Lehrstuhl für Rechnerarchitektur und Parallel Systeme
TUM School of Computation, Information and Technology
Technische Universität München

EdgeLessPart: Distributed and Progressive Inference for the Edge-Cloud Continuum

Isaac David Núñez Araya

Michael Gerndt

Mohak Chadha

CODECO
Cognitive Decentralised
Edge Cloud Orchestration

EDGELESS

ML SysOps

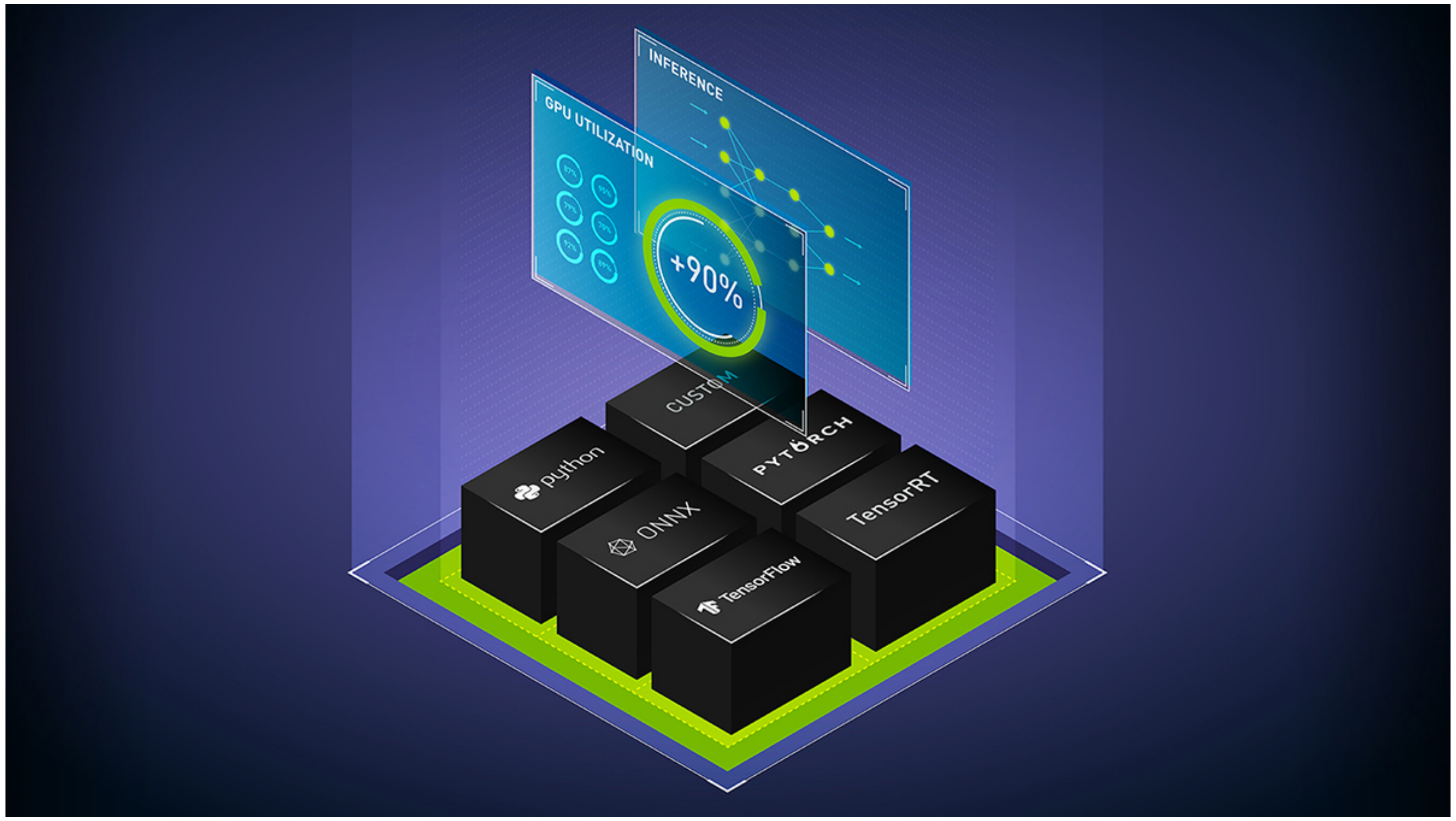
HiPEAC

Today’s Agenda

1. Early Exits and Split Computing
2. Our focus
3. EdgeLessPart
4. Some results
5. Conclusions and Future Work

Towards Distributed Machine Learning

- As the field of Machine Learning changes, it taps into the computational capabilities of heterogeneous devices.
- For the Cloud, it has experienced a fast adaptation of GPUs/TPUs



NVIDIA's Cloud Computing Solutions [1]

Towards Distributed Machine Learning

- As the field of Machine Learning changes, it taps into the computational capabilities of heterogeneous devices.

