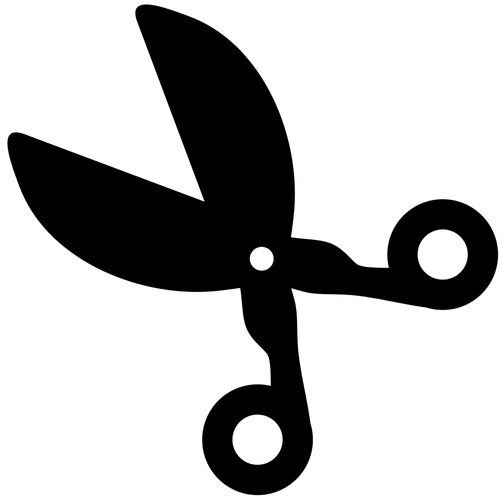
Too **small** and we cannot catch as many relatively easy images

An Answer

How big should the Early Exit Neural Network be?

Answer: Use an Architecture Search Schema and find out

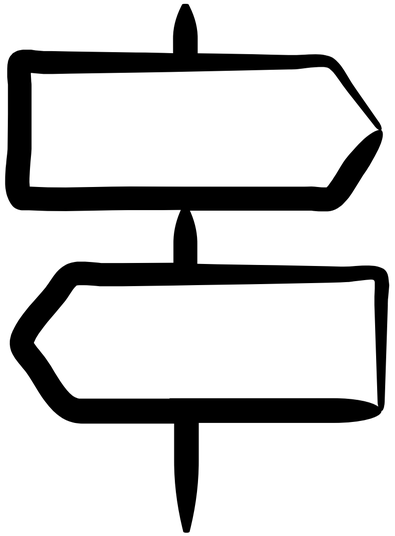
FlexDNN prunes an over-parameterized network until the optimal network architecture is found



A Dilemma

When and where should the Early Exits Be?

Too Many and you incur large overheads without significant benefit.

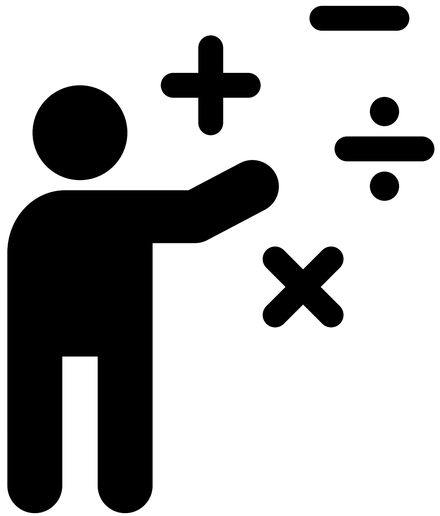


Too few and you miss opportunities for performance improvement.

An Answer

When and where should the Early Exits Be?

Answer: They should not be inserted when the overhead is greater than the benefit.

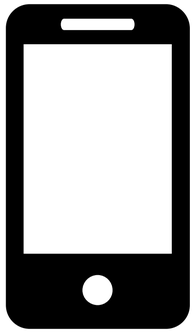


Agenda

1. Introduction
2. Background and Motivation
3. **FlexDNN Design**
4. Evaluation
5. Related Work
6. Conclusion

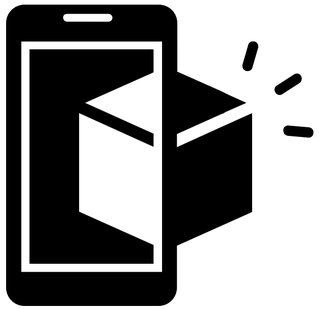
Agenda

1. Introduction
2. Background and Motivation
3. FlexDNN Design
4. **Evaluation**
5. Related Work
6. Conclusion



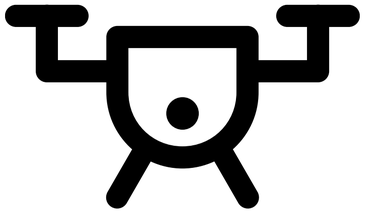
Activity Detection on Mobile Phone

UCF-101



Scene Understanding for Mobile Augmented Reality.

