Comprehensive and Efficient Data Labeling via Adaptive Model Scheduling

Mu Yuan[1], Lan Zhang[1], Xiang-Yang Li[1], Hui Xiong[2]

[1]University of Science and Technology of China

[2]Rutgers University

中国科学技术大学 University of Science and Technology of China

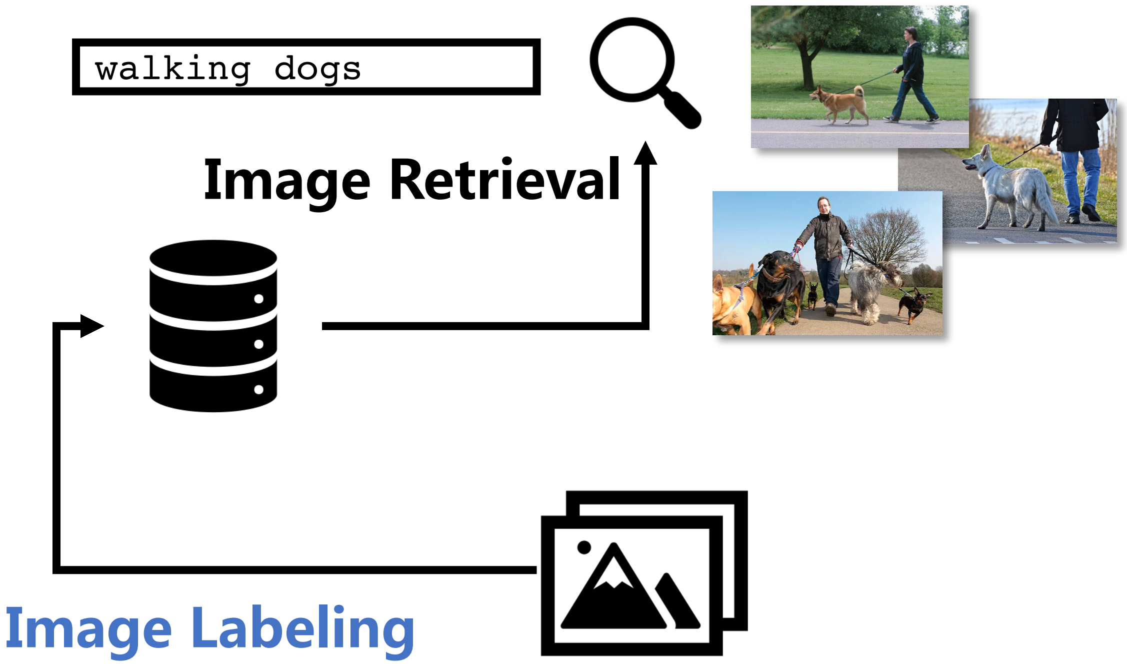
RUTGERS

LINKE Lab for Intelligent Networking and Knowledge Engineering

Outline

- **Resource-wasting multi-model inference workloads**
- Rule-based scheduler
- Learning-based scheduler
- Evaluation
- Conclusion

Multi-Model Inference Workloads

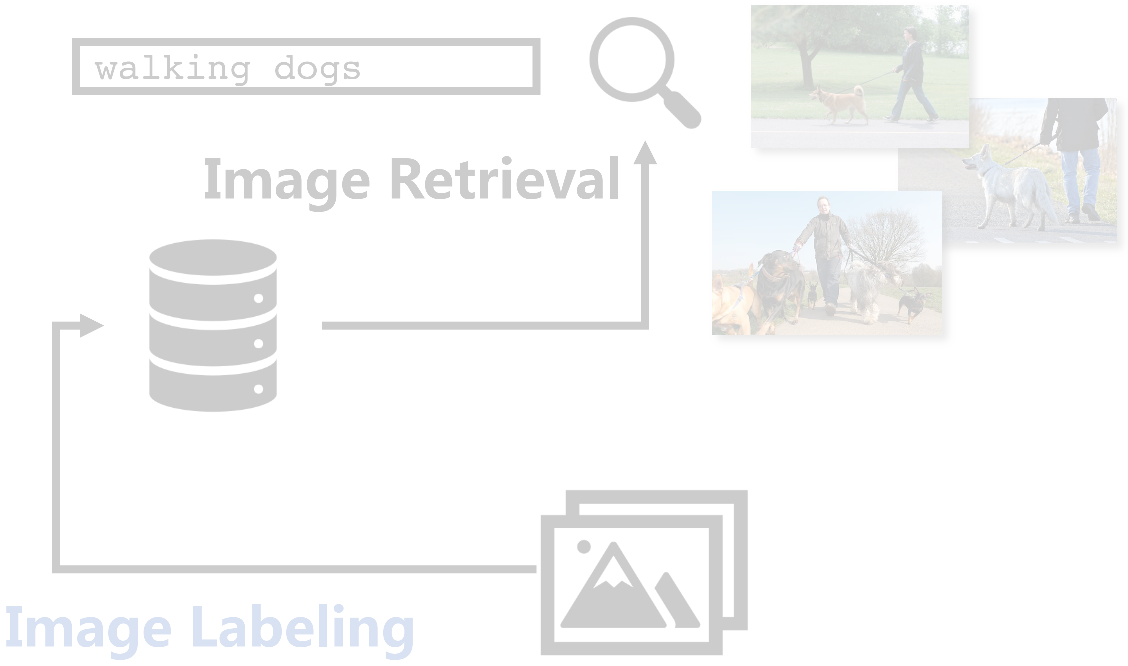


walking dogs

Image Retrieval

Image Labeling

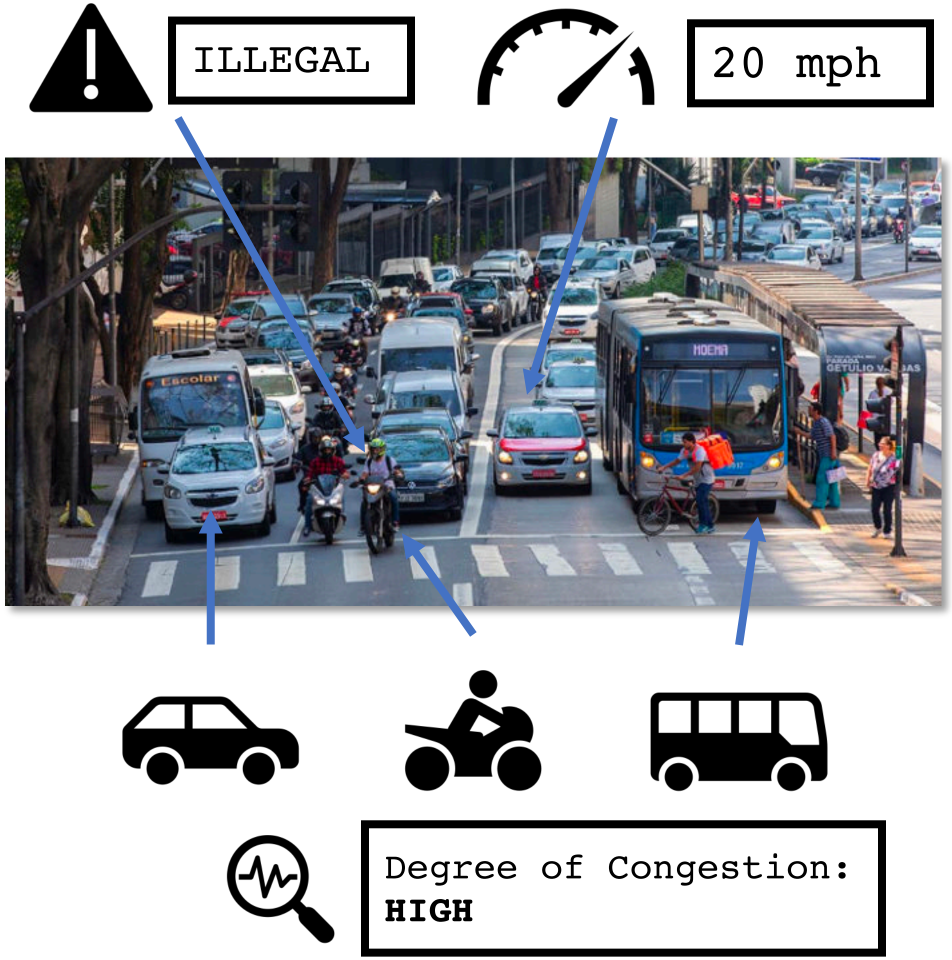
Multi-Model Inference Workloads



walking dogs

Image Retrieval

Image Labeling

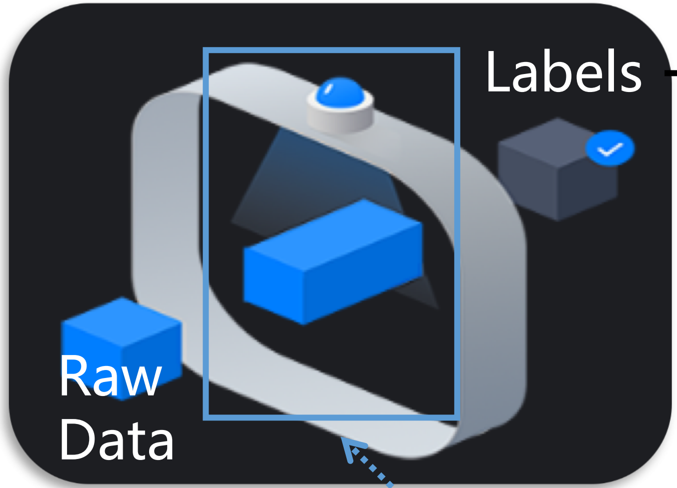


ILLEGAL

20 mph

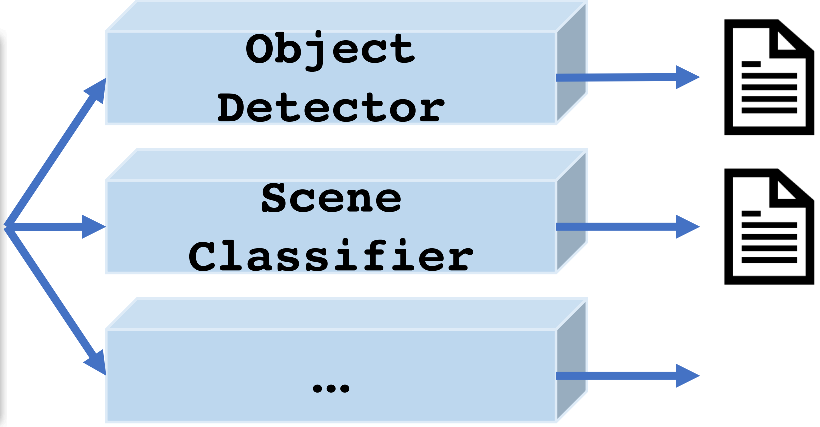
Degree of Congestion: **HIGH**

Data Trading Platform



Data Search Engine

Multi-Model Data Labeling Workloads



Object Detector

Scene Classifier

…

```
person, 0.994
chair, 0.565
tv monitor, 0.996
pub, 0.727
beer hall, 0.198

...
```

Observation

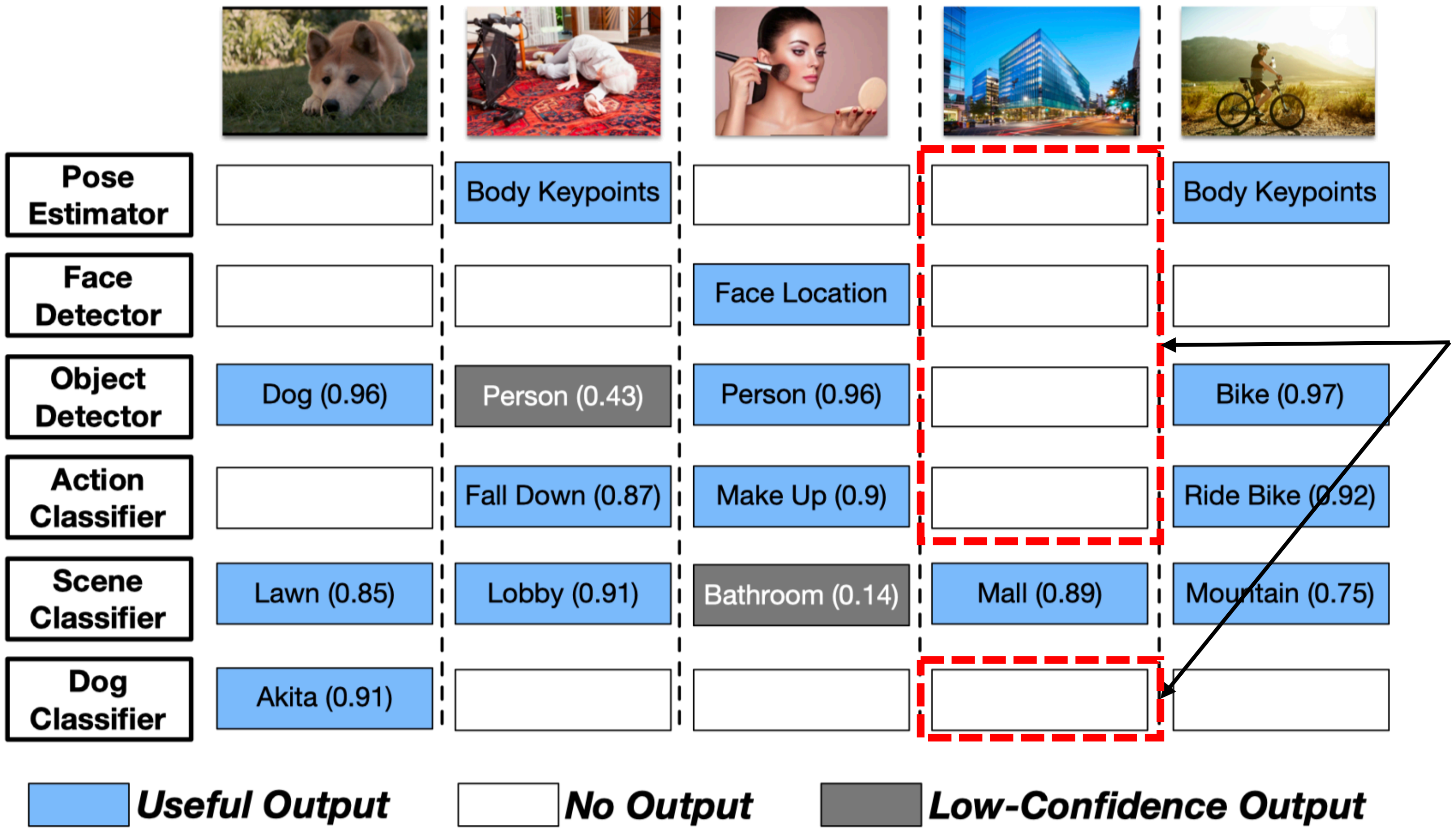
CV Models

Raw Images

Pose Estimator		Body Keypoints			Body Keypoints
Face Detector			Face Location		
Object Detector	Dog (0.96)	Person (0.43)	Person (0.96)		Bike (0.97)
Action Classifier		Fall Down (0.87)	Make Up (0.9)		Ride Bike (0.92)
Scene Classifier	Lawn (0.85)	Lobby (0.91)	Bathroom (0.14)	Mall (0.89)	Mountain (0.75)
Dog Classifier	Akita (0.91)				

🟦 ***Useful Output*** □ ***No Output*** ▨ ***Low-Confidence Output***

Observation



Pose Estimator		Body Keypoints			Body Keypoints
Face Detector			Face Location		
Object Detector	Dog (0.96)	Person (0.43)	Person (0.96)		Bike (0.97)
Action Classifier		Fall Down (0.87)	Make Up (0.9)		Ride Bike (0.92)
Scene Classifier	Lawn (0.85)	Lobby (0.91)	Bathroom (0.14)	Mall (0.89)	Mountain (0.75)
Dog Classifier	Akita (0.91)				

🟦 *Useful Output* □ *No Output* ⬛ *Low-Confidence Output*

Wasted computing resources!