- For the Cloud, it has experienced a fast adaptation of GPUs/TPUs

- Yet, computation becoming ubiquitous has motivated to push them closer to users, near the necessary data
  - Aiming to reduce latency and the risks of overloading servers

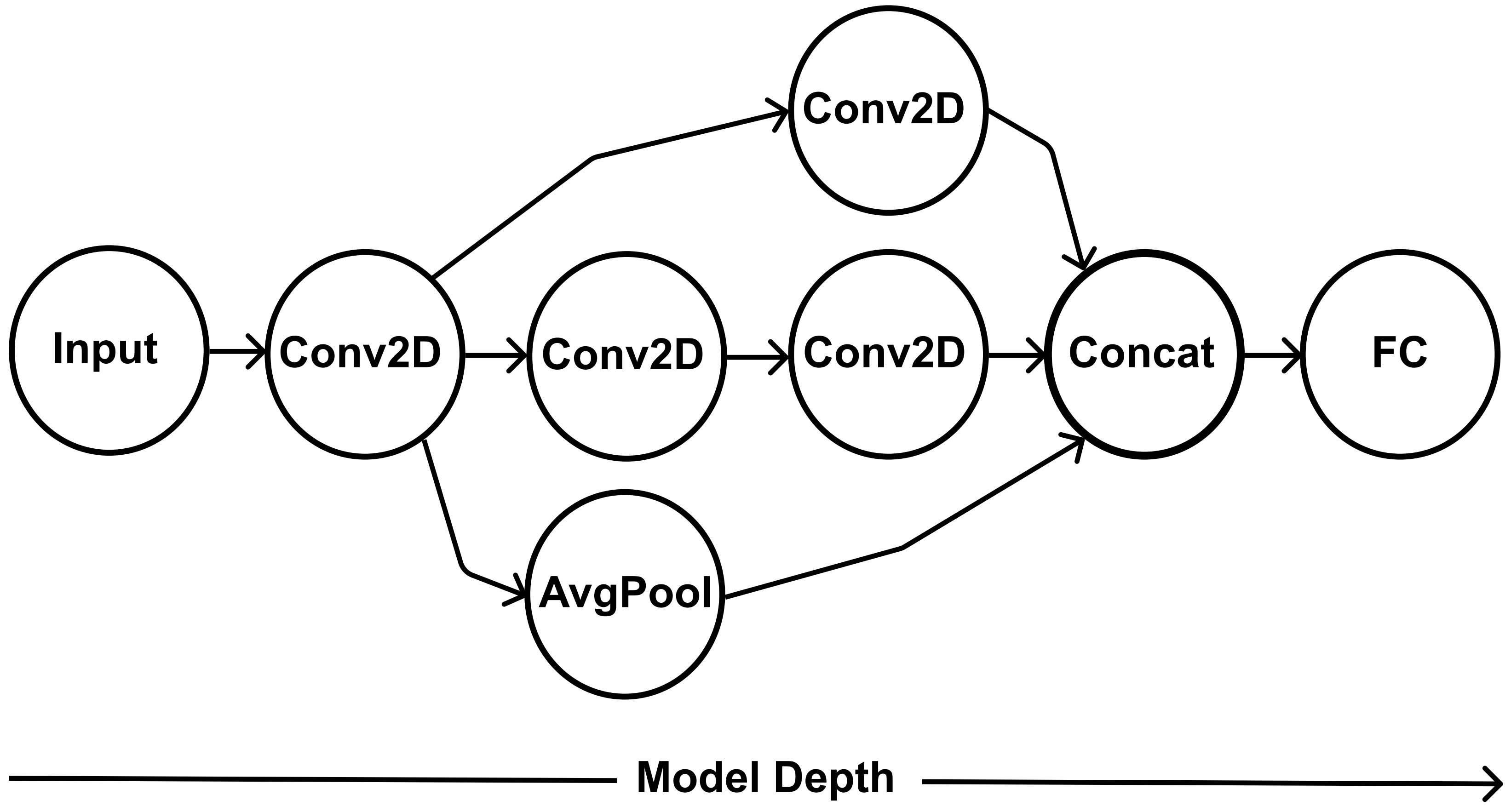
The challenge at the Edge

- We focus on resource-constrained devices in terms of:
  - Computational capabilities
  - Power limitations
    - It also includes passively cooled devices

- On the other hand, Deep Neural Networks (our target field) tend to require more than what is available on such devices. For example:
  - Raspberry Pi, NVIDIA Jetson Family, or similar
  - Small form factor PCs, e.g., Intel NUC and similar

