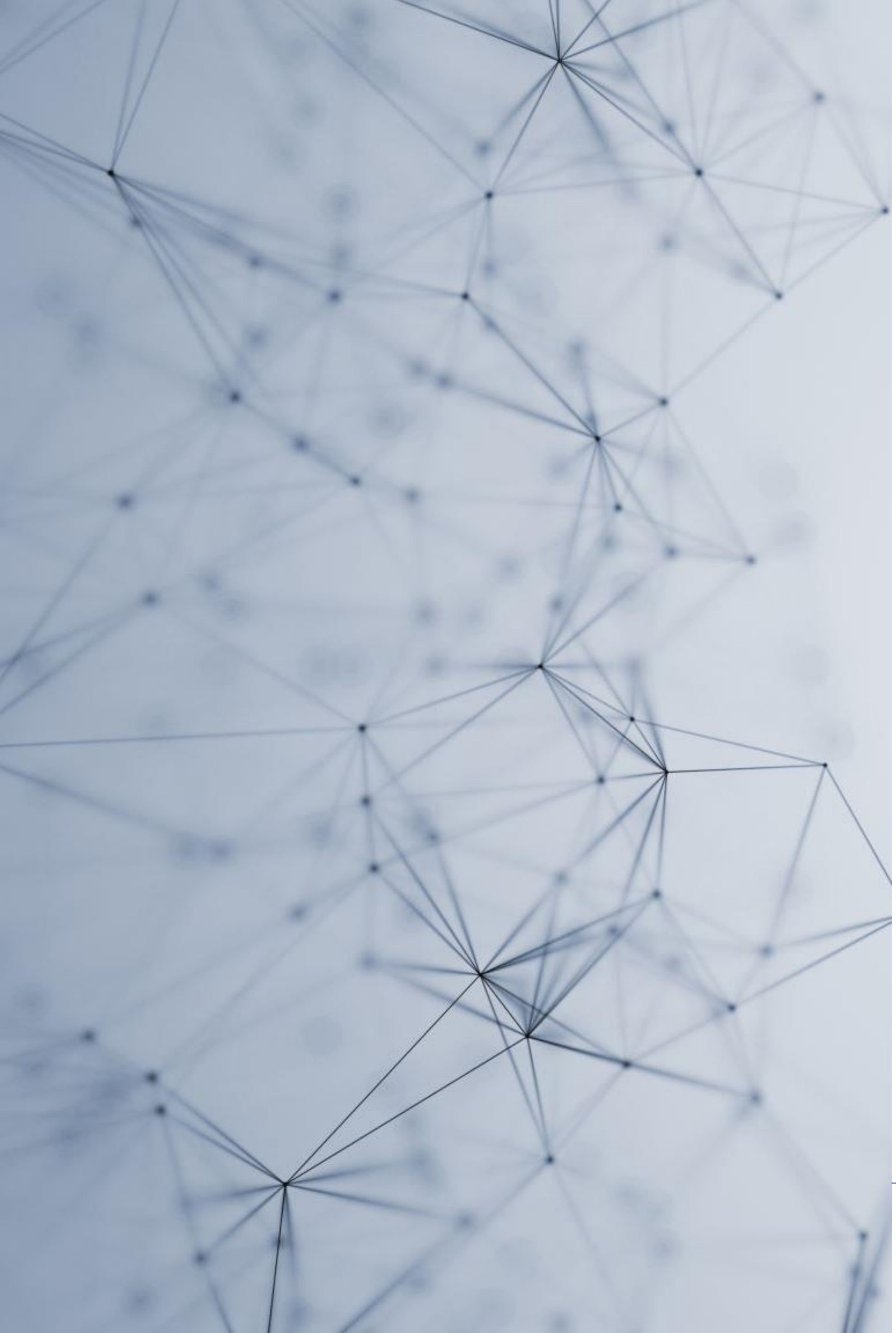


LEARNING WHEN AND WHERE TO ZOOM WITH DEEP REINFORCEMENT LEARNING

Student : LE VAN THE

ID: 20110746

Major: INTELLIGENT MECHATRONICS ENGINEERING

Uzkent, Burak, and Stefano Ermon. "Learning when and where to zoom with deep reinforcement learning." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

