43rd IEEE International Conference on Distributed Computing Systems (ICDCS 2023)
July 18-21, 2023 · Sheraton Hong Kong Hotel & Tower · Hong Kong, China

Elastic DNN Inference with Unpredictable Exit in Edge Computing

Jiaming Huang, Yi Gao, and Wei Dong

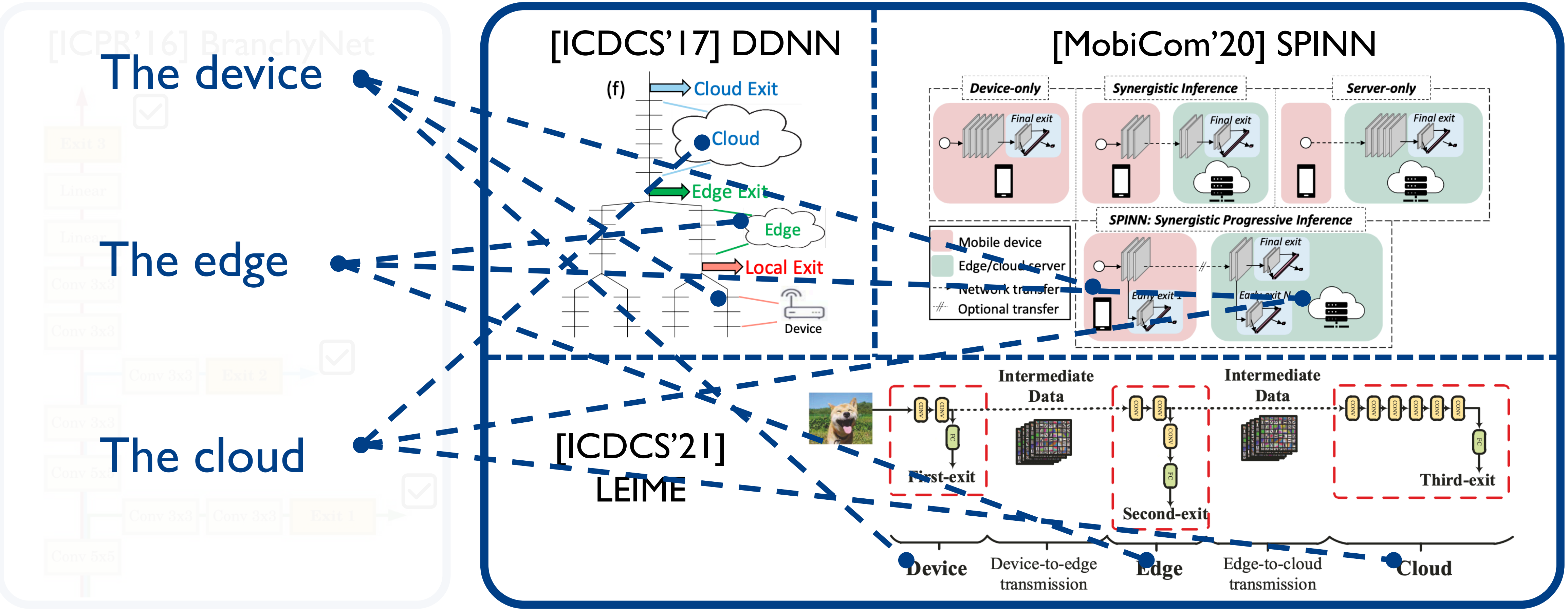
College of Computer Science, Zhejiang University, China

Presenter: Jiaming Huang

浙江大学
ZHEJIANG UNIVERSITY

Background

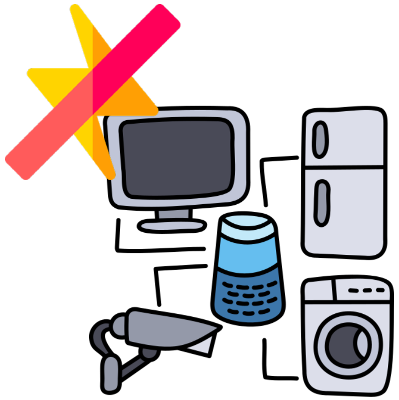
- In recent years, multi-exit neural networks have emerged and flourished in edge computing.