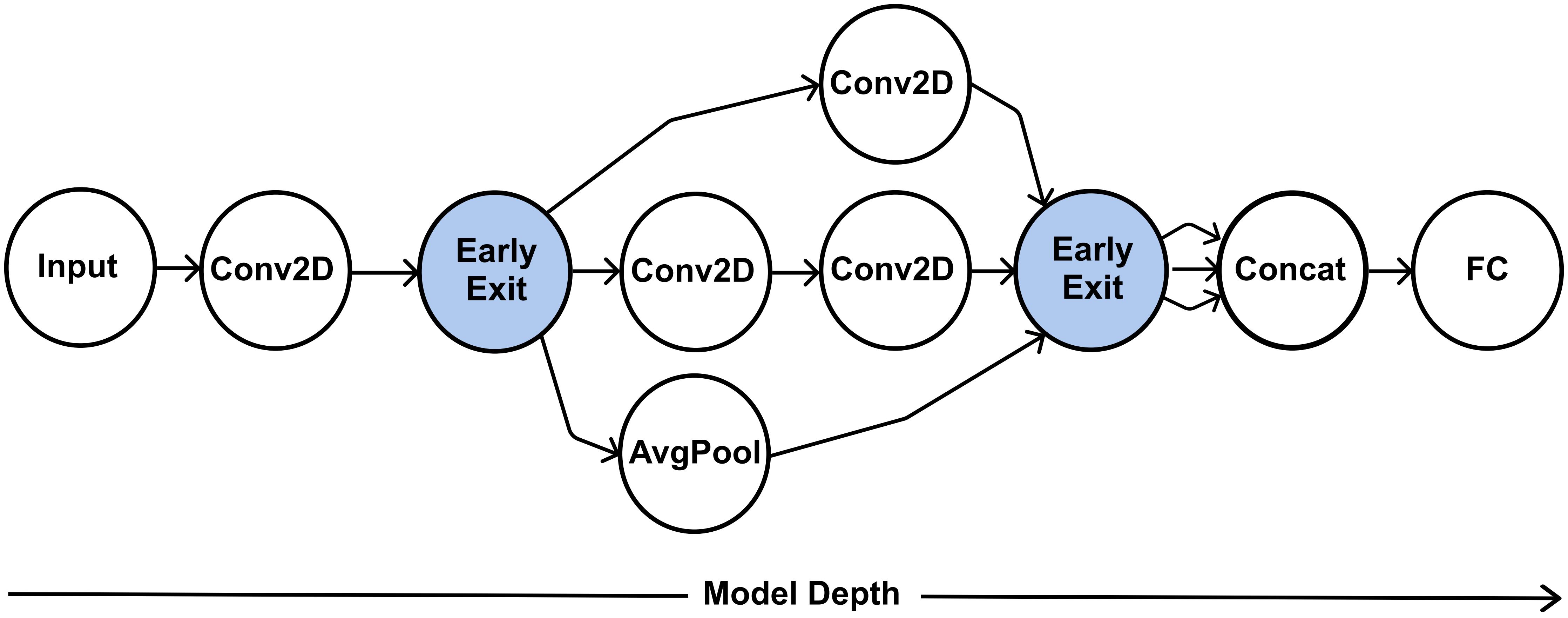
The opportunities for DNN optimization

- Early Exits



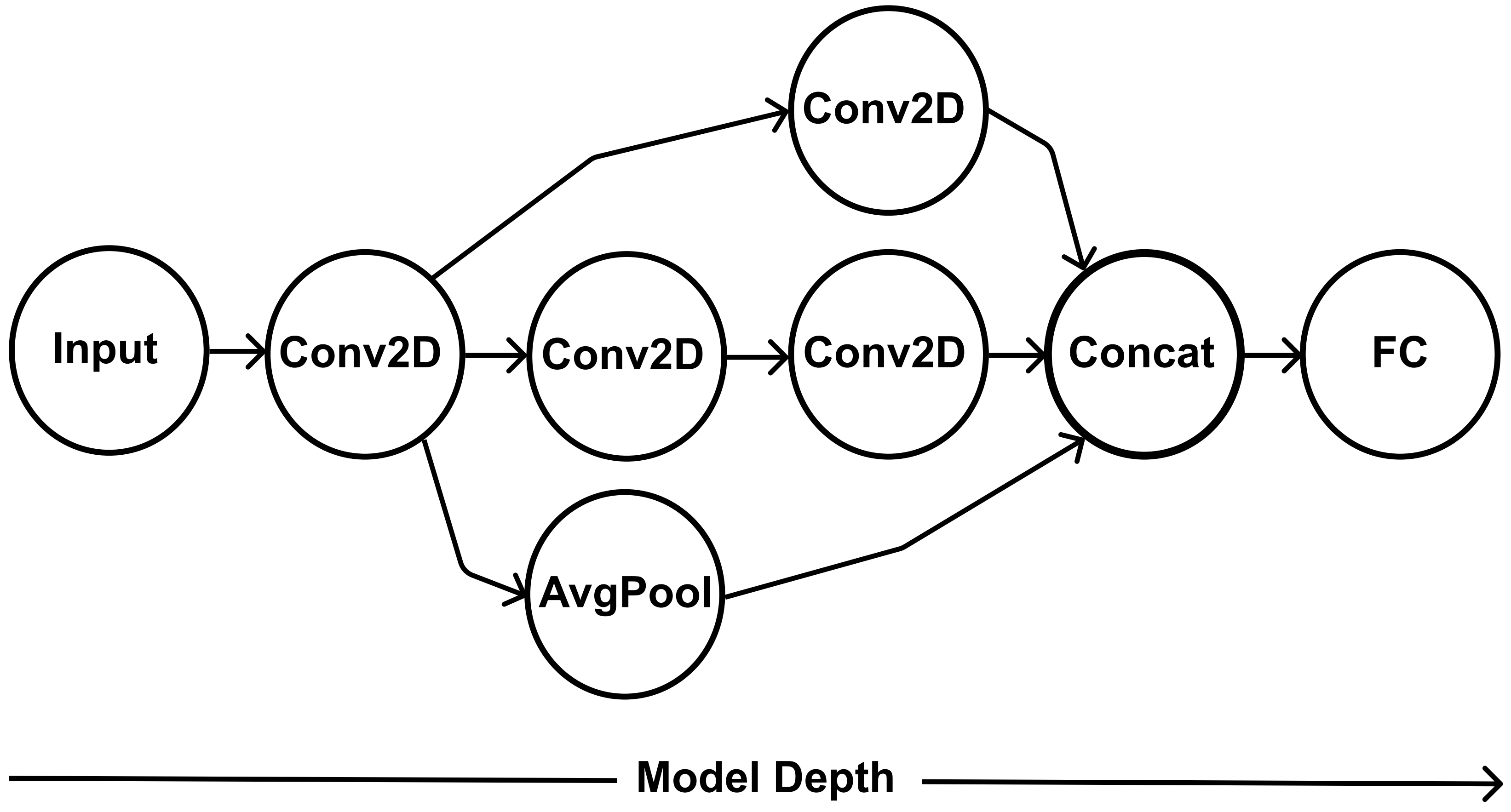
The opportunities for DNN optimization

- Early Exits



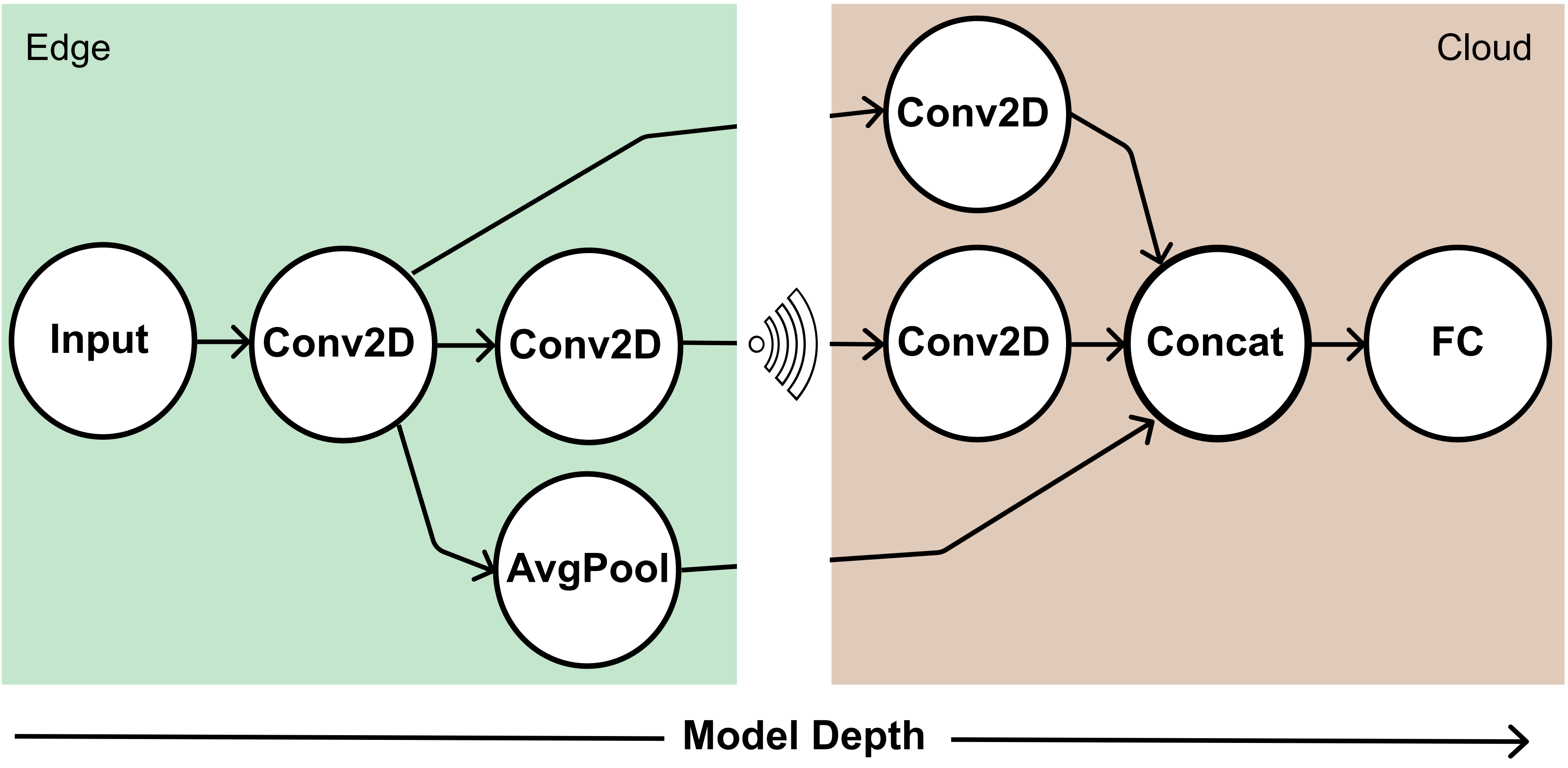
The opportunities for DNN optimization

- Split Computing



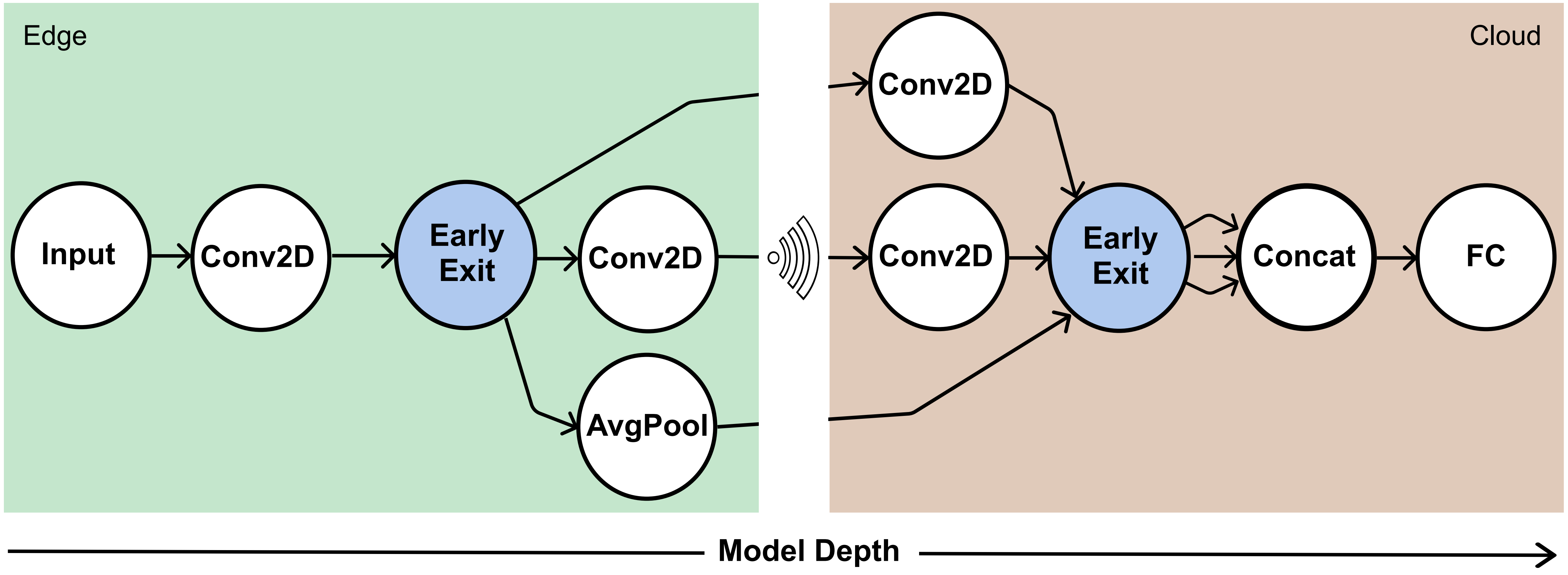