CONTENT

MAIN PROBLEM

High accuracy

High Expense

High Resolution Image
160x160

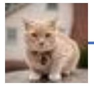
Classifier network

Mouse
Cat
Ship
.
.
Car

Compression

Low accuracy

Low Expense



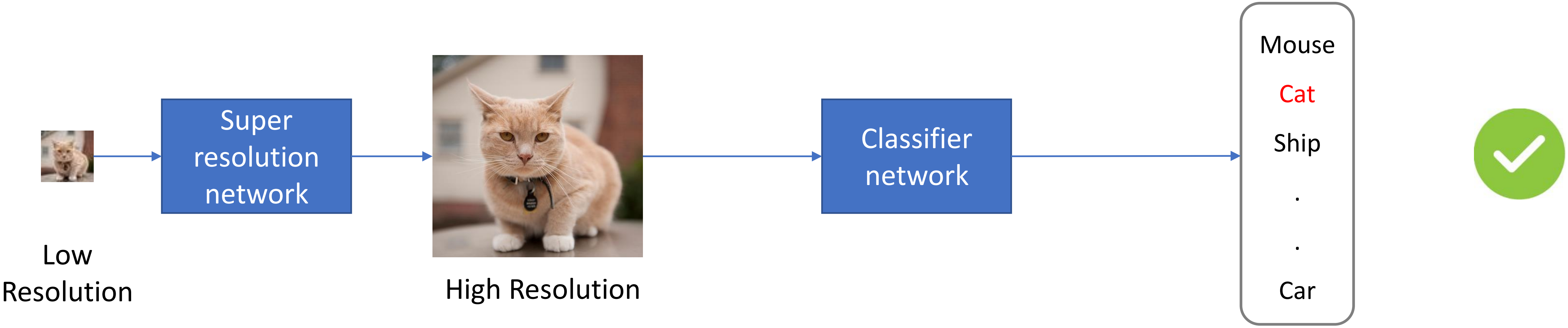
Low Resolution Image
40x40