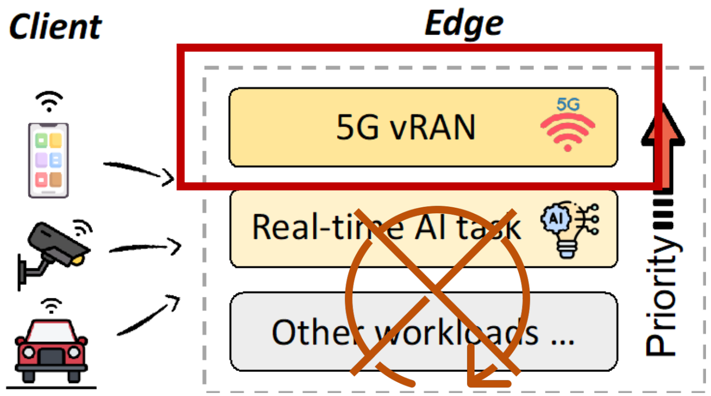
Background

- **However, many tasks often encounter unpredictable exit**

system power outages



preemption by high-priority tasks



[SIGCOMM'21]Concordia

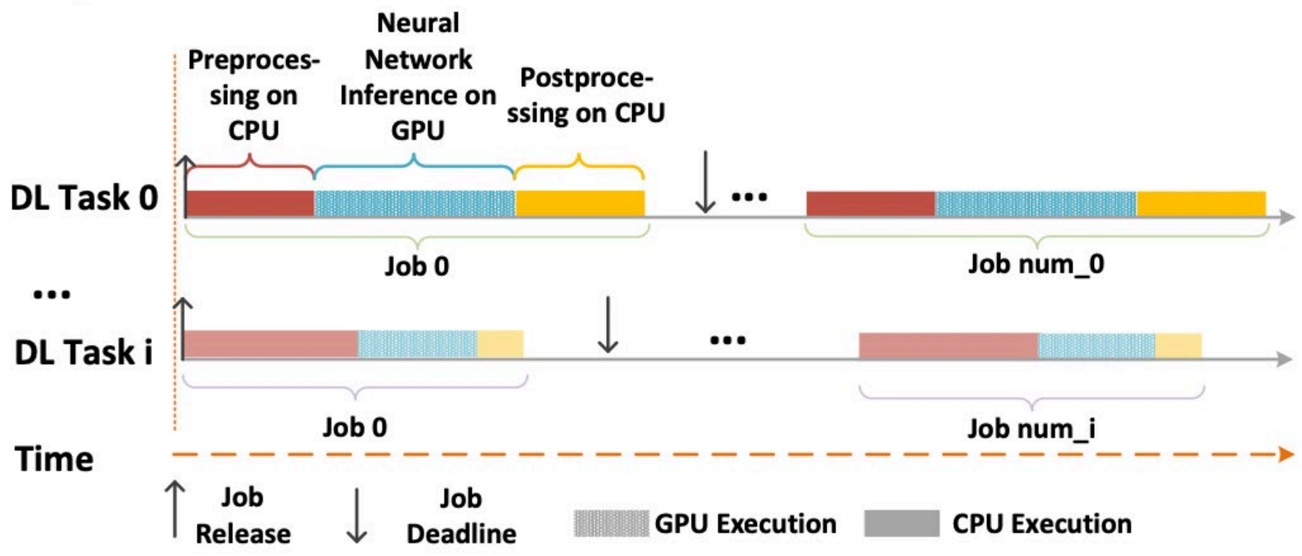
user exit demands

The forced exit real-time inference tasks were overlooked for a long time !

Prior work for efficient inference

- Model Compression
  - compression techniques
  - lightweight models

- Faster and lighter inference
- They still may output no result in r



- **Model Partition and Scheduling**
  - They perform model partitions on different levels and distribute them across multiple heterogeneous processors

- Higher efficiency of the entire system
- They use predictable time for scheduling and fail to adapt to unpredictable exits

Prior work for efficient inference

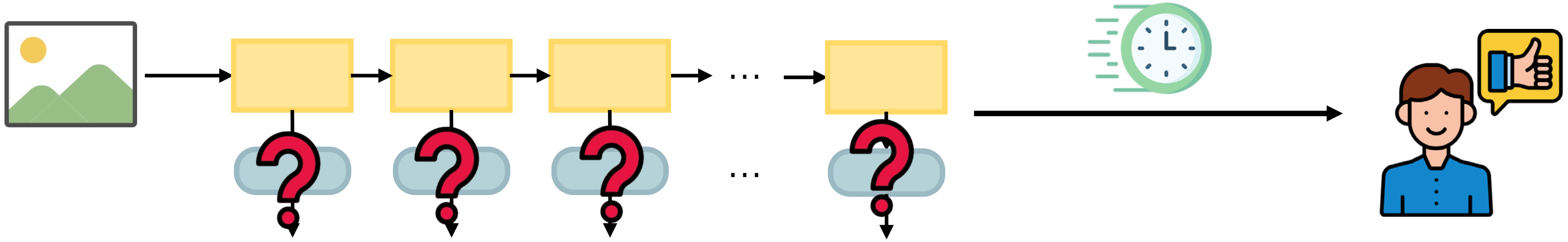
- Multi-exit NNs can exit early to output at least an intermediate result

  - Instance-wise dynamic inference

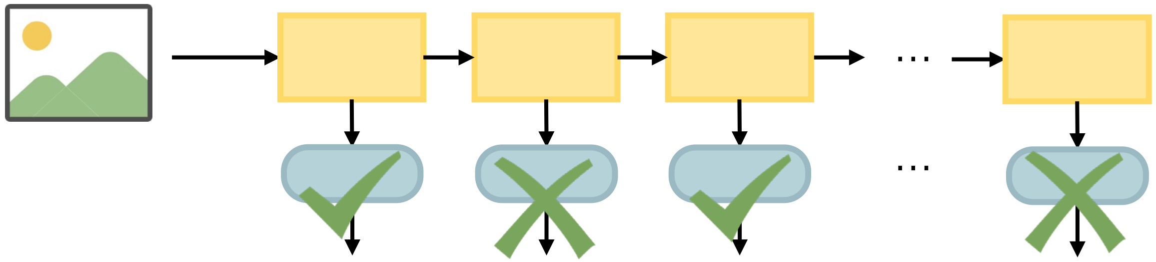
Orthogonal to these techniques, we are the first to propose **Elastic Inference** based on multi-exit models and bring a whole new plan generation solution under unpredictable exit scenarios.

EINet for Elastic Inference

- The **Elastic Inference** is time-insensitive which can make models generate desirable intermediate results no matter when being forced to exit.

