🍹 Cocktail: A Multidimensional 🍹 Optimization for Ensemble Learning

Jashwant Raj Gunasekaran, Cyan Subhra Mishra

Prashanth Thinakaran, Bikash Sharma, Mahmut T. Kandemir, Chita R. Das

Penn State, CSE, High Performance Computing Lab

nsdi '22

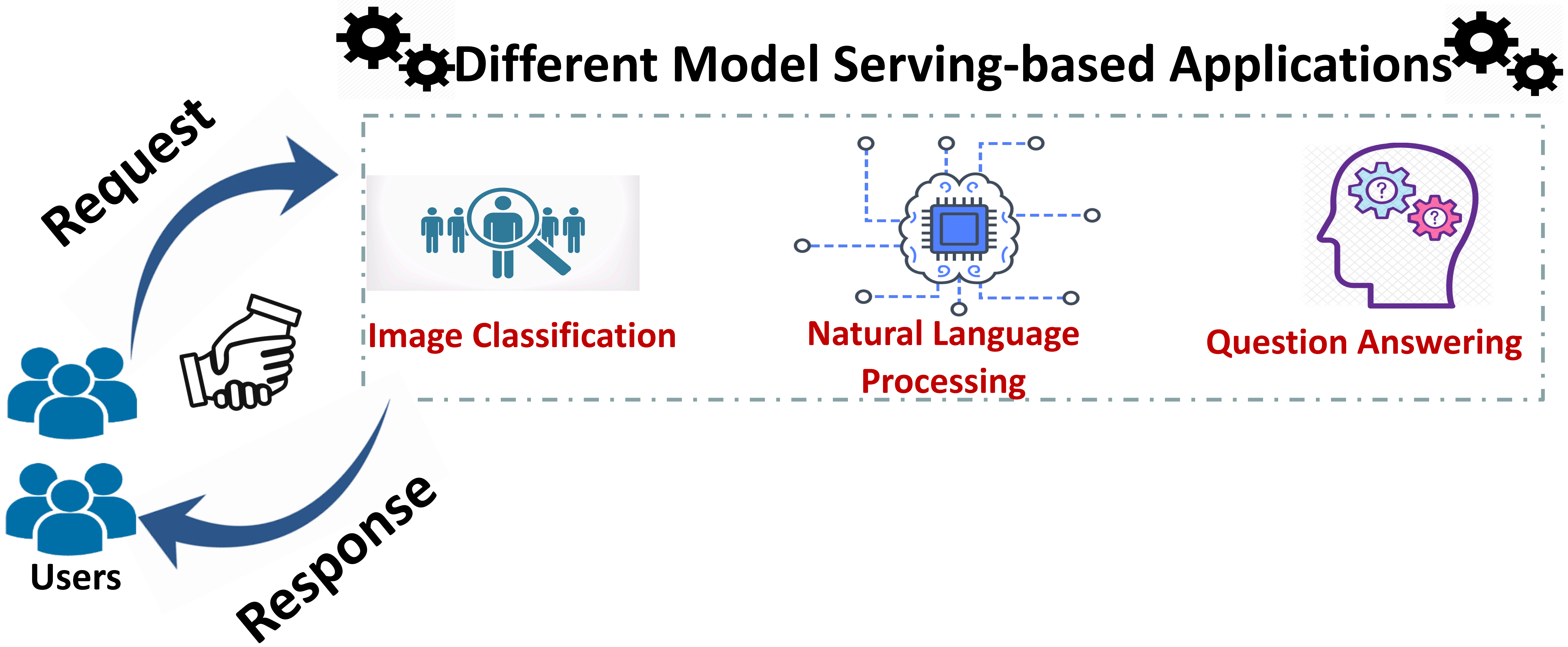
19th USENIX Symposium on Networked Systems Design and Implementation

April'6-2022 | Renton, WA, USA

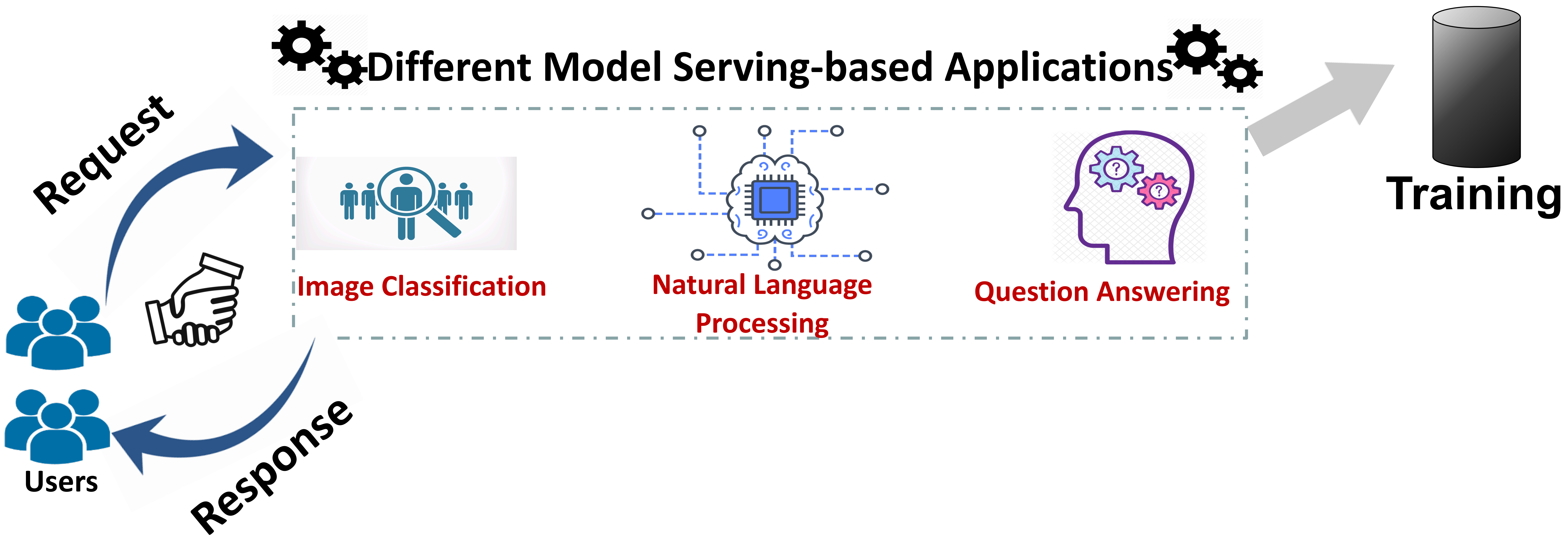
PennState College of Engineering

PennState High Performance Computing Lab

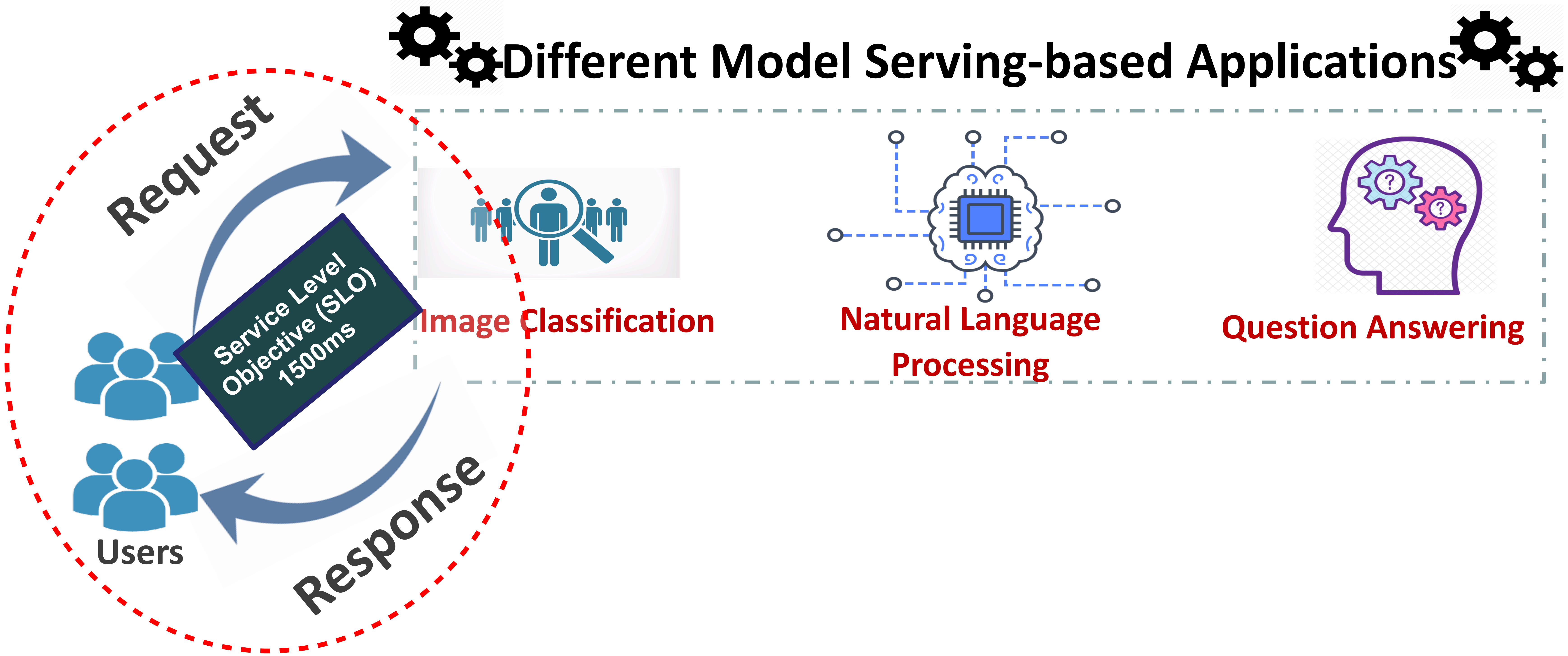
Model Serving Hosted on Cloud



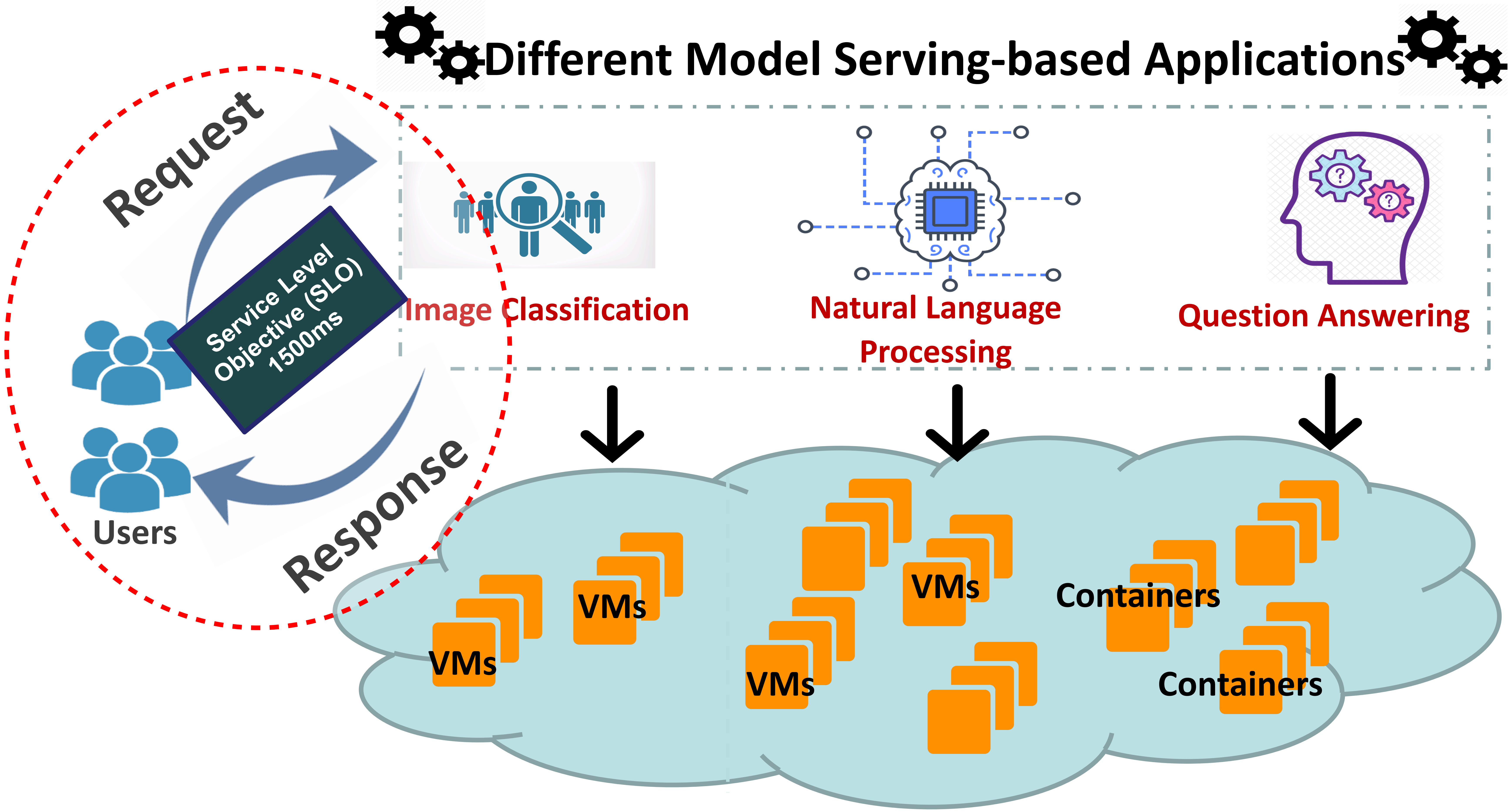
MODEL SERVING HOSTED ON CLOUD



Model Serving Hosted on Cloud

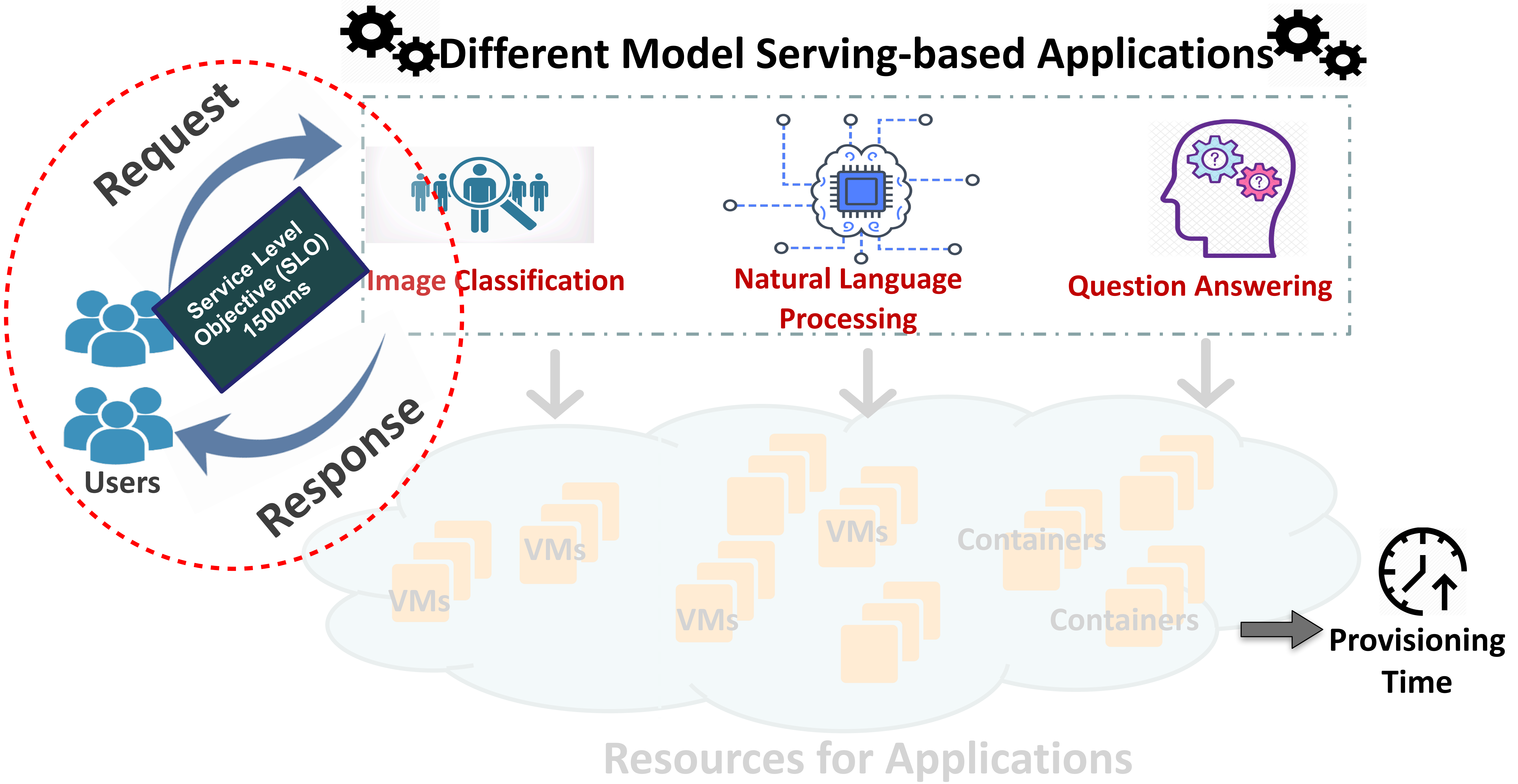


Model Serving Hosted on Cloud

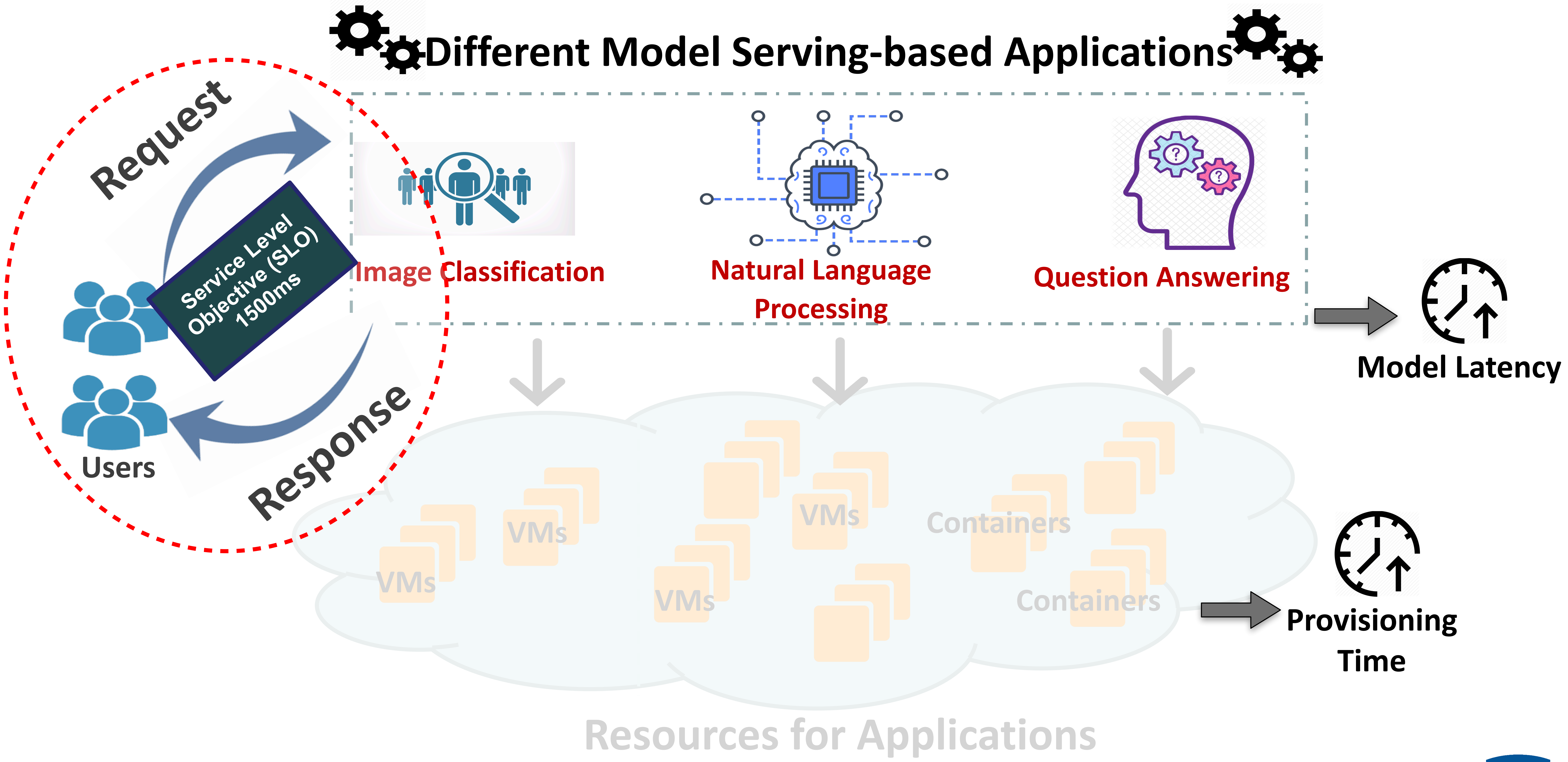


Resources for Applications

Model Serving Hosted on Cloud



MODEL SERVING HOSTED ON CLOUD



PennState College of Engineering

PennState High Performance Computing Lab

MODEL SERVING CHALLENGES

Model Serving

- **Resource Type** → **Provisioning Latency**
- **Model Type** → **Model Latency**, **Accuracy**

Cost

Model Serving Challenges

✅ Wang et al. ATC'19
Jain et al Middleware'19
Li etal, SC'21

Model Serving

- **Resource Type** → **Provisioning Latency**
- **Model Type** → **Model Latency**
- **Model Type** → **Accuracy**

Provisioning Latency, **Model Latency** → **Cost**

Model Serving Challenges

Wang et al. ATC’19
Jain et al Middleware’19
Li etal, SC’21

Model Serving

- **Resource Type** → **Provisioning Latency**
- **Model Type** → **Model Latency**, **Accuracy**

Provisioning Latency, **Model Latency** → **Cost**

Model Serving Challenges

Wang et al. ATC'19
Jain et al Middleware'19
Li etal, SC'21

Model Serving

- **Resource Type** → **Provisioning Latency**
- **Model Type** → **Model Latency**, **Accuracy**

Cost

In Netflix, 75% of viewer activity is based on these accurate suggestions.

Model Serving Challenges

Wang et al. ATC'19
Jain et al Middleware'19
Li etal, SC'21

Provisioning Latency

Resource

How to improve accuracy with low latency and low cost?