The opportunities for DNN optimization

- Split Computing



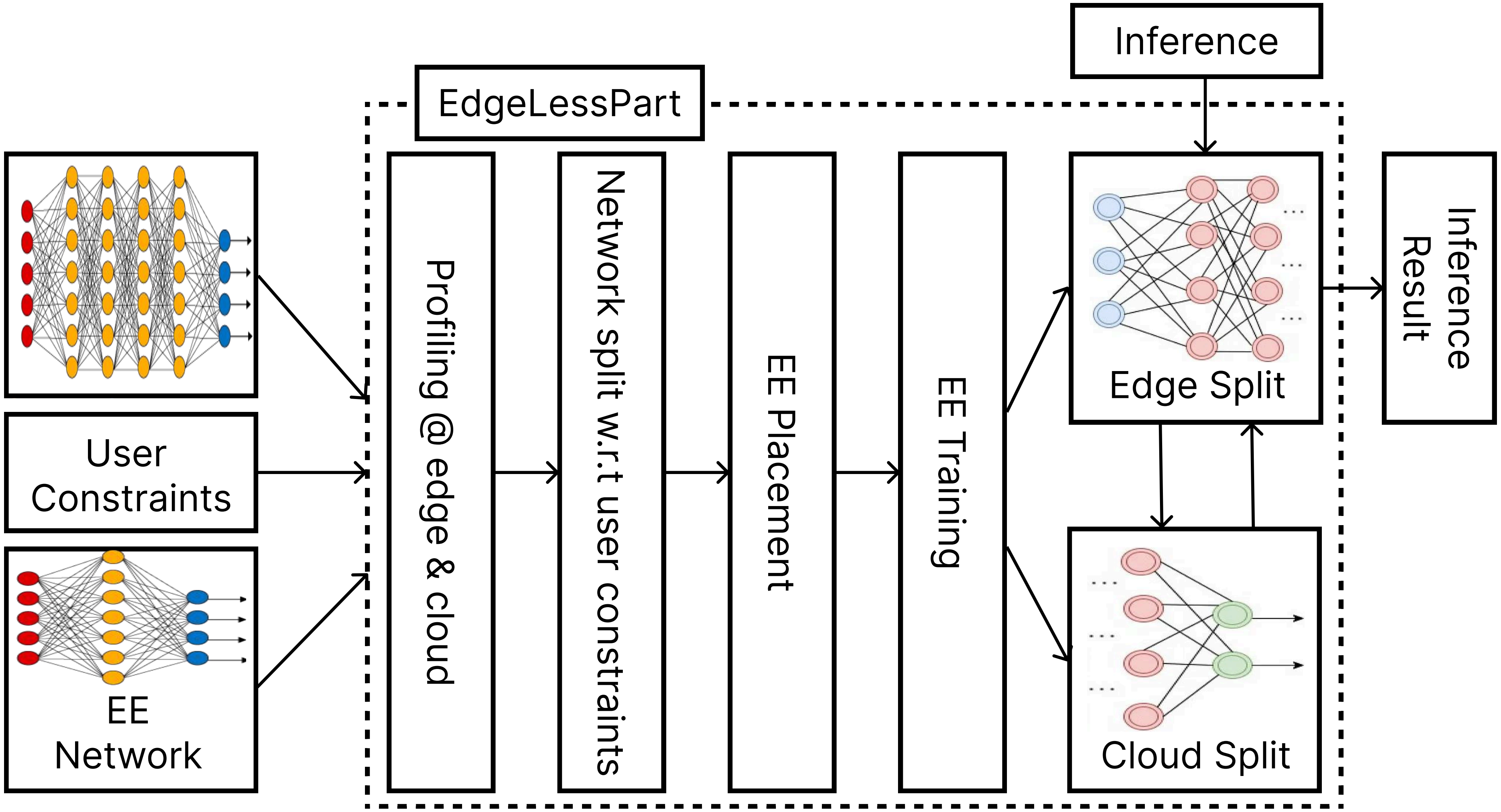
The opportunities for DNN optimization

- Split Computing



EdgeLessPart

- Still open questions:
  - How to select the exit and split points?
  - How to dynamically insert exits?



Profiling