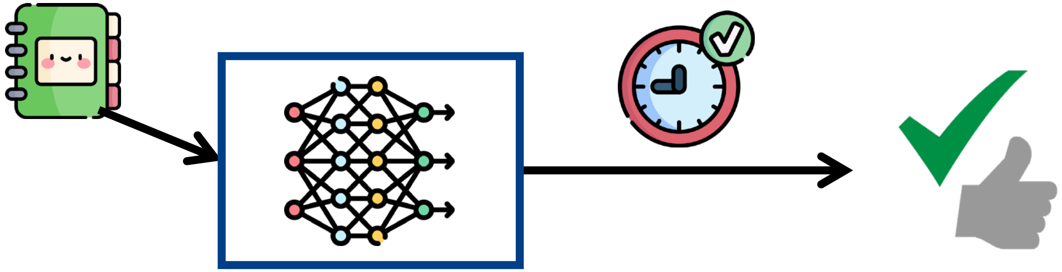


- **EINet**
  - It is a sample-wise **planner** of real-time multi-exit NNs
  - It guides multi-exit NNs to dynamically select branches for different samples

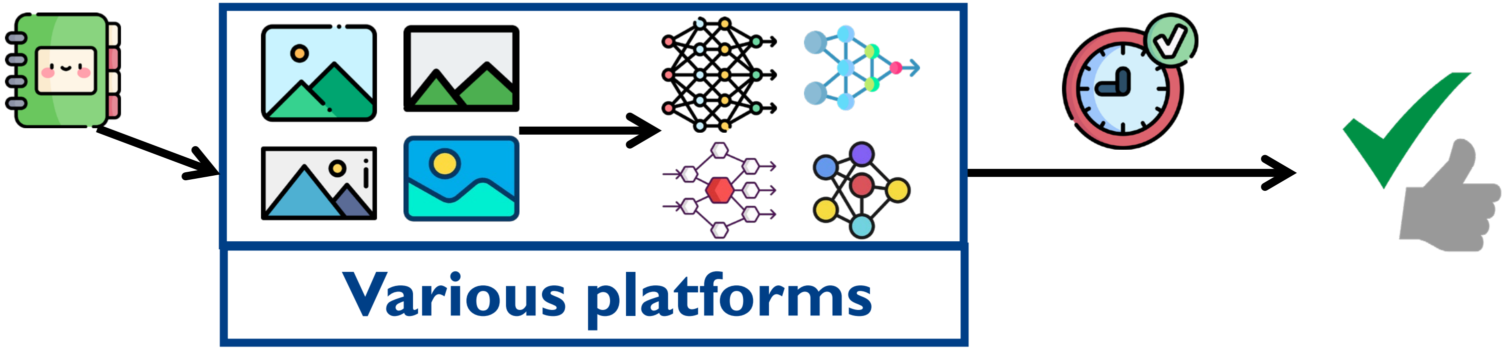


Challenges

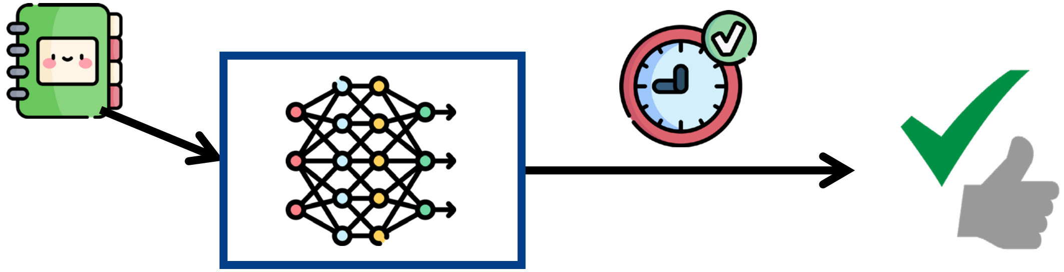
- **How can our planner always guide the model to get desirable results before being stopped to meet real-time demand?**



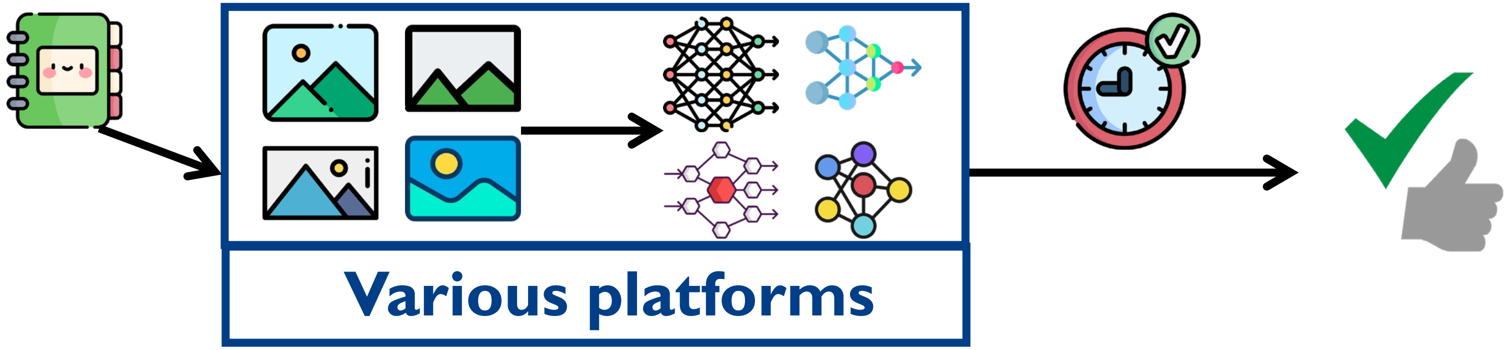
- **How can our planner be general to adapt to all models on various platforms for different input samples?**