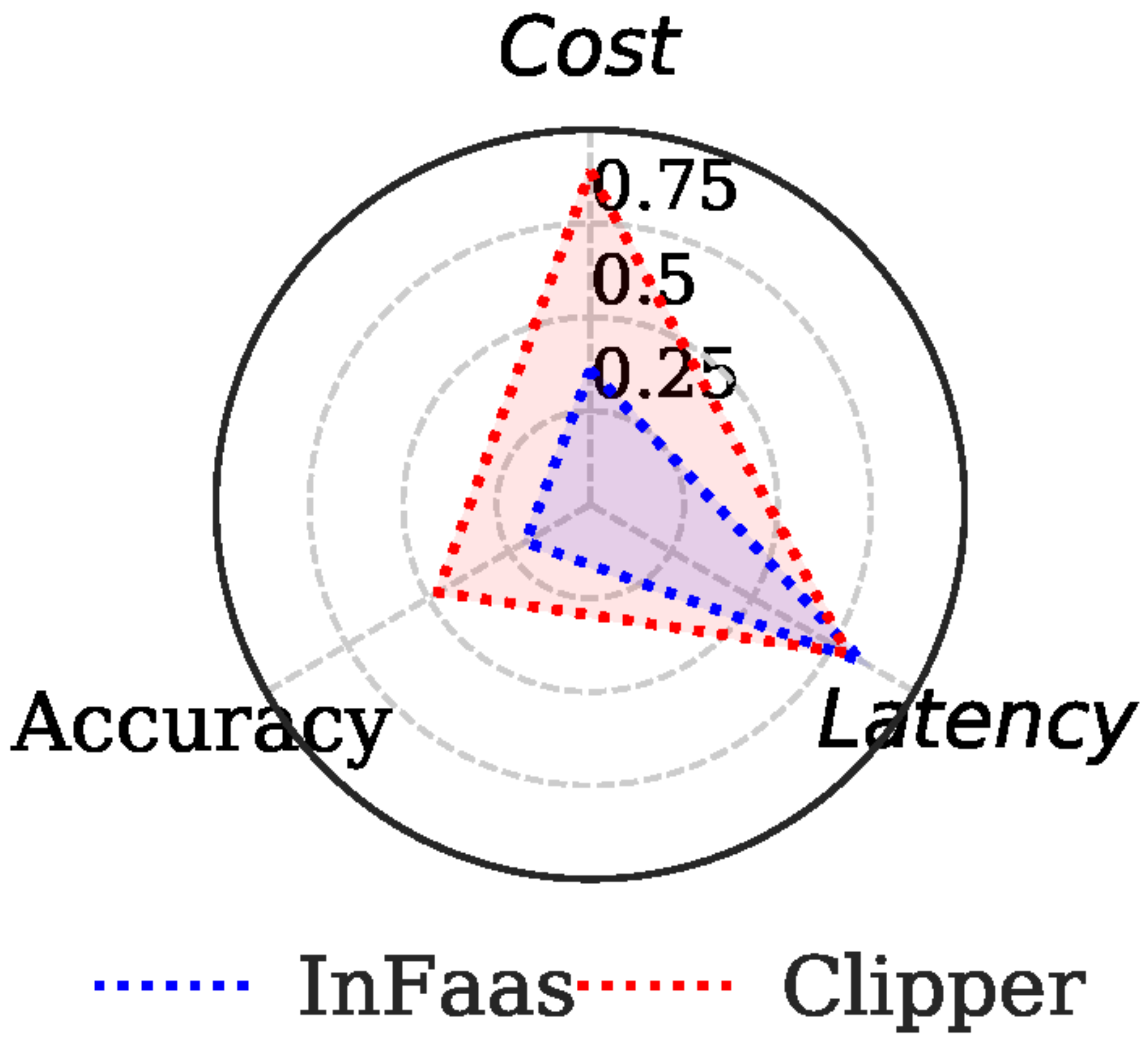
Prior Work in Model Serving

– **InFaas** uses different resource types to ensure low latency at low cost.

– **Clipper** achieves higher accuracy while compromising latency.

Crankshaw et al CIDR'15, NSDI'17, SoCC'20
Yadawkar et al ATC'21

Prior Work in Model Serving

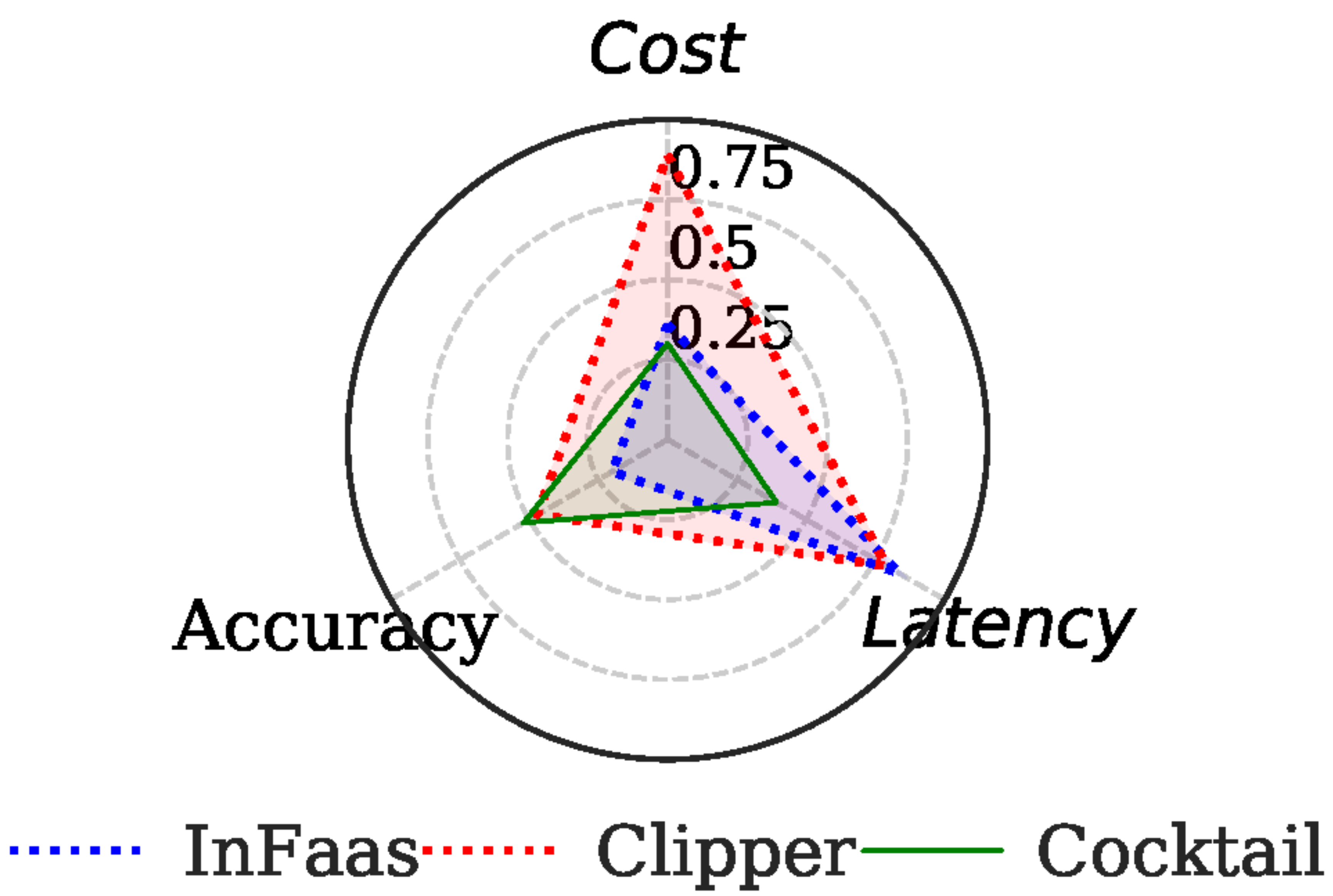


– **InFaas** uses different resource types to ensure low latency at low cost.

– **Clipper** achieves higher accuracy while compromising latency.

Crankshaw et al CIDR'15, NSDI'17, SoCC'20
Yadawkar et al ATC'21

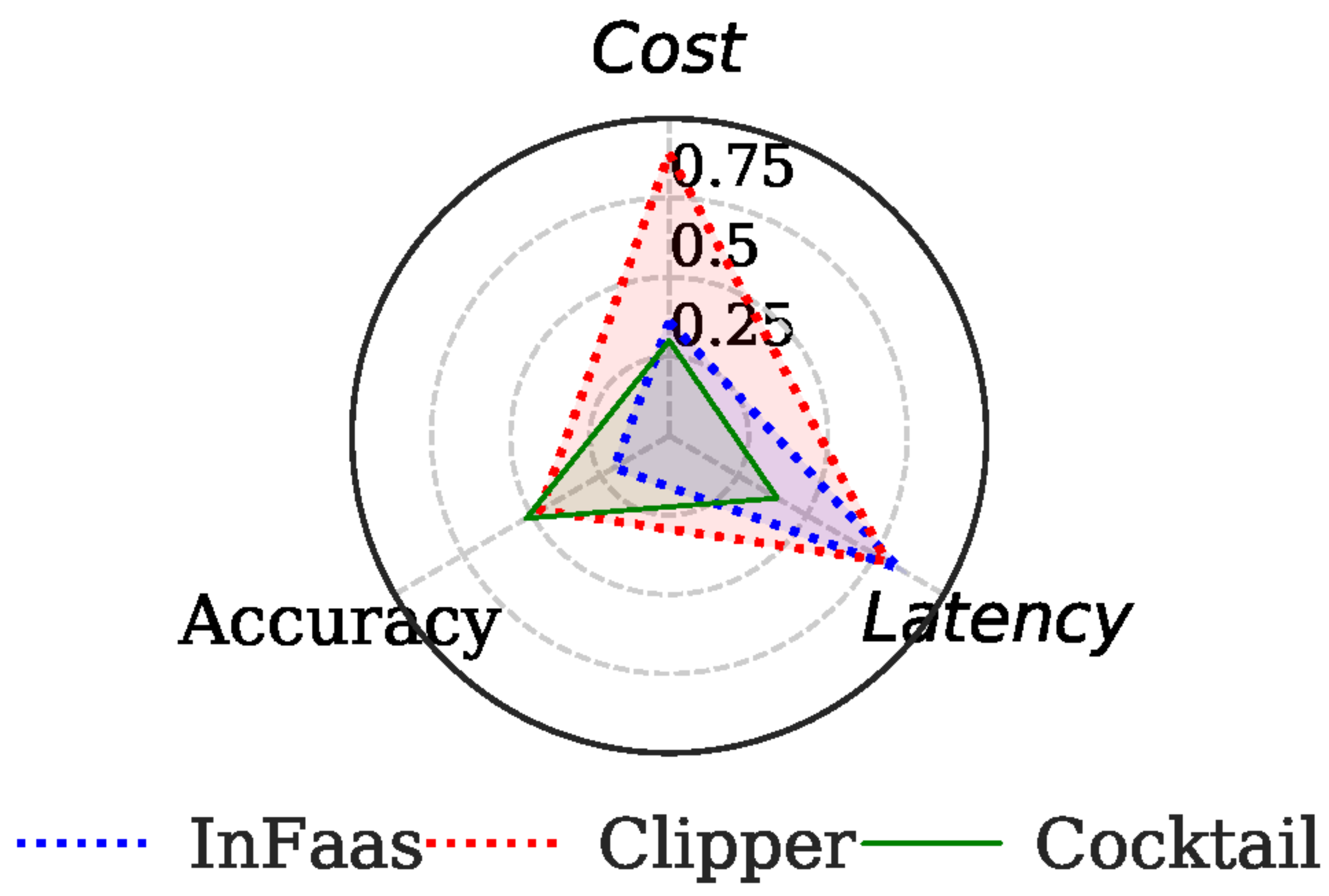
PRIOR WORK IN MODEL SERVING

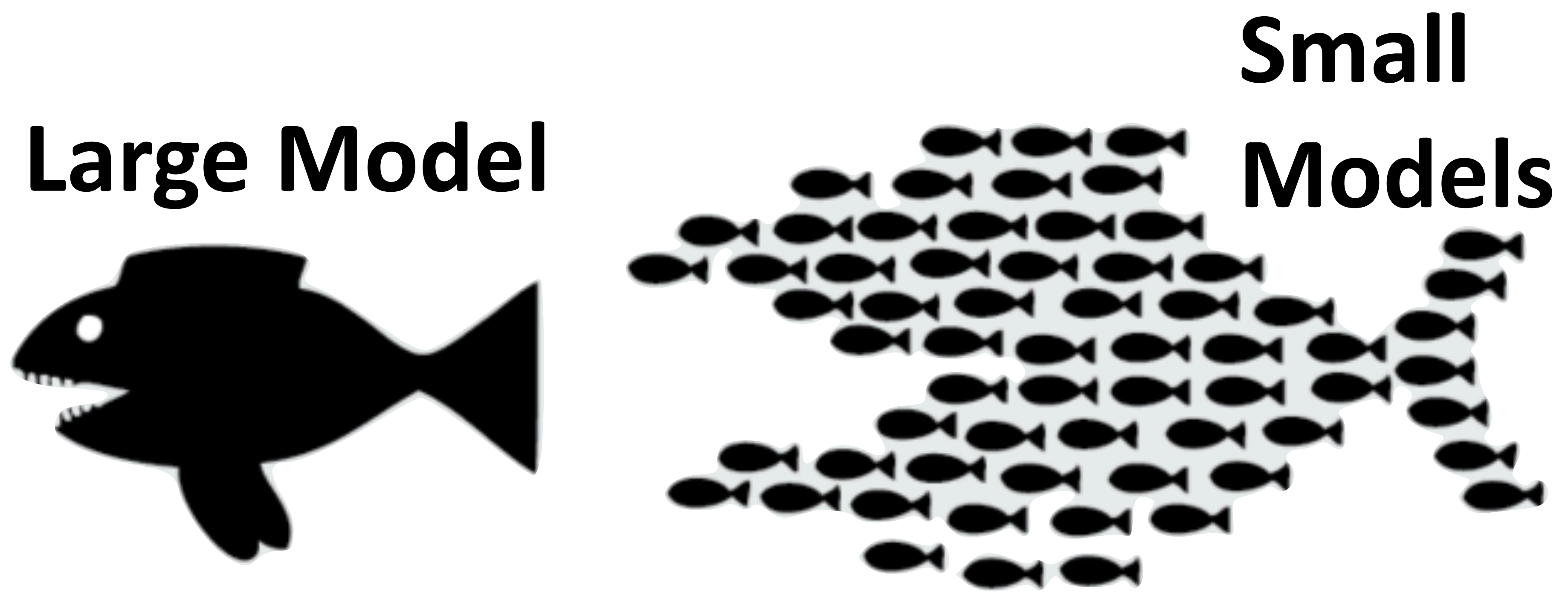


- **InFaas** uses different resource types to ensure low latency at low cost.
- **Clipper** achieves higher accuracy while compromising latency.

Crankshaw et al CIDR'15, NSDI'17, SoCC'20
Yadawkar et al ATC'21

Prior Work in Model Serving



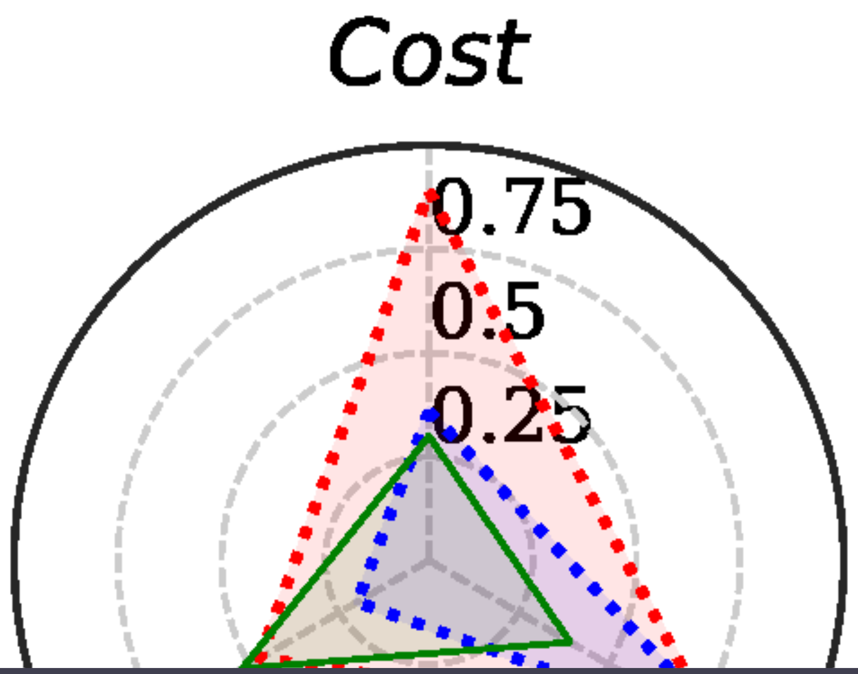


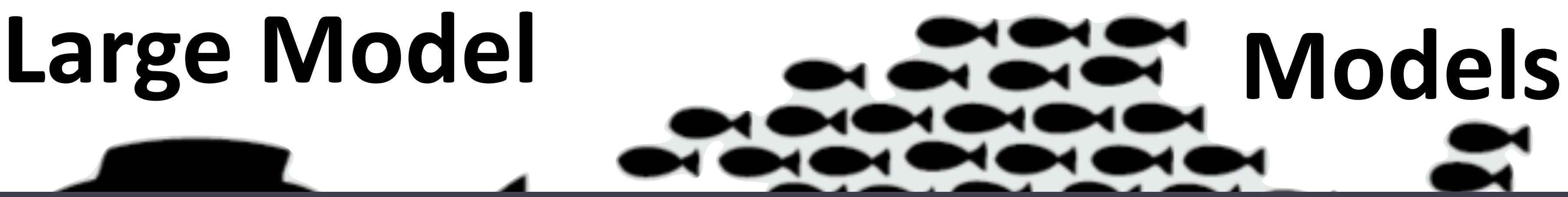
– **InFaas** uses different resource types to ensure low latency at low cost.

– **Clipper** achieves higher accuracy while compromising latency.

Crankshaw et al CIDR'15, NSDI'17, SoCC'20
Yadawkar et al ATC'21

PRIOR WORK IN MODEL SERVING





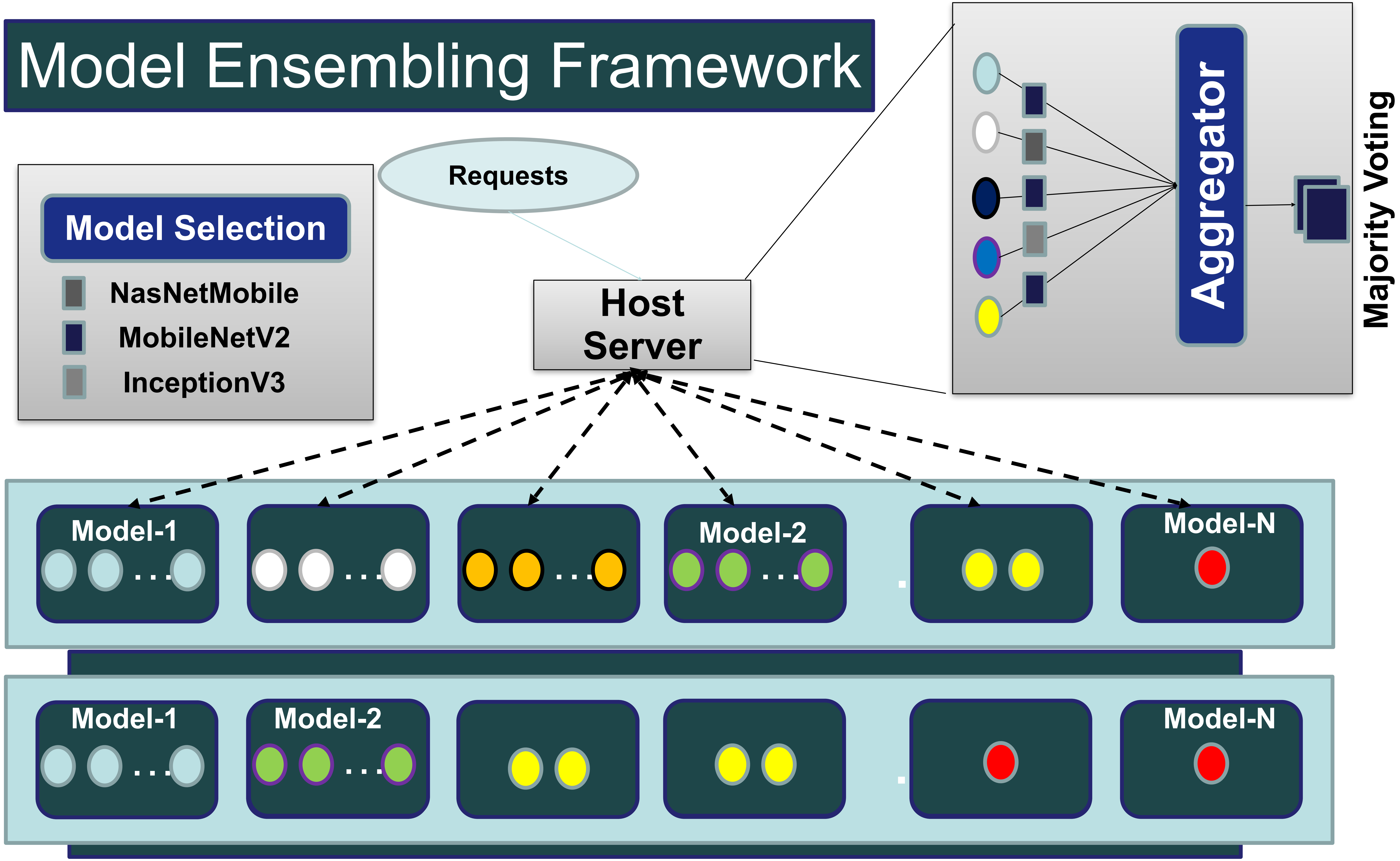
How to do ensembling?

– **InFaas** uses different resource types to ensure low latency at low cost.

– **Clipper** achieves higher accuracy while compromising latency.