- Measures runtime characteristics of the target devices
- Gathers memory usage and inference latency

Split Point Selection

- Combines profiling data and user constraints to balance the partitions at the edge and cloud
- The constraints are based on **latency** or **DNN memory size.**

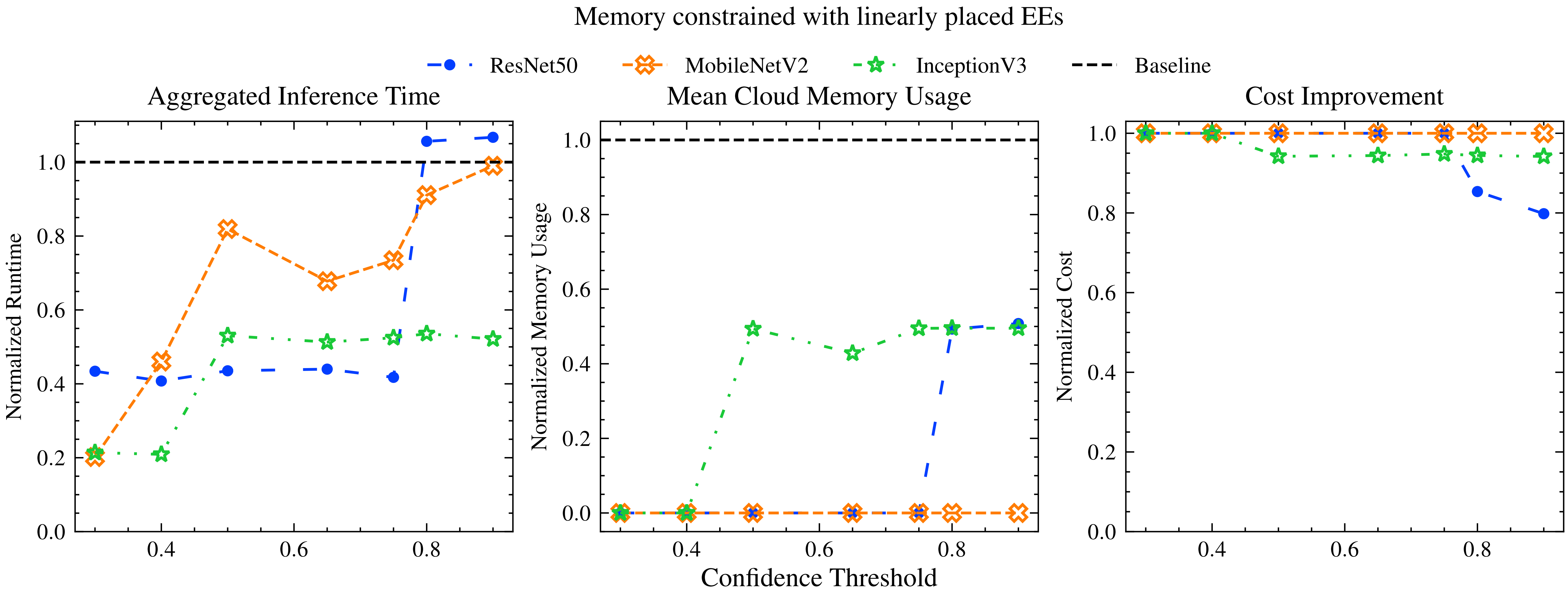
Early Exits with EdgeLessPart

- Once the split point has been determined, EdgeLessPart uses it to limit the exit point search
- For the candidate exit points, it inserts custom (and small) DNN to serve as Early Exits
  - These DNNs are given as part of user constraints.
- If the model has been pre-trained, EdgeLessPart will only tune the new layers