Challenges



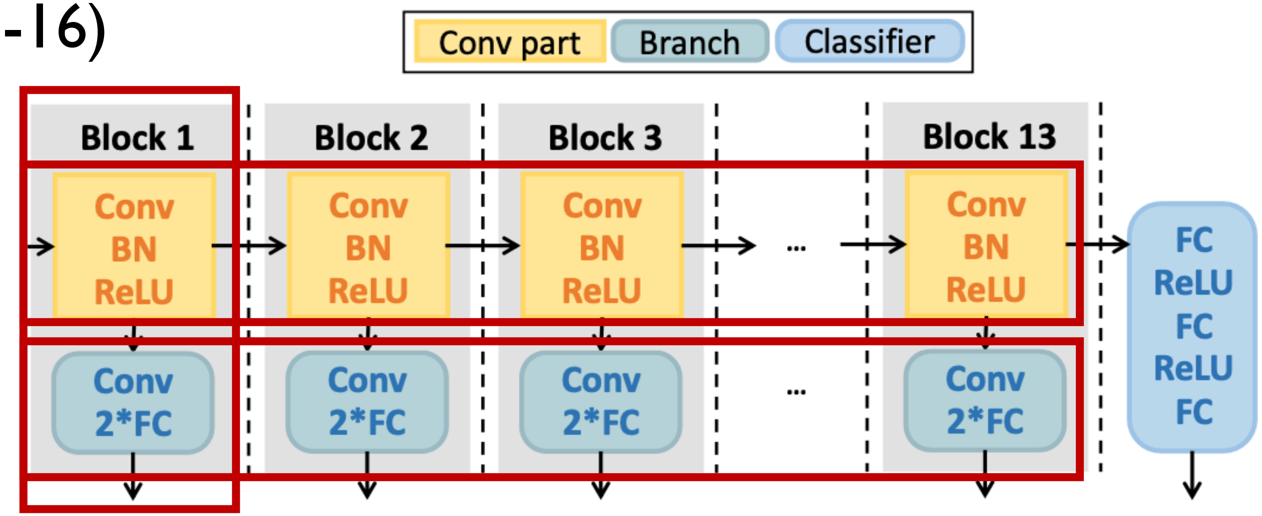
- We design **fine-grained** multi-exit NNs and **Search Engine** to strike the balance between inference latency and accuracy



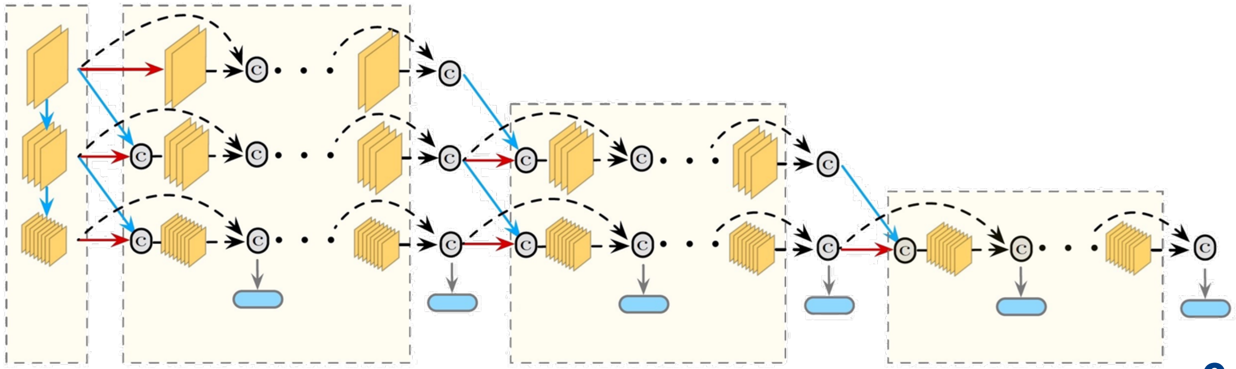
- We present offline **Block-wise Model Profiling** to profile the characteristics of different models on various platforms and **Confidence Score Predictors** to better adapt to the input samples

Design - fine-grained Multi-exit NNs

- **normal single-exit CNNs** (e.g.VGG-16)
  - ***conv*** part: one convolutional layer and its subsequent operations
  - ***branch***: one convolutional layer and two fully connected layers
  - ***block***: *conv* part and its *branch*