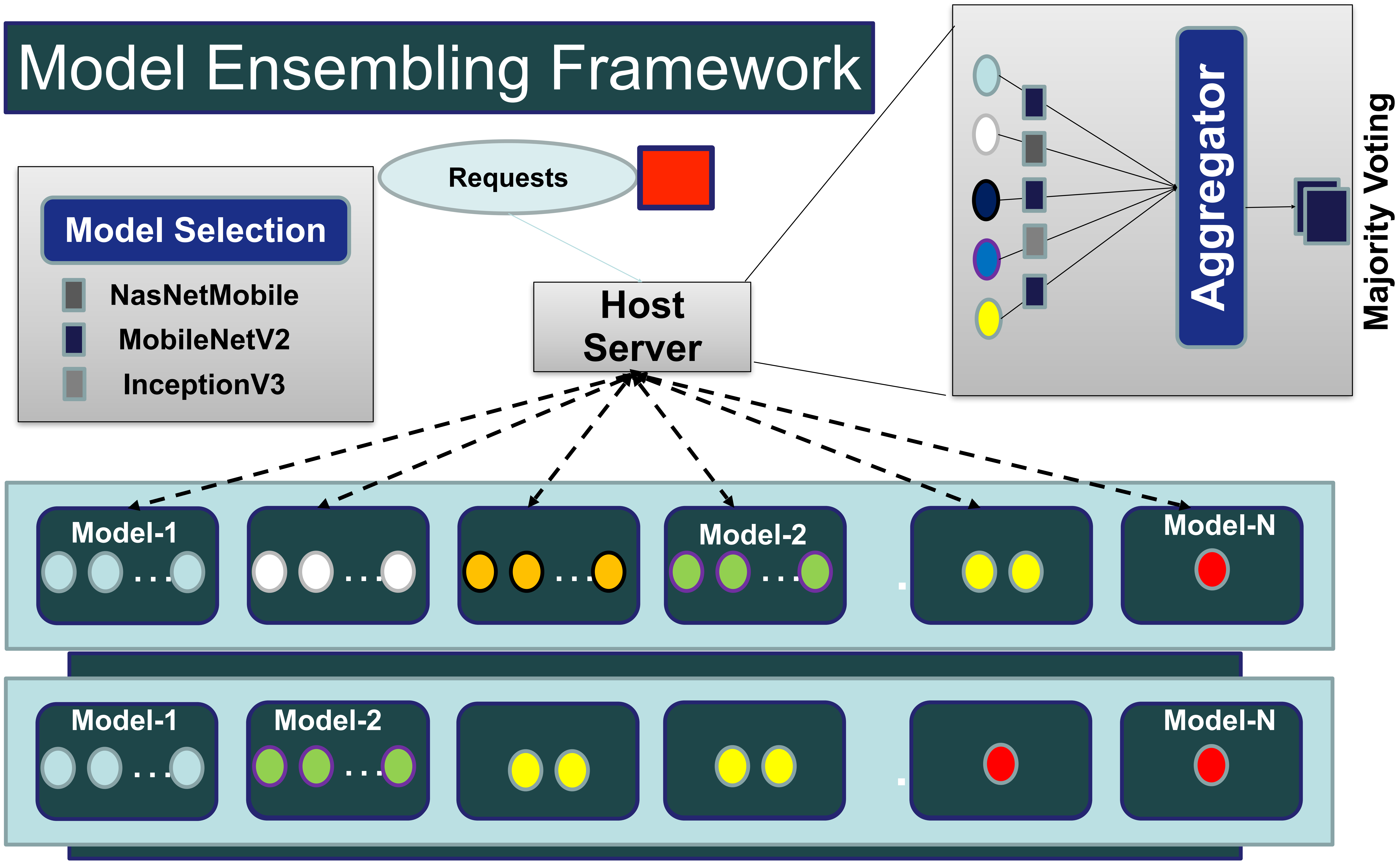
Crankshaw et al CIDR’15, NSDI’17, SoCC’20
Yadawkar et al ATC’21

Model Ensembling Framework



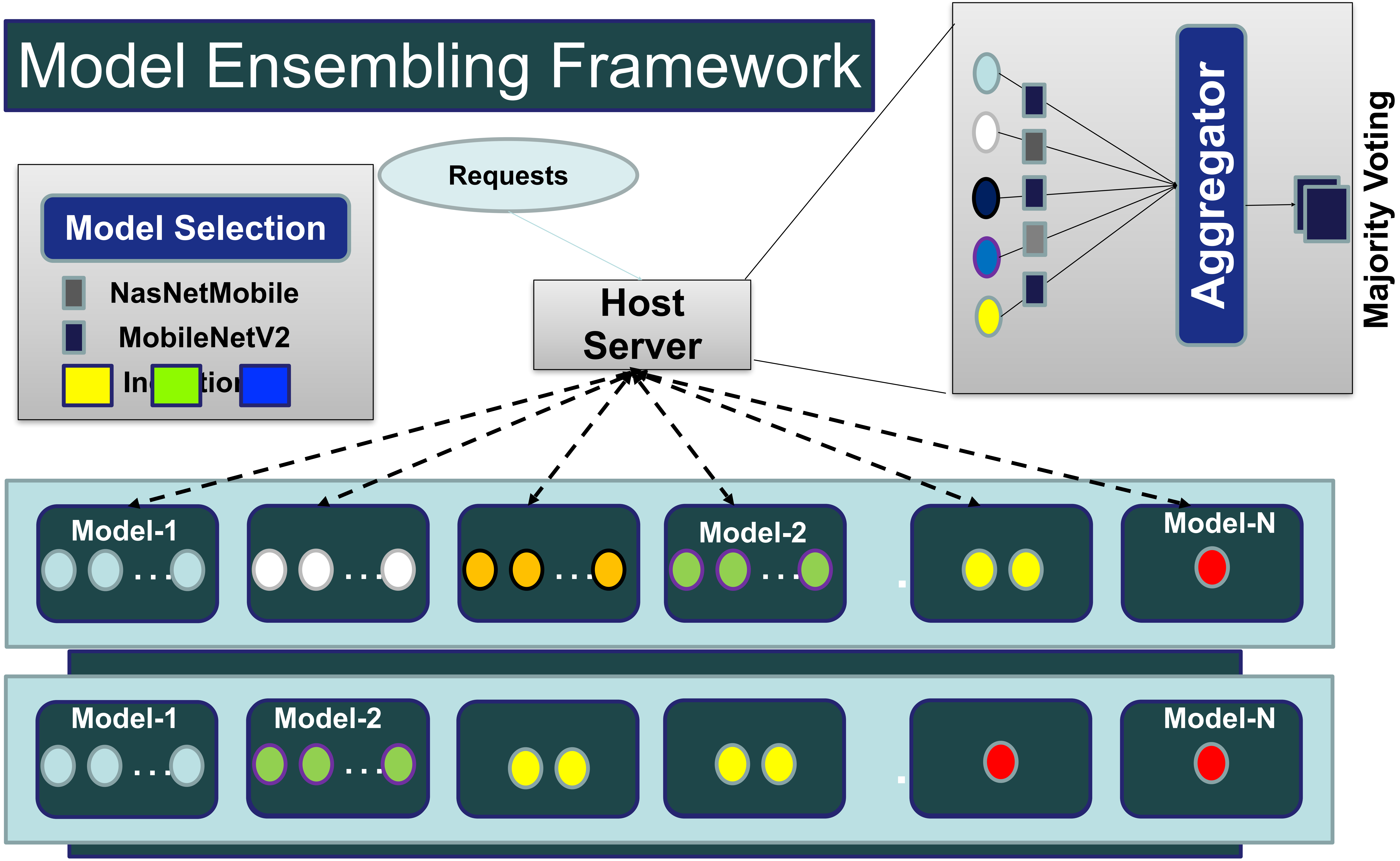
Cloud Resources for Individual Models (Virtual Machines)

Model Ensembling Framework



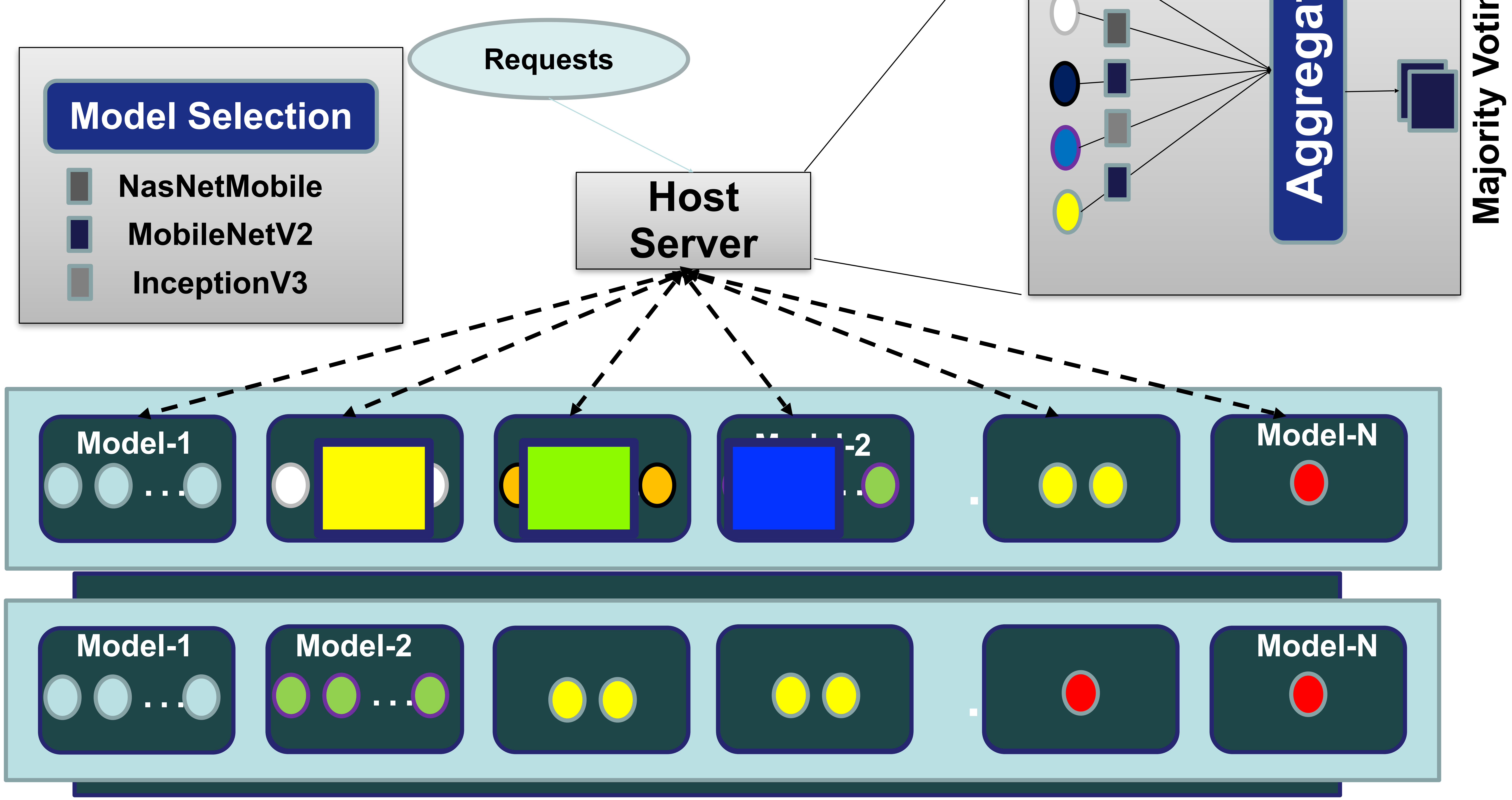
Cloud Resources for Individual Models (Virtual Machines)

Model Ensembling Framework



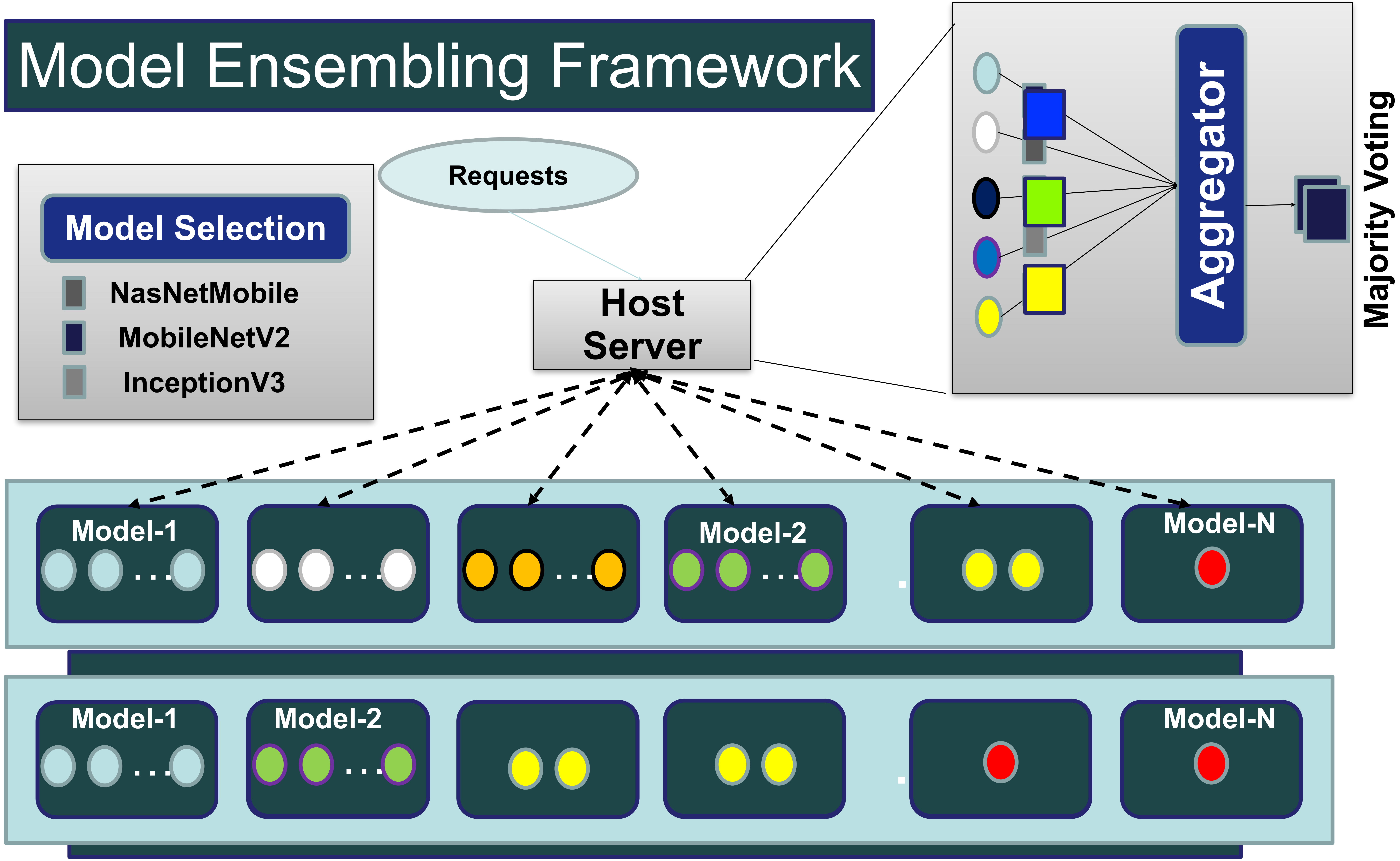
Cloud Resources for Individual Models (Virtual Machines)

Model Ensembling Framework



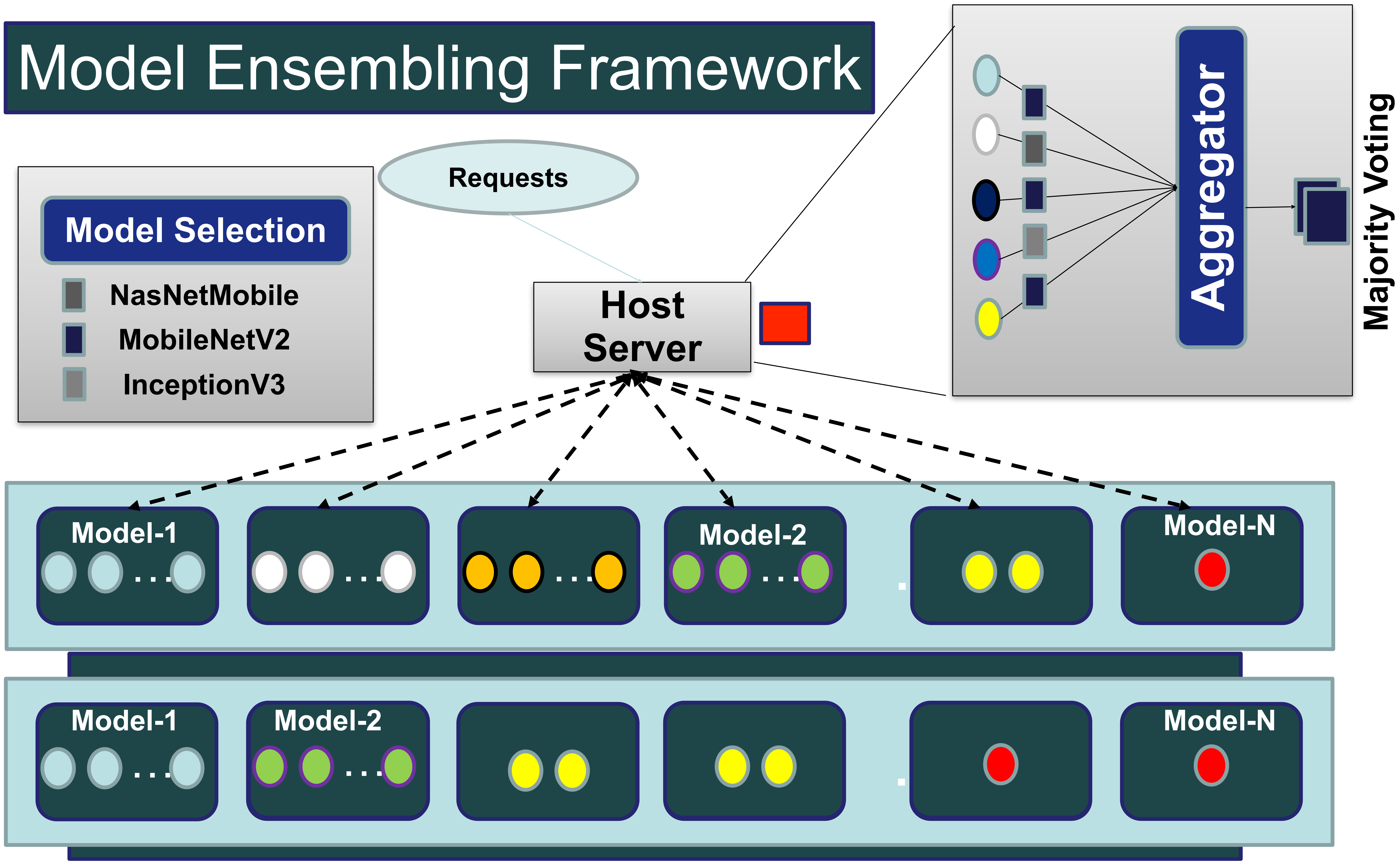
Cloud Resources for Individual Models (Virtual Machines)

Model Ensembling Framework



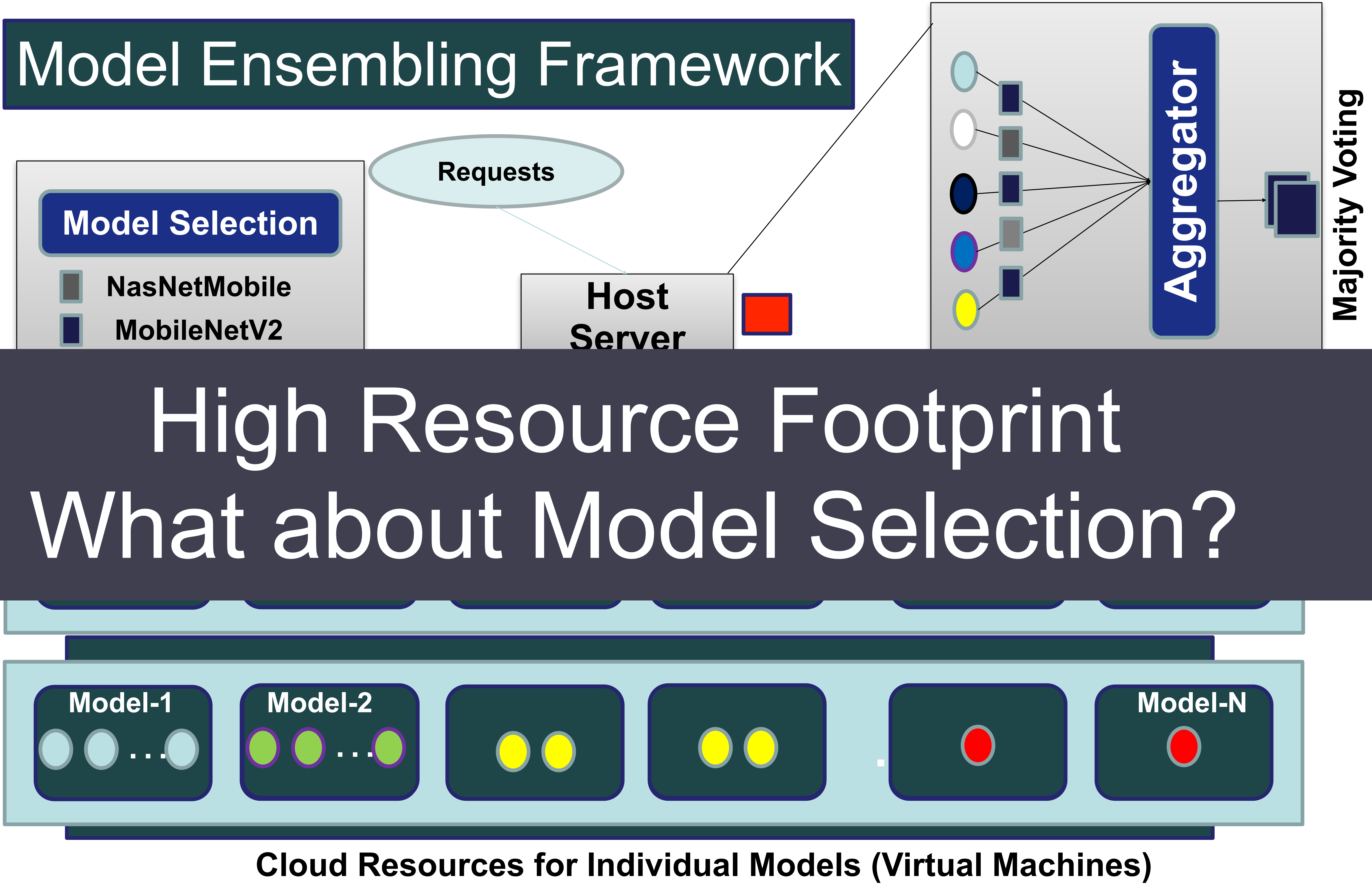
Cloud Resources for Individual Models (Virtual Machines)

Model Ensembling Framework



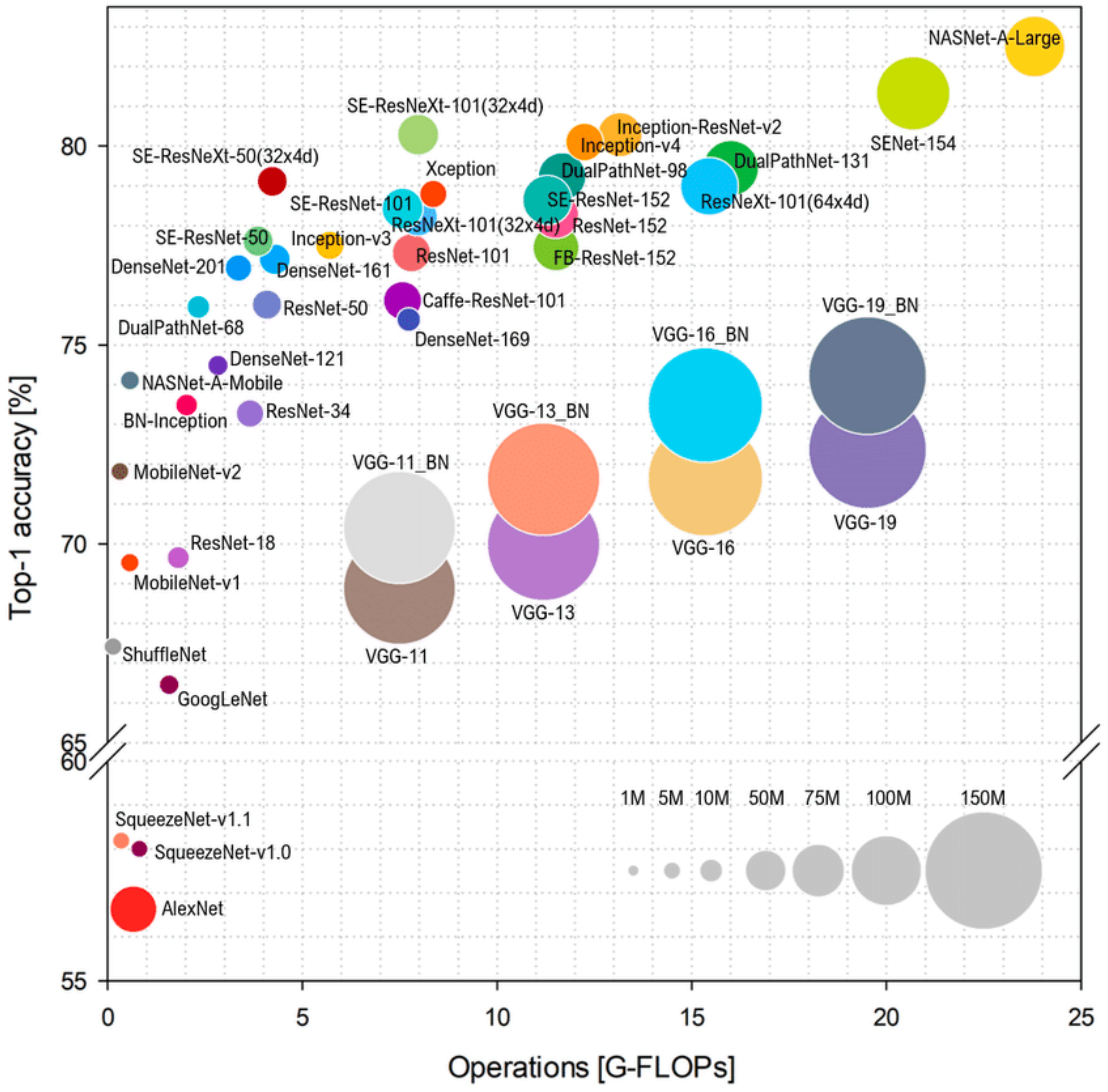
Cloud Resources for Individual Models (Virtual Machines)