Place-8



Drone-based Traffic Surveillance

VeDrone

Model Evaluation

High Early Exit Rate without Accuracy Loss

Compact Memory Footprint

Computation-Efficient Early Exits

High Computational Consumption Reduction

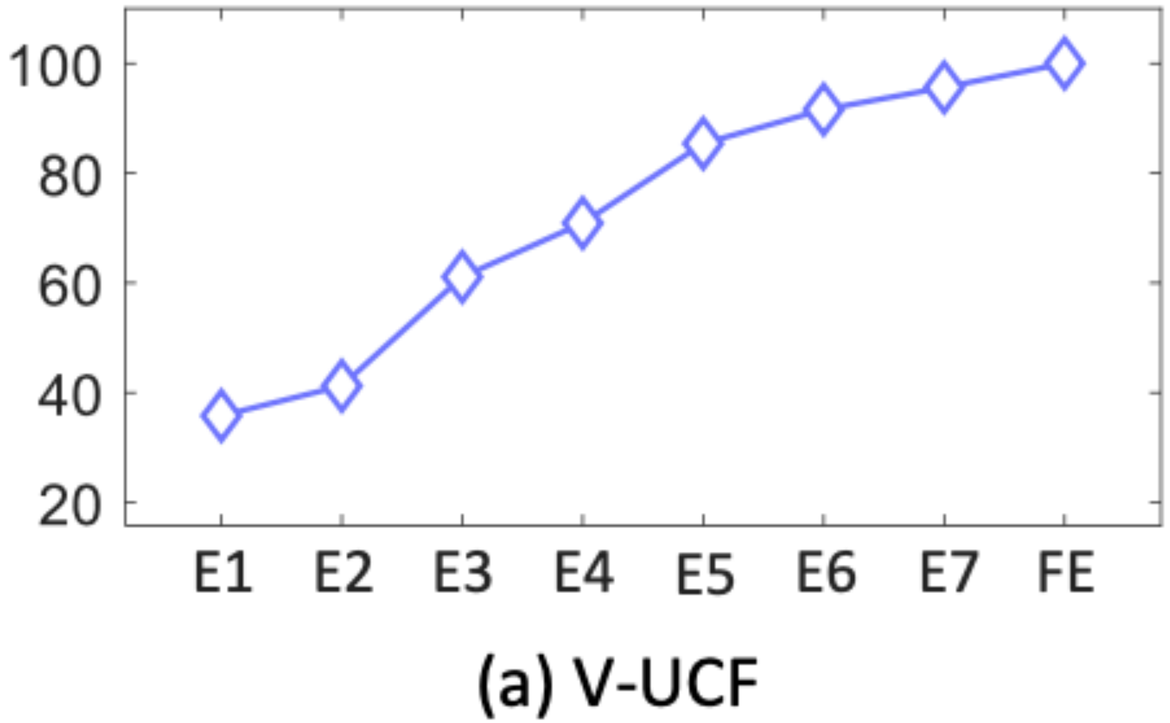
Model Evaluation

High Early Exit Rate without Accuracy Loss

Compact Memory Footprint

Computation-Efficient Early Exits

High Computational Consumption Reduction



(a) V-UCF

Model Evaluation

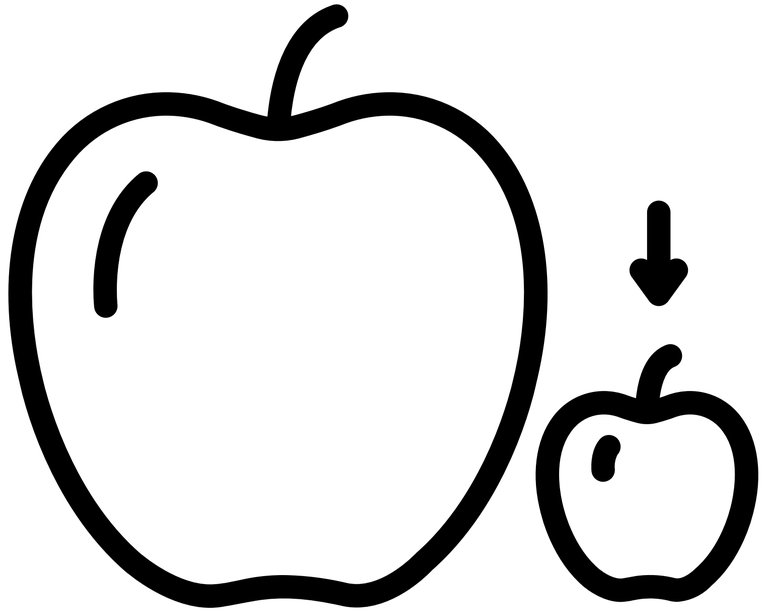
High Early Exit Rate without Accuracy Loss