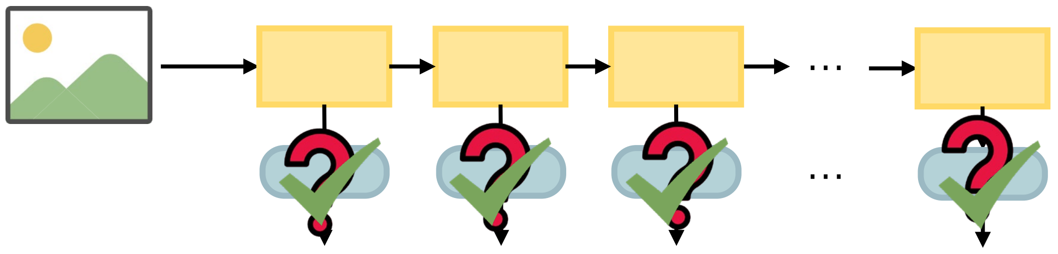


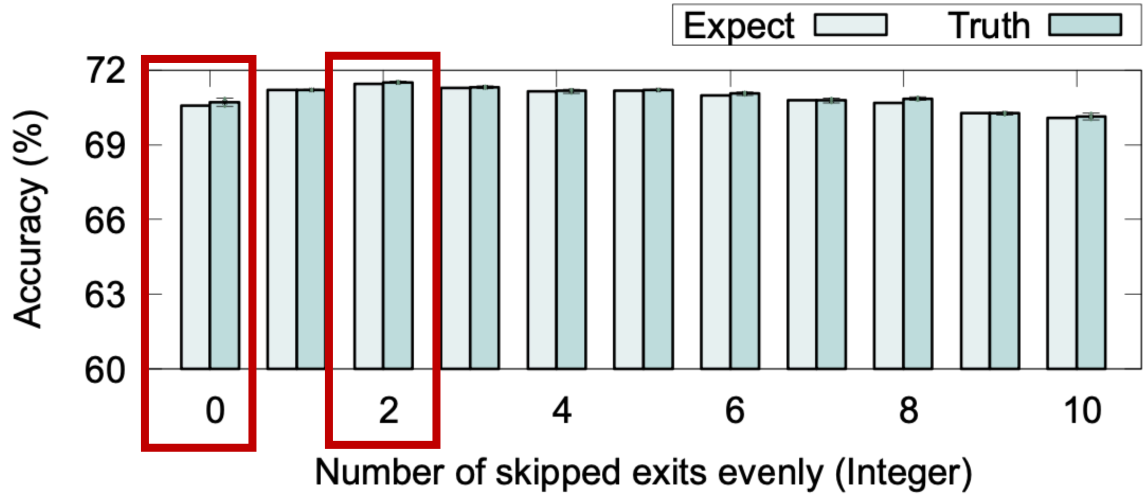
- **well-designed multi-exit model** (e.g. MSDNet)
  - manually adjust their structures to make them more fine-grained



Design - Search Engine

- **Skipping several exits will lead to different accuracy**
  - performing all the branches may have a time overhead that prevents the inference from going deeper for better accuracy

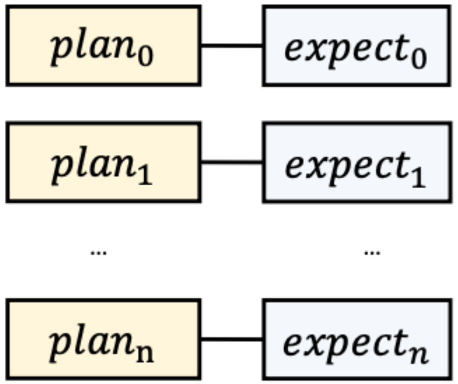


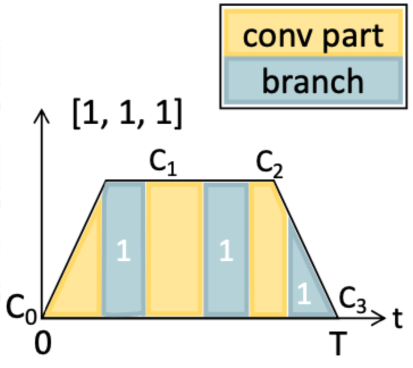


Can we find the optimal exit plan to guide models to skip several exits for better accuracy?

Design - Search Engine-Accuracy expectation

- **Evaluate each exit plan**
  - Unpredictable exit fall in which an inference time interval is a probabilistic event





Irregular Distribution

$$E = \sum_{i=0}^{len(C)} \frac{C_i t_i}{T}$$

$area_i$ (weighted time)

Area

Search space is huge

If a multi-exit NN has 40 exits, there will be **2^{40}** plans and the enumeration time is up to **about 40 days**

Searching is difficult

Accuracy expectation algorithm is **non-linear** which also aggravates the difficulty of heuristic searching

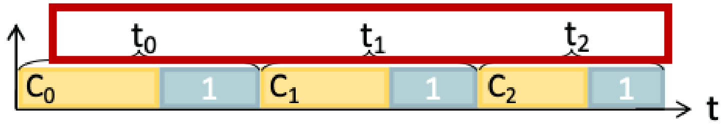
Design - Search Engine - Hybrid Search

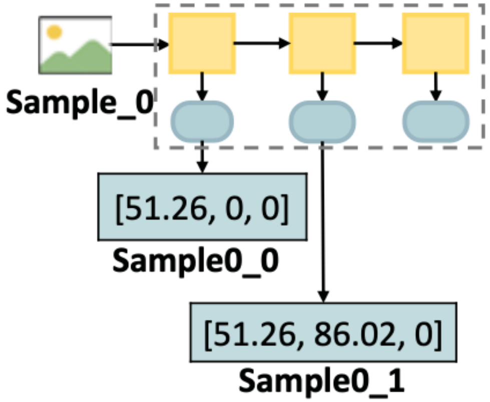
- **Greedy search**
  - It continuously explores plans by increasing branches to execute one by one
  - It tends to fall into the local optimum in many cases.