Model Ensembling Framework



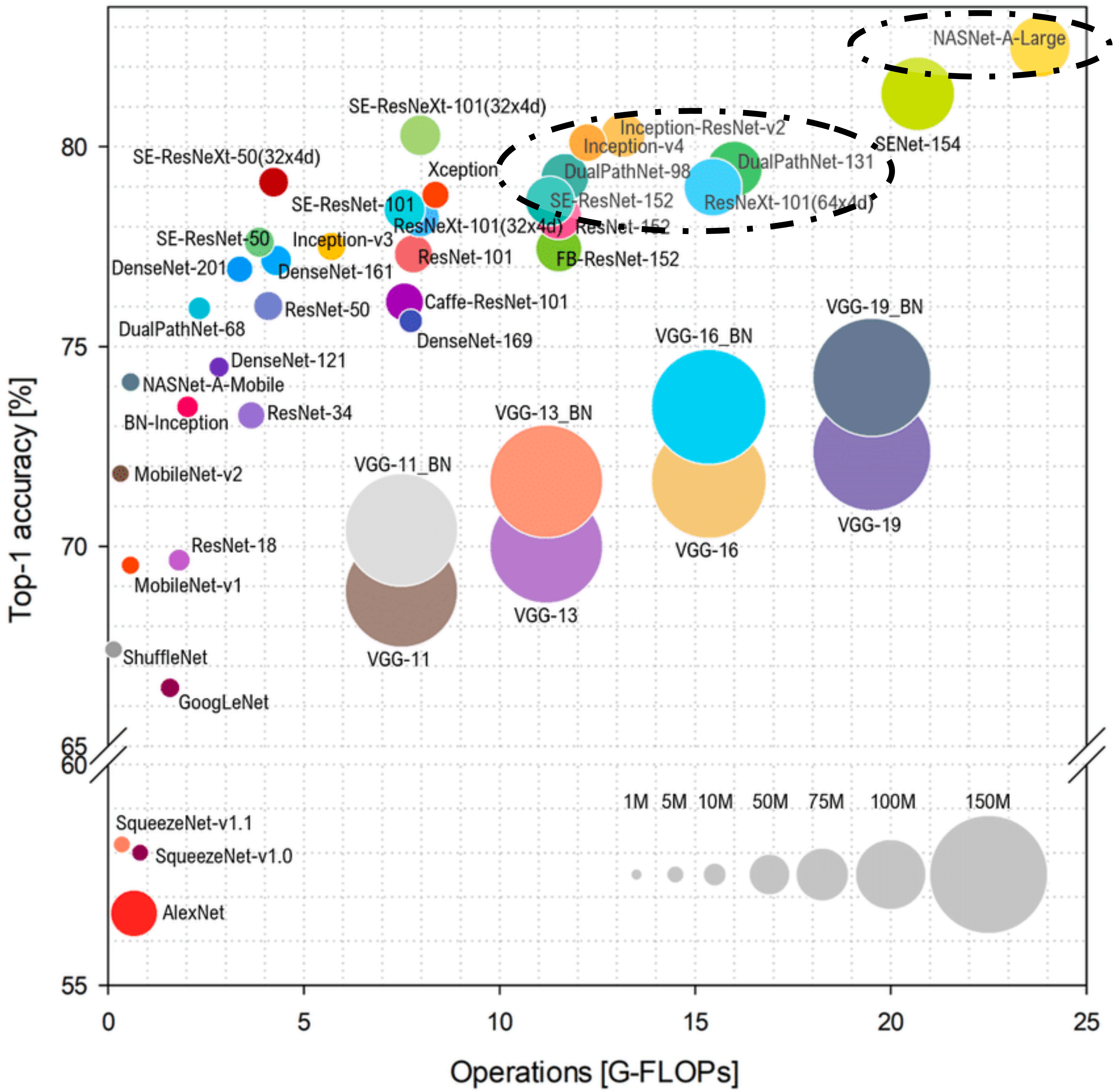
High Resource Footprint
What about Model Selection?

MODEL SPACE EXPLORATION



IEEE Access'18 Benchmark Analysis of Representative Deep Neural Network Architectures

MODEL SPACE EXPLORATION

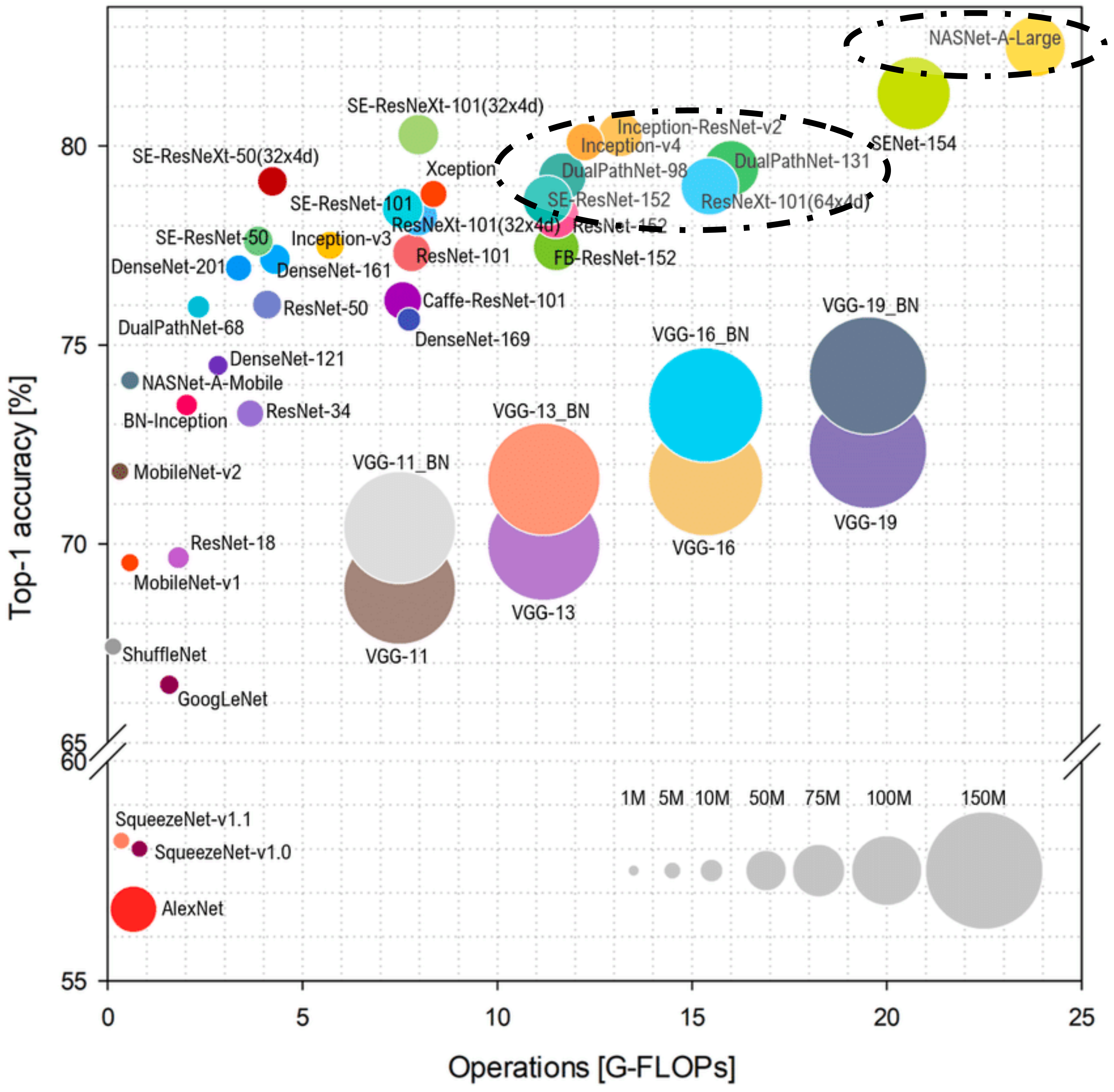


IEEE Access'18 Benchmark Analysis of Representative Deep Neural Network Architectures

Most accurate model

- ✴**~2x** parameters, latency
- ✴**~2%** more accuracy

Model Space Exploration



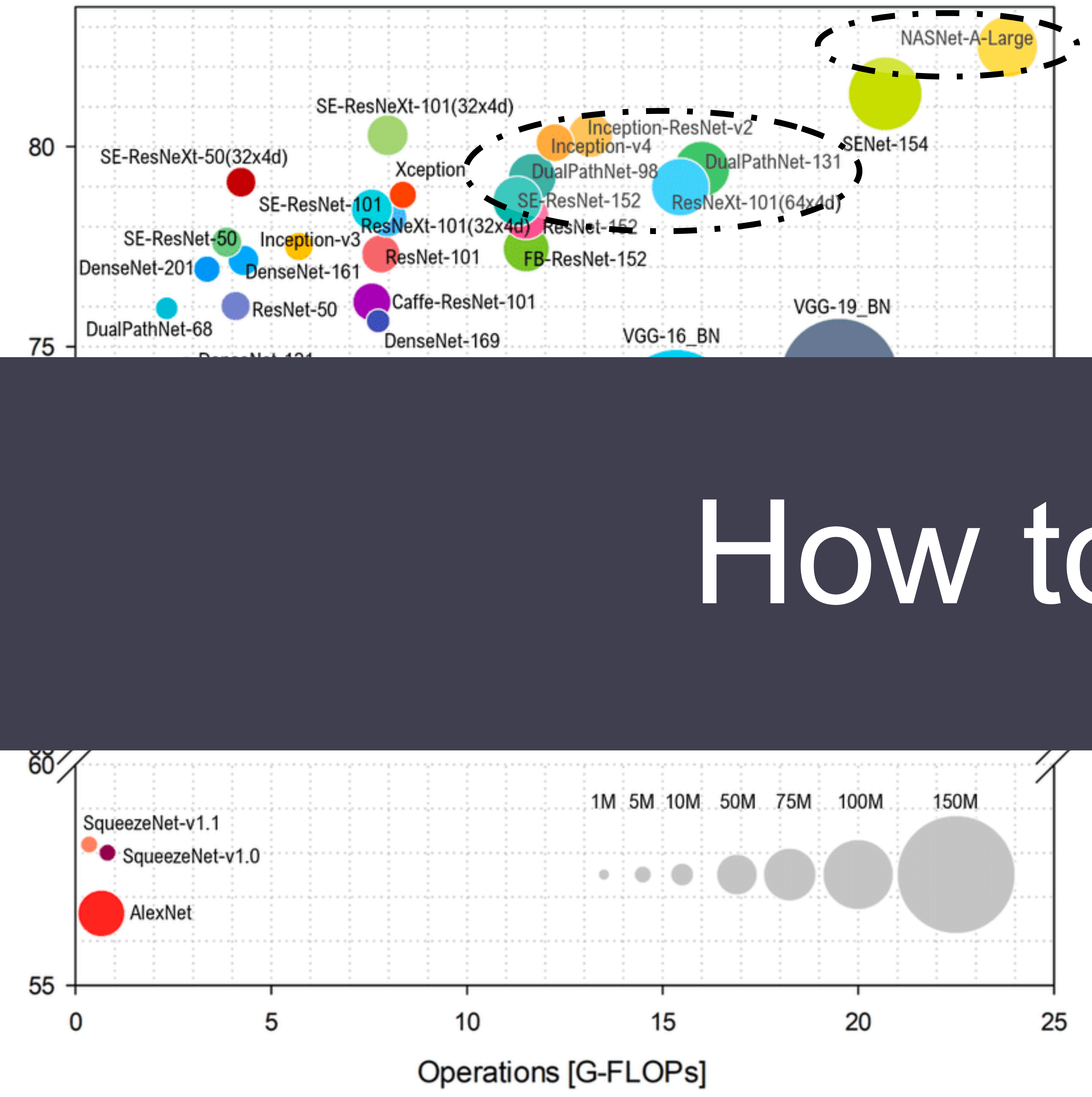
IEEE Access'18 Benchmark Analysis of Representative Deep Neural Network Architectures

Most accurate model

- ✶**~2x** parameters, latency
- ✶**~2%** more accuracy

- How to bridge the 2% accuracy gap?
- What about cost?

Model Space Exploration



IEEE Access'18 Benchmark Analysis of Representative Deep Neural Network Architectures

Most accurate model

- ✴**~2x** parameters, latency
- ✴**~2%** more accuracy

How to ensemble?

- What about cost?

Full Ensemble

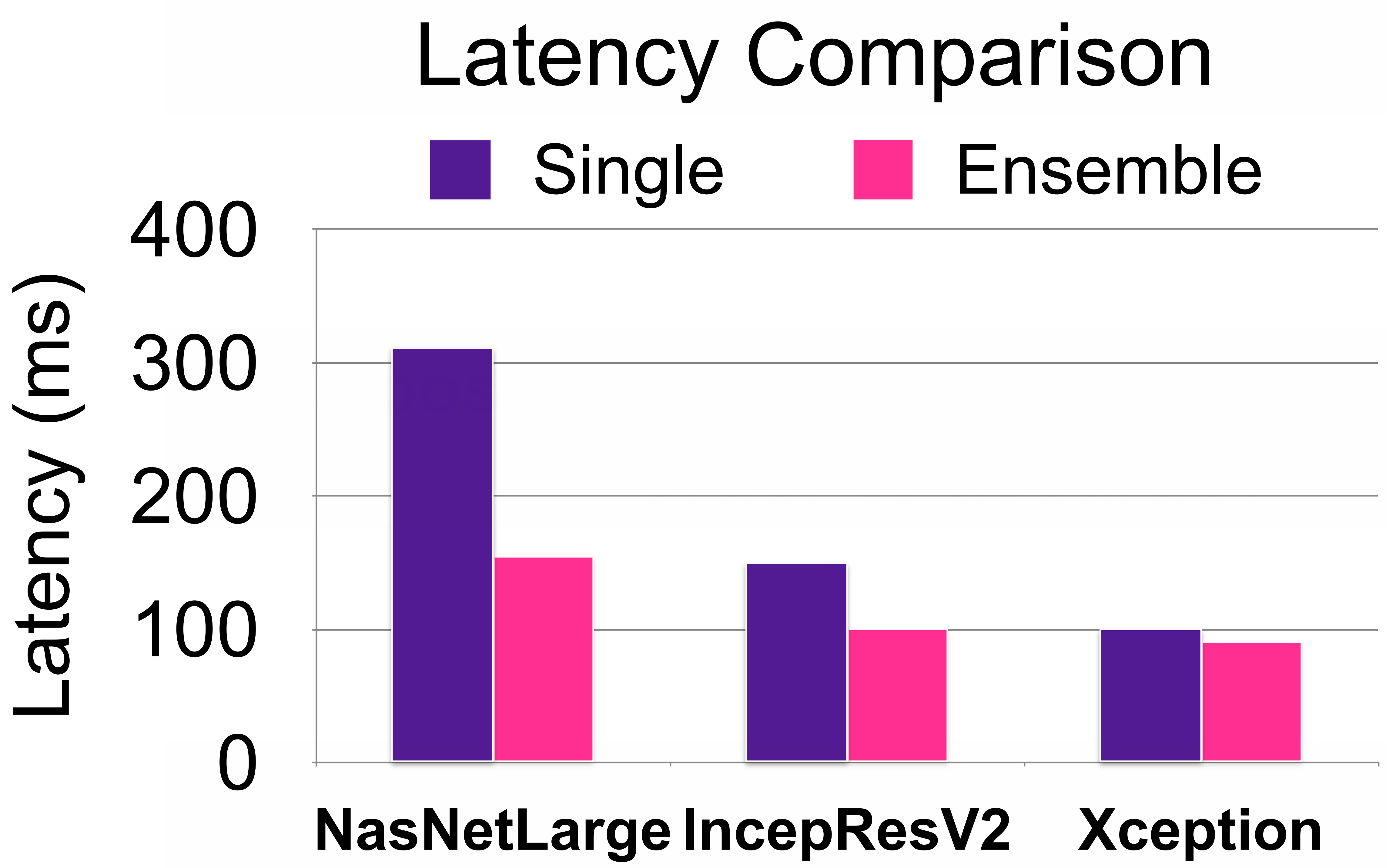
Model Set: Top 12 frequently used models from Keras Tensorflow

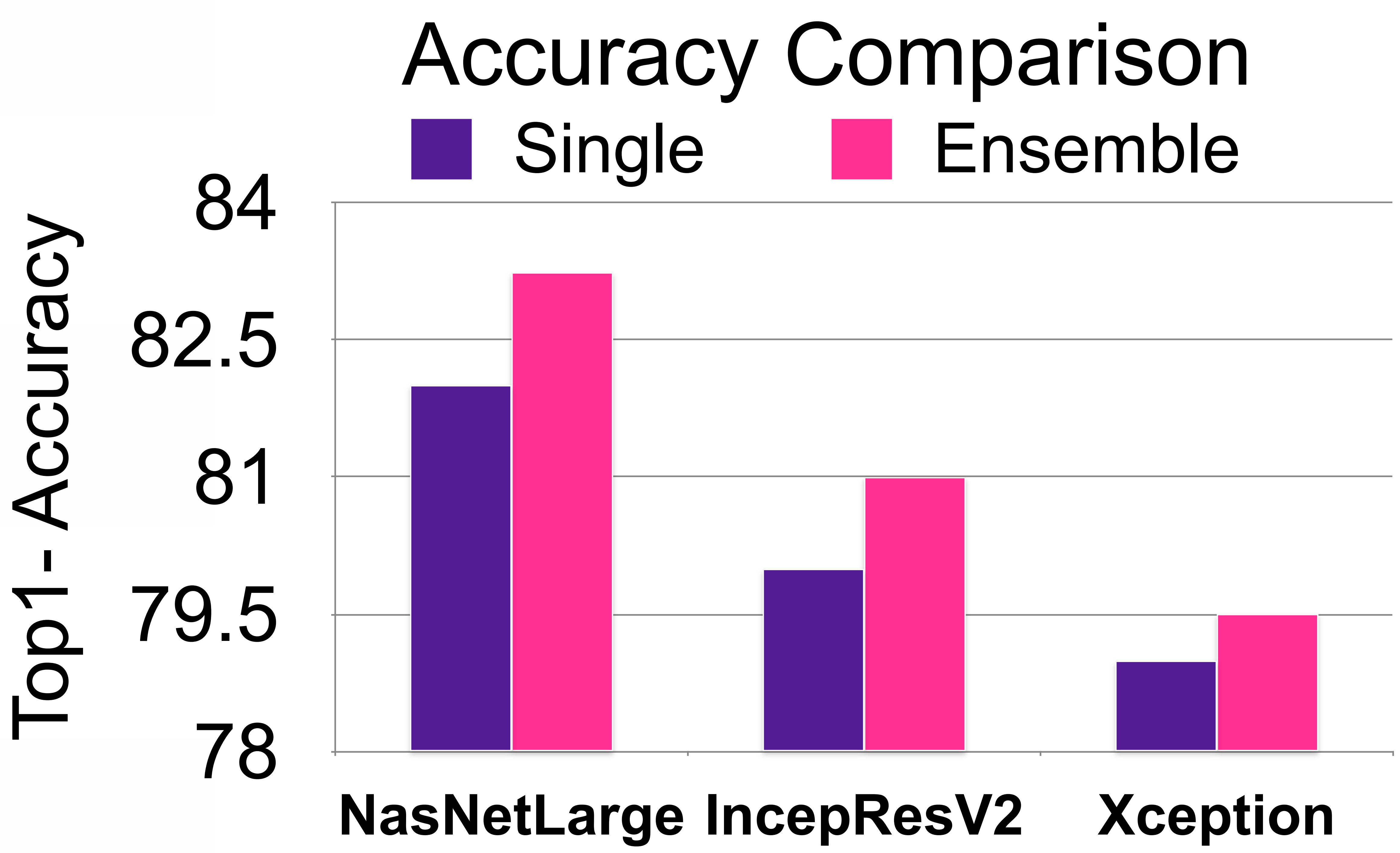
Choose baseline models in decreasing order of accuracy

Combine all models which are under the latency of baseline model.