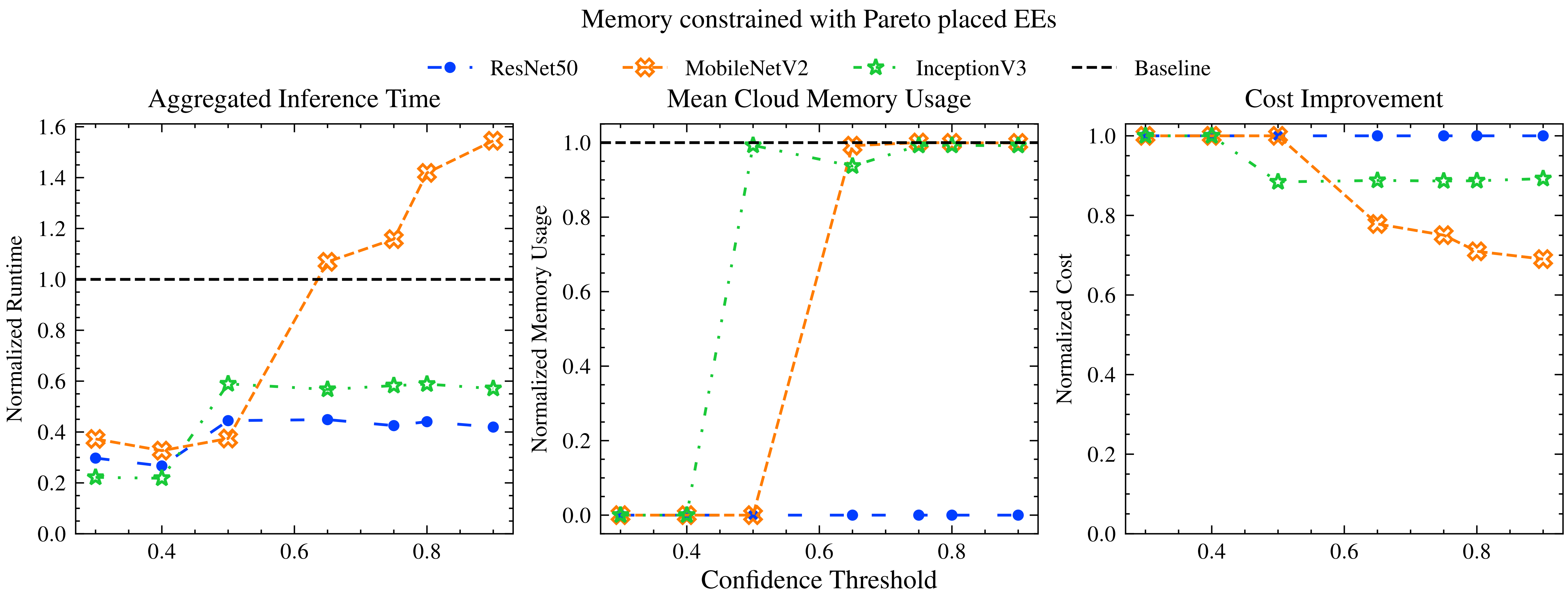
During Inference

- As EdgeLessPart opts for leveraging the Edge first, all inferences will start there
- EdgeLessPart uses per request confidence threshold to drive the early exiting
- If the Edge split does not reach enough accuracy, EdgeLessPart will send the intermediate results to the cloud for further processing