Observation

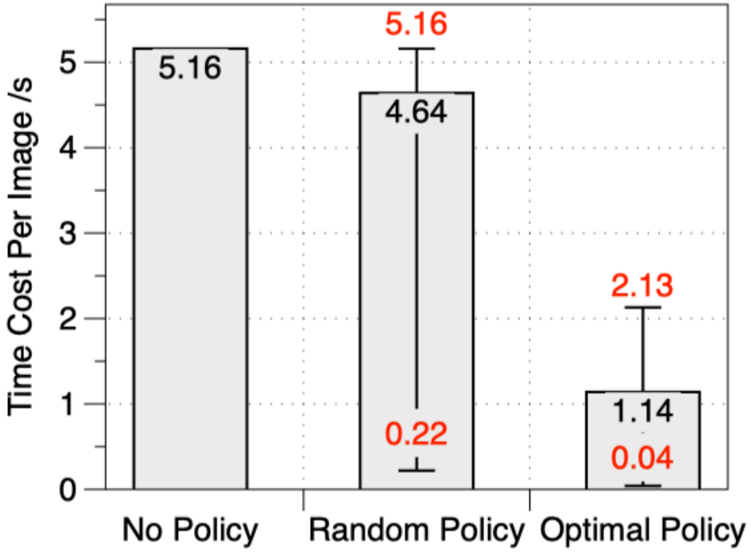
Pose Estimator		Body Keypoints			Body Keypoints
Face Detector			Face Location		
Object Detector	Dog (0.96)	Person (0.43)	Person (0.96)		Bike (0.97)
Action Classifier		Fall Down (0.87)	Make Up (0.9)		Ride Bike (0.92)
Scene Classifier	Lawn (0.85)	Lobby (0.91)	Bathroom (0.14)	Mall (0.89)	Mountain (0.75)
Dog Classifier	Akita (0.91)				

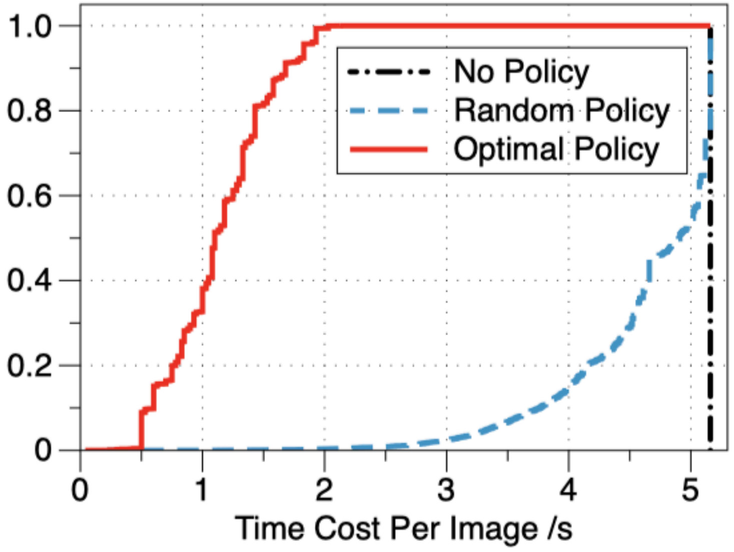
🟦 ***Useful Output*** ⬜ ***No Output*** ⬛ ***Low-Confidence Output***

Wasted computing resources!

Data-Driven Analysis

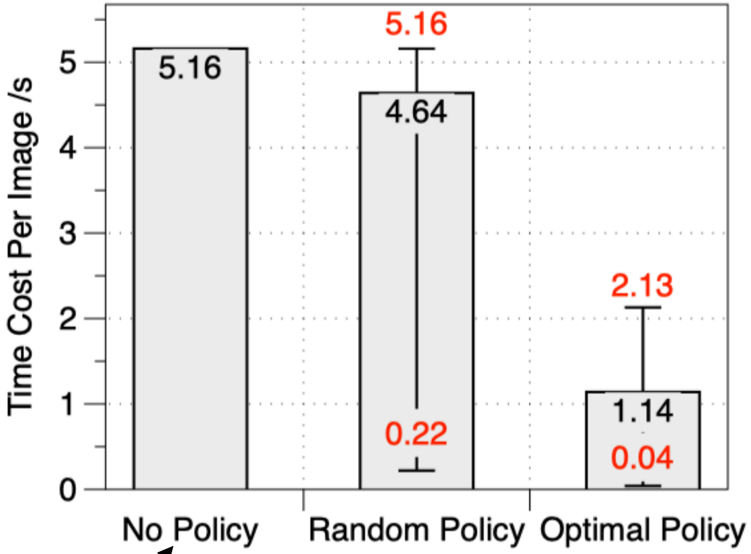
394,170 images
30 CV models

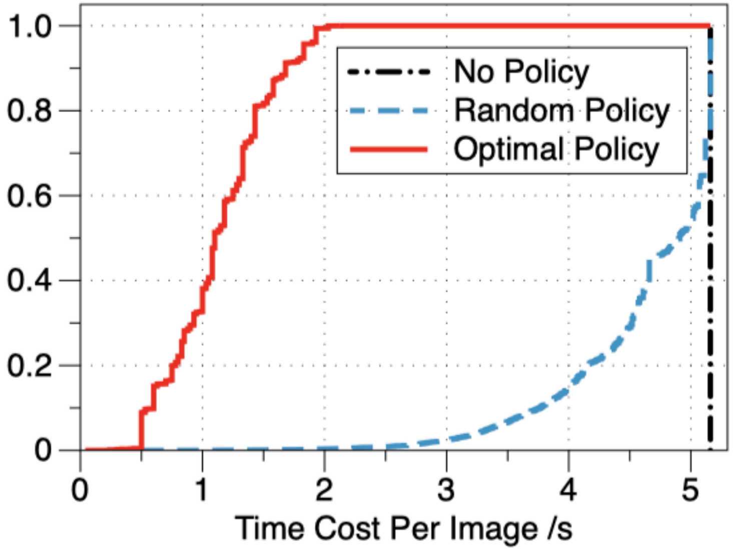




Data-Driven Analysis

394,170 images
30 CV models

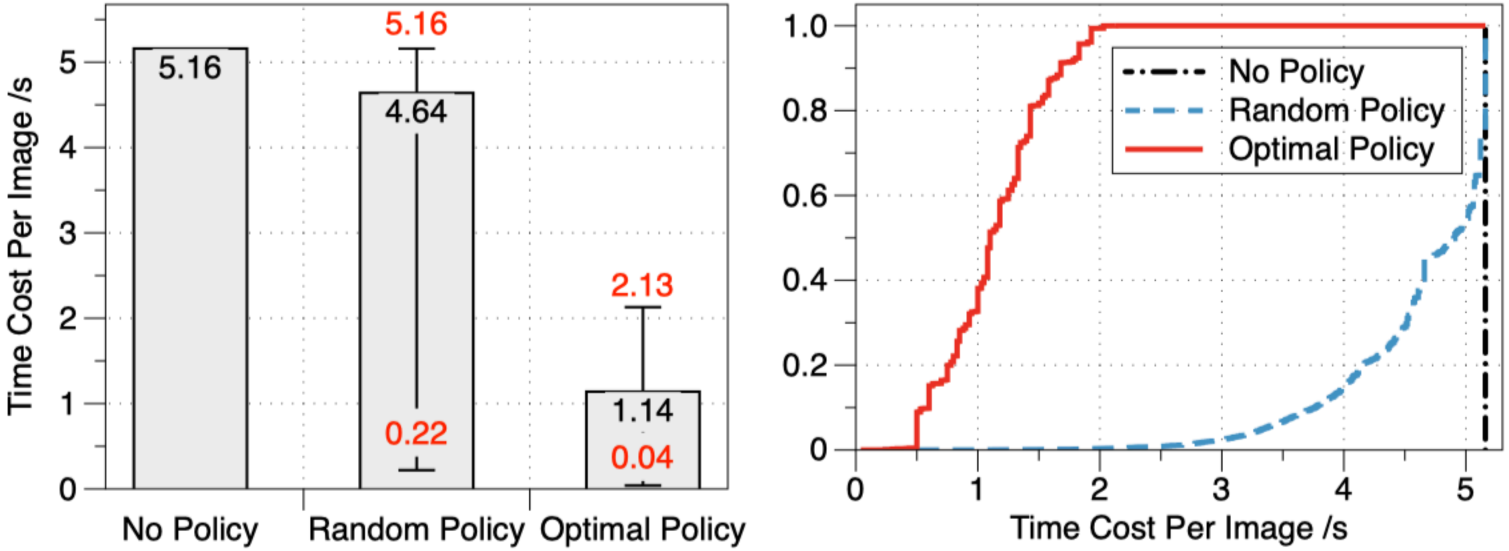




Executing **ALL** models.

Data-Driven Analysis

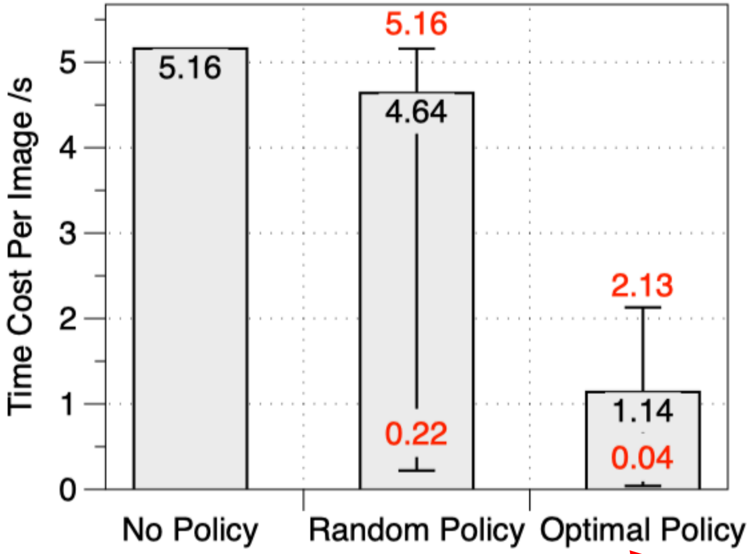
394,170 images
30 CV models

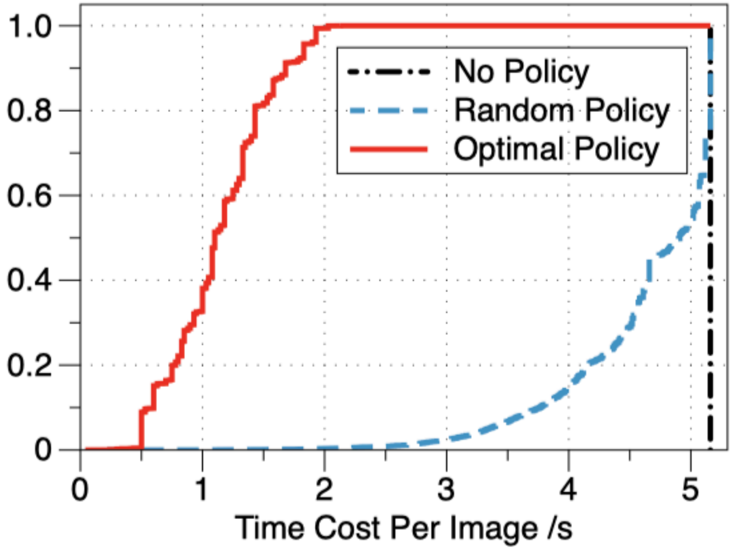


Executing models that output **VALUABLE** labels.

Data-Driven Analysis

394,170 images
30 CV models

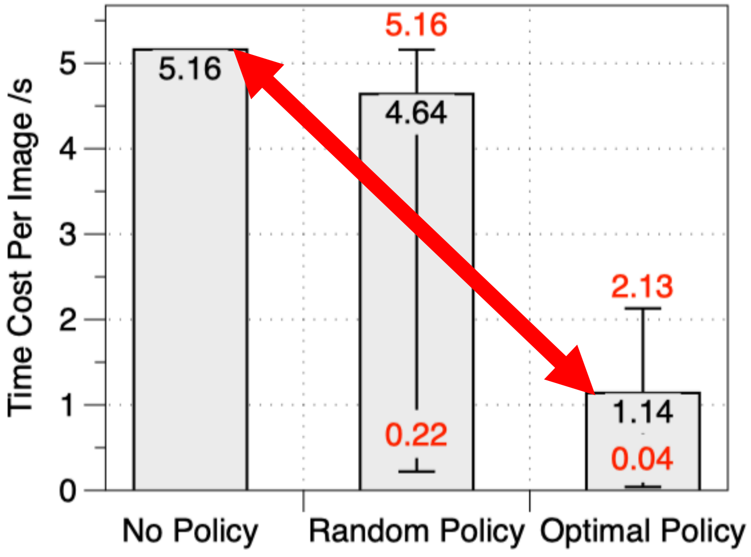


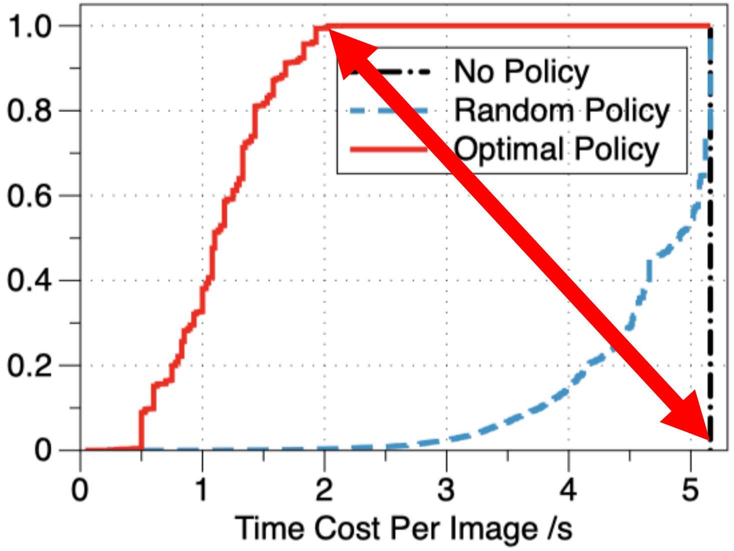


```
Valuable(model, img) = (Output(model, img).conf > 0.5).any() ? True : False;
```