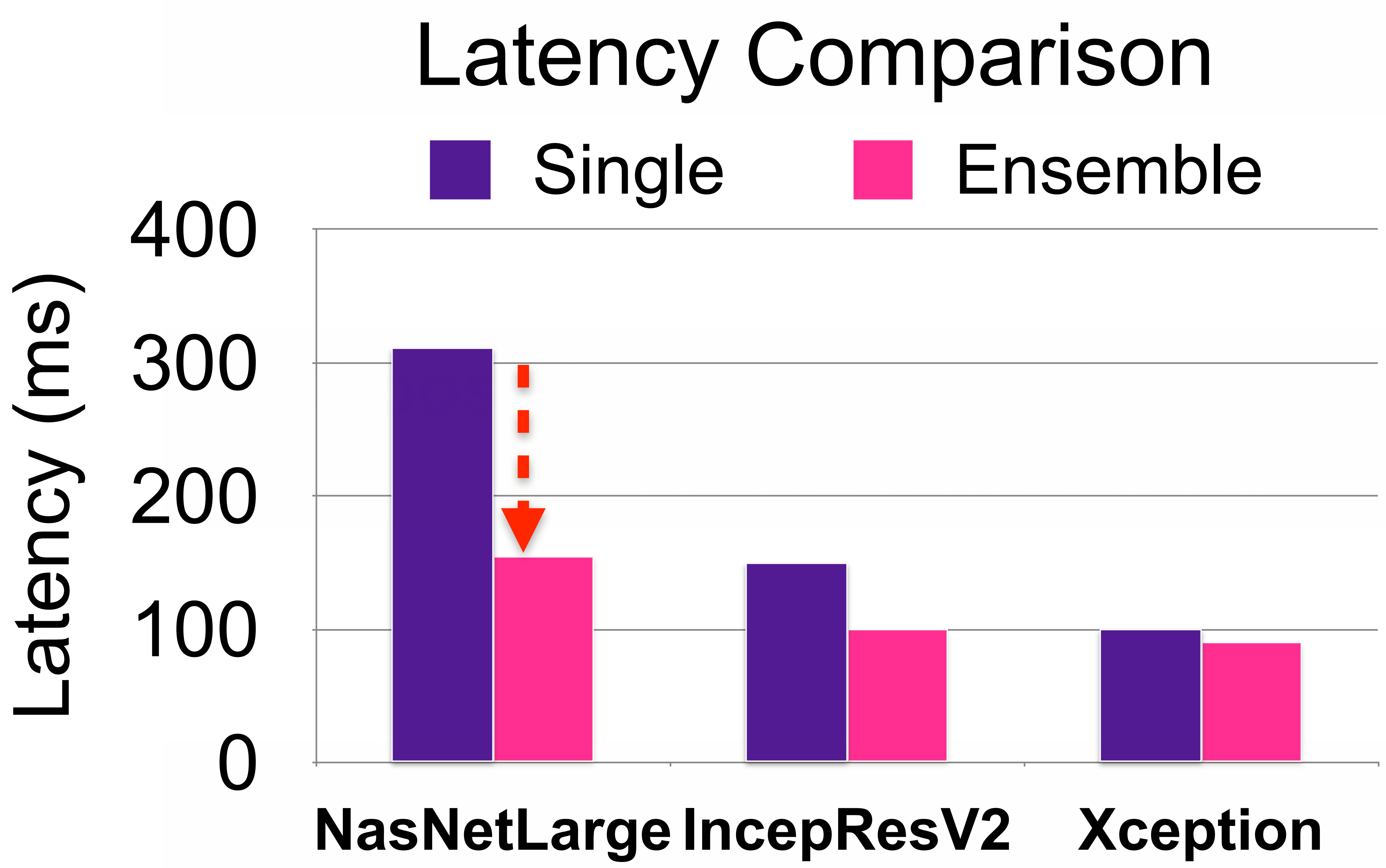
Model (Acronym)	Params (10k)	Top-1 Accuracy(%)	Latency (ms)	P_f
MobileNetV1 (MNet)	4,253	70.40	43.45	10
MobileNetV2 (MNetV2)	4,253	71.30	41.5	10
NASNetMobile (NASMob)	5,326	74.40	78.18	3
DenseNet121 (DNet121)	8,062	75.00	102.35	3
DenseNet201 (DNet201)	20,242	77.30	152.21	2
Xception (Xcep)	22,910	79.00	119.2	4
Inception V3 (Incep)	23,851	77.90	89	5
ResNet50-V2 (RNet50)	25,613	76.00	89.5	6
Resnet50 (RNet50)	25,636	74.90	98.22	5
IncepResnetV2 (IRV2)	55,873	80.30	151.96	1
NasNetLarge (NasLarge)	343,000	82.00	311	1

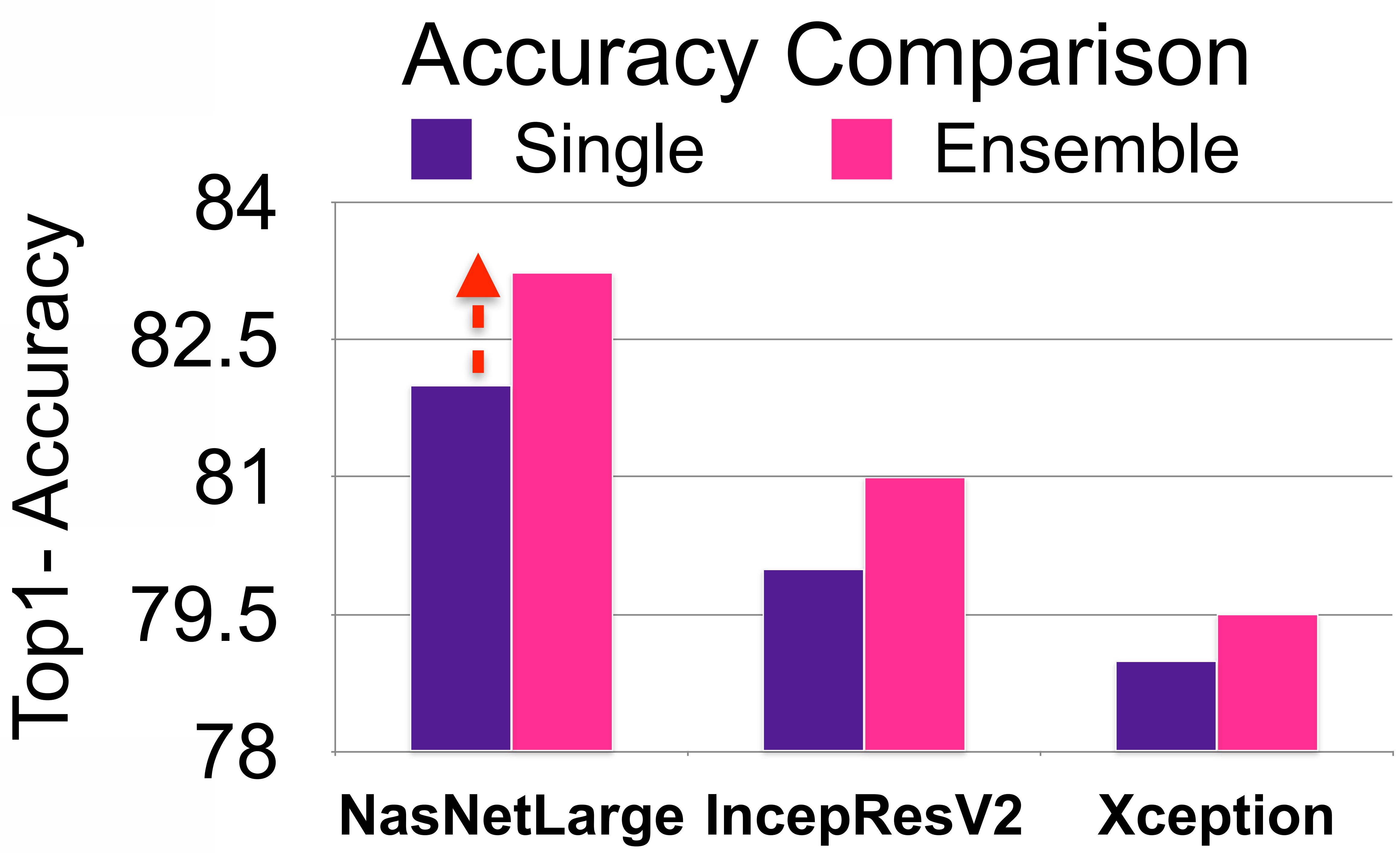
FULL ENSEMBLE





Full Ensemble

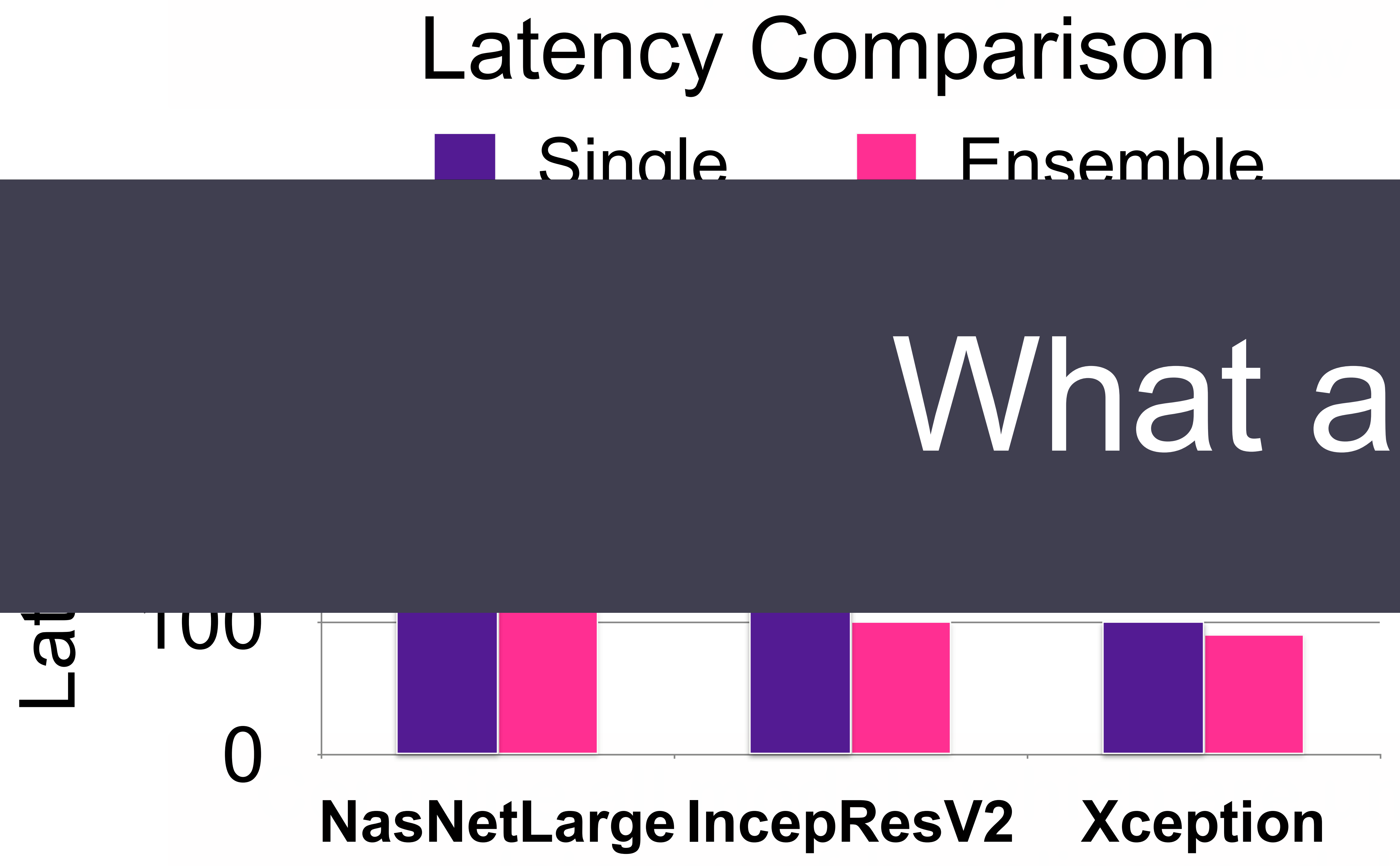


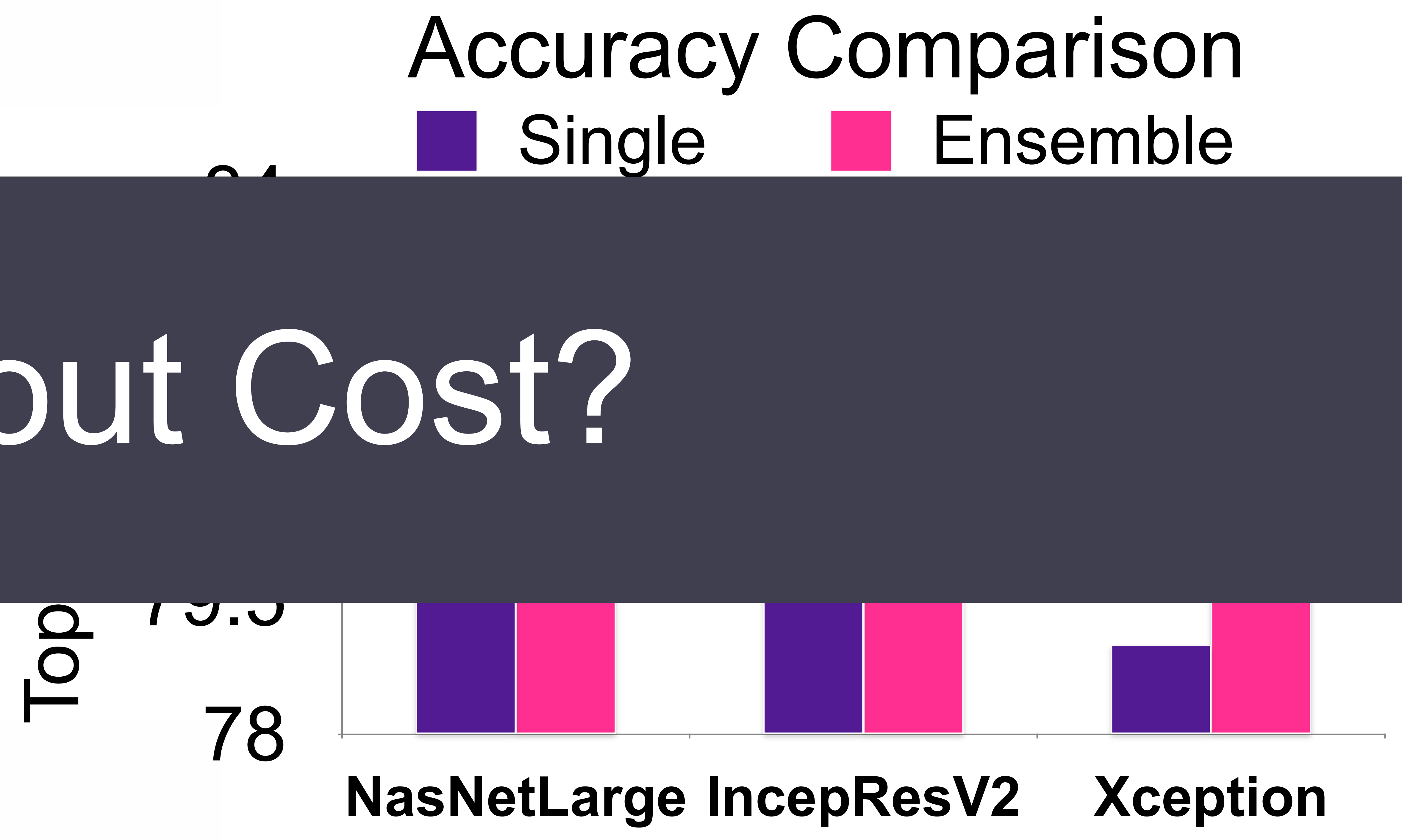


PennState College of Engineering

PennState High Performance Computing Lab

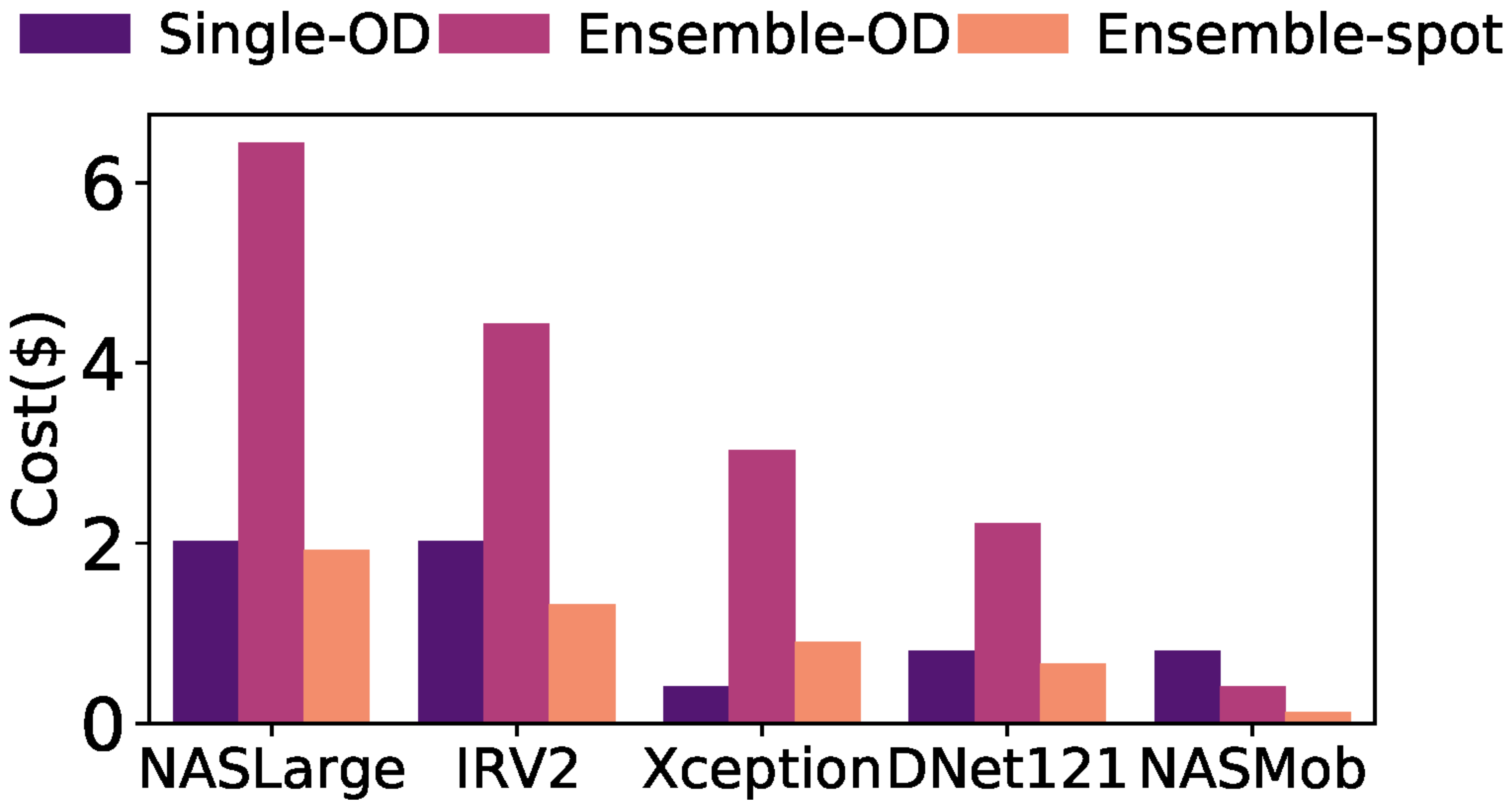
Full Ensemble



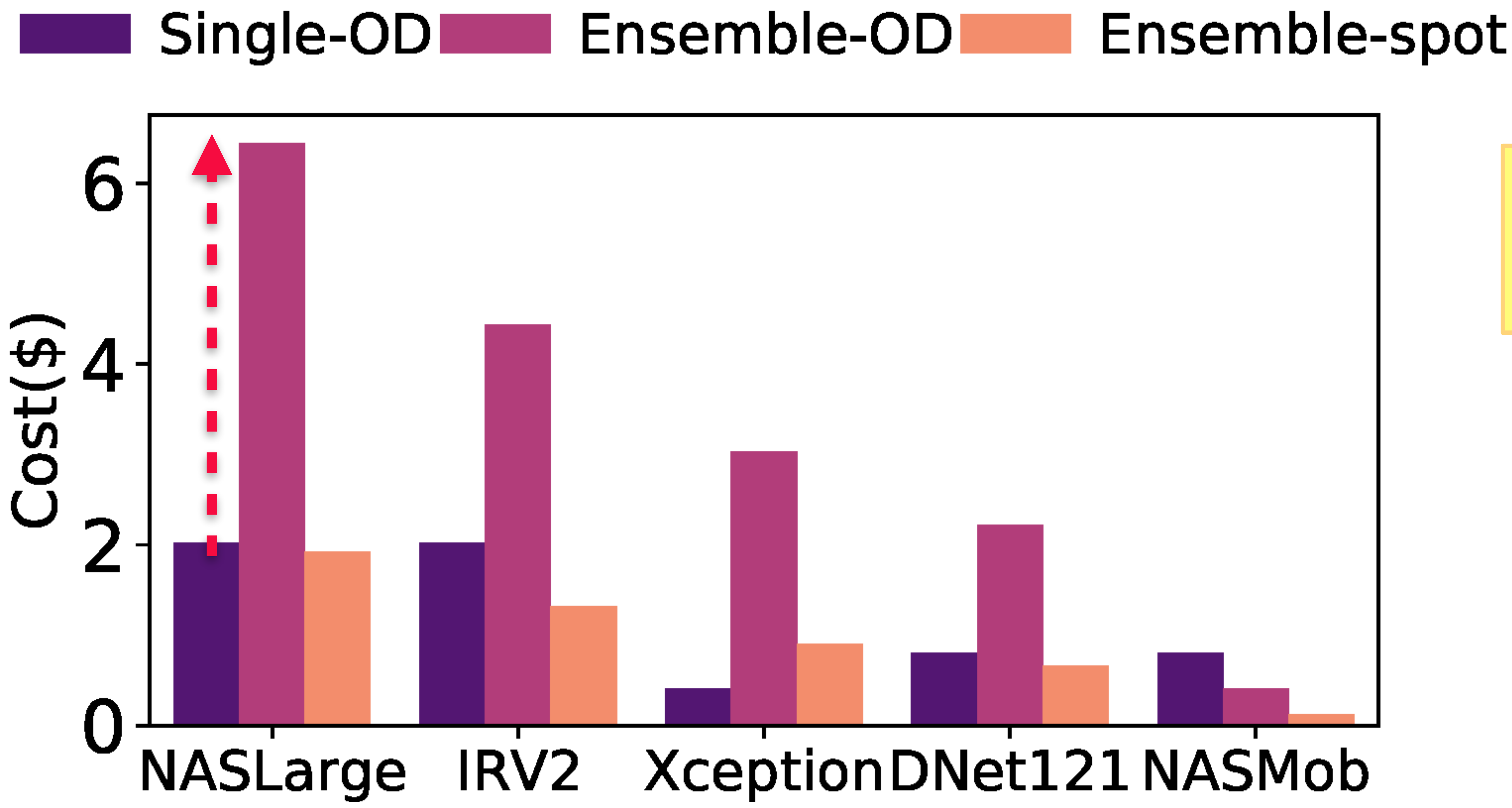


What about Cost?

FULL ENSEMBLING COST

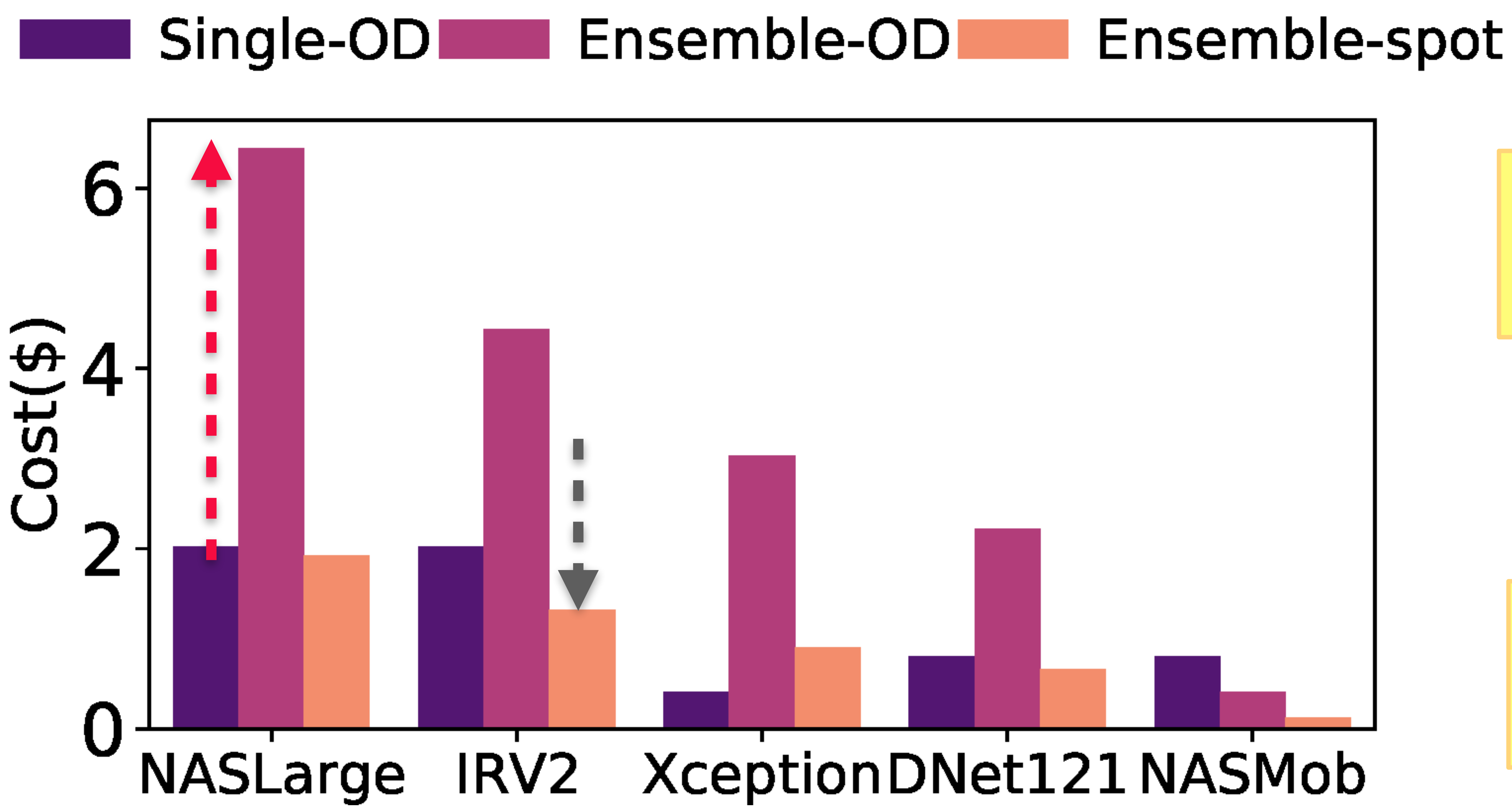


FULL ENSEMBLING COST



Ensembling is up-to 2x expensive.