Classifier network

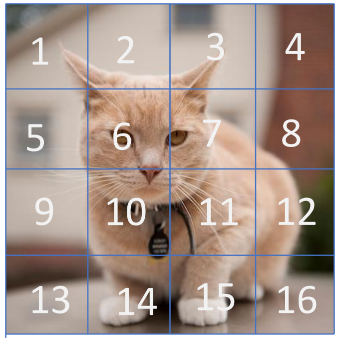
Mouse
Cat
Ship
.
.
Car

Cause: *many Information loss.*

MAIN PROBLEM



Limitation? ***- Increase the computational cost with unmeaning information.***

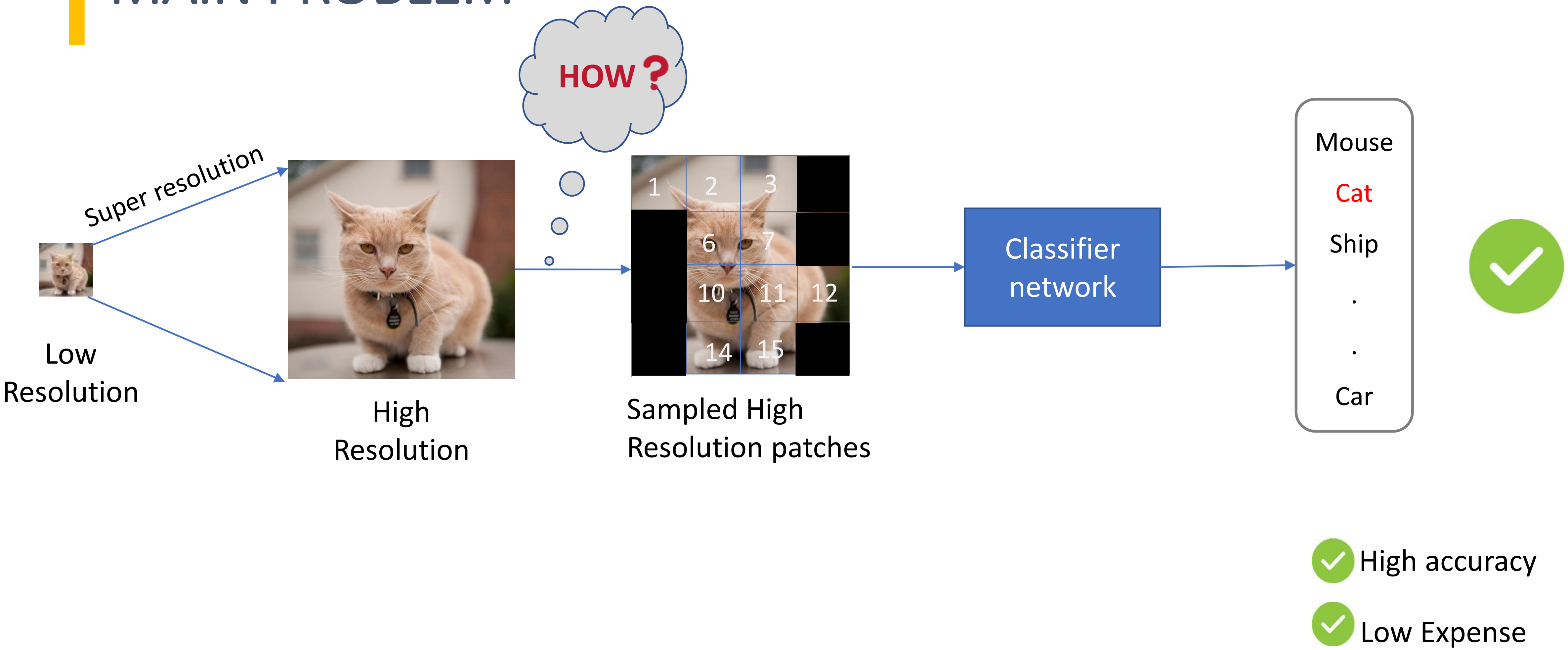