Data-Driven Analysis

394,170 images
30 CV models

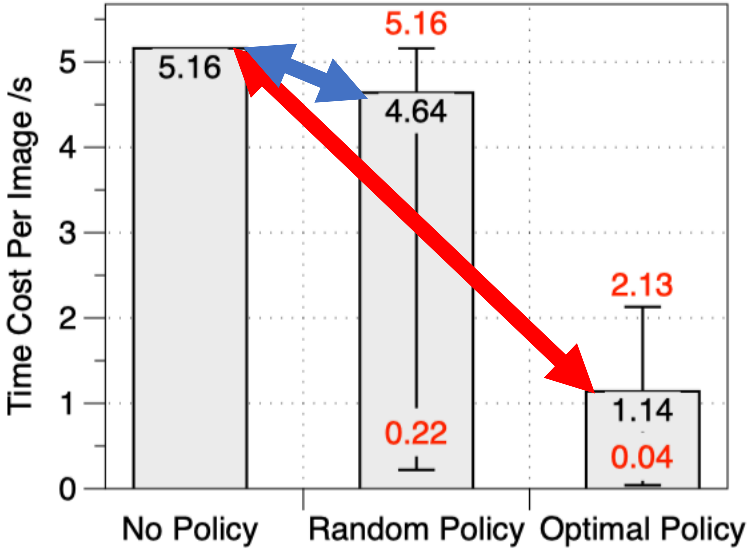


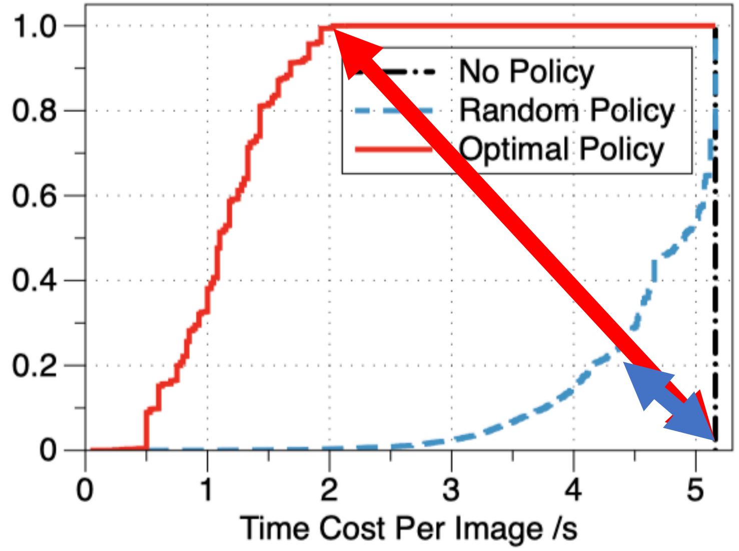


```
Valuable(model, img) = (Output(model, img).conf > 0.5).any() ? True : False;
```

Data-Driven Analysis

```
394,170 images
30 CV models
```





```
Valuable(model, img) = (Output(model, img).conf > 0.5).any() ? True : False;
```

Executing models **RANDOMLY**.

Key Idea

Without assumption of input data distribution,
how to predict the value of models before execution?

Key Idea

**Without assumption of input data distribution,
how to predict the value of models before execution?**

IMPOSSIBLE for single-model workloads ...

But not for the multi-model tasks!

Key Idea

Without assumption of input data distribution, how to predict the value of models before execution?

IMPOSSIBLE for single-model workloads ...

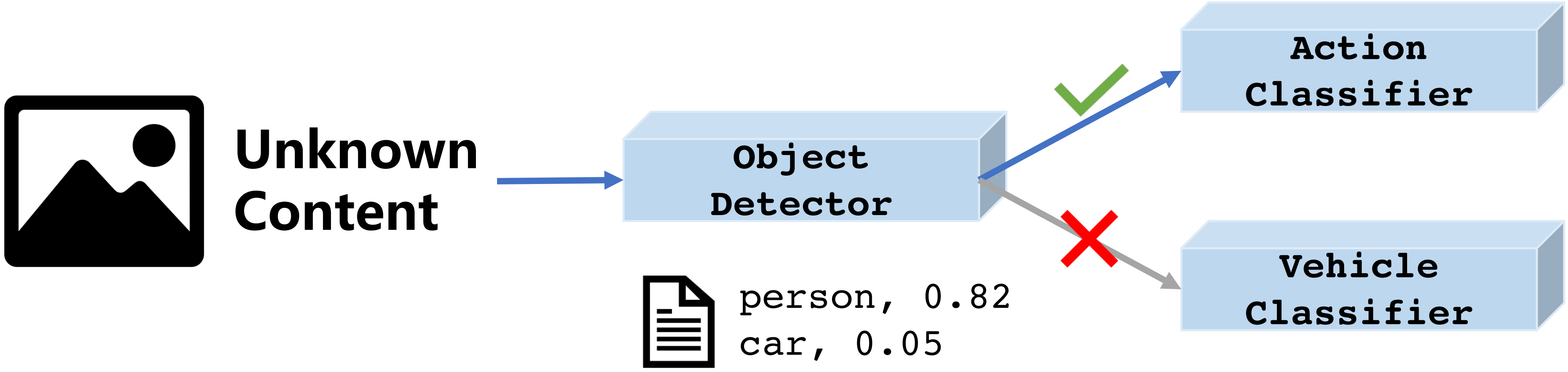
But not for the multi-model tasks!

We can get hints from executed models.

Outline

- Resource-wasting multi-model inference workloads
- **Rule-based scheduler**
- Learning-based scheduler
- Evaluation
- Conclusion

Manual Rules



Unknown Content

Object Detector

Action Classifier

Vehicle Classifier

```
person, 0.82
car, 0.05
```

Limitations

- Hard to express the non-pairwise rules.
- Too expensive for large-scale semantic labels.
- Challenging to tune the effects of multiple rules.

Limitations

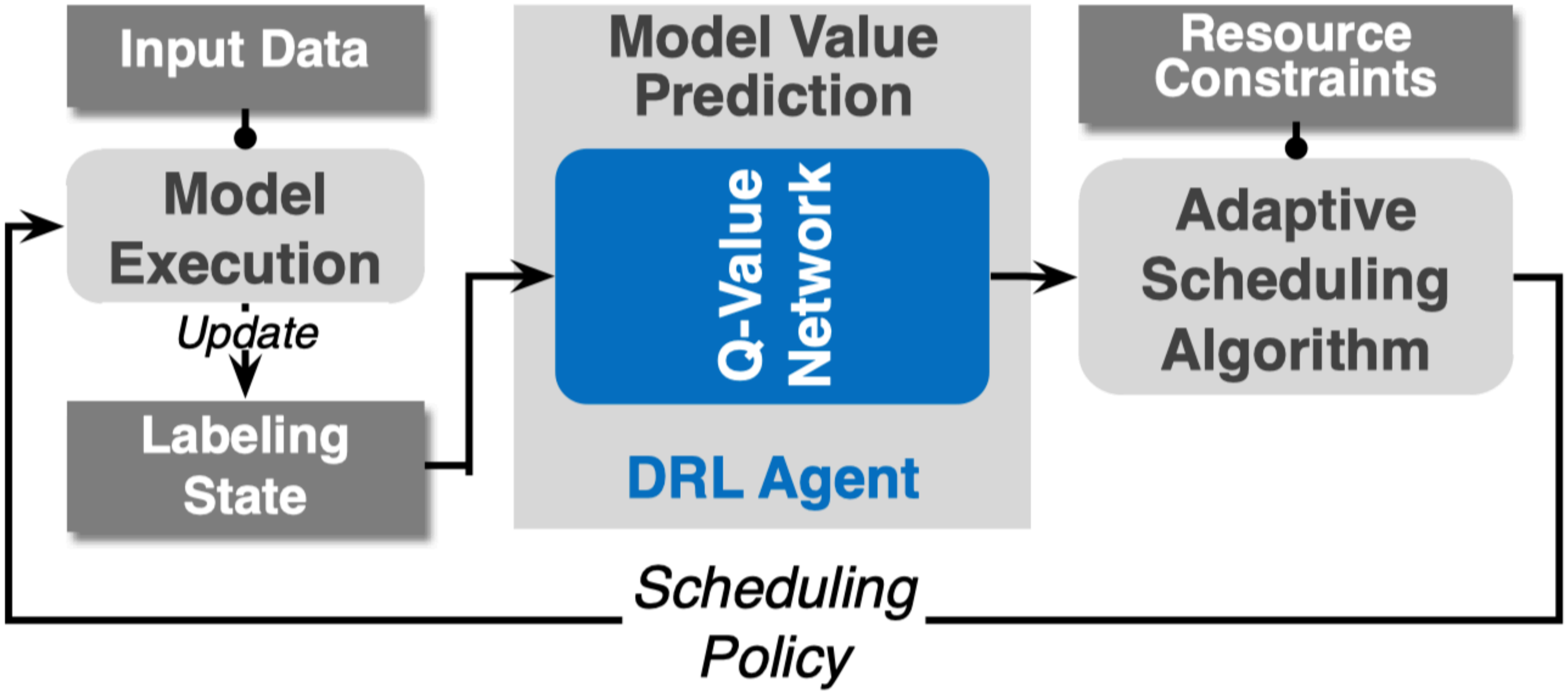
- Hard to express the non-pairwise rules.
- Large-scale semantic labels (>1000 labels in our workloads).
- Effects of multiple rules are difficult to tune.

Deep learning could help!

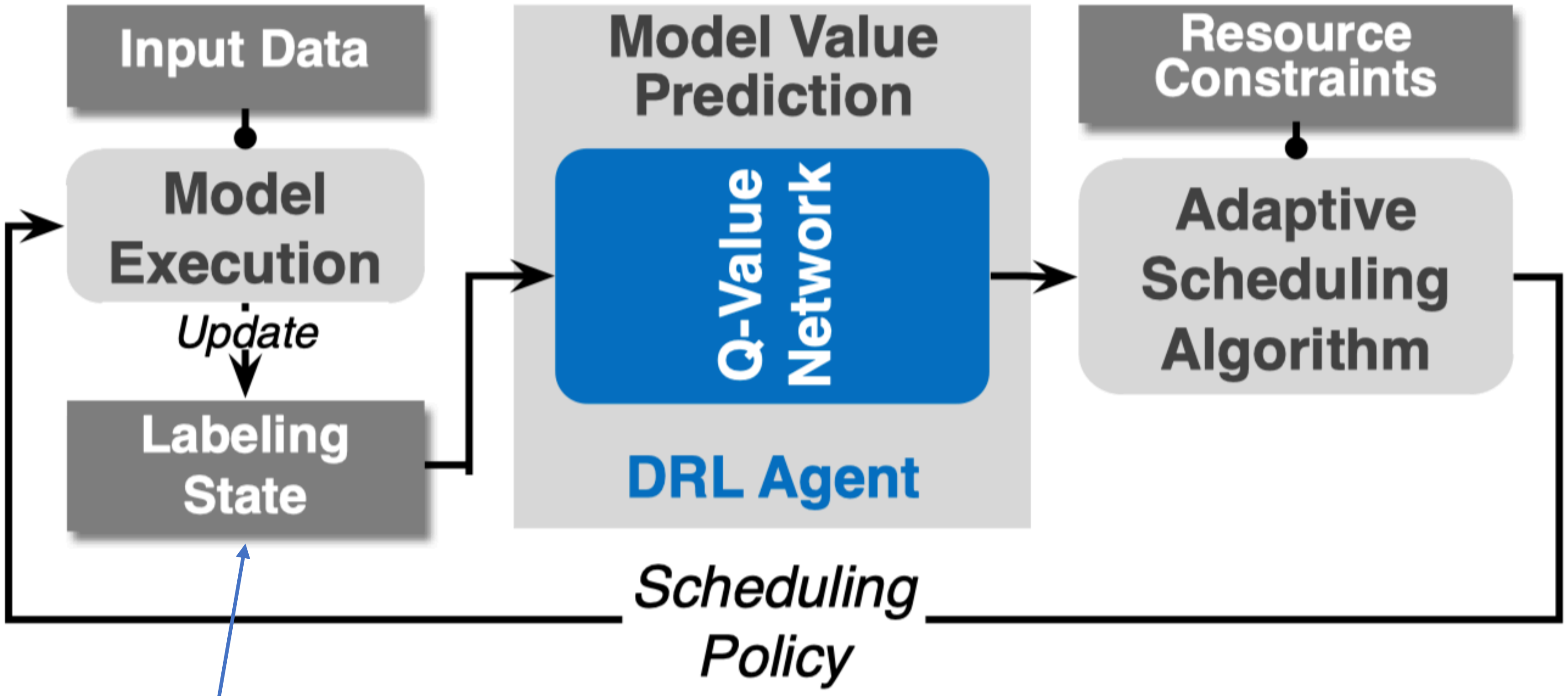
Outline

- Resource-wasting multi-model inference workloads
- Rule-based scheduler
- **Learning-based scheduler**
- Evaluation
- Conclusion

Framework

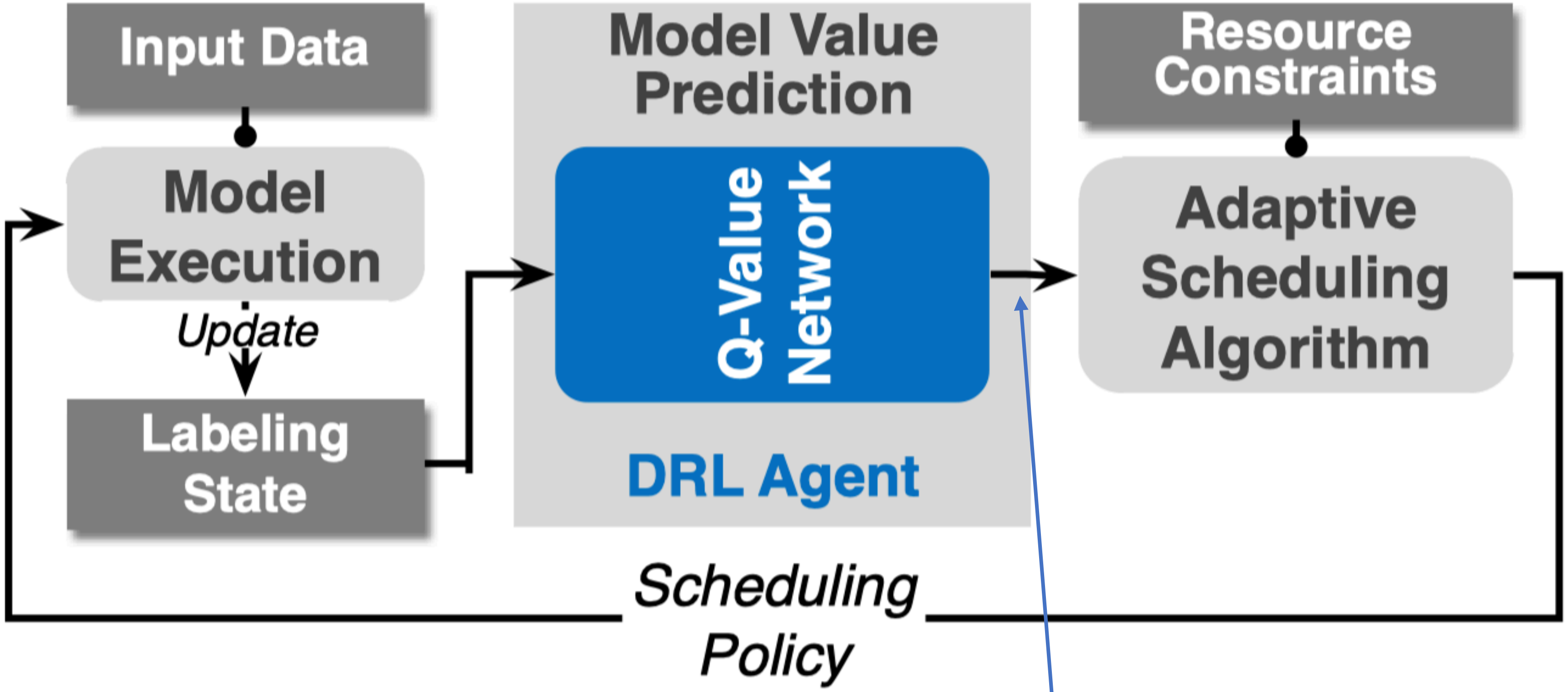

Input Data
Model Execution
Update
Labeling State
Model Value Prediction
Q-Value Network
DRL Agent
Resource Constraints
Adaptive Scheduling Algorithm
Scheduling Policy