Compact Memory Footprint

Computation-Efficient Early Exits

High Computational Consumption Reduction

Bag-of-models



FlexDNN

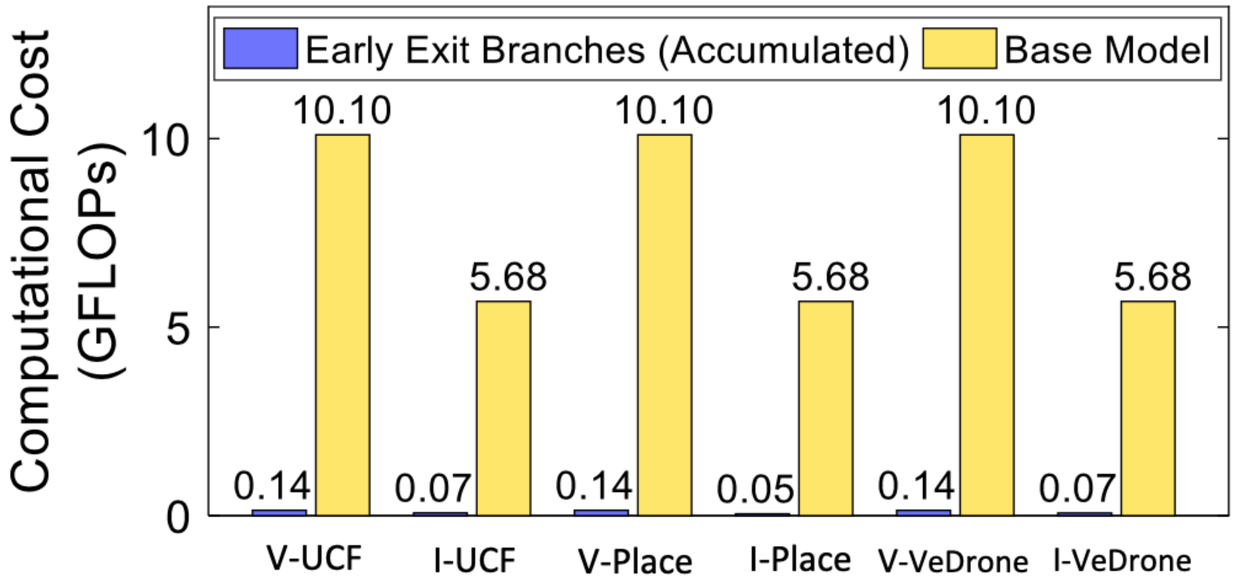
Model Evaluation

High Early Exit Rate without Accuracy Loss

Compact Memory Footprint