Block 4,8,9,13,16 are not meaning in cat classification

➔ no need to use these block as input of Classifier Network

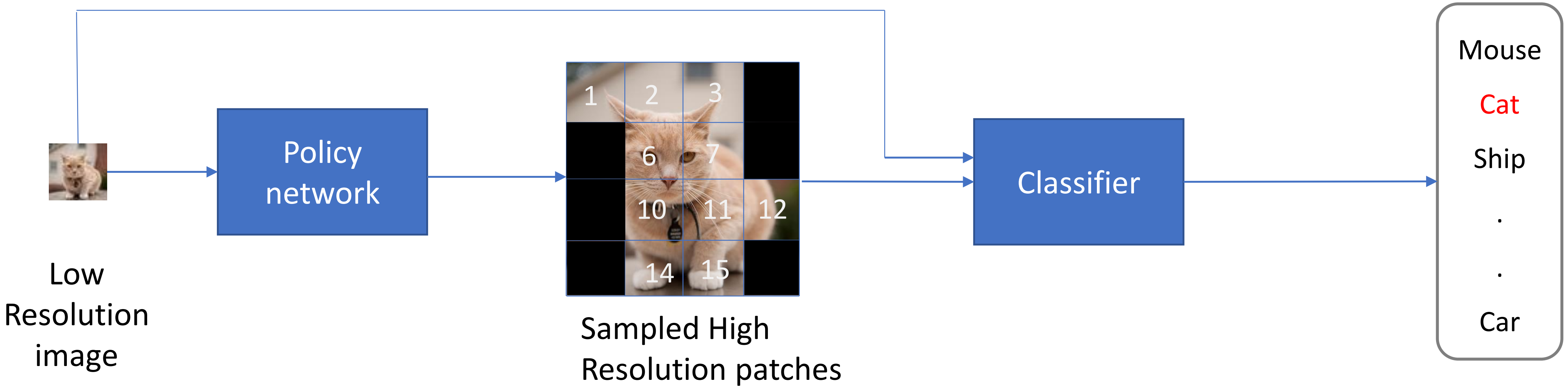
MAIN PROBLEM

MAIN PROBLEM



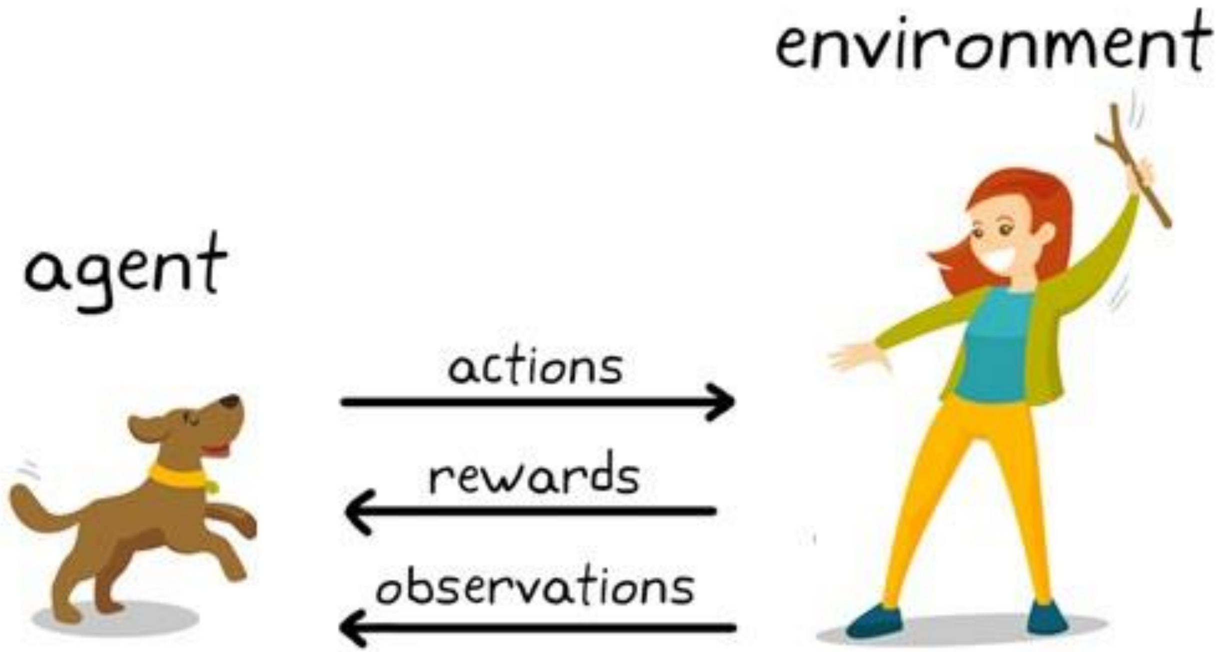
In example, the method can reduce of 6/16 memory cost and computation