Framework



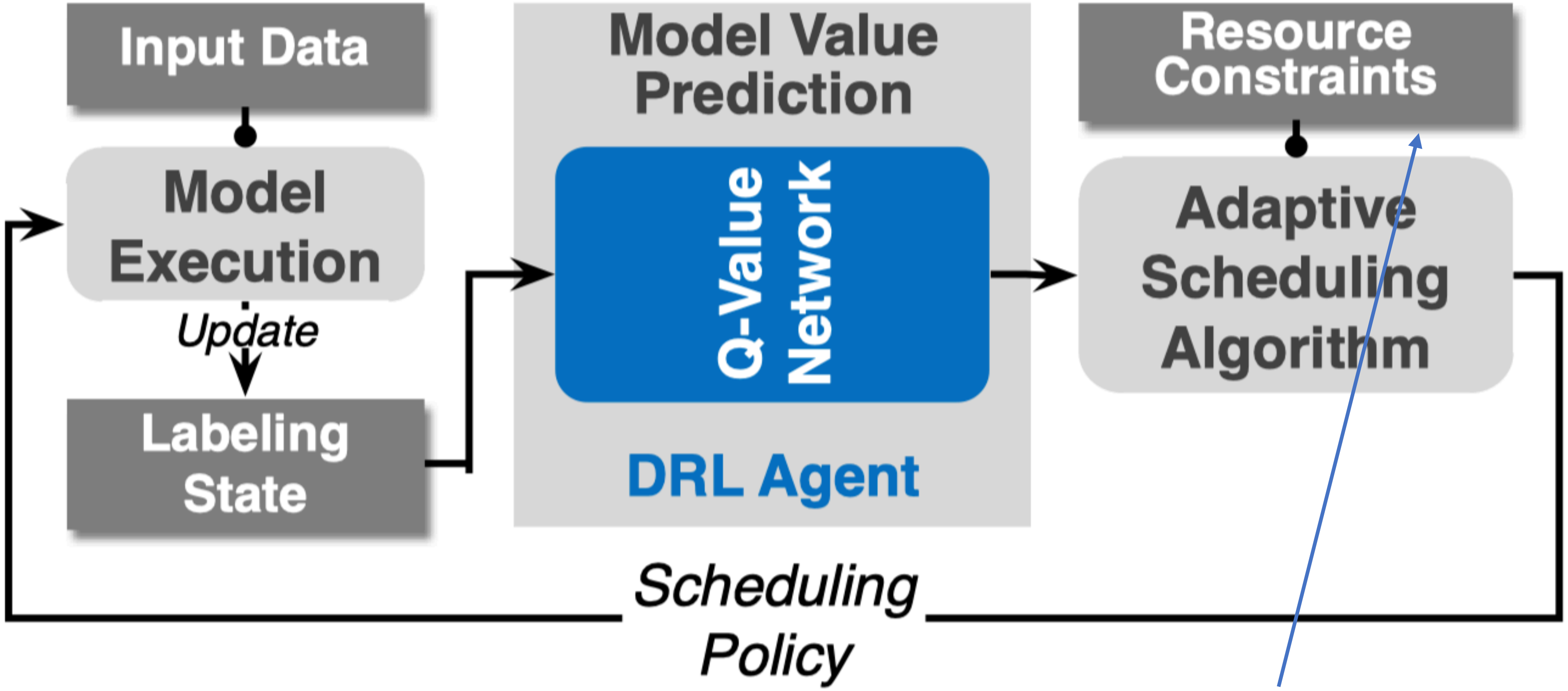
One dimension per label.

Framework



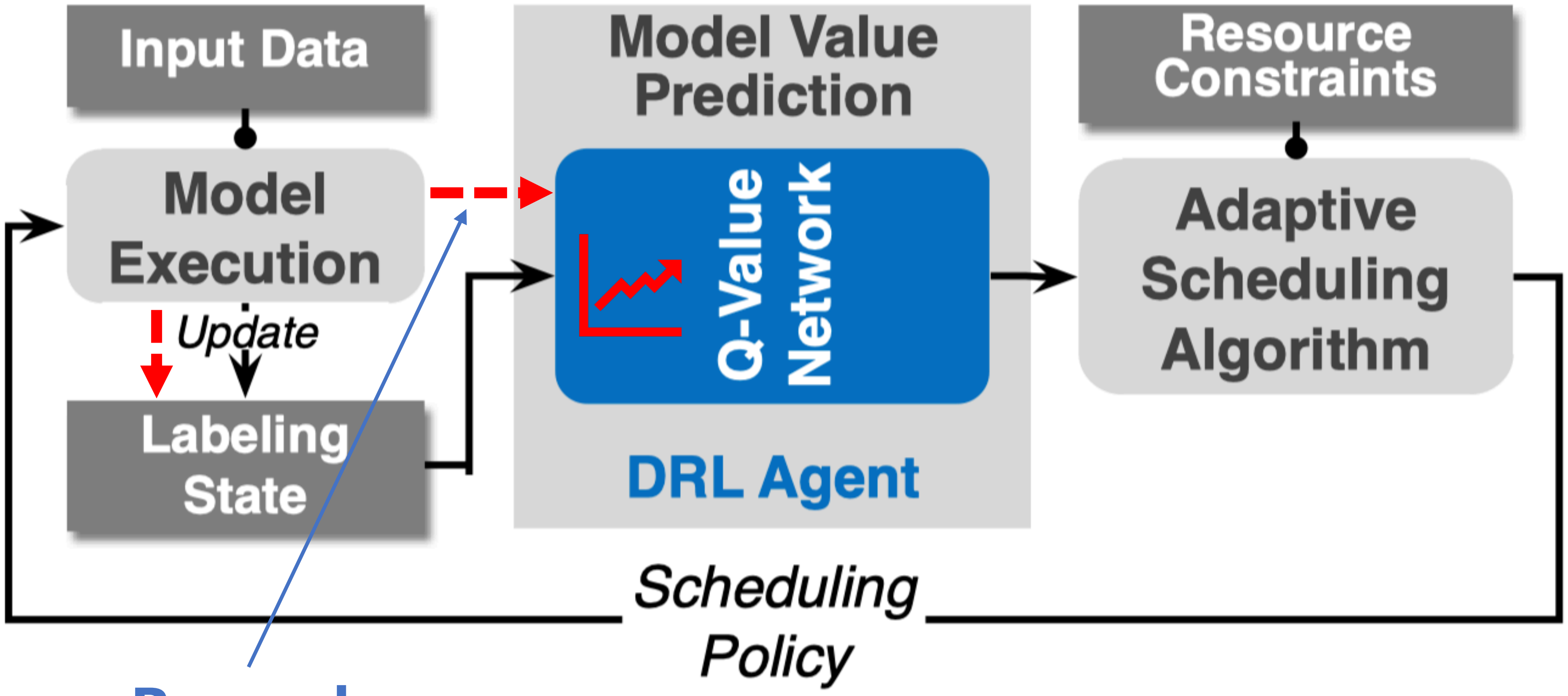
One *action* per model.

Framework



For **versatility** of DRL agent, resource constraints are left to scheduling algorithms.

Framework

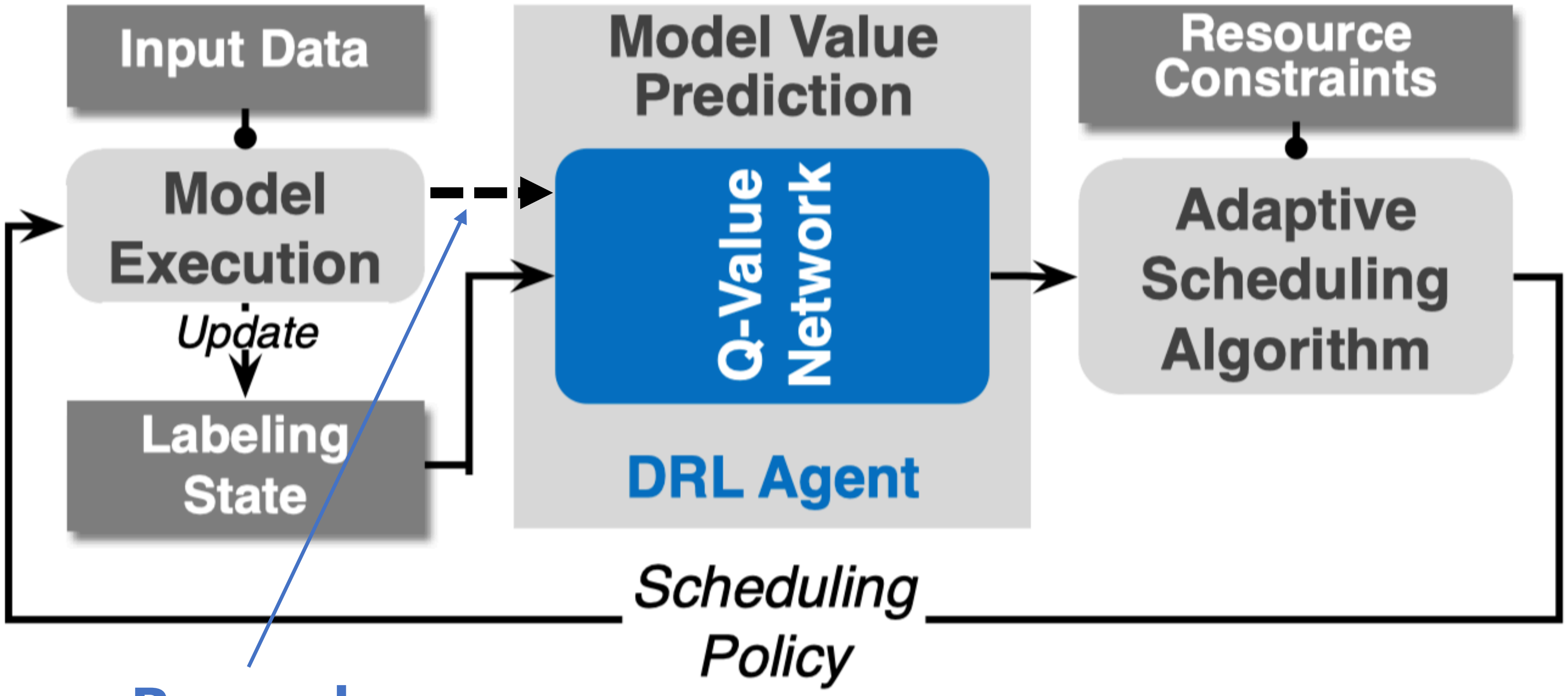


Reward:

$$r(w,d) = log(\theta_m \sum_{l_i \in O'(\{m\},d)} l_i.conf + 1), O'(\{m\},d) \neq \emptyset$$

$$r(w,d) = -1, O'(\{m\},d) = \emptyset$$

Framework



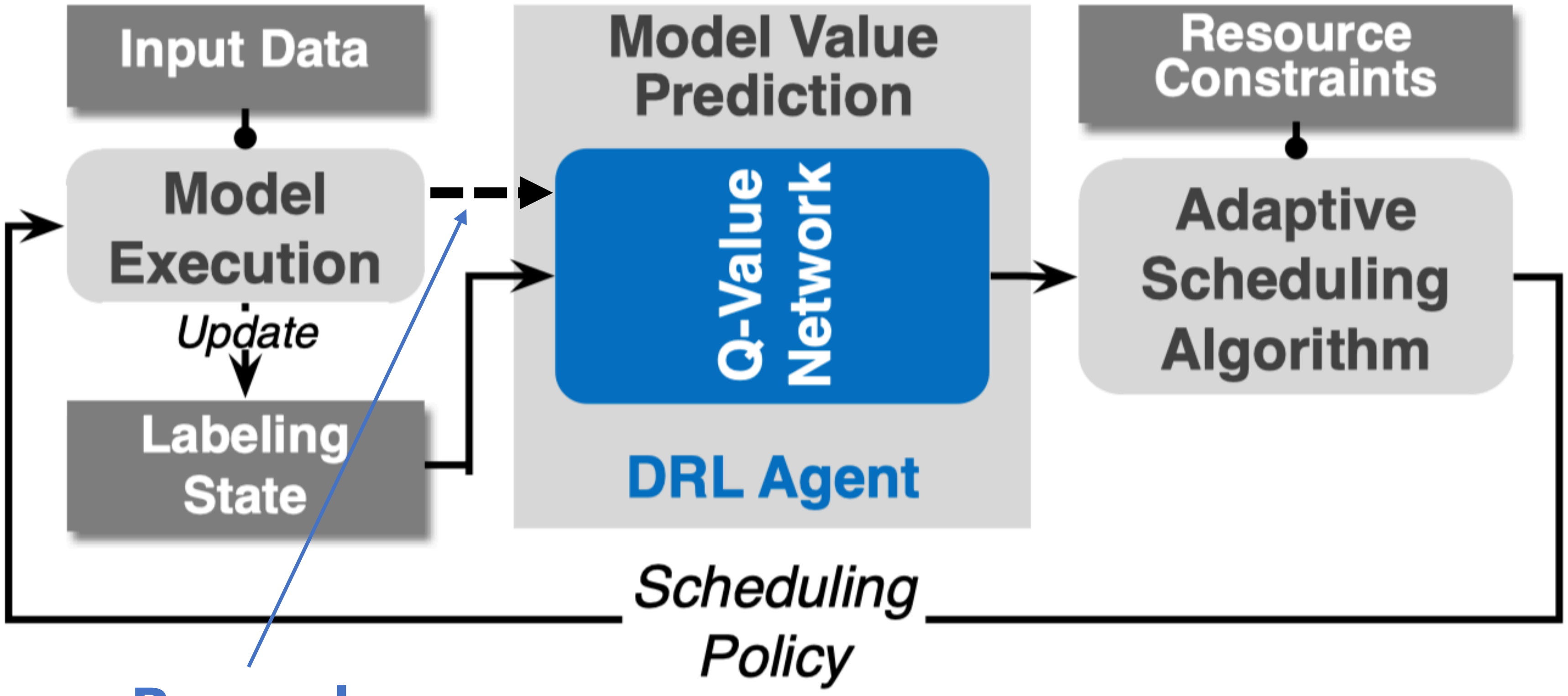
Reward:

$$r(w,d) = log(\theta_m \sum_{l_i \in O'(\{m\},d)} l_i.conf + 1), \boxed{O'(\{m\},d)} \neq \emptyset$$

$$r(w,d) = -1, O'(\{m\},d) = \emptyset$$

newly updated labels

Framework



Reward:

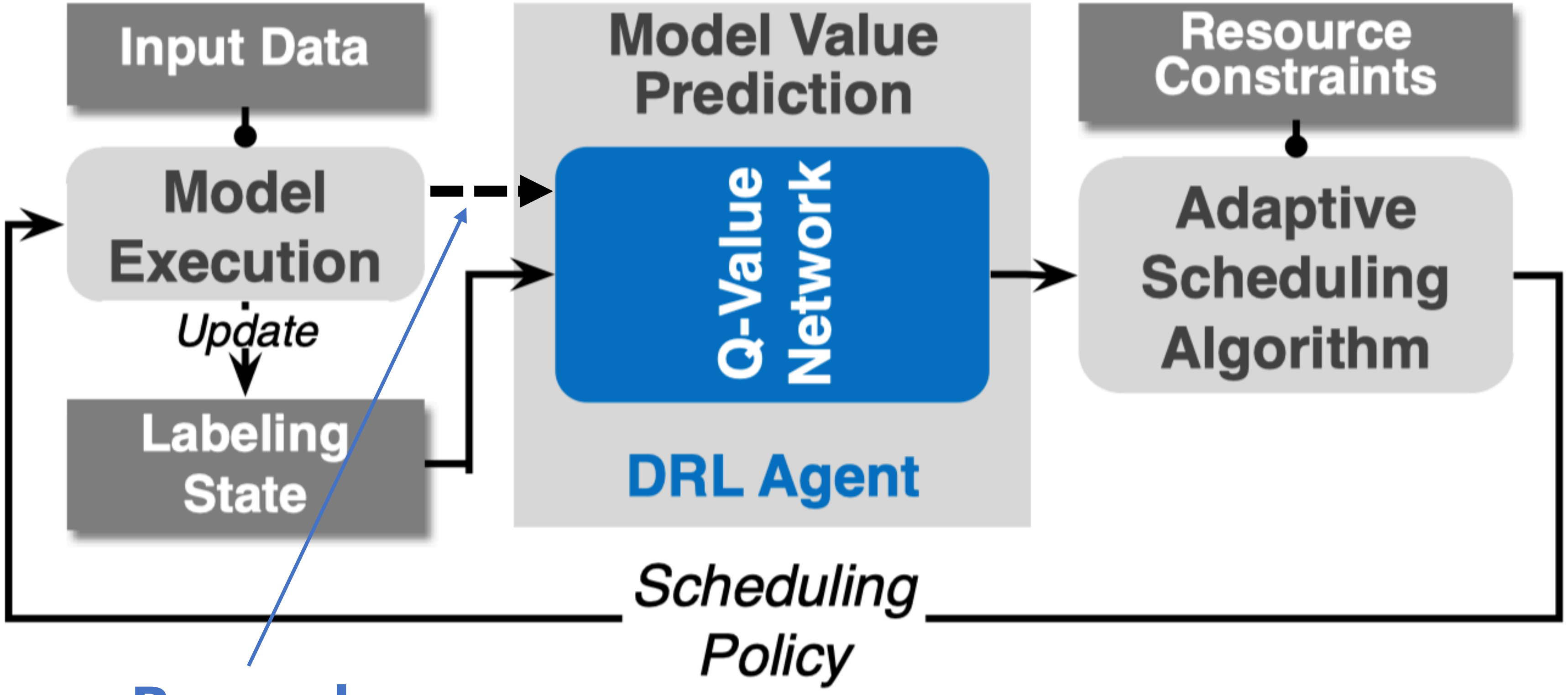
$$r(w,d) = log(\boxed{\theta_m}\boxed{\sum_{l_i \in O'(\{m\},d)} l_i.conf} + 1), O'(\{m\},d) \neq \emptyset$$

$$r(w,d) = -1, O'(\{m\},d) = \emptyset$$

Priority parameter

Sum of label confidence

Framework



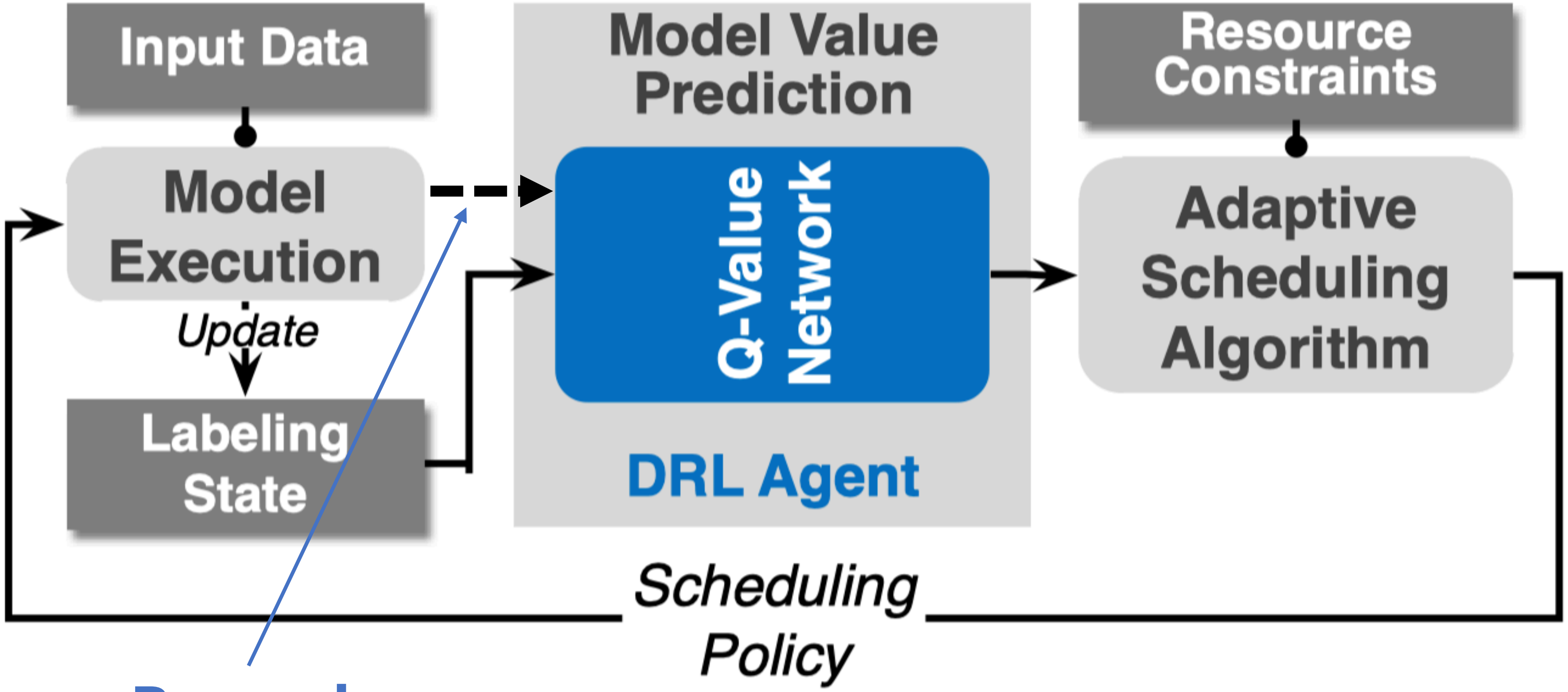
Reward:

$$r(w,d) = \boxed{log}(\theta_m \sum_{l_i \in O'(\{m\},d)} l_i.conf + 1), O'(\{m\},d) \neq \emptyset$$

$$r(w,d) = -1, O'(\{m\},d) = \emptyset$$

Logarithmic smoothing

Framework



Reward:

$$r(w,d) = log(\theta_m \sum_{l_i \in O'(\{m\},d)} l_i.conf + 1), O'(\{m\},d) \neq \emptyset$$

$$r(w,d) = \boxed{-1}, O'(\{m\},d) = \emptyset$$

Punishment

DRL Agent

- Deep Q-Network
  - input -> dense-layer -> ReLU -> (N+1)-output
  - The ***+1*** action is an ***END*** action

DRL Agent

- Deep Q-Network
  - 1104-input -> 256-dense -> ReLU -> (30+1)-output
  - The ***+1*** action is an ***END*** action
  - The reward of selecting ***END*** is ***0***

DRL Agent

- Deep Q-Network
  - 1104-input -> 256-dense -> ReLU -> (30+1)-output
  - The ***+1*** action is an ***END*** action
  - The reward of selecting ***END*** is ***0***
- Training mechanisms:
  - Original DQN
  - Double DQN
  - Dueling DQN
  - Deep SARSA

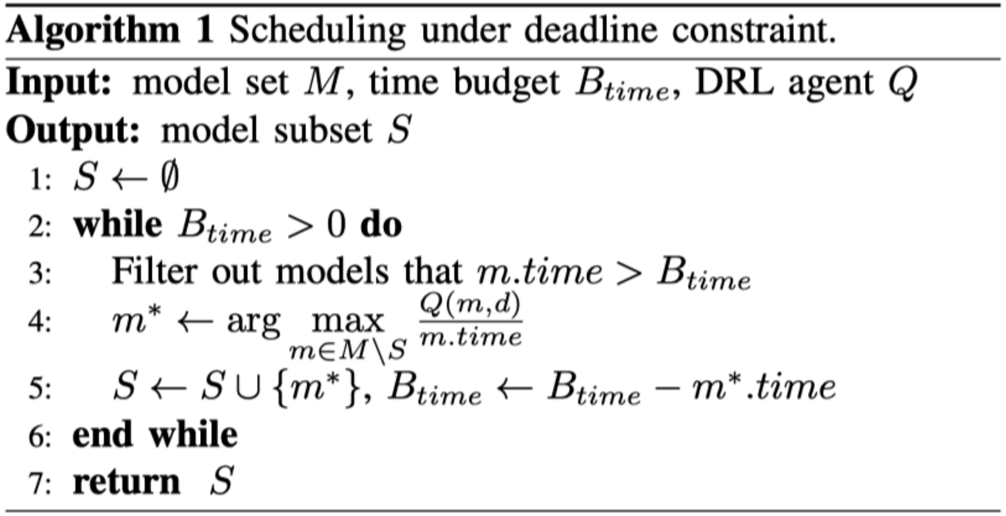
Adaptive Scheduling

- An **adaptive submodular function maximization** problem.
- Existing approximate algorithms with performance guarantee is infeasible for our problem, since they require partial permutation on items.

Adaptive Scheduling

- Two common constraints of computing resources are studied:

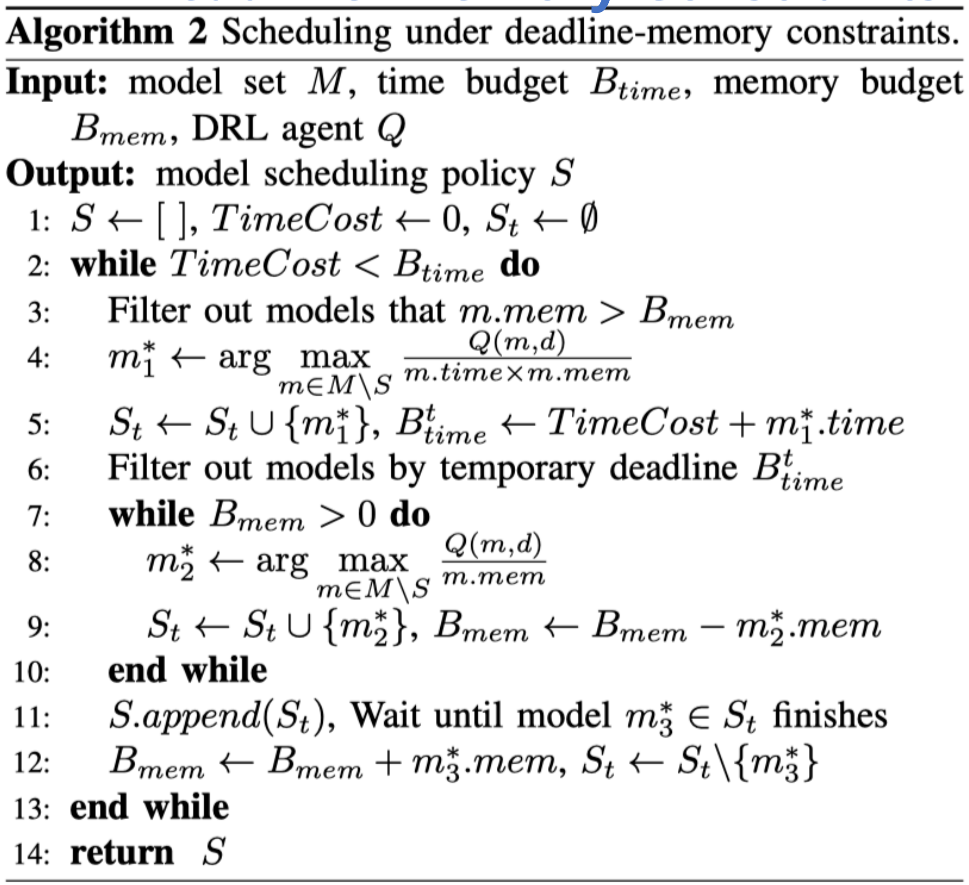
1-D Deadline Constraint



Algorithm 1 Scheduling under deadline constraint.

```
Input: model set $M$, time budget $B_{time}$, DRL agent $Q$
Output: model subset $S$
$S \leftarrow \emptyset$
while $B_{time} > 0$ do
    Filter out models that $m.time > B_{time}$
    $m^* \leftarrow \arg\max_{m \in M \backslash S} \frac{Q(m,d)}{m.time}$
    $S \leftarrow S \cup \{m^*\}$, $B_{time} \leftarrow B_{time} - m^*.time$
end while
return $S$
```