Computation-Efficient Early Exits

High Computational Consumption Reduction



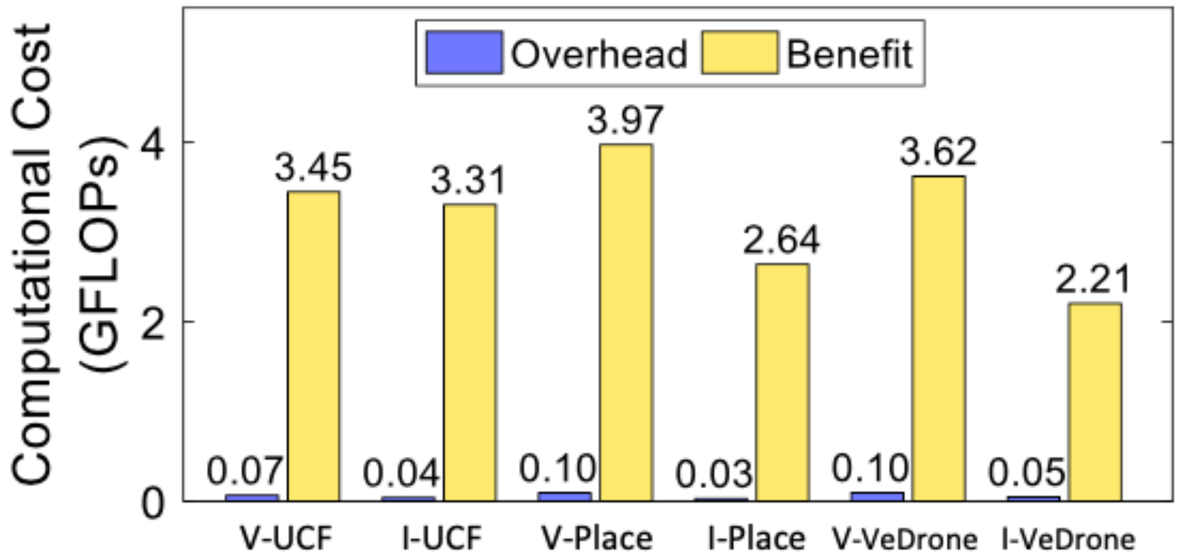
Model Evaluation

High Early Exit Rate without Accuracy Loss

Compact Memory Footprint

Computation-Efficient Early Exits

High Computational Consumption Reduction