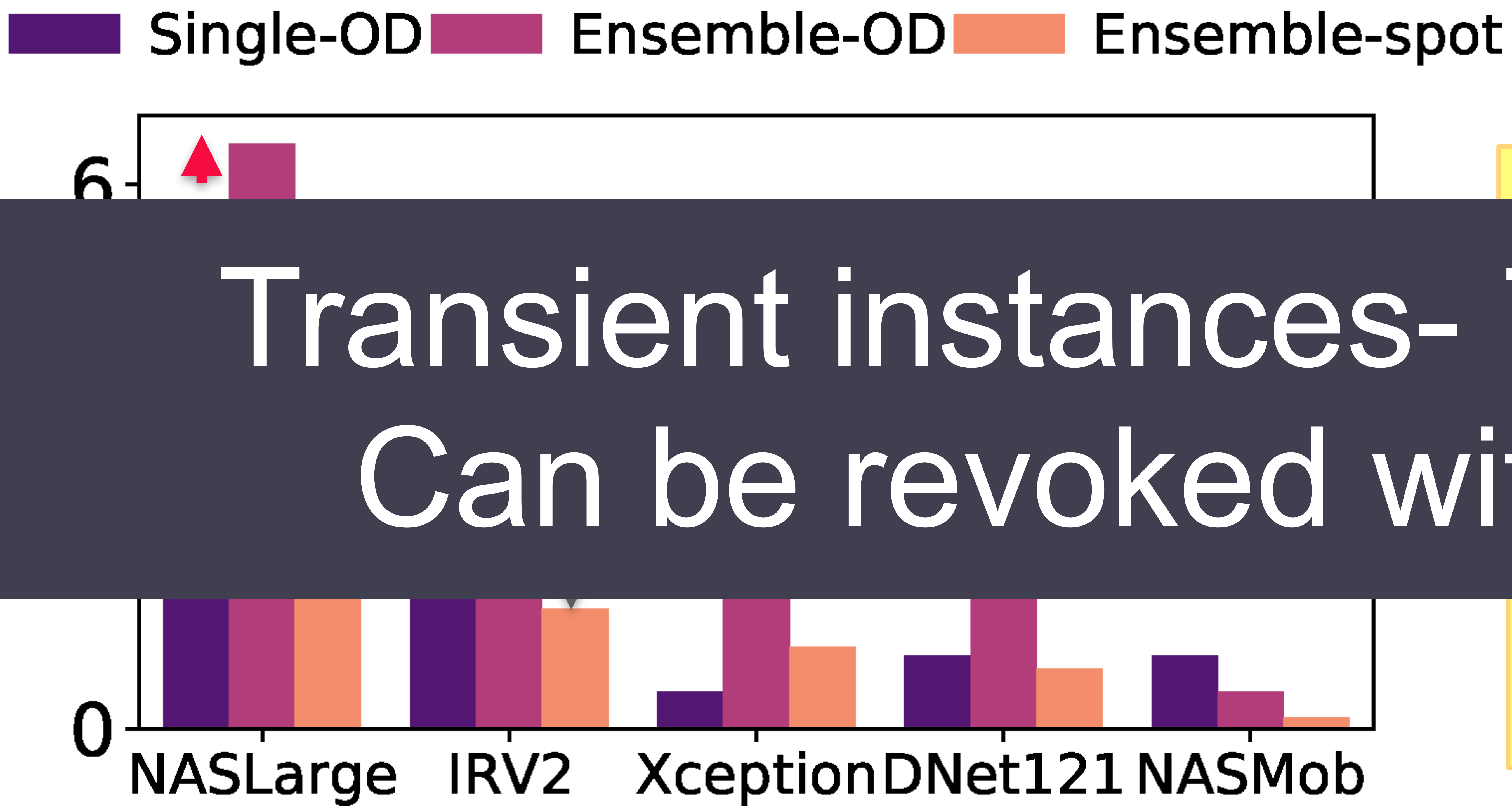
FULL ENSEMBLING COST



Ensembling is up-to 2x expensive.

Spot instances can reduce cost by 2x.

FULL ENSEMBLING COST



Ensembling is up to 2x

Spot instances can reduce cost by 2x.

Transient instances- 70-80% cheaper.
Can be revoked with short notice.

What can we do?

Baseline(BL)	NASLarge	IRV2	Xception	DNet121	NASMob
#Models	10	8	7	5	2

PennState College of Engineering

PennState High Performance Computing Lab

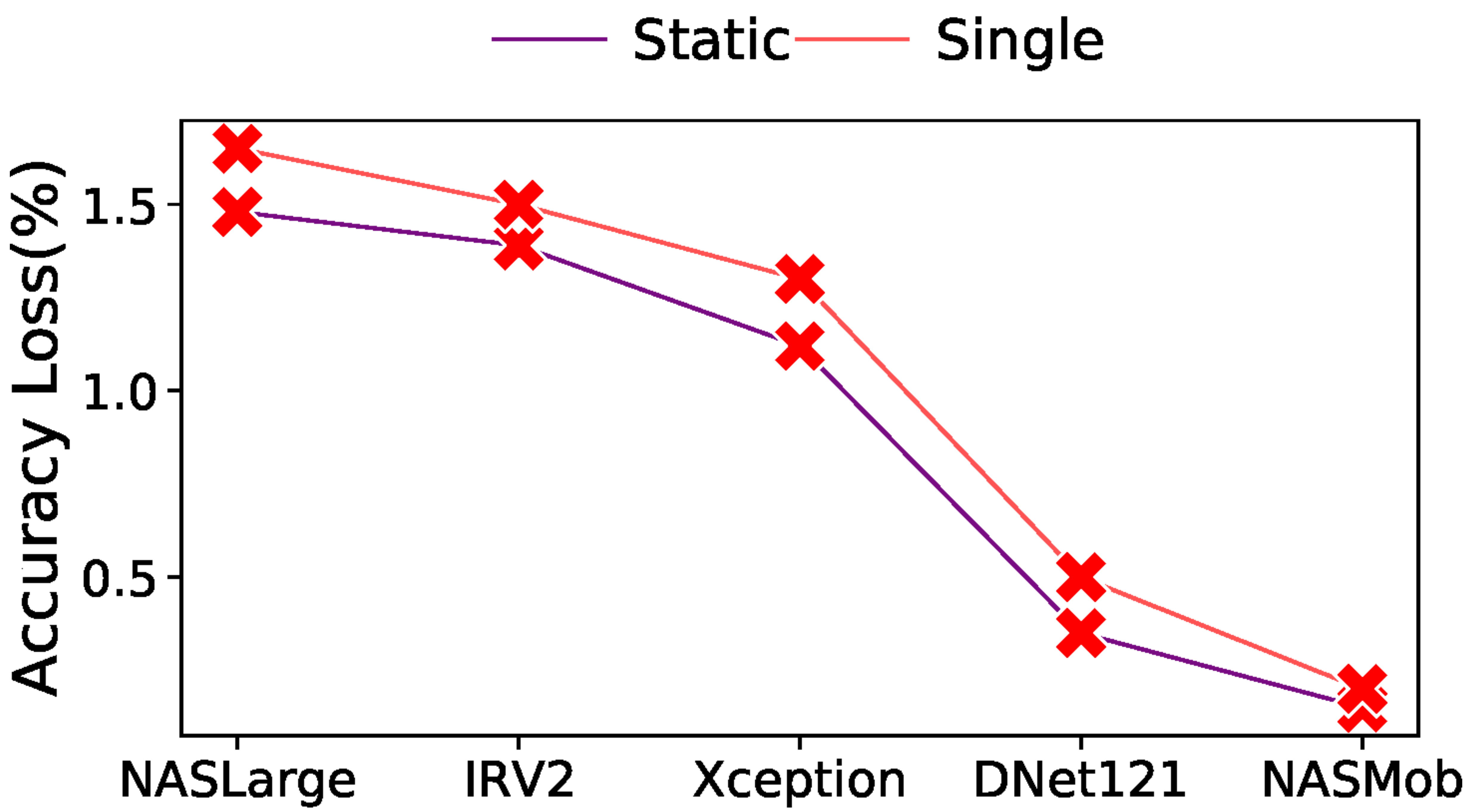
What can we do?

Baseline(BL)	NASLarge	IRV2	Xception	DNet121	NASMob
#Models	10	8	7	5	2

- Do we need so many models?
- How to autoscale resources for each model?
- How to handle instance failures?

Static Ensembling

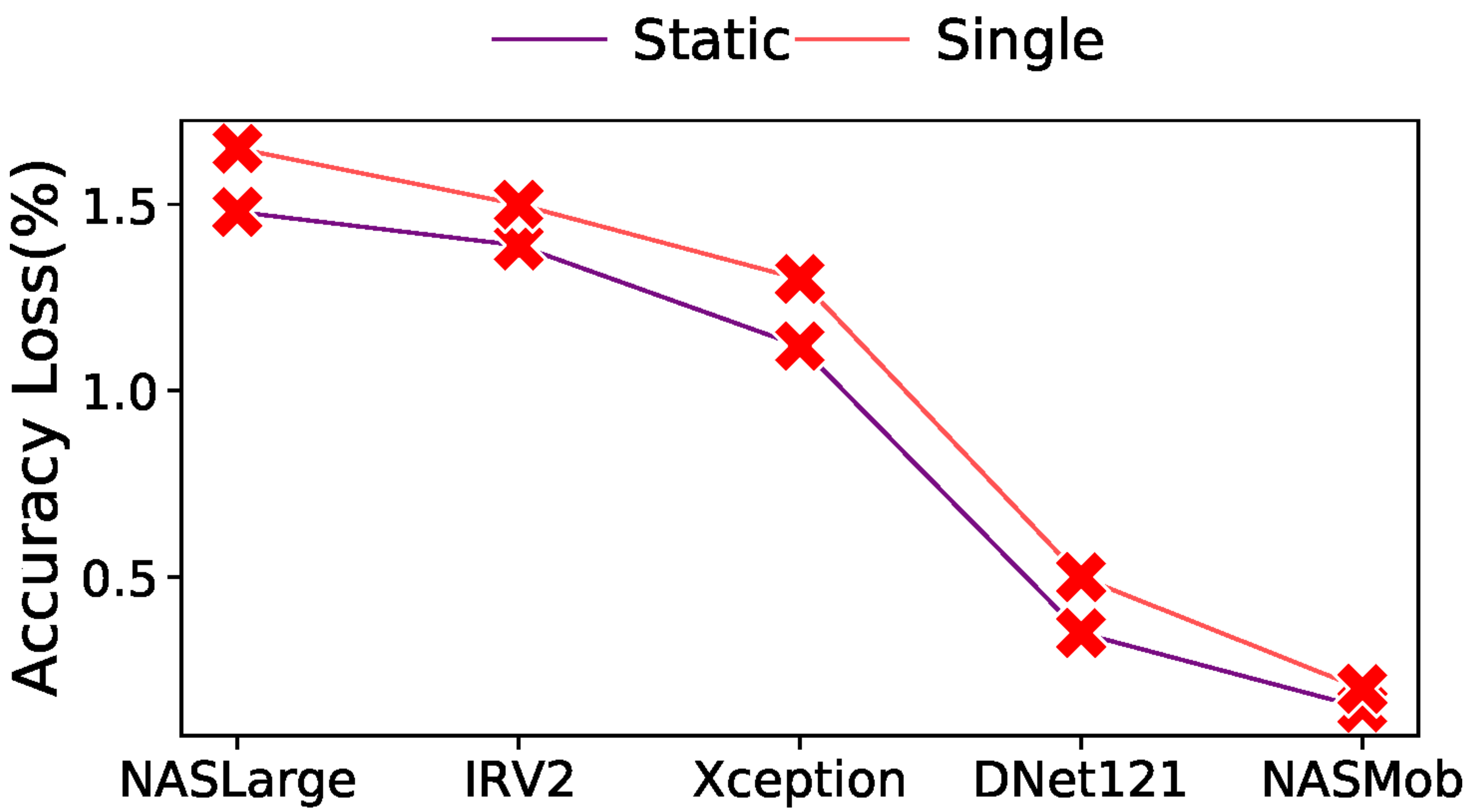
Compared to Full-Ensemble (N models)



Most accurate N/2 models

STATIC ENSEMBLING

Compared to Full-Ensemble (N models)

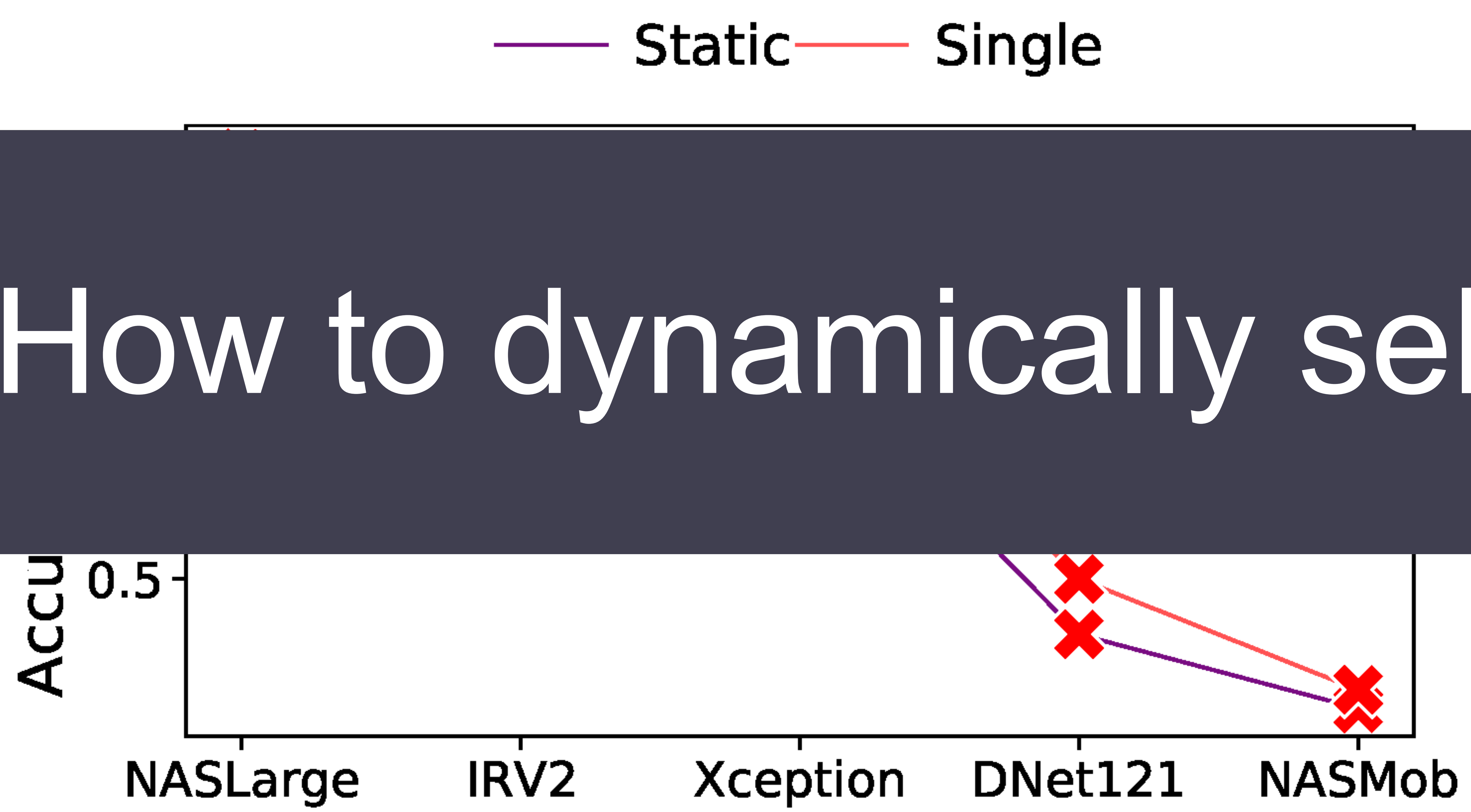


Most accurate N/2 models

Accuracy

Static Ensembling

Compared to Full-Ensemble (N models)



Most accurate N/2

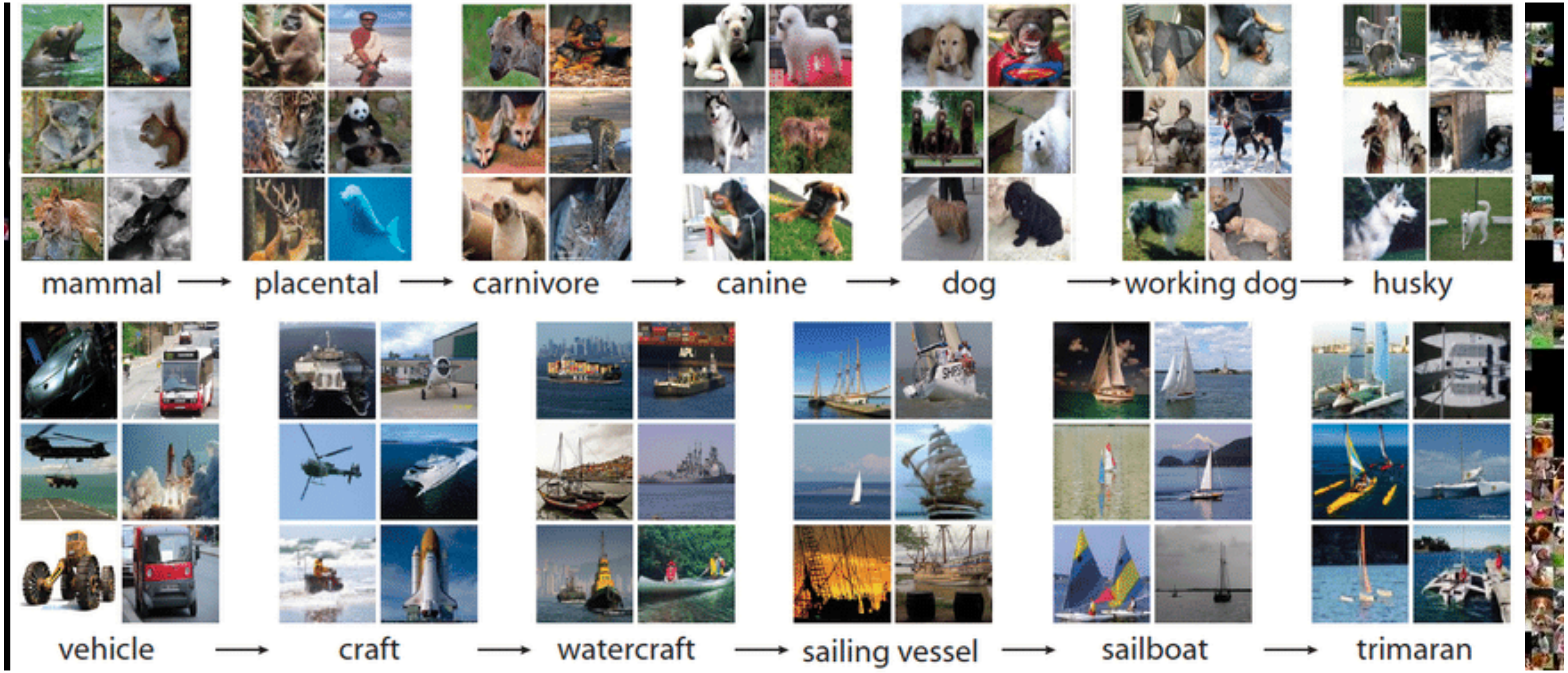
Accuracy ☺

☹

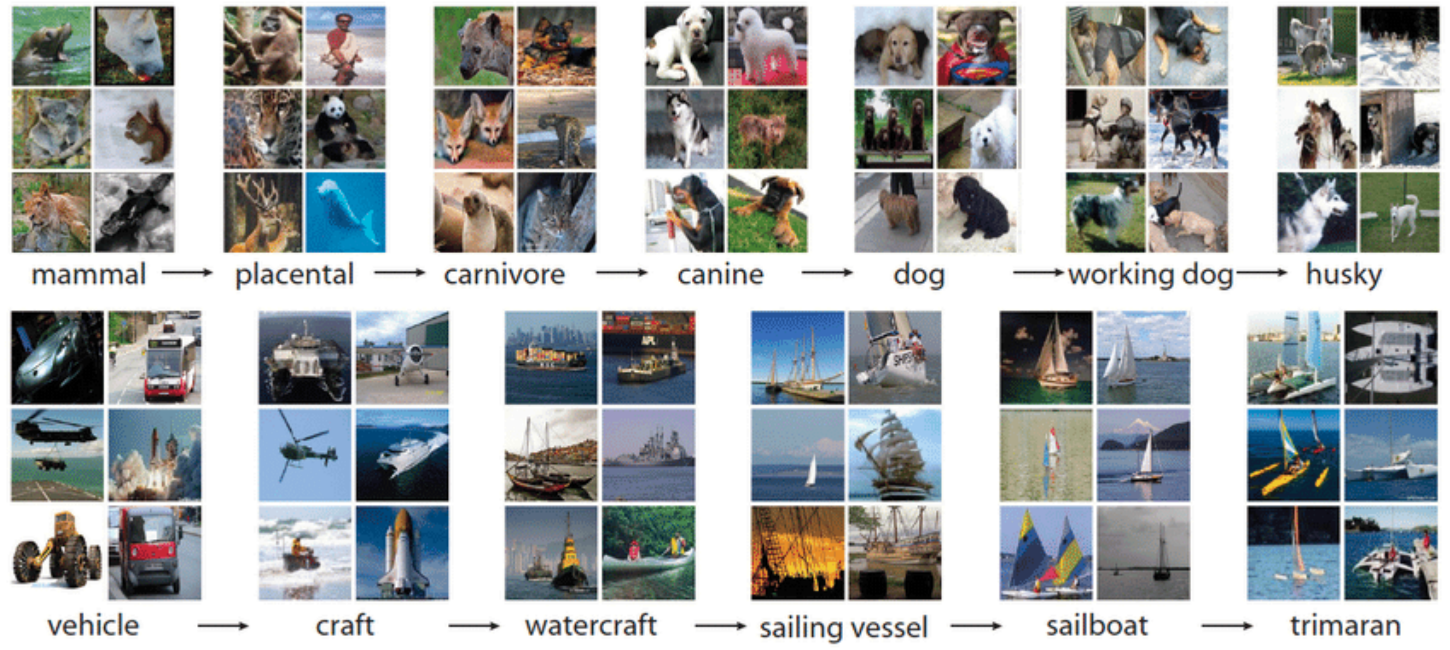
How to dynamically select the models?