2-D Deadline-Memory Constraints



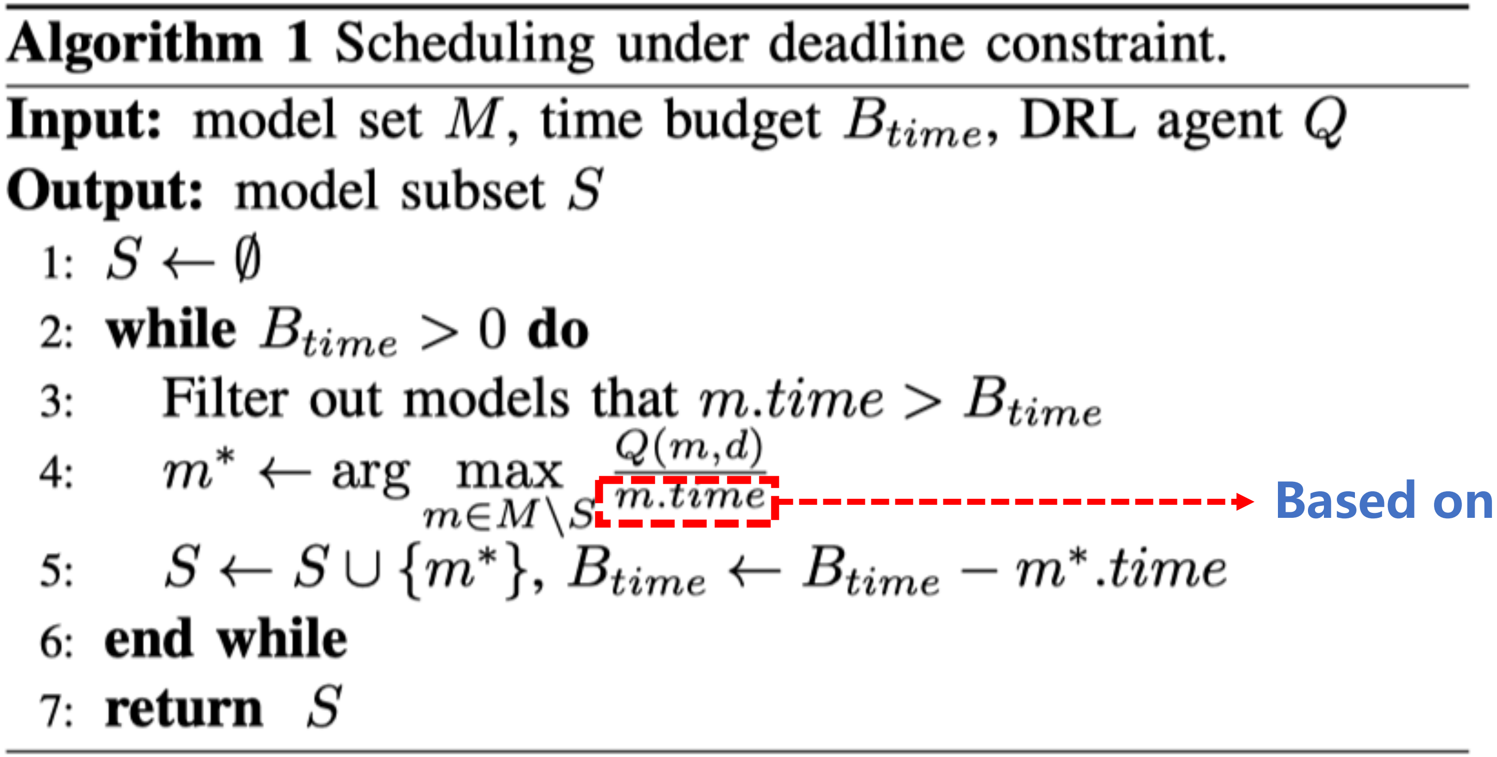
Algorithm 2 Scheduling under deadline-memory constraints.

```
Input: model set $M$, time budget $B_{time}$, memory budget
    $B_{mem}$, DRL agent $Q$
Output: model scheduling policy $S$
$S \leftarrow [\,]$, $TimeCost \leftarrow 0$, $S_t \leftarrow \emptyset$
while $TimeCost < B_{time}$ do
    Filter out models that $m.mem > B_{mem}$
    $m_1^* \leftarrow \arg\max_{m \in M \backslash S} \frac{Q(m,d)}{m.time \times m.mem}$
    $S_t \leftarrow S_t \cup \{m_1^*\}$, $B_{time}^t \leftarrow TimeCost + m_1^*.time$
    Filter out models by temporary deadline $B_{time}^t$
    while $B_{mem} > 0$ do
        $m_2^* \leftarrow \arg\max_{m \in M \backslash S} \frac{Q(m,d)}{m.mem}$
        $S_t \leftarrow S_t \cup \{m_2^*\}$, $B_{mem} \leftarrow B_{mem} - m_2^*.mem$
    end while
    $S.append(S_t)$, Wait until model $m_3^* \in S_t$ finishes
    $B_{mem} \leftarrow B_{mem} + m_3^*.mem$, $S_t \leftarrow S_t \backslash \{m_3^*\}$
end while
return $S$
```

Adaptive Scheduling

- An adaptive submodular function maximization problem.
- Two common constraints are studied:

1-D Deadline Constraint



Algorithm 1 Scheduling under deadline constraint.

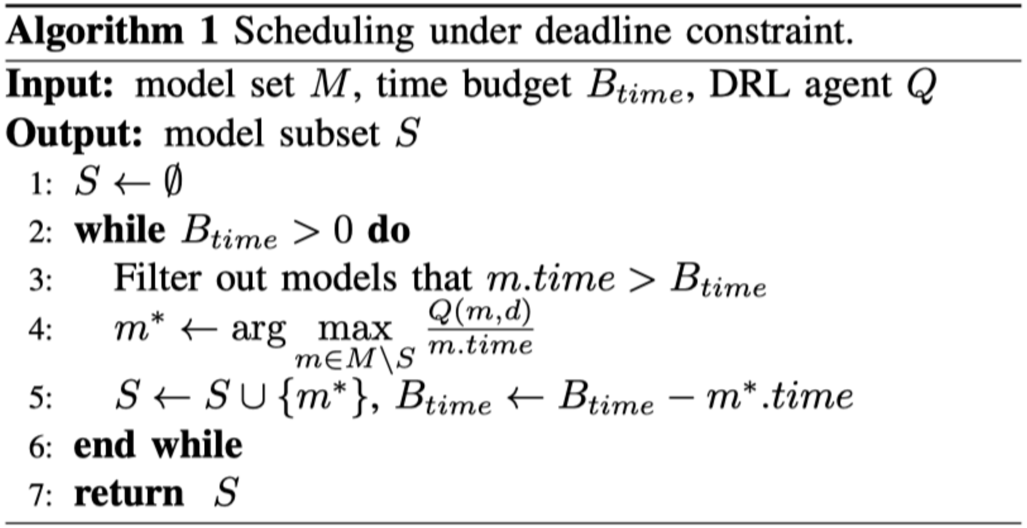
```
Input: model set $M$, time budget $B_{time}$, DRL agent $Q$
Output: model subset $S$
$S \leftarrow \emptyset$
while $B_{time} > 0$ do
    Filter out models that $m.time > B_{time}$
    $m^* \leftarrow \arg\max_{m \in M \setminus S} \frac{Q(m,d)}{\boxed{m.time}}$
    $S \leftarrow S \cup \{m^*\}$, $B_{time} \leftarrow B_{time} - m^*.time$
end while
return $S$
```

Based on model profiling on sampled data.

Adaptive Scheduling

- An adaptive submodular function maximization problem.
- Two common constraints are studied:

1-D Deadline Constraint

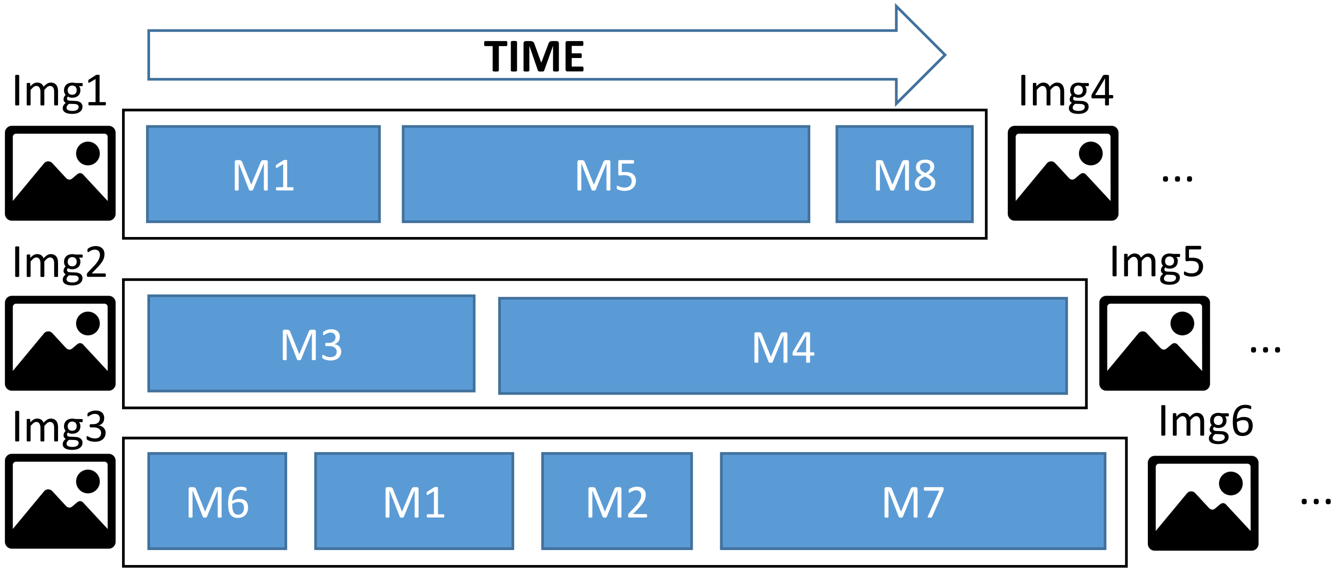


Algorithm 1 Scheduling under deadline constraint.

Input: model set M, time budget B_{time}, DRL agent Q

Output: model subset S

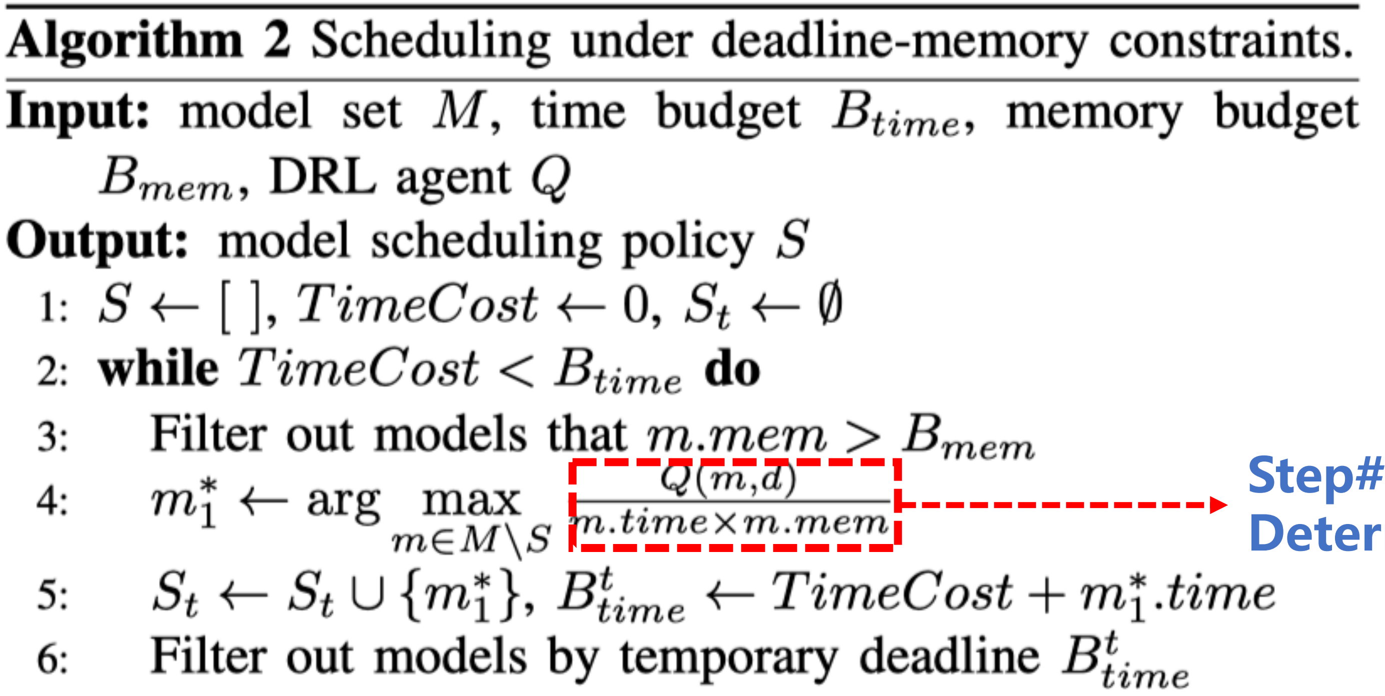
```
$S \leftarrow \emptyset$
while $B_{time} > 0$ do
    Filter out models that $m.time > B_{time}$
    $m^* \leftarrow \arg\max_{m \in M \setminus S} \frac{Q(m,d)}{m.time}$
    $S \leftarrow S \cup \{m^*\}$, $B_{time} \leftarrow B_{time} - m^*.time$
end while
return $S$
```



Adaptive Scheduling

- An adaptive submodular function maximization problem.
- Two common constraints are studied:

2-D Deadline-Memory Constraints



Algorithm 2 Scheduling under deadline-memory constraints.

Input: model set M, time budget B_{time}, memory budget B_{mem}, DRL agent Q

Output: model scheduling policy S

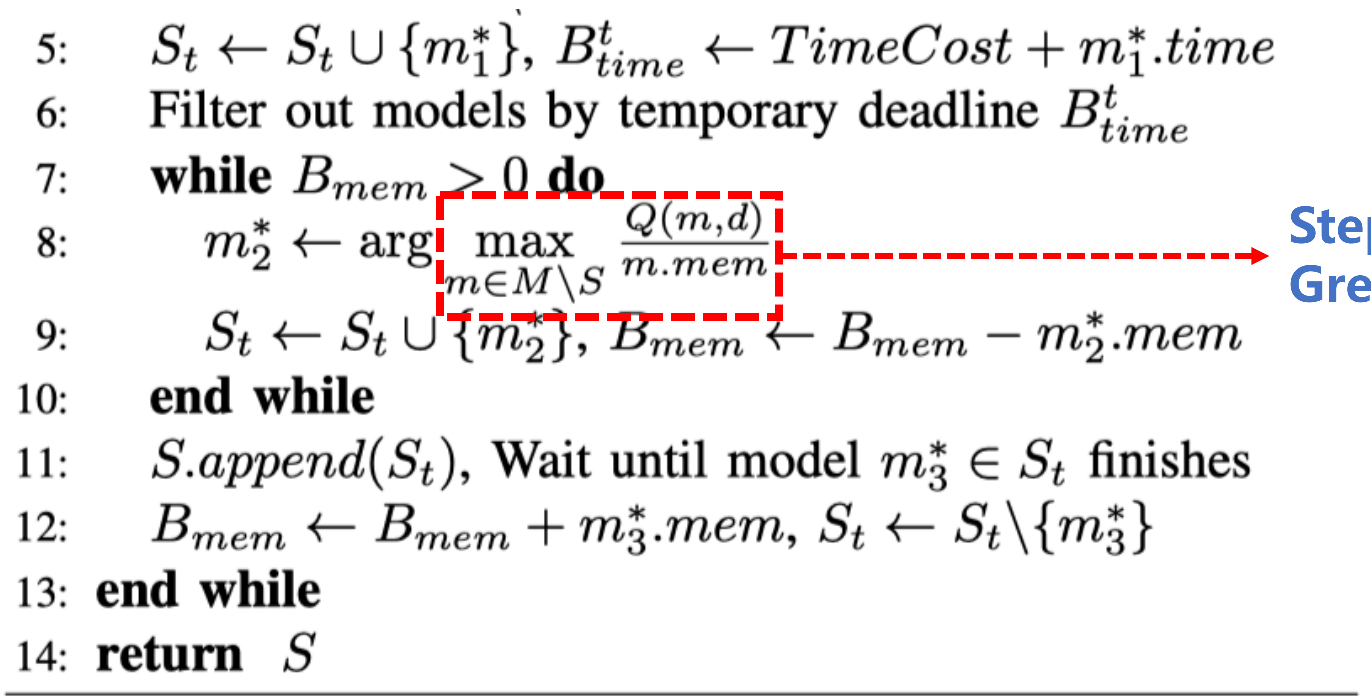
```
$S \leftarrow [\,]$, $TimeCost \leftarrow 0$, $S_t \leftarrow \emptyset$
while $TimeCost < B_{time}$ do
    Filter out models that $m.mem > B_{mem}$
    $m_1^* \leftarrow \arg\max_{m \in M \setminus S} \boxed{\frac{Q(m,d)}{m.time \times m.mem}}$
    $S_t \leftarrow S_t \cup \{m_1^*\}$, $B_{time}^t \leftarrow TimeCost + m_1^*.time$
    Filter out models by temporary deadline $B_{time}^t$
```

Step#1:
Determine the temporary *deadline*.