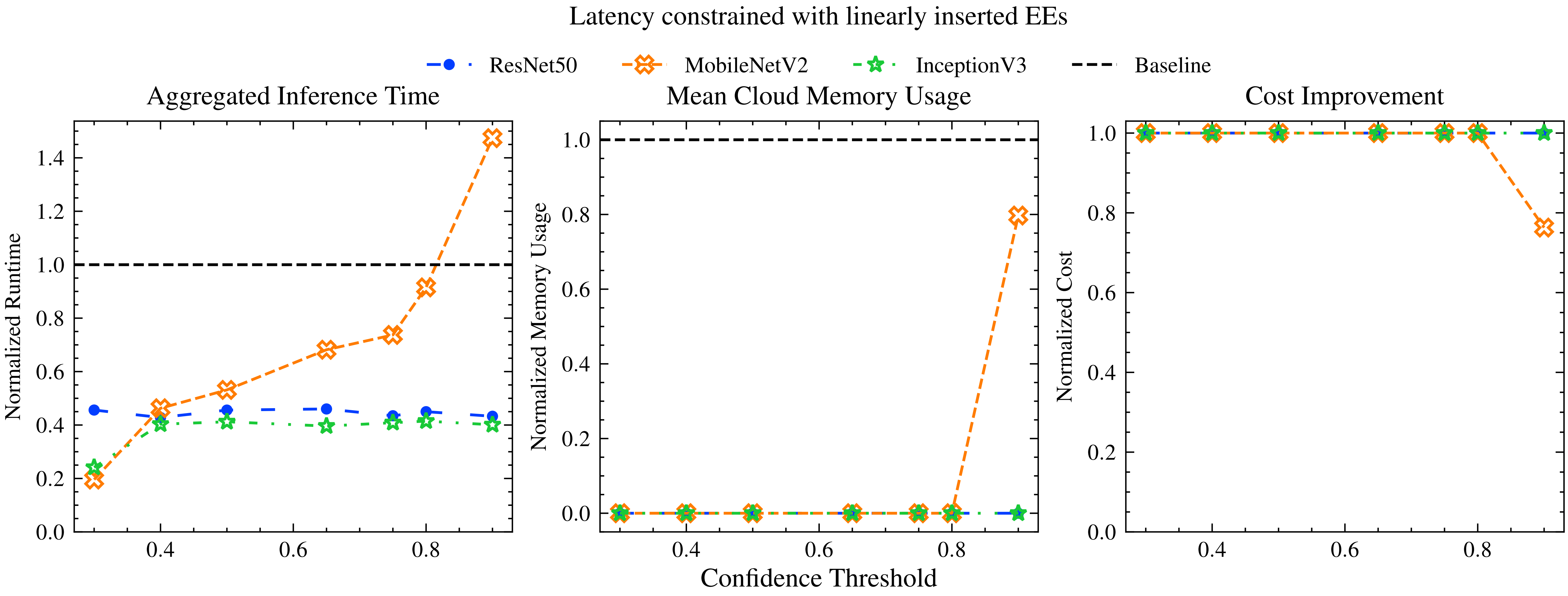
Our Preliminary Results



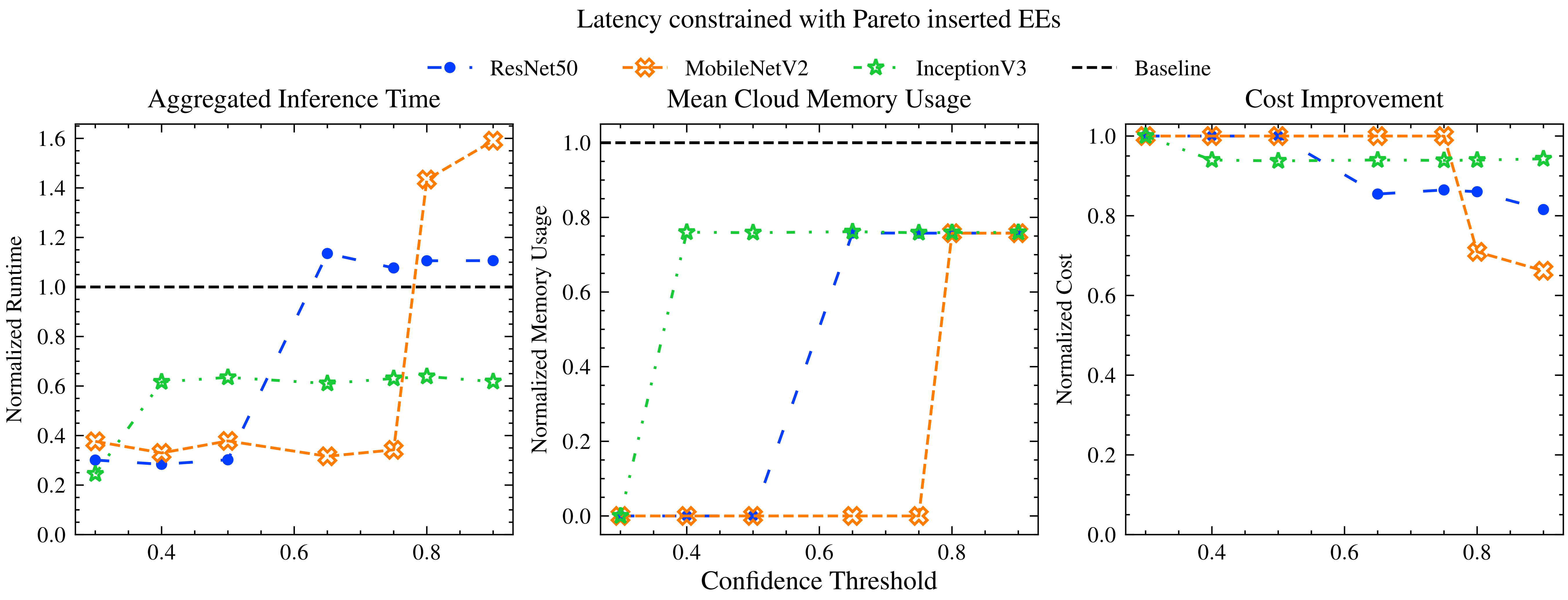
Our Preliminary Results



Our Preliminary Results



Our Preliminary Results



Conclusions and Future Work

- From our results, balancing between the Edge and Cloud is not trivial.
  - That includes DNN selection

- Nonetheless, EdgeLessPart can generate optimized DNNs that reduce cost by relying on edge devices

- Our current research shows that more optimizations are necessary:
  - Early Exits are still manually designed
  - EdgeLessPart uses just one network
    - We plan to address this by including Neural Architecture Search
  - EdgeLessPart still relies of user input

References

[1] NVIDIA GPU Cloud Computing Solutions. [Available online] https://www.nvidia.com/en-us/data-center/gpu-cloud-computing/