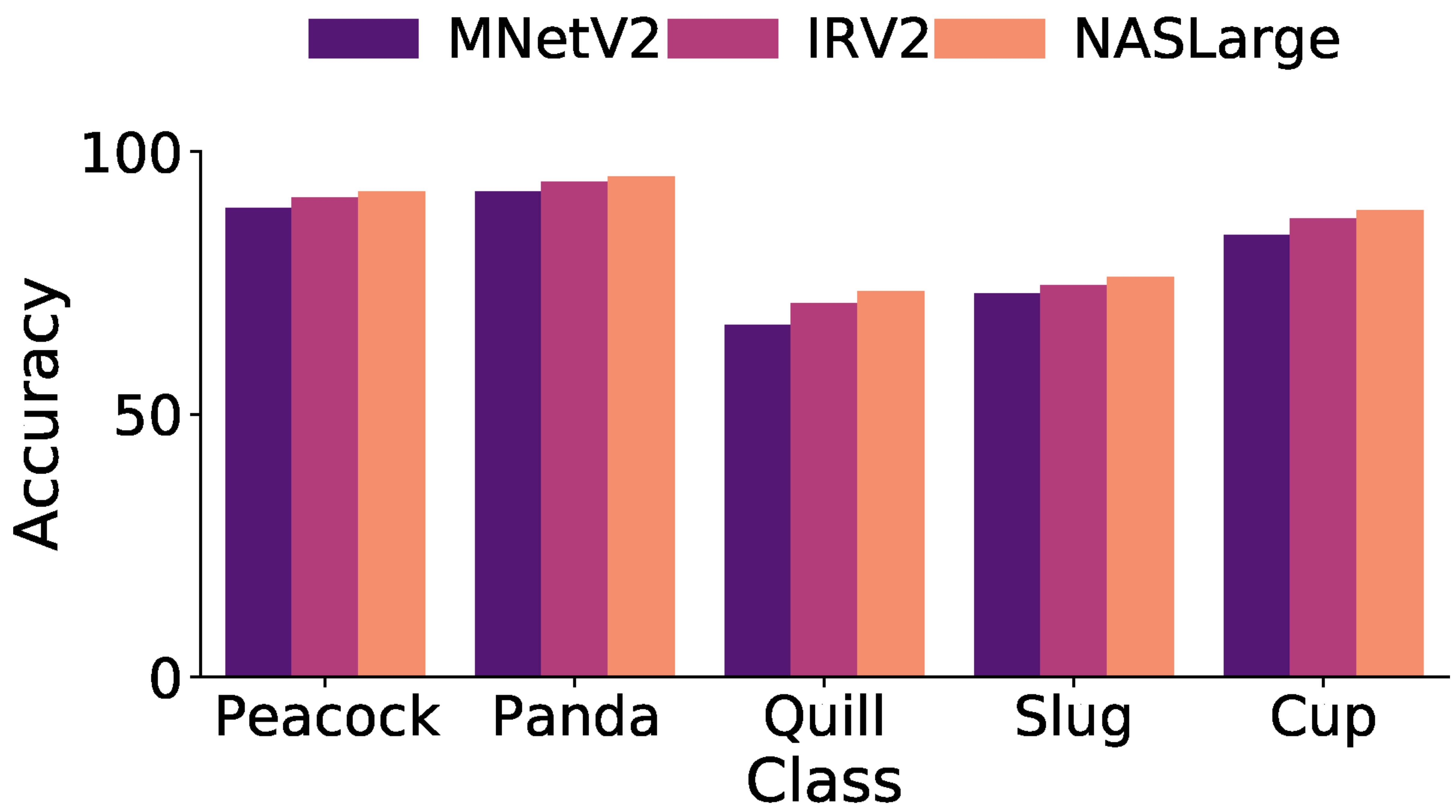
DYNAMIC MODEL SELECTION

Dynamic Model Selection



Dynamic Model Selection

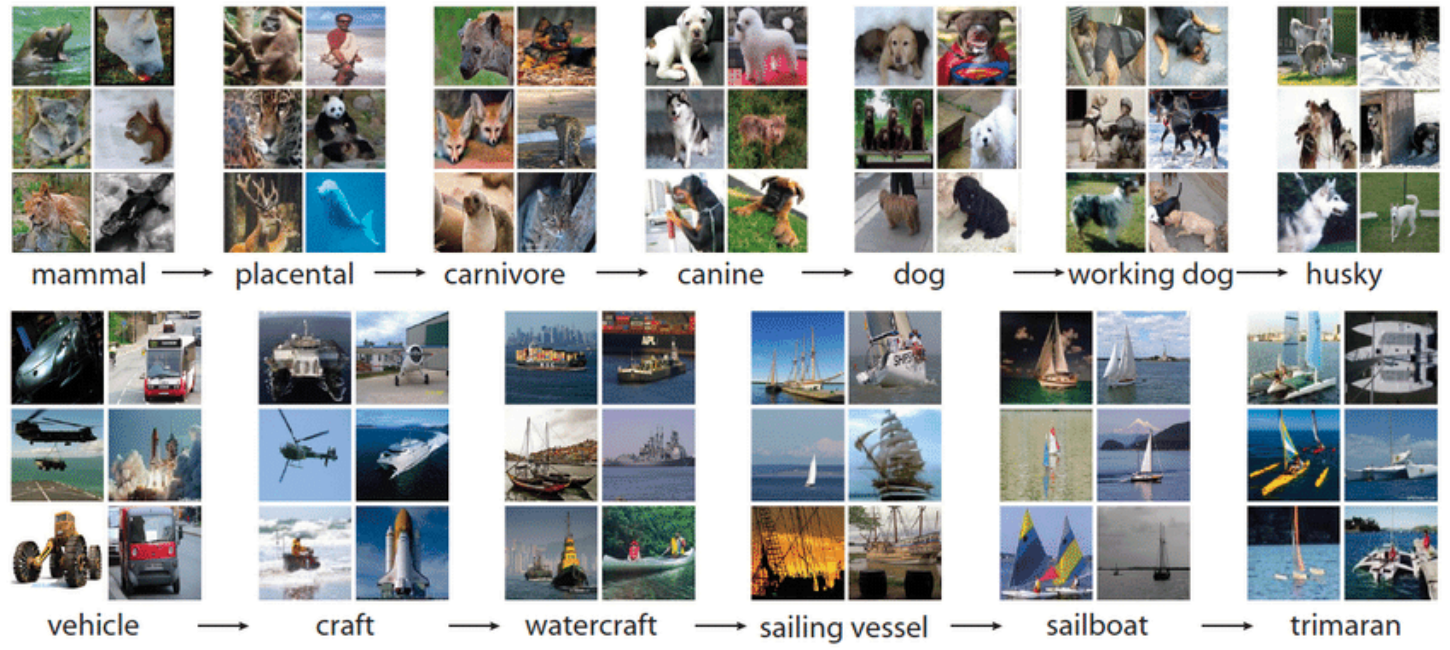


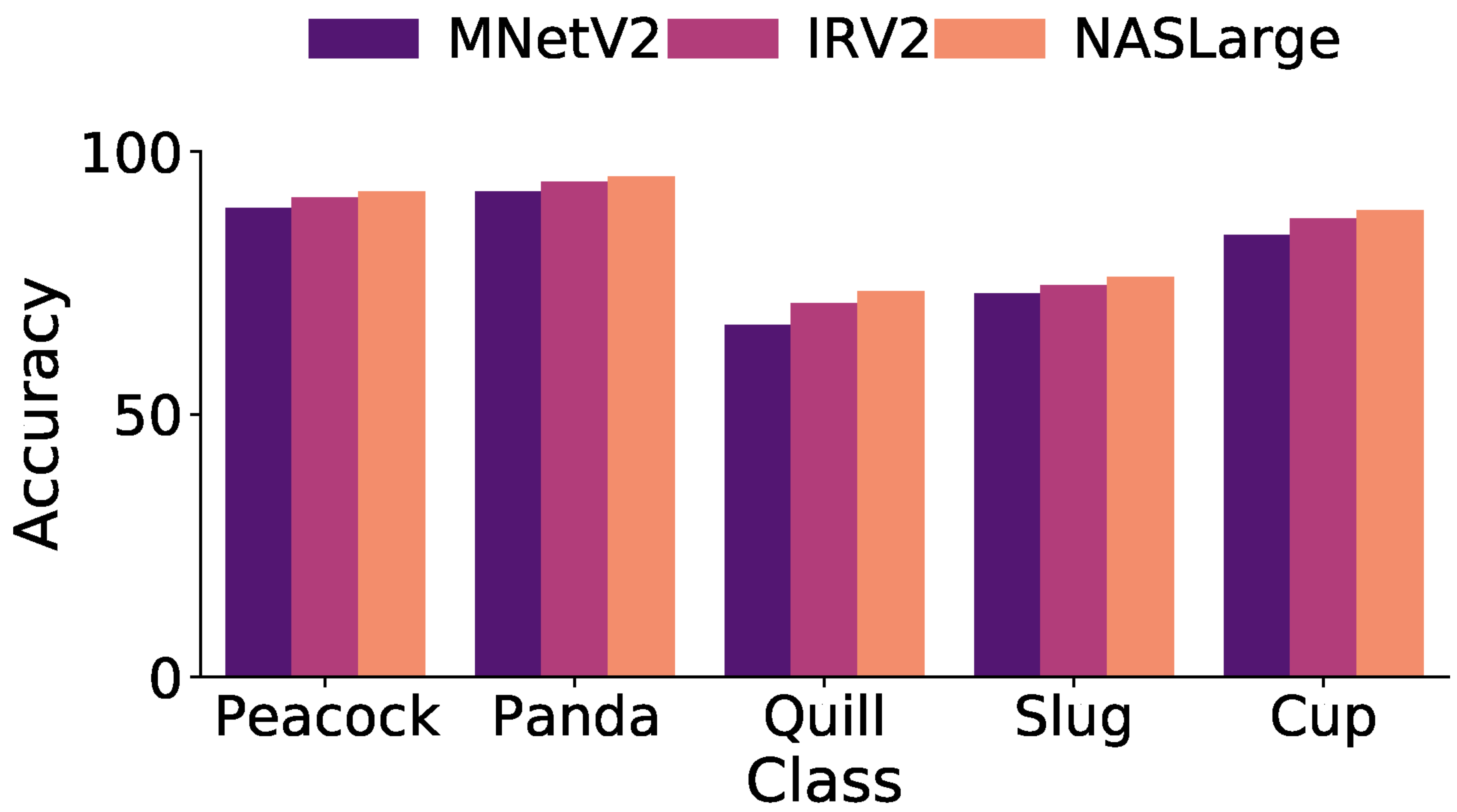


PennState College of Engineering

PennState High Performance Computing Lab

Dynamic Model Selection

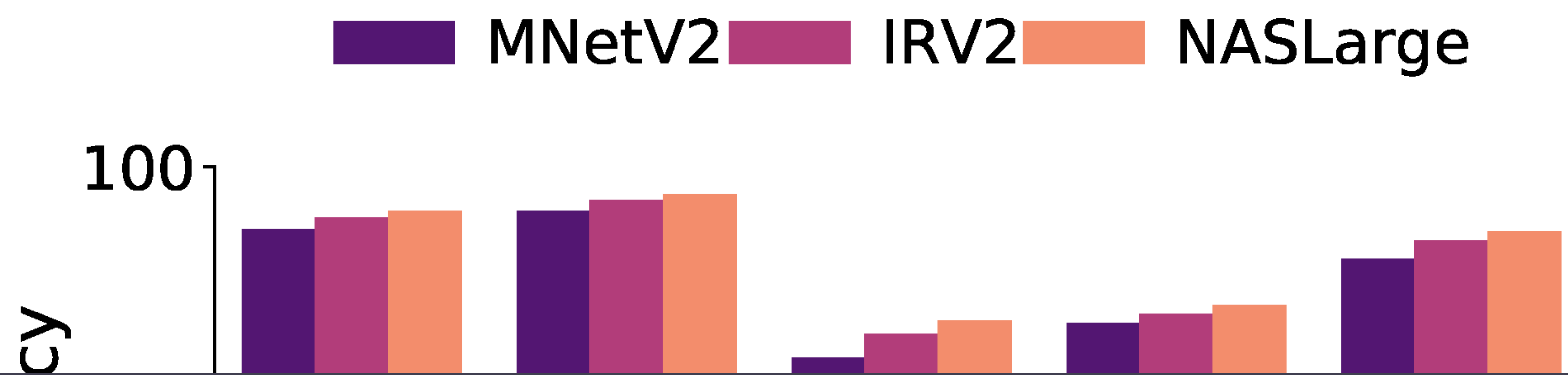




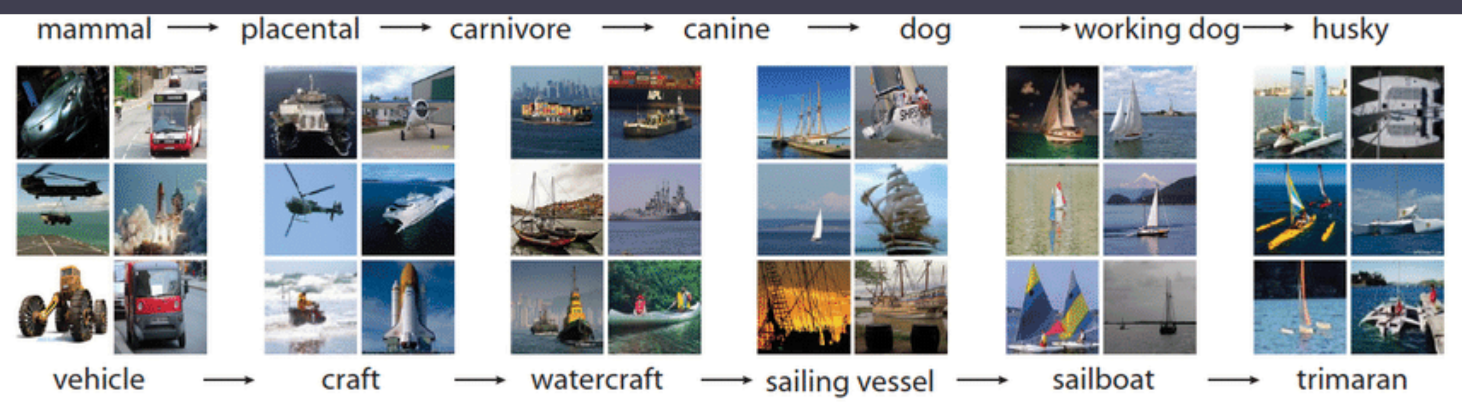
Mobilenet (MNet) Slug

Mobilenet (MNet) Quill

Dynamic Model Selection



Leverage Class-wise Accuracy

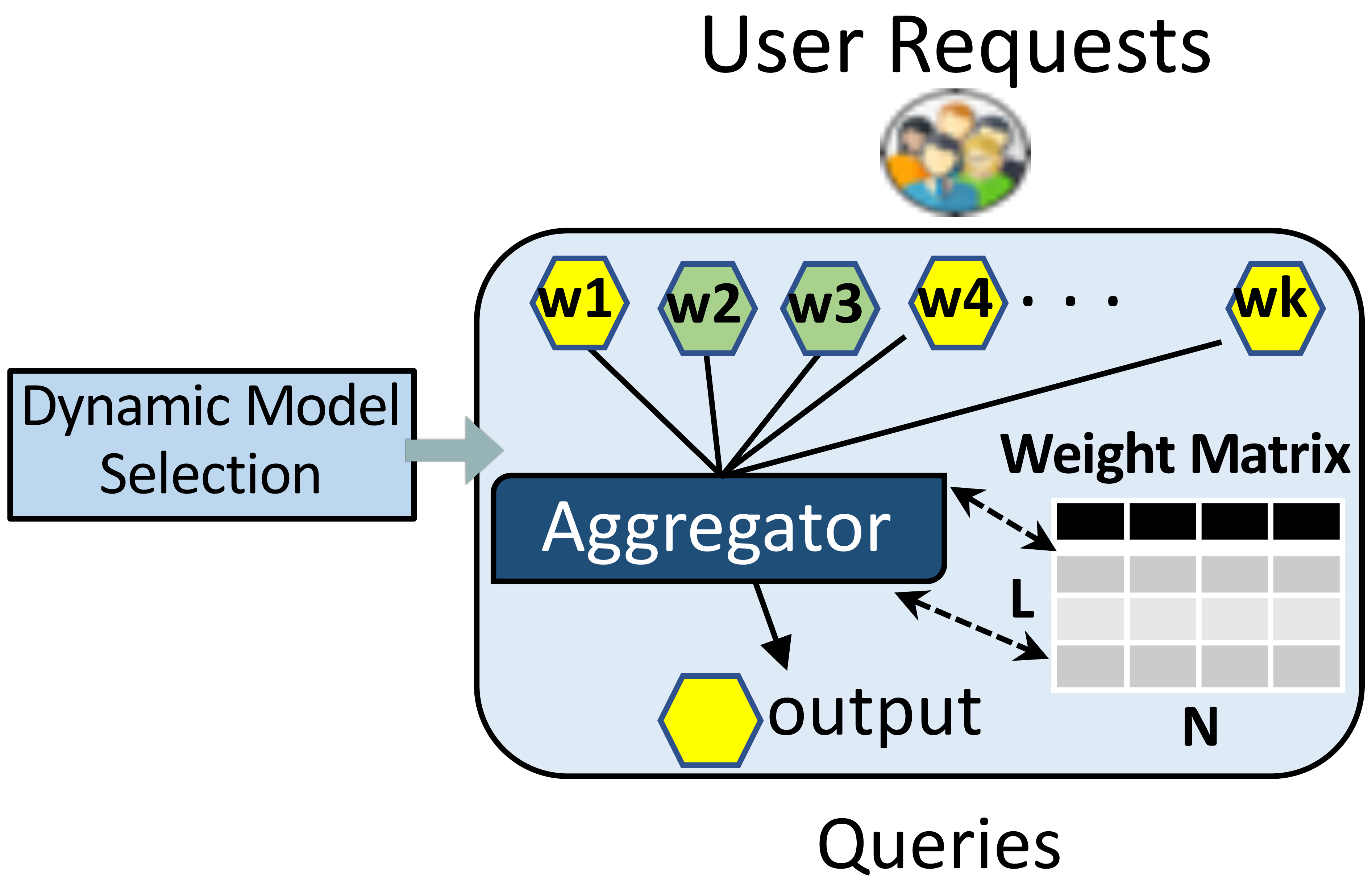


Class

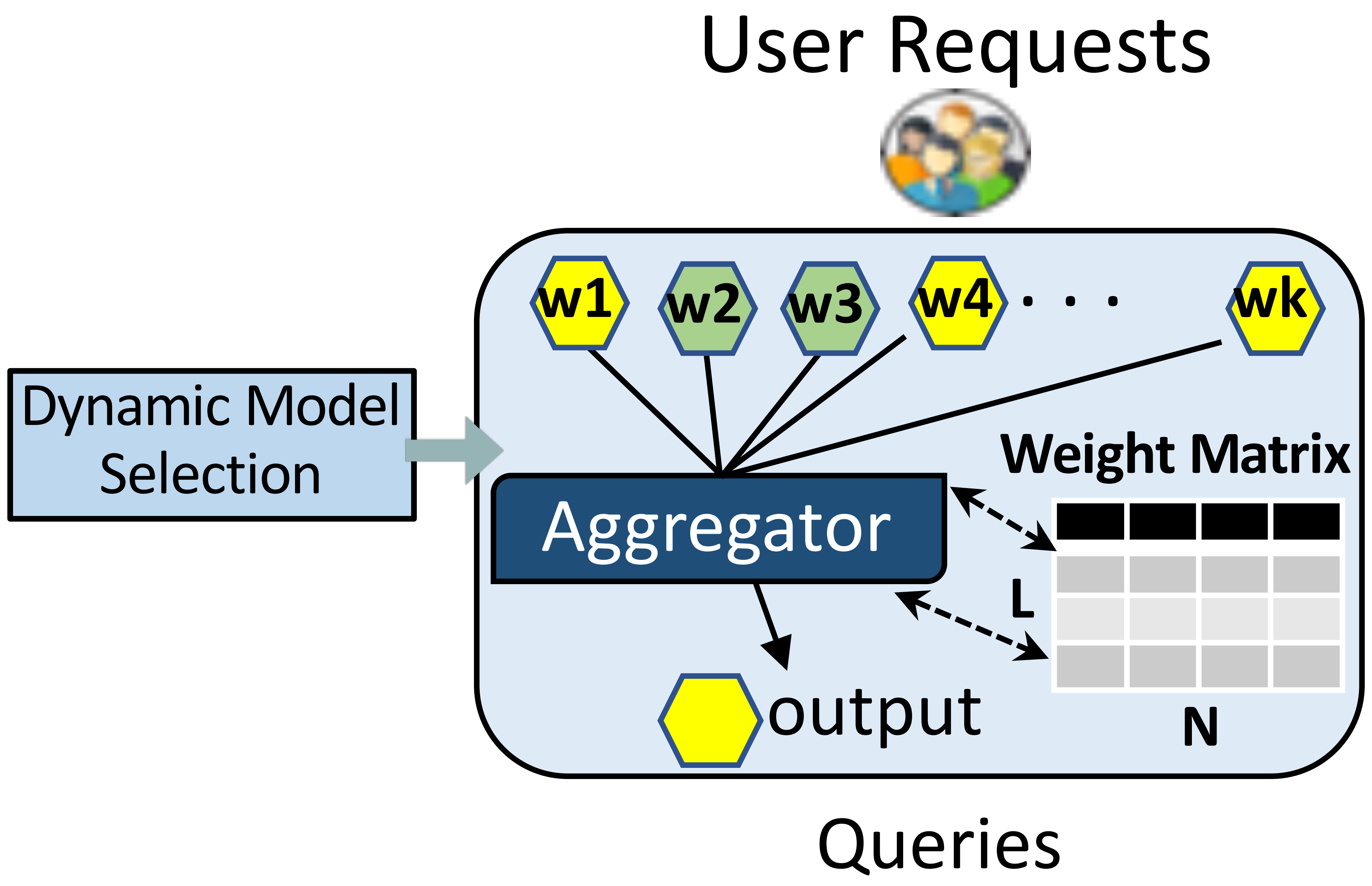
Mobilenet (MNet) ⇨ Slug ☺

Mobilenet (MNet) ⇨ Quill ☹

COCKTAIL- MULTIDIMENSIONAL OPTIMIZATION FOR ENSEMBLE LEARNING IN CLOUD



COCKTAIL- MULTIDIMENSIONAL OPTIMIZATION FOR ENSEMBLE LEARNING IN CLOUD



Class-wise dictionary

Weighted Selection

Objective Function

- Three optimization points: cost, latency and accuracy
- Metrics $\mu_1 = \frac{Acc}{Lat}$; $\mu_2 = k \sum_{i=1}^{n} \frac{inst\_cost}{p_{m_i}}$
  - Where we use n models (model $m_i, i = 1\ to\ n$) to ensemble
  - Each model m_i has a packing factor of p_{m_i}. k is a constant which is dependent on the resource and the instance type
- Our objective:

Obj1:

$$\max \mu_1 \ : \begin{cases} Acc \ \geq Acc_{SLO} \pm \ Acc_{margin} \\ Lat \ \leq Lat_{SLO} \pm \ Lat_{margin} \end{cases}$$

Obj2:

$$\min_{Cost \leq Cost_{Baseline}} \mu_2 : Acc \ \geq Acc_{SLO} \pm \ Acc_{margin}$$