Adaptive Scheduling

- An adaptive submodular function maximization problem.
- Two common constraints are studied:

2-D Deadline-Memory Constraints



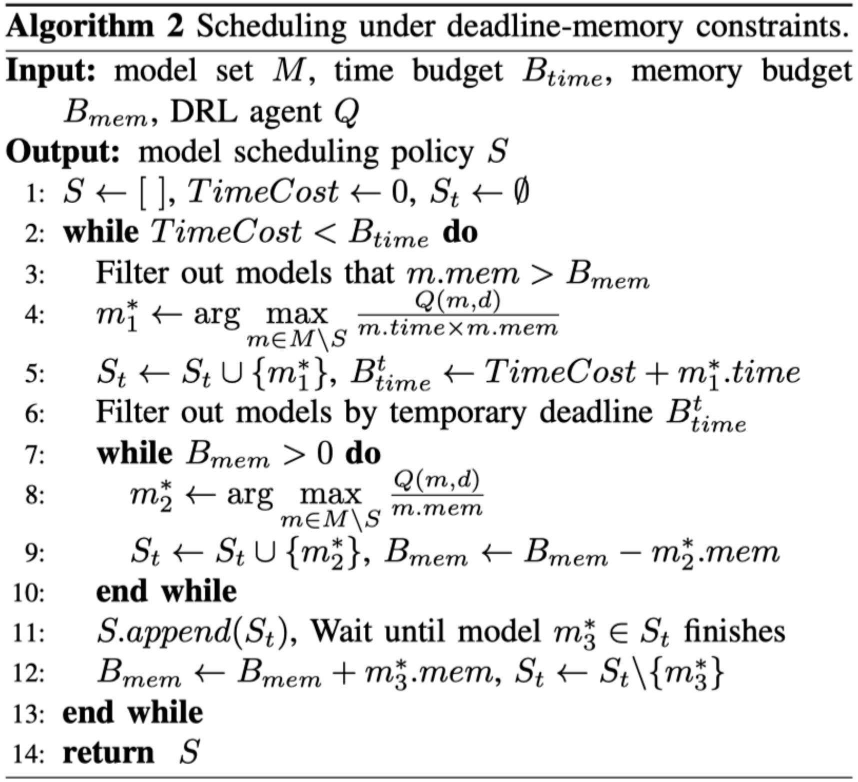
```
    $S_t \leftarrow S_t \cup \{m_1^*\},\ B_{time}^t \leftarrow TimeCost + m_1^*.time$
    Filter out models by temporary deadline $B_{time}^t$
    while $B_{mem} > 0$ do
        $m_2^* \leftarrow \arg\boxed{\max_{m \in M \setminus S} \frac{Q(m,d)}{m.mem}}$
        $S_t \leftarrow S_t \cup \{m_2^*\},\ B_{mem} \leftarrow B_{mem} - m_2^*.mem$
    end while
    $S.append(S_t)$, Wait until model $m_3^* \in S_t$ finishes
    $B_{mem} \leftarrow B_{mem} + m_3^*.mem,\ S_t \leftarrow S_t \setminus \{m_3^*\}$
end while
return $S$
```

Step#2:
Greedily fill in the memory pool.

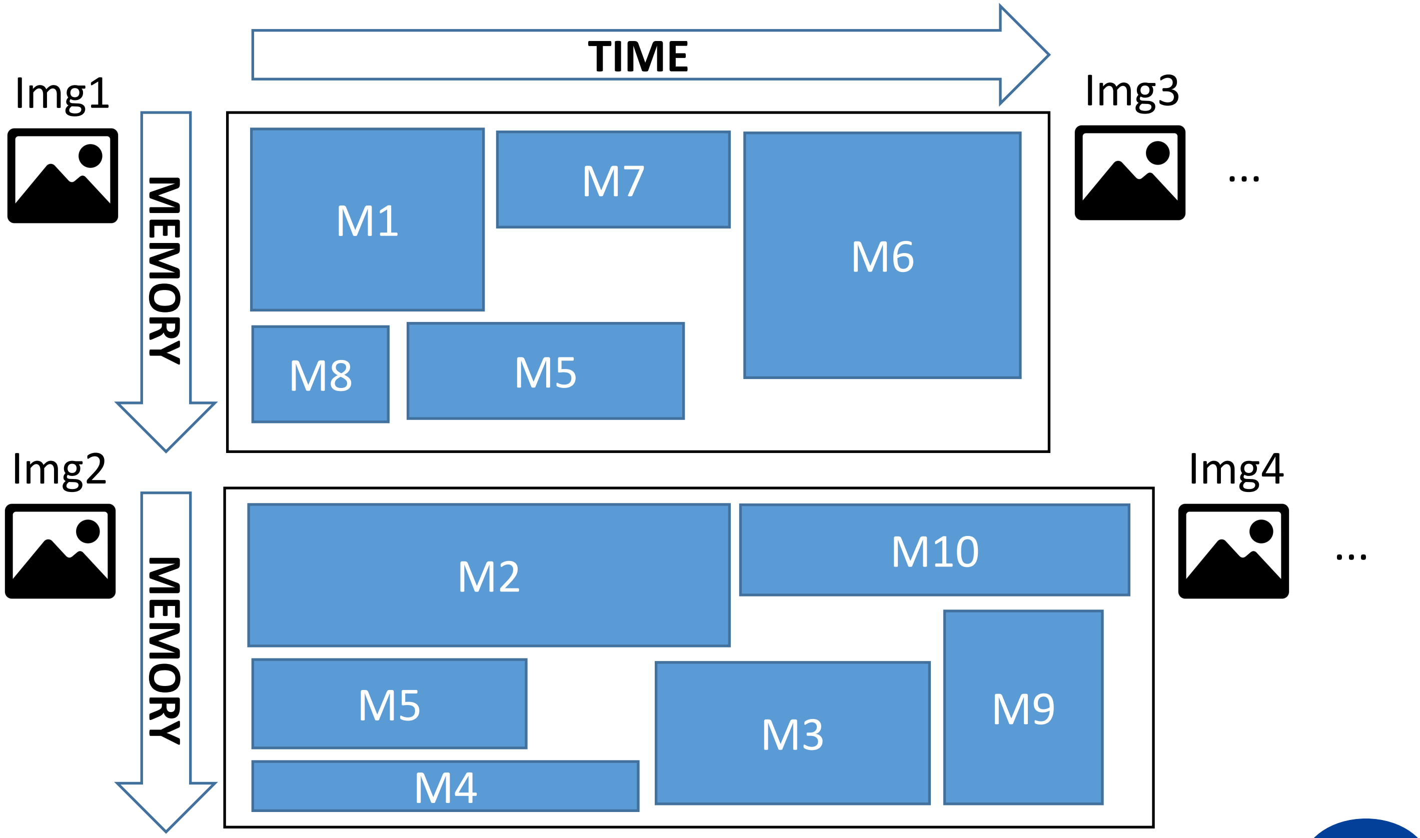
Adaptive Scheduling

- An adaptive submodular function maximization problem.
- Two common constraints are studied:

2-D Deadline-Memory Constraints



```
Algorithm 2 Scheduling under deadline-memory constraints.
Input: model set $M$, time budget $B_{time}$, memory budget
    $B_{mem}$, DRL agent $Q$
Output: model scheduling policy $S$
$S \leftarrow [\,]$, $TimeCost \leftarrow 0$, $S_t \leftarrow \emptyset$
while $TimeCost < B_{time}$ do
    Filter out models that $m.mem > B_{mem}$
    $m_1^* \leftarrow \arg\max_{m \in M \setminus S} \frac{Q(m,d)}{m.time \times m.mem}$
    $S_t \leftarrow S_t \cup \{m_1^*\}$, $B_{time}^t \leftarrow TimeCost + m_1^*.time$
    Filter out models by temporary deadline $B_{time}^t$
    while $B_{mem} > 0$ do
        $m_2^* \leftarrow \arg\max_{m \in M \setminus S} \frac{Q(m,d)}{m.mem}$
        $S_t \leftarrow S_t \cup \{m_2^*\}$, $B_{mem} \leftarrow B_{mem} - m_2^*.mem$
    end while
    $S.append(S_t)$, Wait until model $m_3^* \in S_t$ finishes
    $B_{mem} \leftarrow B_{mem} + m_3^*.mem$, $S_t \leftarrow S_t \backslash \{m_3^*\}$
end while
return $S$
```



Outline

- Resource-wasting multi-model inference workloads
- Rule-based scheduler
- Learning-based scheduler
- **Evaluation**
- Conclusion

Experimental Setup

Multi-Model Image Labeling Workloads

Task	Label#
Object Detection	80
Scene Classification	365
Face Detection	1
Face Landmark Localization	70
Pose Estimation	17
Emotion Classification	7
Gender Classification	2
Action Classification	40
Hand Landmark Localization	42
Dog Classification	120
10 Tasks	**1104 Labels**