- **Hybrid search, a two-stage search approach, combines enumeration and greedy search.**
  - use **enumeration** for the first few branches
    - the time is few and the optimal results can be guaranteed
  - use **greedy** search for the later branches
    - find the better exit plan with higher performance in less search time

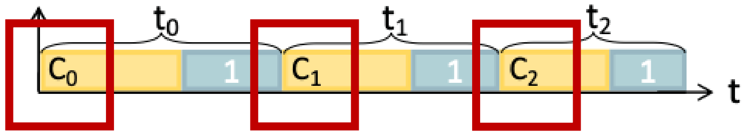
Design - Block-wise Model Profiles

- Adapt to all models on various platforms
  - Execute specified models on specific platforms and records their block-wise profiles
  - *Execution Time profiles* (ET-profiles)
    - the **execution time** of each block of models
    - depends on models and platforms





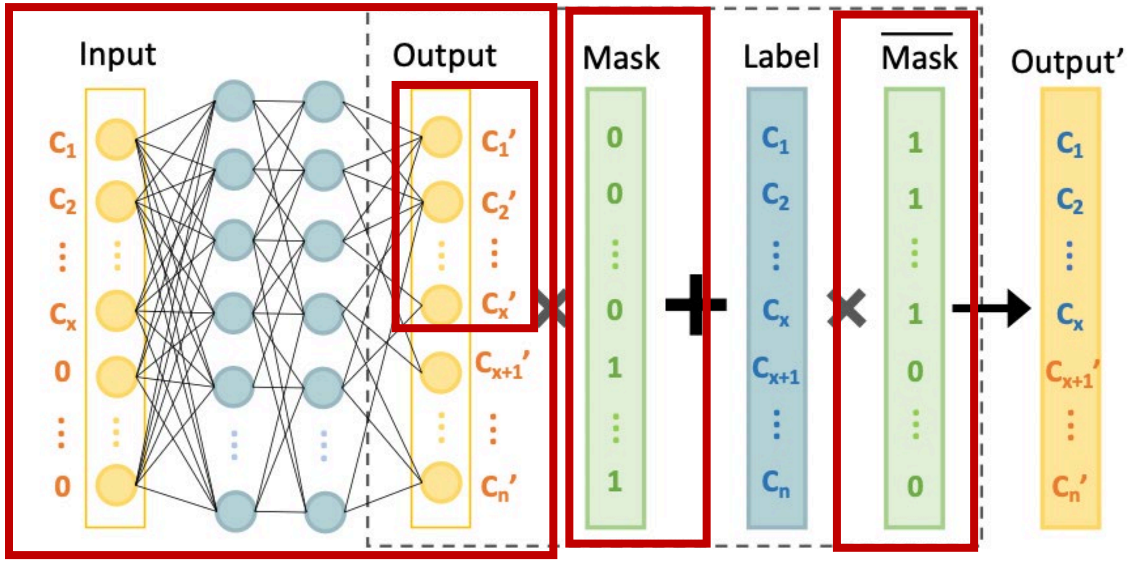
  - *Confidence Score profiles* (CS-profiles)
    - the **confidence scores** of each exit of the models
    - change with input samples



ID	Training data	Labels
Sample0_0	[51.26, 0, 0]	[51.26, 86.02, 99.99]
Sample0_1	[51.26, 86.02, 0]	[51.26, 86.02, 99.99]
Sample1_0	[78.77, 0, 0]	[78.77, 99.99, 100.0]
Sample1_1	[78.77, 99.99, 0]	[78.77, 99.99, 100.0]
…	[… , … , …]	[… , … , …]

Design - Confidence Score Predictors

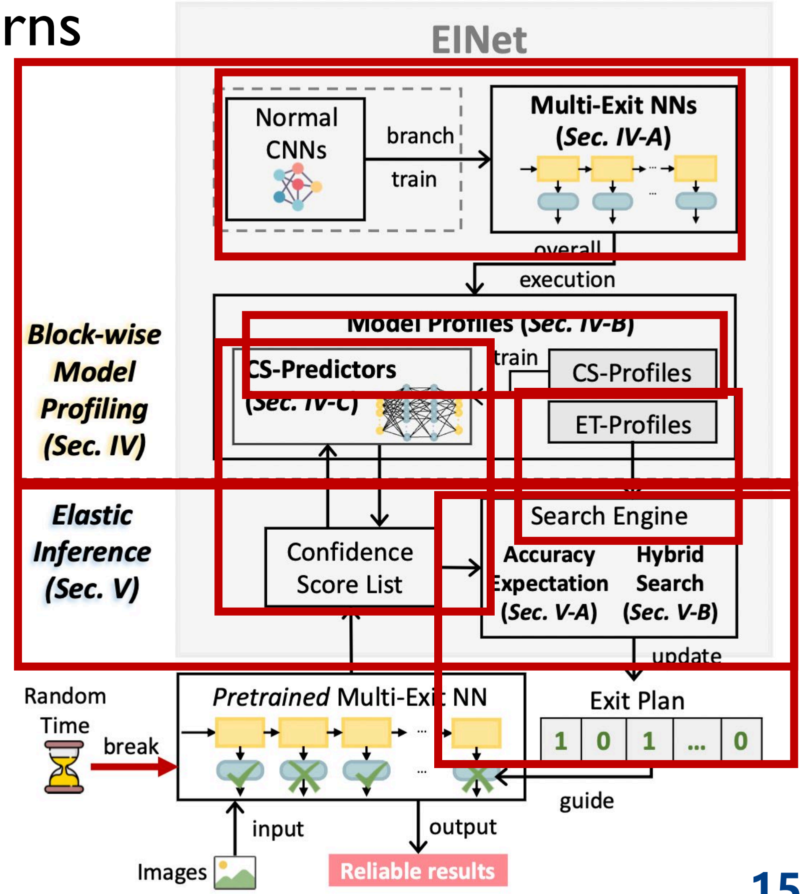
- **Adapt to the different input samples**
  - Since both training data and labels are one-dimensional, **MLP** is suitable.
  - To achieve dynamic output size, we use a mask to update outputs