Objective Functions

Can we select the models apriori? ☹

Objective Functions

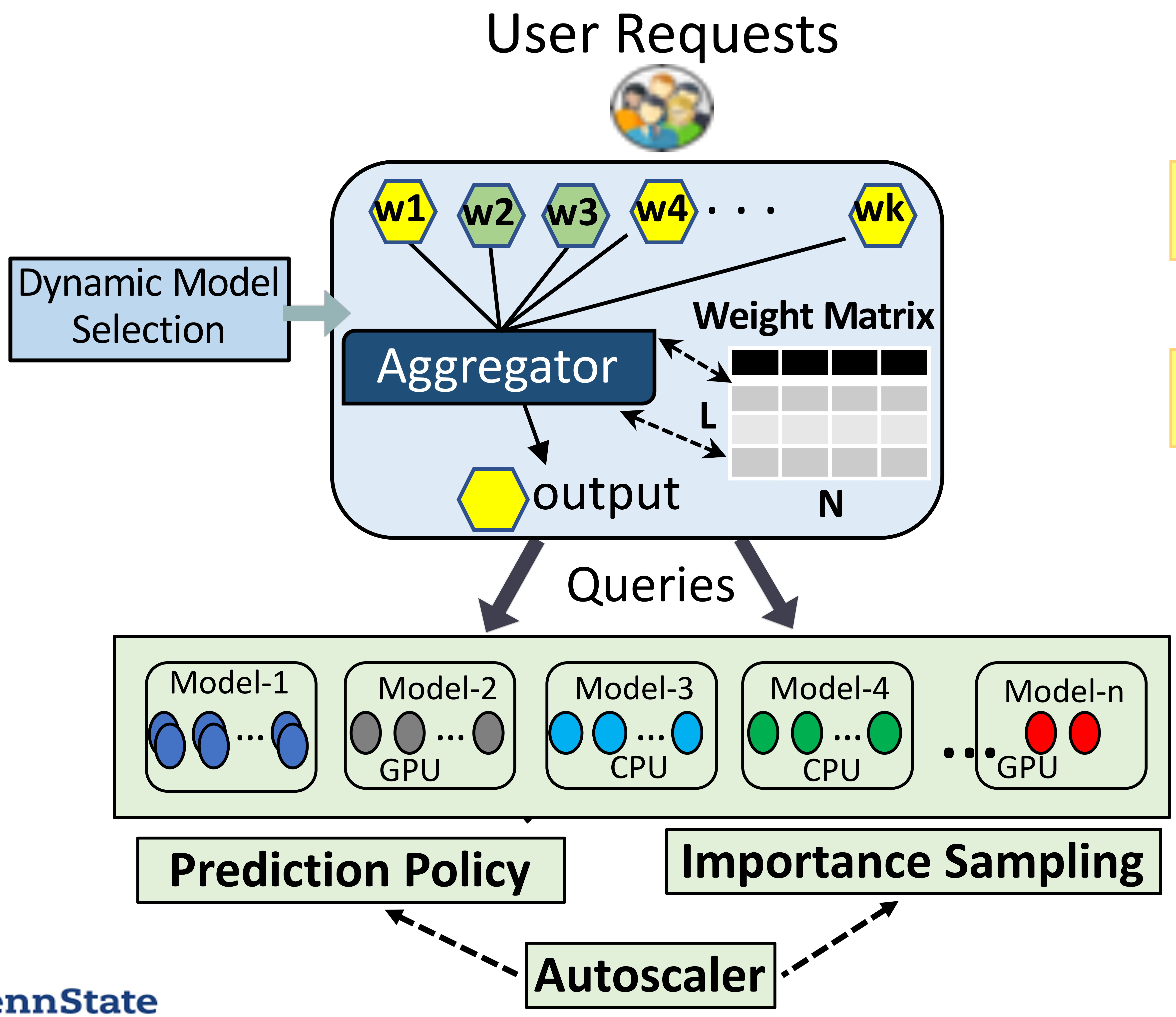
Can we select the models apriori? ☹

Can we drop models after selection? ☺

Ensuring $Acc \geq Acc_{SLO} \pm Acc_{margin}$

- Say we have 'n' models with minimum accuracy of 'a'
- We use majority voting ensemble : we need at least $\frac{n}{2} + 1$ give correct results.
- Prob correct = $\binom{n}{\lfloor \frac{n}{2}+1 \rfloor + i} a^{\lfloor \frac{n}{2}+1 \rfloor + i} (1-a)^{n - (\lfloor \frac{n}{2}+1 \rfloor + i)}$; $for\ i = 0\ to \lceil n/2 + 1 \rceil$

COCKTAIL- MULTIDIMENSIONAL OPTIMIZATION FOR ENSEMBLE LEARNING IN CLOUD



Class-wise dictionary

Weighted Selection

Dedicated Pools

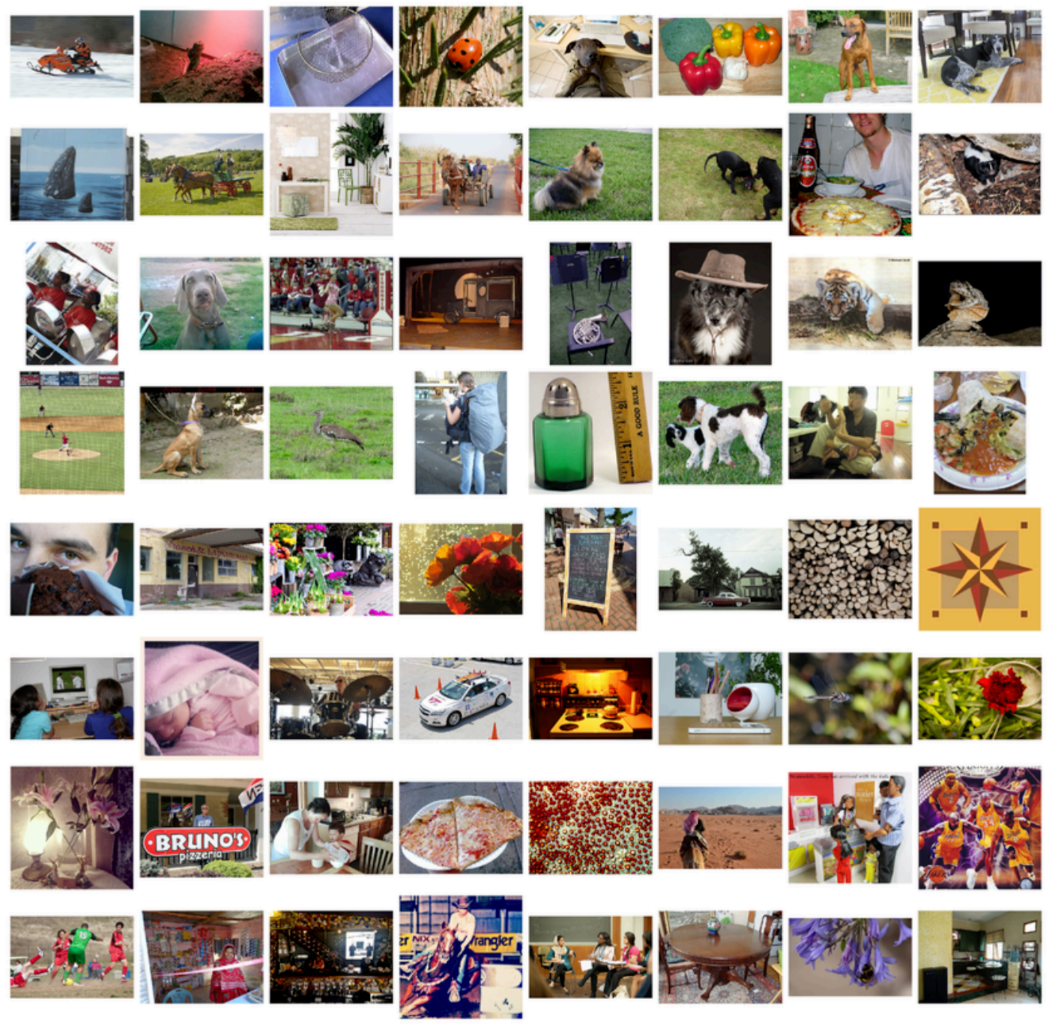
Per model Scaling

Fault tolerant

Evaluation and Setup

Amazon EC2

TensorFlow 2.0



Workloads

Dataset	Application	Classes	Train-set	Test-set
ImageNet [56]	Image	1000	1.2M	50K
CIFAR-100 [116]	Image	100	50K	10K
SST-2 [117]	Text	2	9.6K	1.8K
SemEval [118]	Text	3	50.3K	12.2K

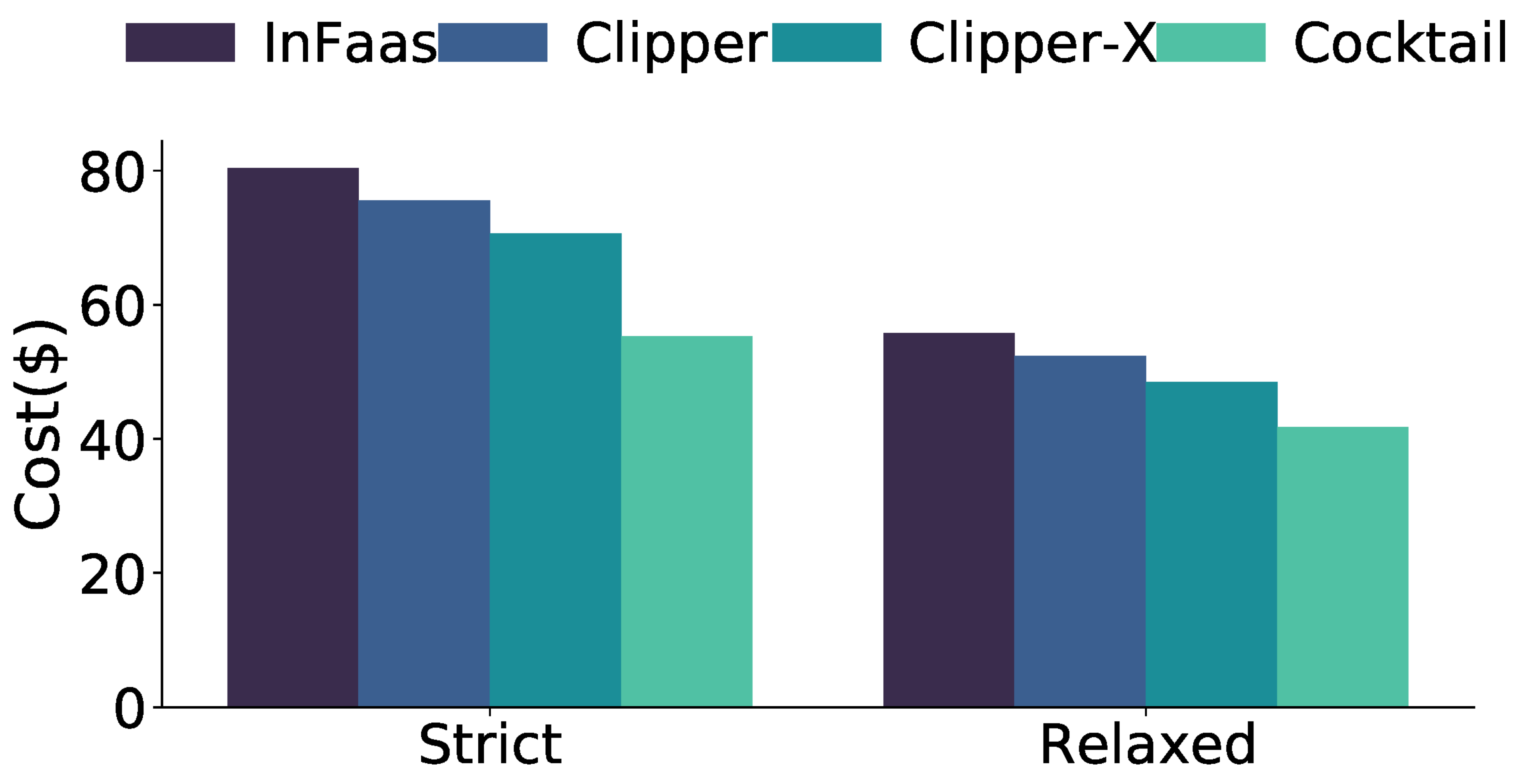
Baselines

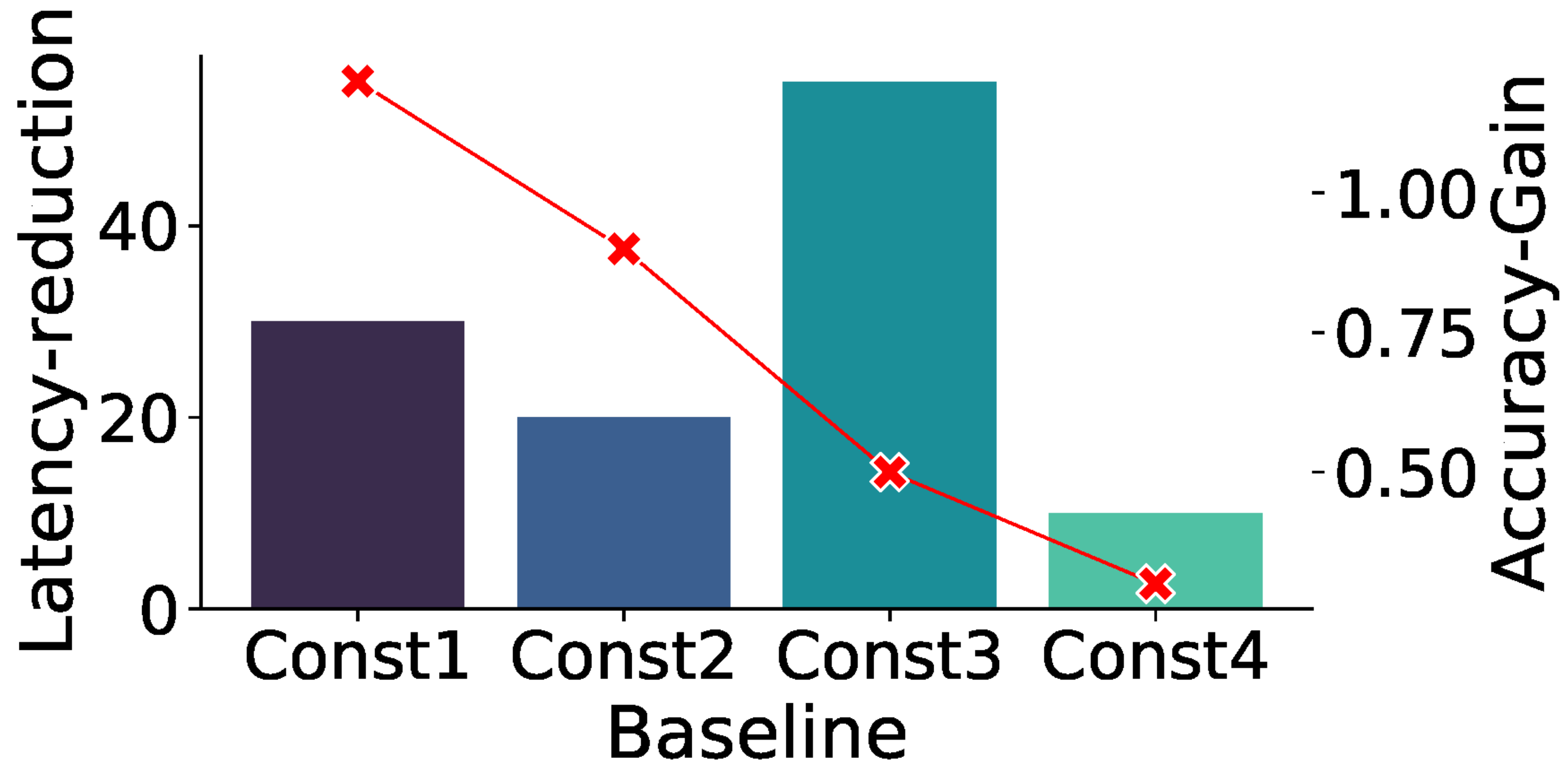
- **InFaaS**: Single Models
- **Clipper**: Static Ensemble
- **Clipper-X**: Dynamic ensemble

Experiment Setup

- *40 EC2 CPU/GPU VMs*
- *Wiki Twitter Traces*

Major Results

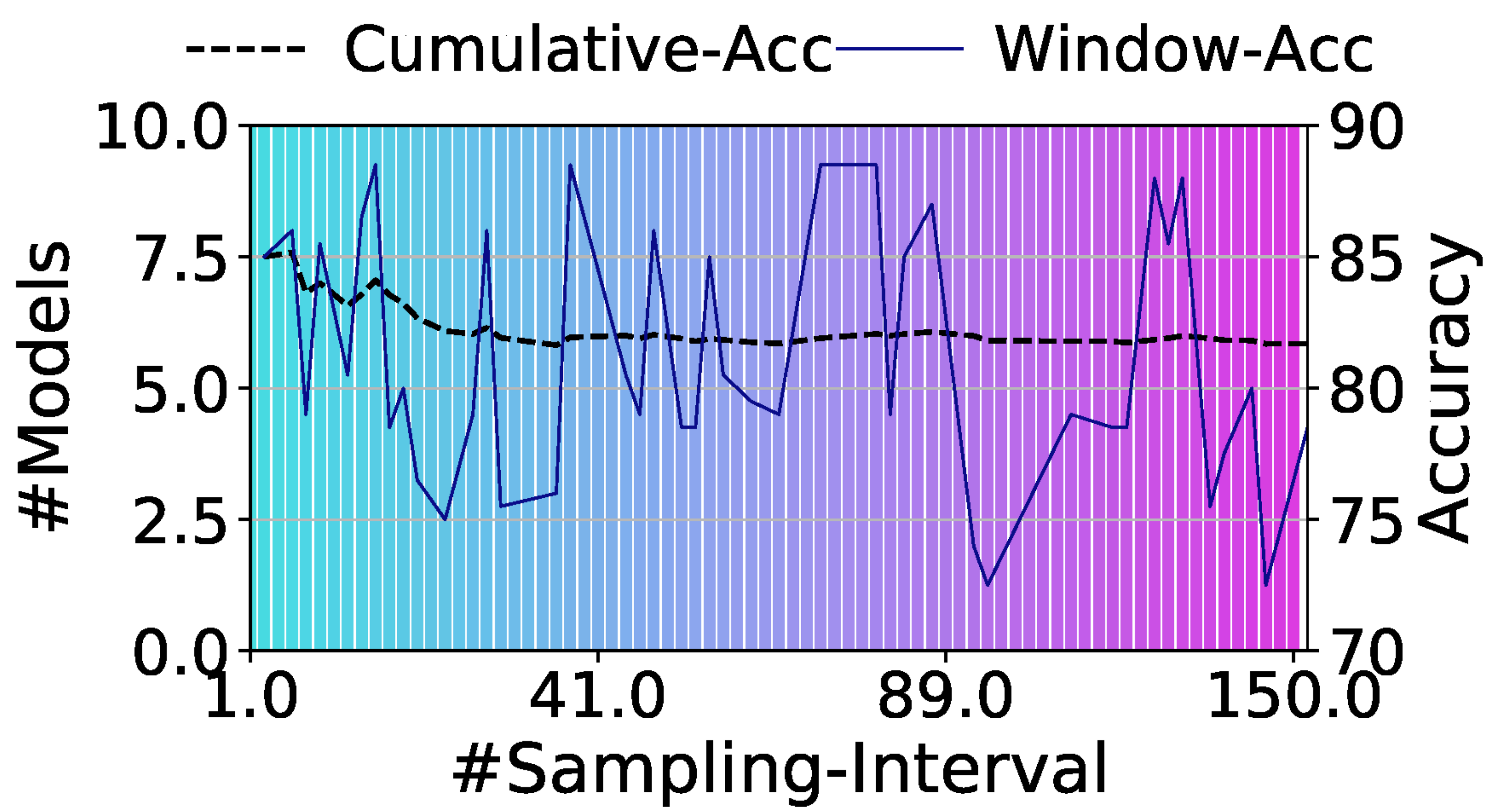




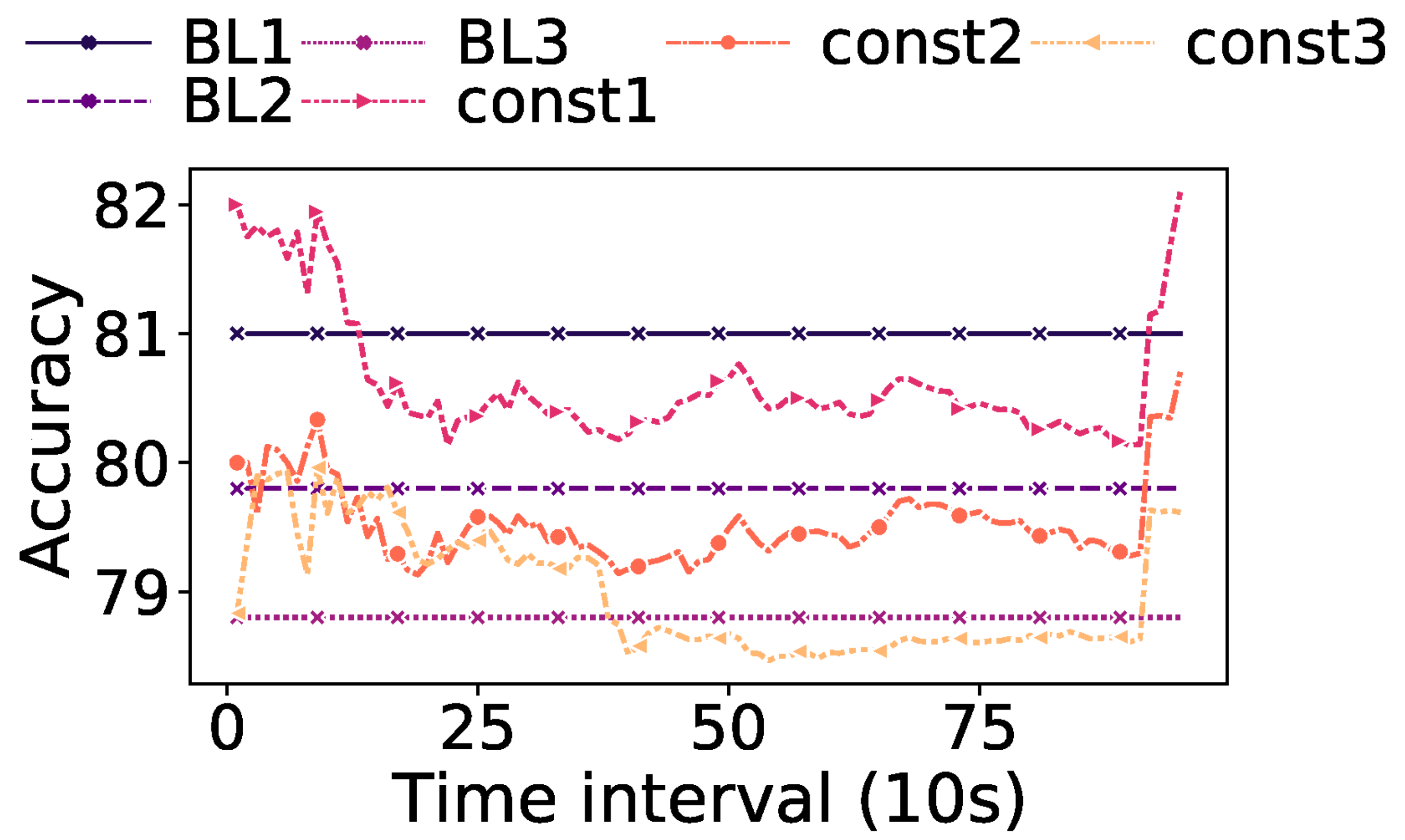
- ☑ Cocktail incurs **~32%** lower cost
- ☑ Cocktail reduces #models by **~50%** on average

- ☑ Cocktail yields **~2x** lower latency
- ☑ Cocktail gains upto **~1.25%** more accuracy

MAJOR RESULTS



- Cocktail quickly adjusts #Models
- Cocktail on average uses **5** models



- Cocktail incurs modest accuracy loss upto **0.7%**
- Cocktail avoid inference failures while compromising accuracy.

SUMMARY

Cocktail leverages ensembling to achieve higher accuracy at lower latency

Cocktail dynamically adjusts the #models in the ensemble without compromising accuracy.

Cocktail leverages transient instances to reduce the deployment cost.

Thank You

Code: https://github.com/jashwantraj92/cocktail.git

Contact: jashwant.raj92@gmail.com, cyanmishra92@gmail.com