MSCOCO-2017

COCO
Common Objects in Context

MIRFLICKR-25k

by Silke Gerstenkorn

by Dave Wild

by Hugo A.B. Olivas

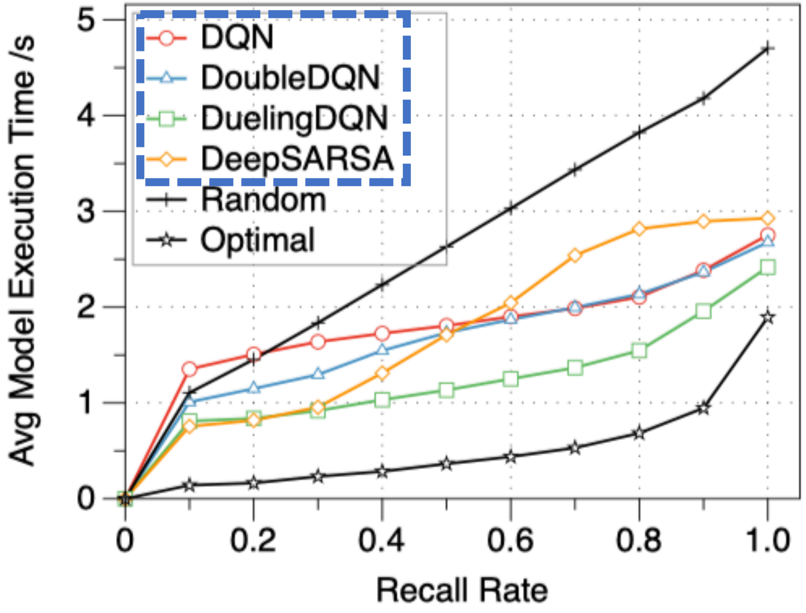
by Martin P. Szymczak

by Mani Babbar

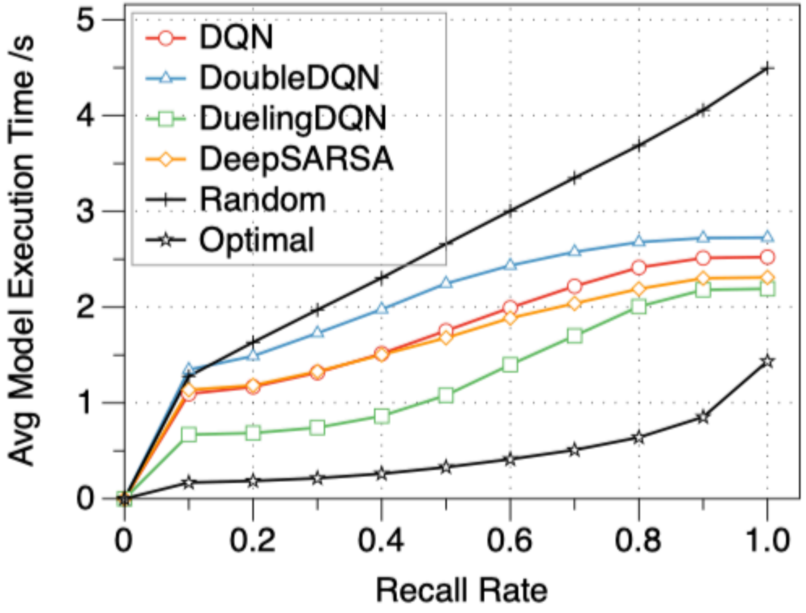
by Lee Otis

Evaluation: DRL Agent

Different Training Mechanisms

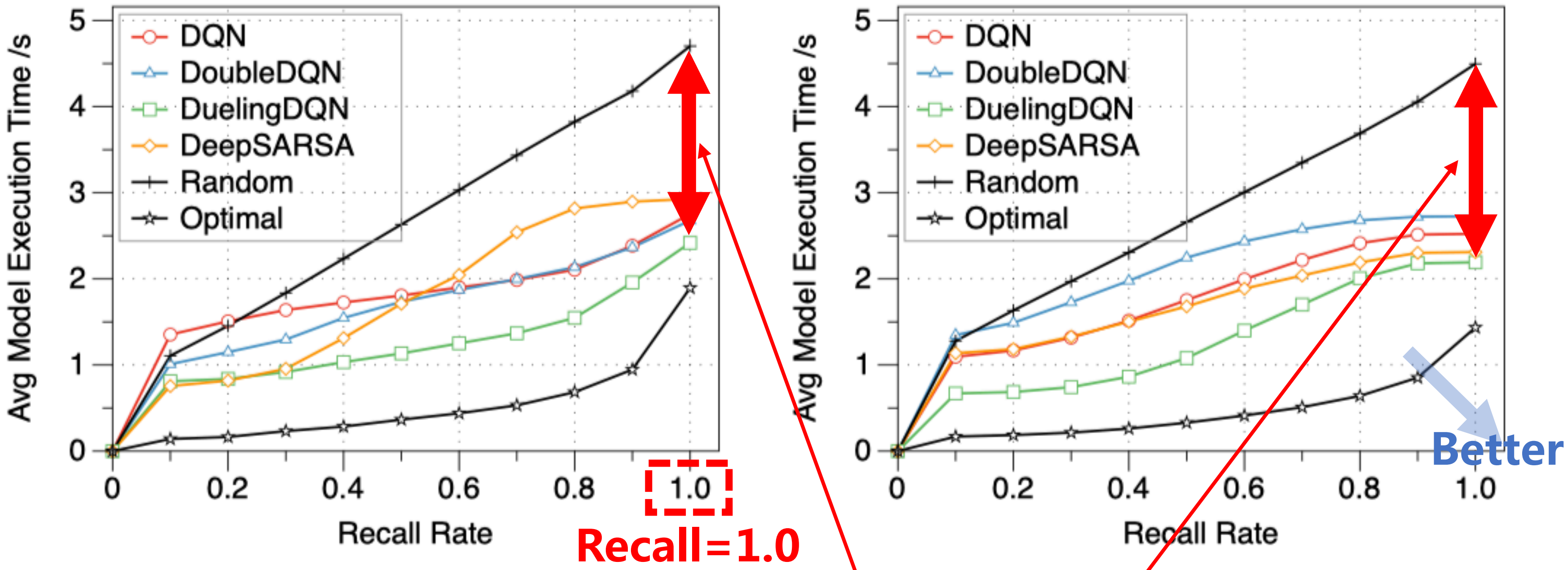


(a) MSCOCO 2017



(b) MirFlickr25

Evaluation: DRL Agent

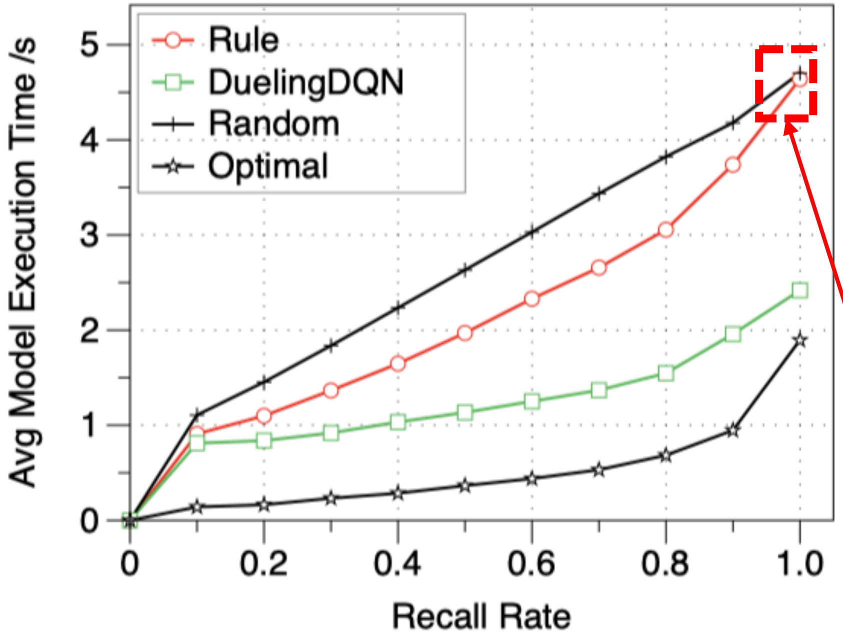


(a) MSCOCO 2017

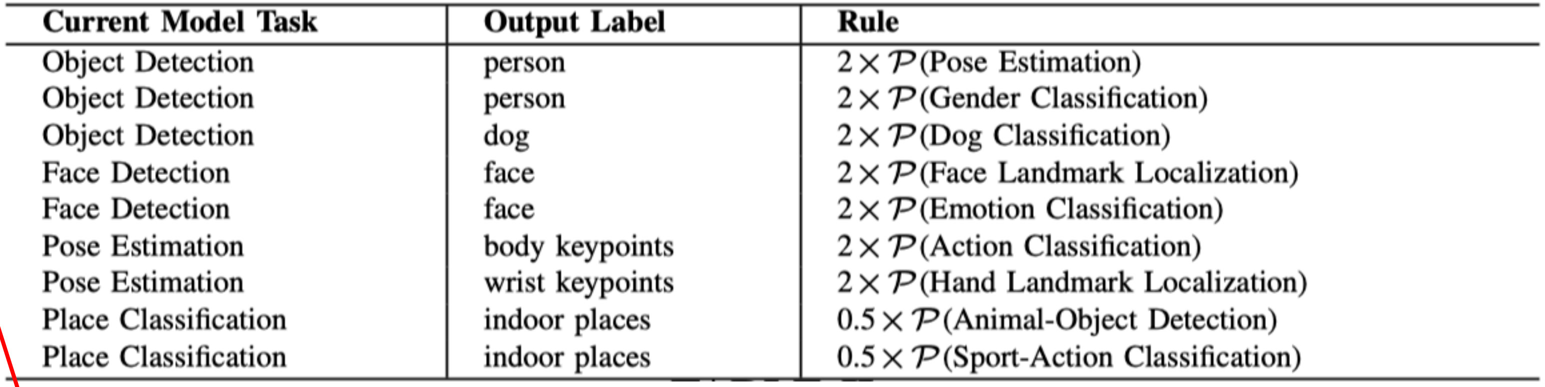
(b) MirFlickr25

Saving **48.6-51.2%** execution time **without loss** of valuable labels.

Evaluation: DRL vs. Rules



10 Manual Rules



Current Model Task	Output Label	Rule
Object Detection	person	$2 \times \mathcal{P}$(Pose Estimation)
Object Detection	person	$2 \times \mathcal{P}$(Gender Classification)
Object Detection	dog	$2 \times \mathcal{P}$(Dog Classification)
Face Detection	face	$2 \times \mathcal{P}$(Face Landmark Localization)
Face Detection	face	$2 \times \mathcal{P}$(Emotion Classification)
Pose Estimation	body keypoints	$2 \times \mathcal{P}$(Action Classification)
Pose Estimation	wrist keypoints	$2 \times \mathcal{P}$(Hand Landmark Localization)
Place Classification	indoor places	$0.5 \times \mathcal{P}$(Animal-Object Detection)
Place Classification	indoor places	$0.5 \times \mathcal{P}$(Sport-Action Classification)

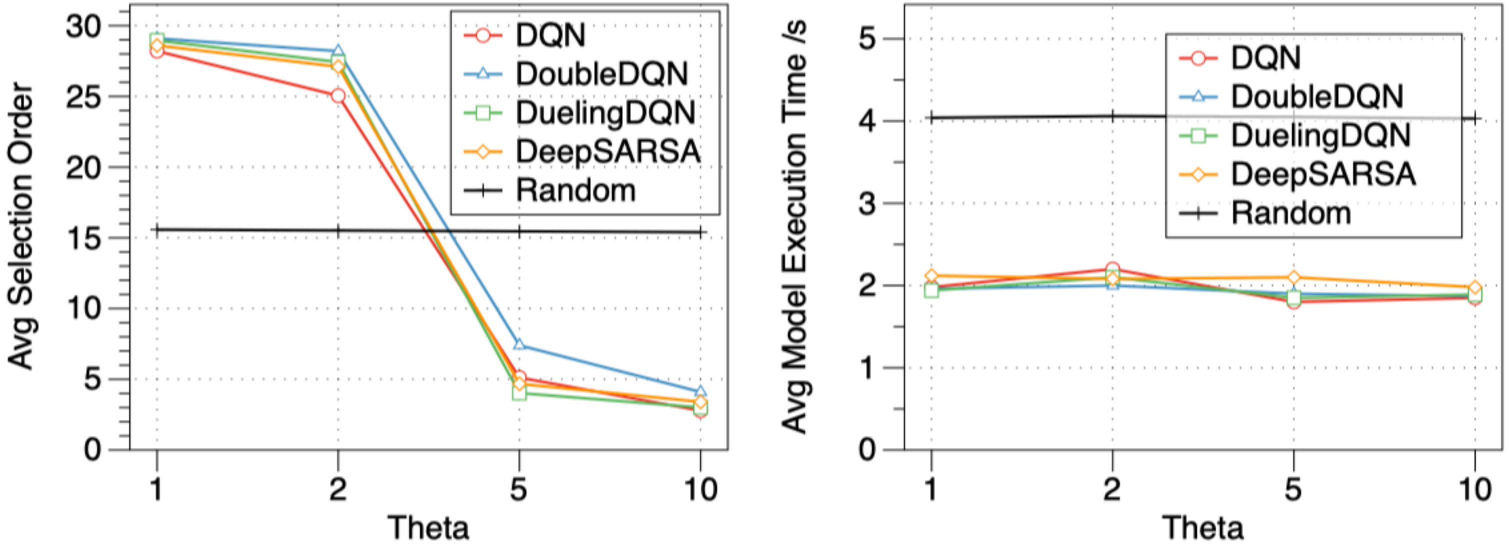
Saving only **1.4%** execution time
when required recall is 1.0.

Evaluation: Model Priority

Adjusting priority of the face-detection model.

$$r(w,d) = log(\boxed{\theta_m}\sum_{l_i \in O'(\{m\},d)} l_i.conf + 1), O'(\{m\},d) \neq \emptyset$$

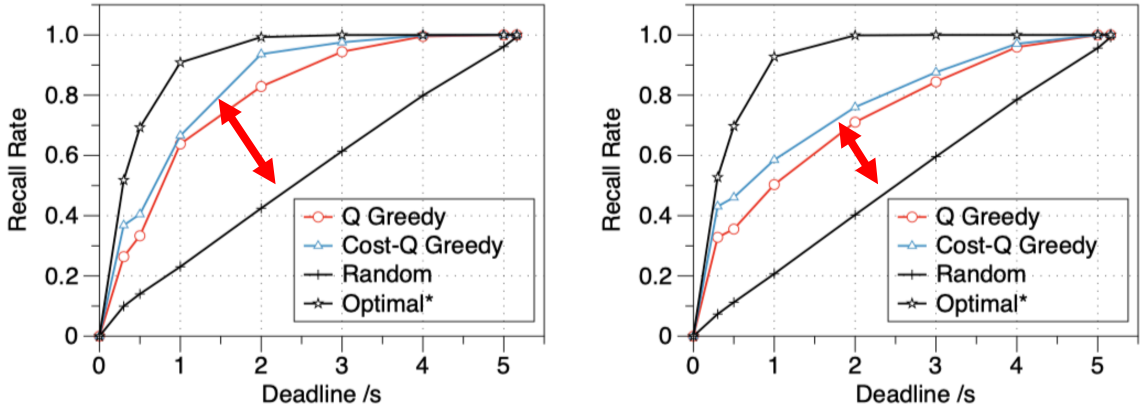
Priority parameter



(a) Average execution order **(b) Average time cost**

Evaluation: Scheduling Algorithms

1-D Deadline Constraint

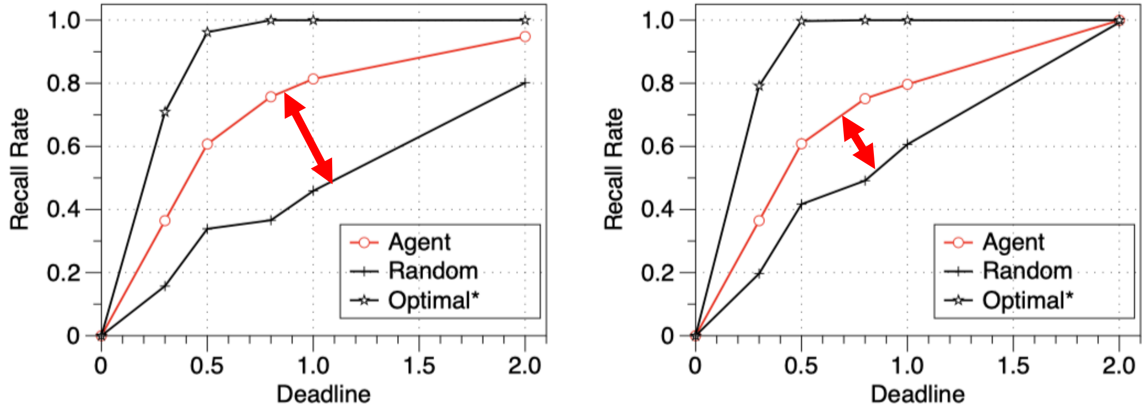


(a) MSCOCO 2017

(b) MirFlickr25

Boosts **188.7-309.5%** recall of valuable labels under 0.5s deadline constraint.

2-D Deadline-Memory Constraints



(a) 8GB Memory

(b) 12GB Memory

Boosts **106.9%/52.8%** recall of valuable labels under 0.8s deadline and 8/12GB memory constraints.

Outline

- Resource-wasting multi-model inference workloads
- Rule-based scheduler
- Learning-based scheduler
- Evaluation
- **Conclusion**

The Take-Home Message

Adaptive model scheduling can improve the efficiency of multi-model inference workloads by avoiding valueless execution.

Lab for Intelligent Networking & Knowledge Engineering

12 Faculty members, 2 Post-Doc, 3 Secretary; 7 with PhD from abroad

XiangYang Li

IEEE Fellow，
ACM Fellow，
ACM China Co-Chair

Yanyong Zhang

IEEE Fellow，
Prof. in Rutgers，
NSF Career

Panlong Yang

CCF Dist Speaker
Wireless network、
Mobile computing

Bei Hua

High performance
computing 、
Edge computing

Haisheng Tan

HK, Tsinghua Post-Doc
Cloud computing、
Algorithms Analysis

Xin Guo

Edge computing，
Security of IoT

Nikolaos M.Freris

USA NYU A.P.
CPS, Algorithms,
Distributed optimization
Machine learning

Yu Zhang

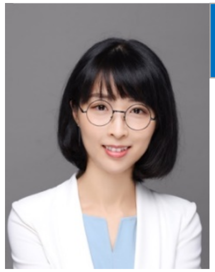
system software ,Software
optimization/security,
Quantum software

YuBo Yan

Wireless/Passive
network， IntelliSense，
IoT，SDR

Xuerong Huang

Master in HKBU
Research assistant



Lan Zhang

CCF, ACM China Doctor
Thesis Award, Youqing,
Qingcheng Award
Data understanding/trading ，
privacy protection

Hao Zhou

Japan NTII
Wireless Network Resource
Management

Xin He

Doc. University of Oulu
Passive network，
Theories of Information
and Coding

Ludi Xue

Research assistant

Comprehensive and Efficient Data Labeling via Adaptive Model Scheduling

Mu Yuan

University of Science and Technology of China

School of Computer Science and Technology

ym0813@mail.ustc.edu.cn

Lan Zhang

University of Science and Technology of China

School of Computer Science and Technology

zhanglan@ustc.edu.cn