ID	Training data	Labels
Sample0_0	[51.26, 0, 0]	[51.26, 86.02, 99.99]
Sample0_1	[51.26, 86.02, 0]	[51.26, 86.02, 99.99]
Sample1_0	[78.77, 0, 0]	[78.77, 99.99, 100.0]
Sample1_1	[78.77, 99.99, 0]	[78.77, 99.99, 100.0]
…	[… , … , …]	[… , … , …]

Design of EINet

- **Block-wise Model Profiling *(offline)***
  - EINet generates model profiles by executing multi-exit NNs
    - For single-exit CNNs that lack multiple exits, EINet turns them into **fine-grained** multi-exit NNs
    - **CS-profiles** : train CS-Predictors
    - **ET-profiles** : calculate accuracy expectation
- **Elastic Inference *(online)***
  - CS-Predictors predict the score during the inference
  - EINet will execute such search and update process repeatedly until the inference is exited unpredictably.



Implementation

- **Language:**
  - PyTorch

Algorithm	Py/C	Max(ms)	Avg(ms)	Min(ms)
Accuracy Expectation	Python	0.0610	0.0594	0.0584
	C	0.0003	0.0003	0.0003
Hybrid Search	Python	4.9145	4.6599	4.3861
	C	0.1292	0.1277	0.1267

- **Datasets:** MNIST, CIFAR-10, CIFAR-100
- **Hardware:** NVIDIA GeForce RTX-3090 GPUs
- **Metric:** **O**verall accuracy
- **Baseline**

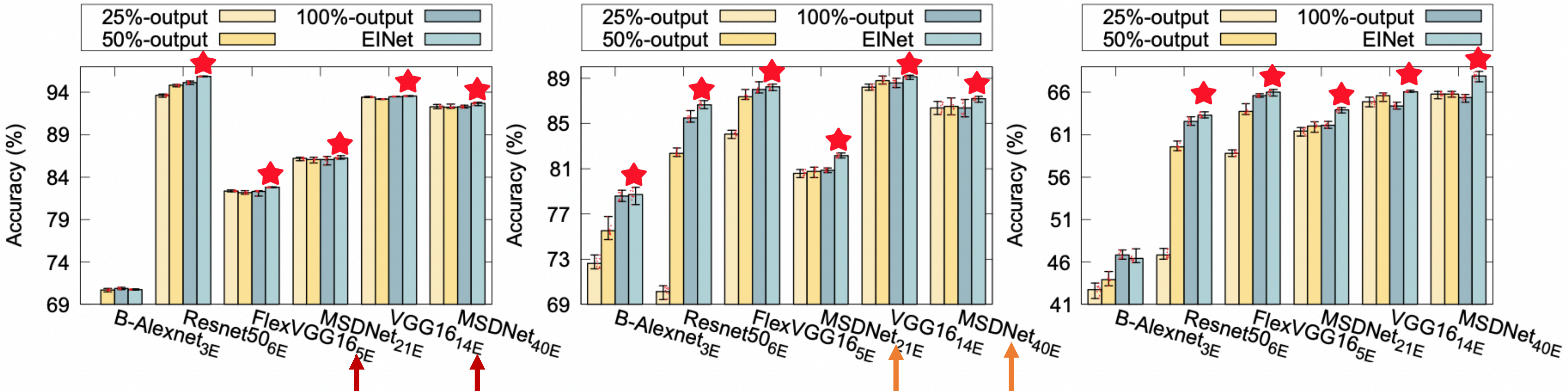
- **Models**
  - B-Alexnet with three exits
  - FlexVGG-16 with five exits
  - fine-grained VGG-16 with 14 exits
  - fine-grained Resnet-50 with six exits
  - MSDNet with 21 and 40 blocks

Exit plans	static	25% exits output	50% exits output	100% exits output	statistics near-optimal
	dynamic	90% confidence-based	95% confidence-based	99% confidence-based	EINet with random search
Common models		classic single-exit models	compressed single-exit models	Multi-exit models without skipping	

Evaluation

- **Compared to static exit plans**
  - EINet has about 0.13%-16.5% performance gain
  - the more fine-grained network has higher accuracy