Runtime Evaluation

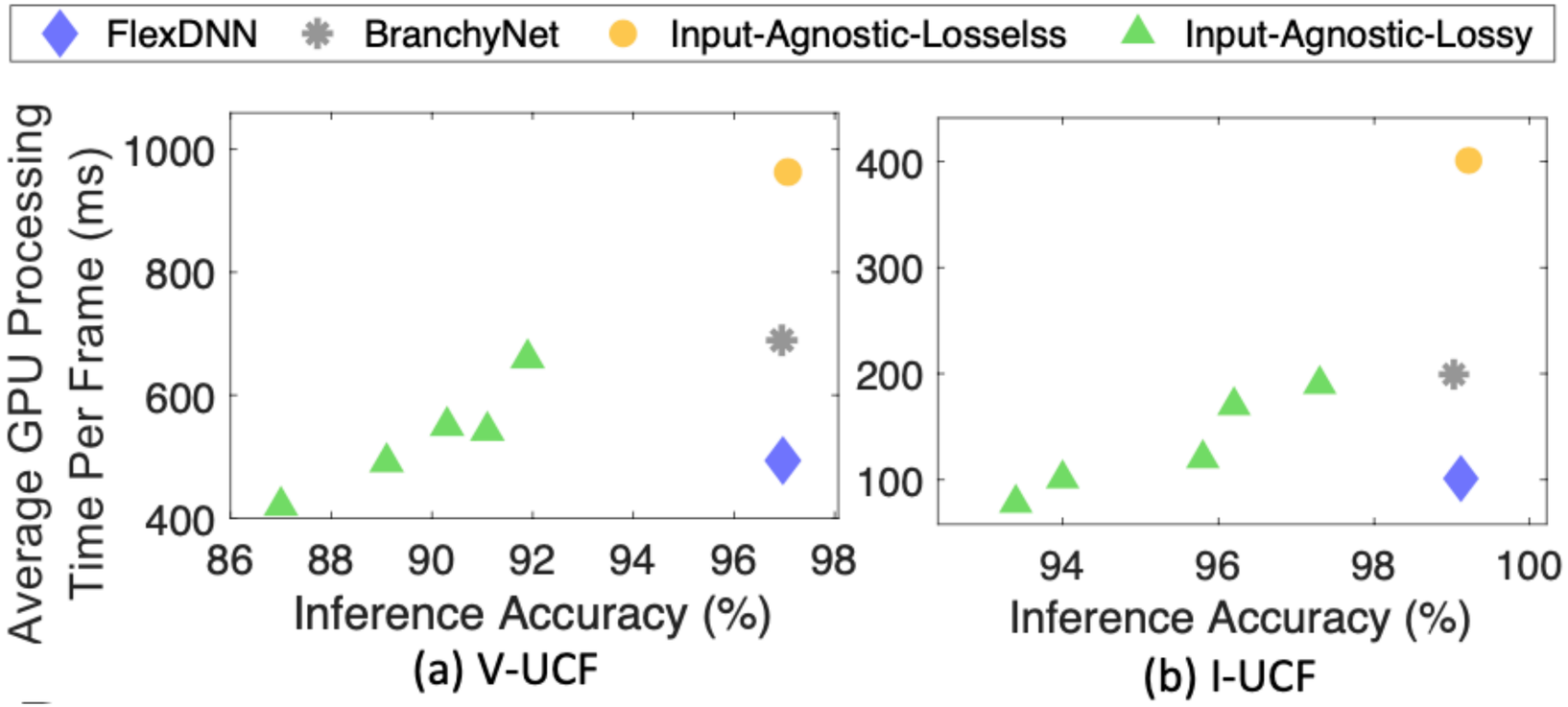
Accuracy and Compute

Frame Drop Rate

Runtime Evaluation

Accuracy and Compute

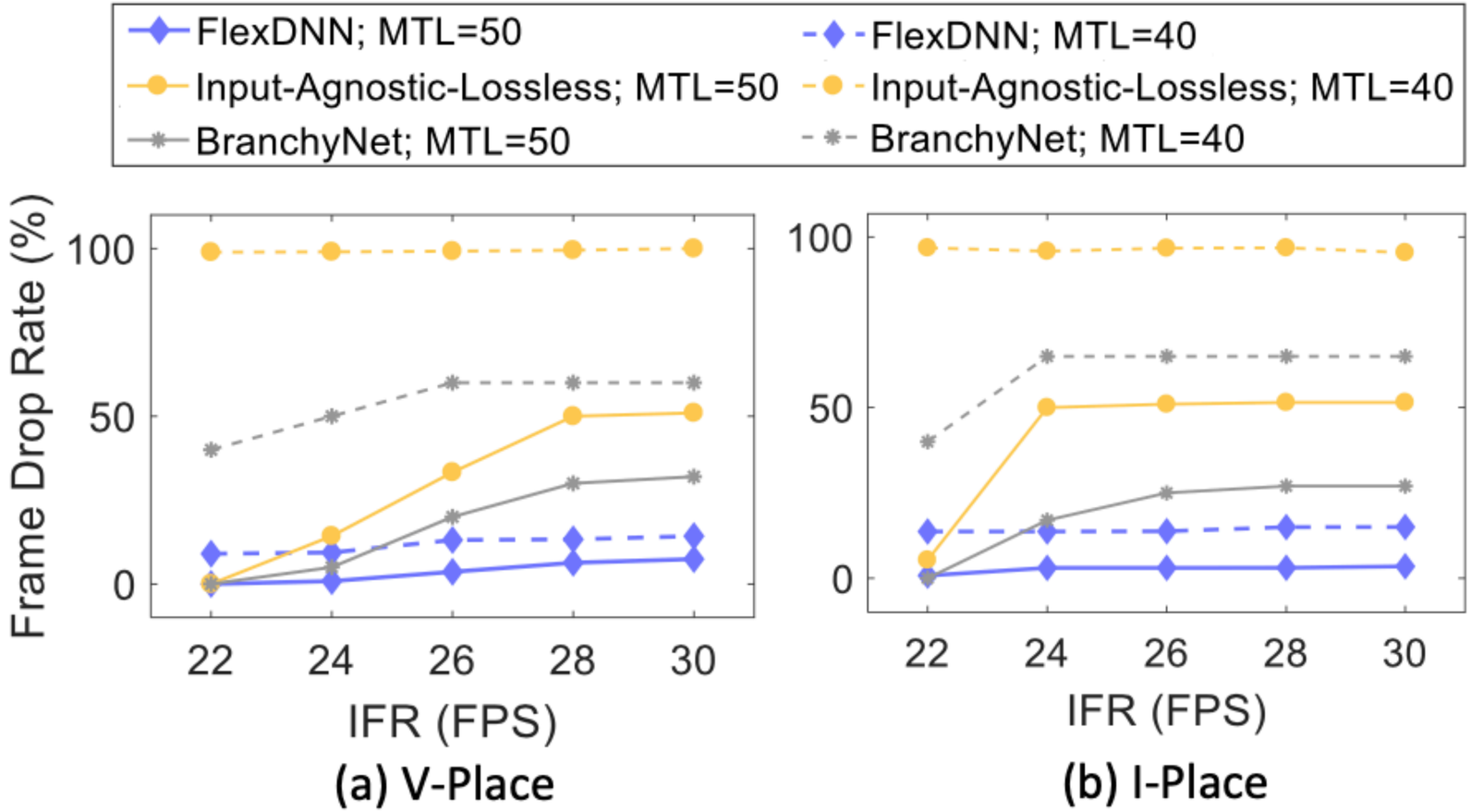
Frame Drop Rate



Runtime Evaluation

Accuracy and Compute

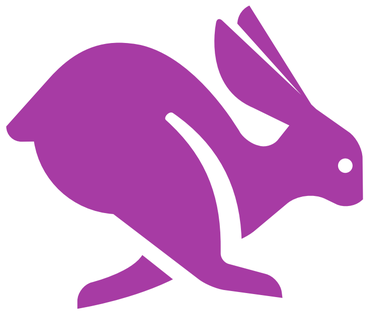
Frame Drop Rate

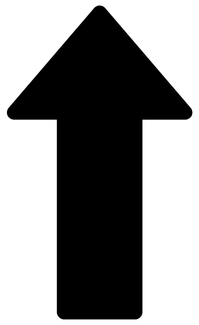


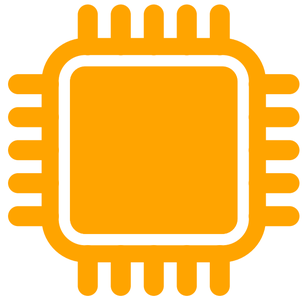
Agenda

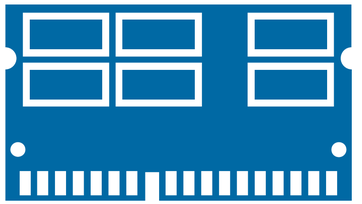
1. Introduction
2. Background and Motivation
3. FlexDNN Design
4. **Evaluation**
5. Related Work
6. Conclusion