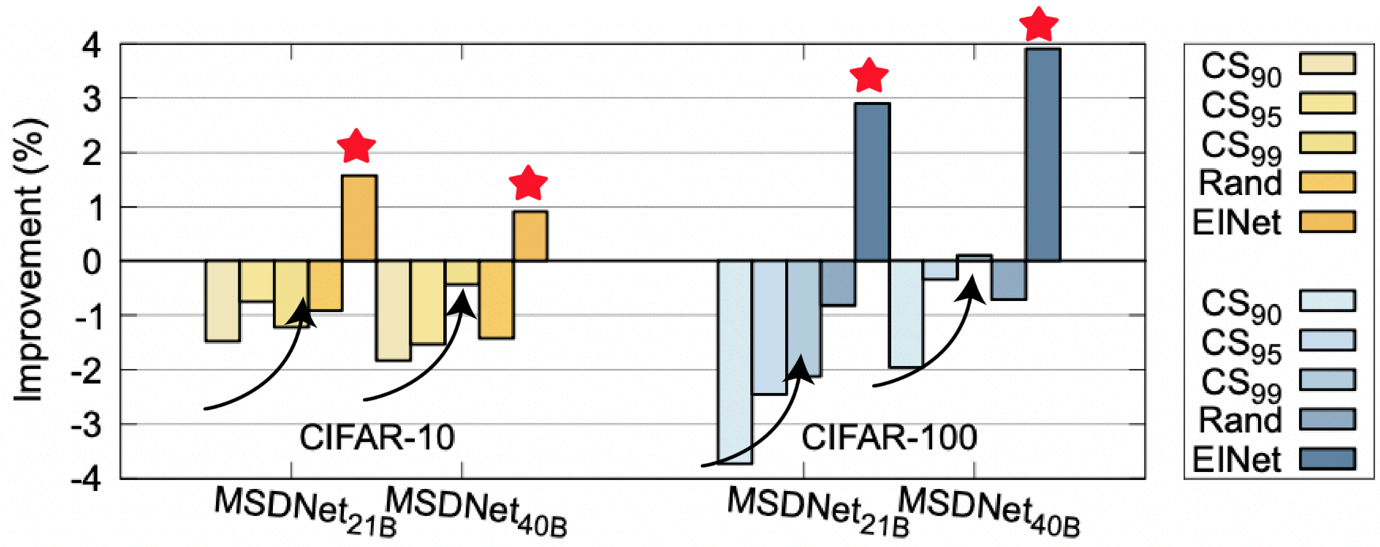
Datasets	Models	Statis(%)	**Ours(%)**
CIFAR-10	B-AlexNet	78.43	78.71 (+0.28)
	ResNet50	85.62	86.65 (**+1.03**)
	FlexVGG16	88.10	88.23 (+0.13)
	MSDNet21	80.87	81.11 (+0.24)
	VGG16	88.98	89.12 (+0.14)
	MSDNet40	86.38	86.60 (+0.22)
CIFAR-100	B-AlexNet	46.40	46.41 (+0.01)
	ResNet50	62.73	63.29 (+0.56)
	FlexVGG16	65.88	66.03 (+0.15)
	MSDNet21	62.25	63.92 (**+1.67**)
	VGG16	65.63	66.08 (+0.45)
	MSDNet40	66.14	67.93 (**+1.79**)



(a) MNIST (b) CIFAR-10 (c) CIFAR-100

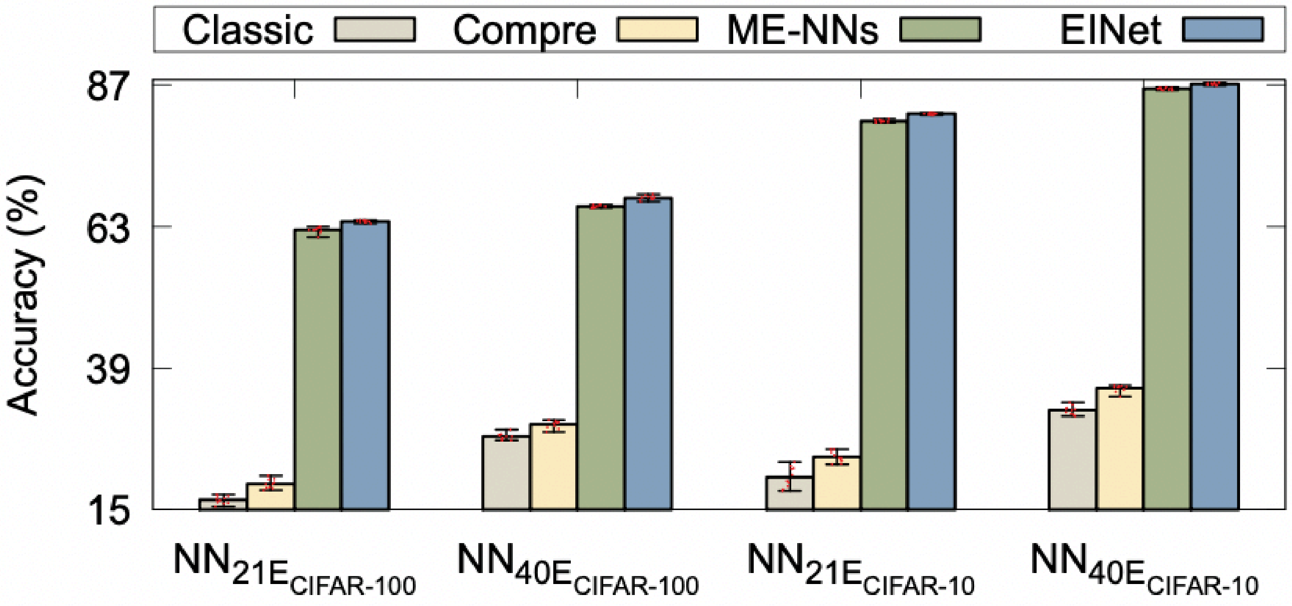
Evaluation

- **Compared to dynamic exit plans**
  - ElNet has about 0.79%- 4.1% performance gain
  - for confidence-based dynamic plans, raising the confidence threshold for early exit has a better effect on elastic inference



Evaluation

- **Compared to common neural networks**
  - ElNet can achieve over 50% accuracy improvement
    - 40.4% to 61.5% improvement in accuracy compared to **classic** models
    - 38.5% to 58.2% performance improvement compared to the compressed models
    - 0.8% to 1.5% improvement compared to the multi-exit models without skipping



EINet

a sample-wise planner of real-time multi-exit DNNs, which achieves efficient **E**lastic **I**nference with unpredictable exit while guaranteeing best-effort accuracy on different edge platforms.

Thank you for your attention!

Jiaming Huang

huangjm@zju.edu.cn

浙江大学 ZHEJIANG UNIVERSITY