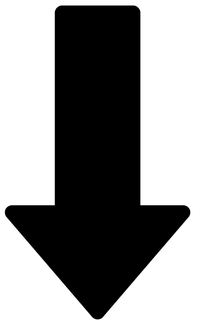
The Tradeoff



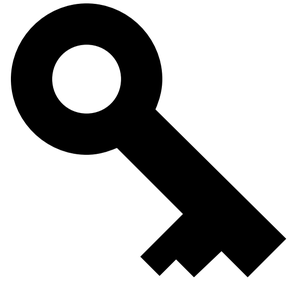








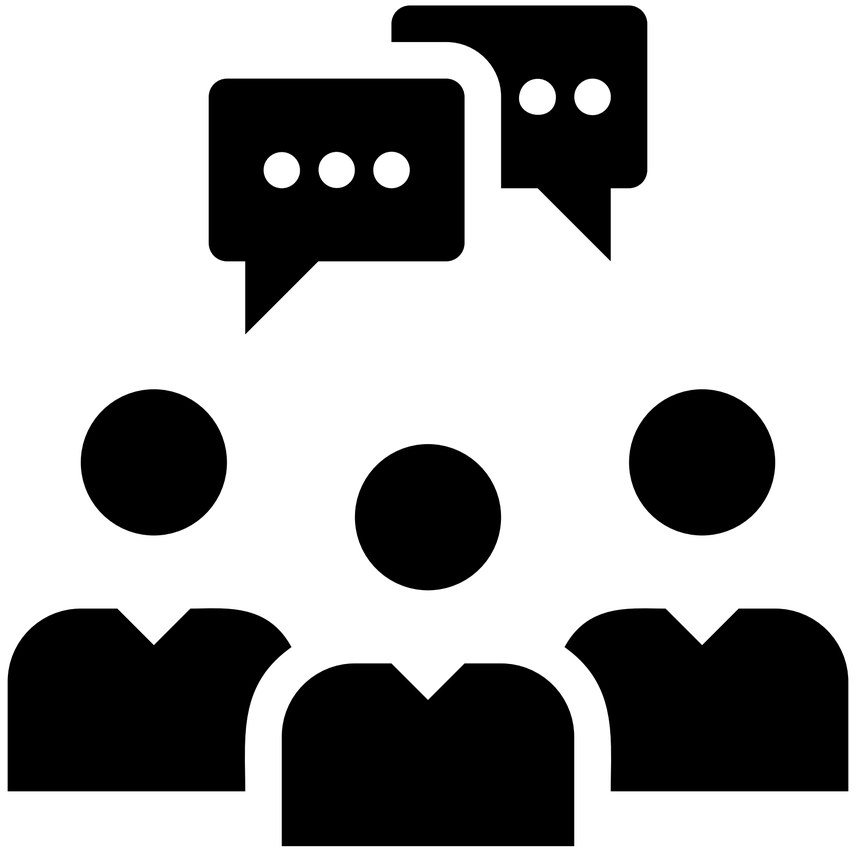
Key Contribution



FlexDNN finds an optimized answer to the questions:

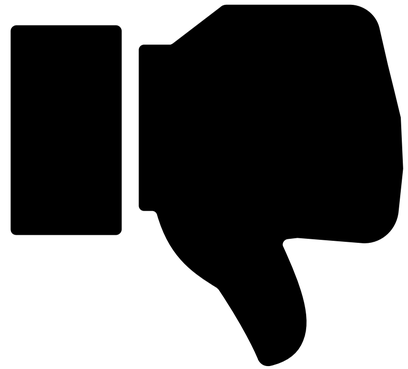
- **How much compute should I spend checking early exits?**
- **When and where in the neural network should I check?**

Discussion



- **Quantified the Problem.** Gathered data to show the inefficiencies.
- **Broadly Tested.** Multiple models and datasets.
- **User Friendly.** You don’t need to be a DNN wizard to use this.

- **Narrowly Applicable.** Not inherently bad, just less interesting.
- **Created own small datasets.** Less trustworthy than larger datasets.



Any Questions?

What other domains could benefit from this technique? They focused exclusively on video processing…

How long do we think this technique will remain relevant?