SOLUTION BY DEEP REINFORCEMENT LEARNING

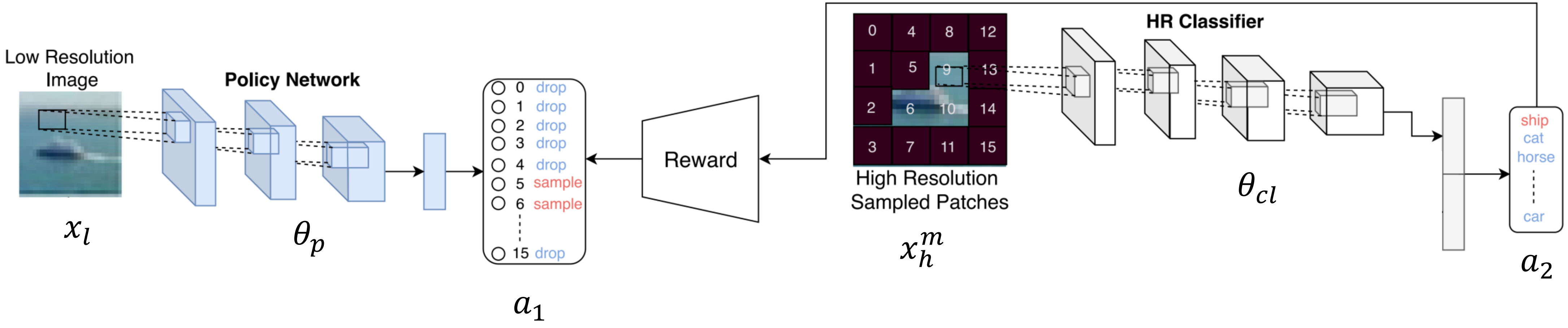


Proposed framework which dynamically drops image patches conditioned on the low-resolution image.

SOLUTION BY DEEP REINFORCEMENT LEARNING

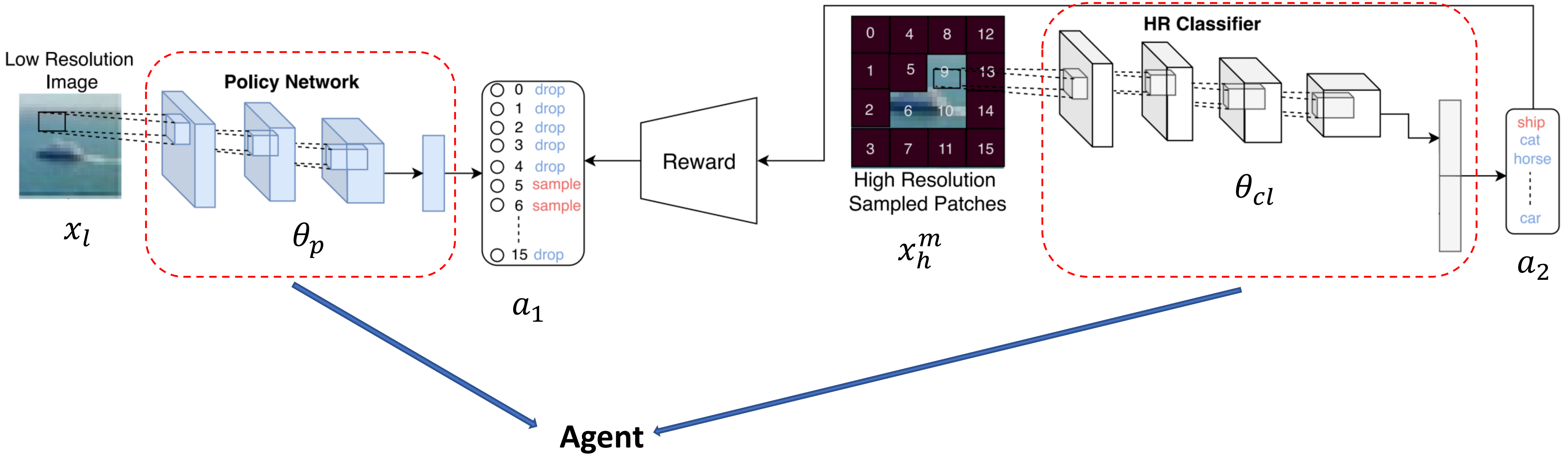


SOLUTION BY DEEP REINFORCEMENT LEARNING



The workflow of the *PatchDrop*

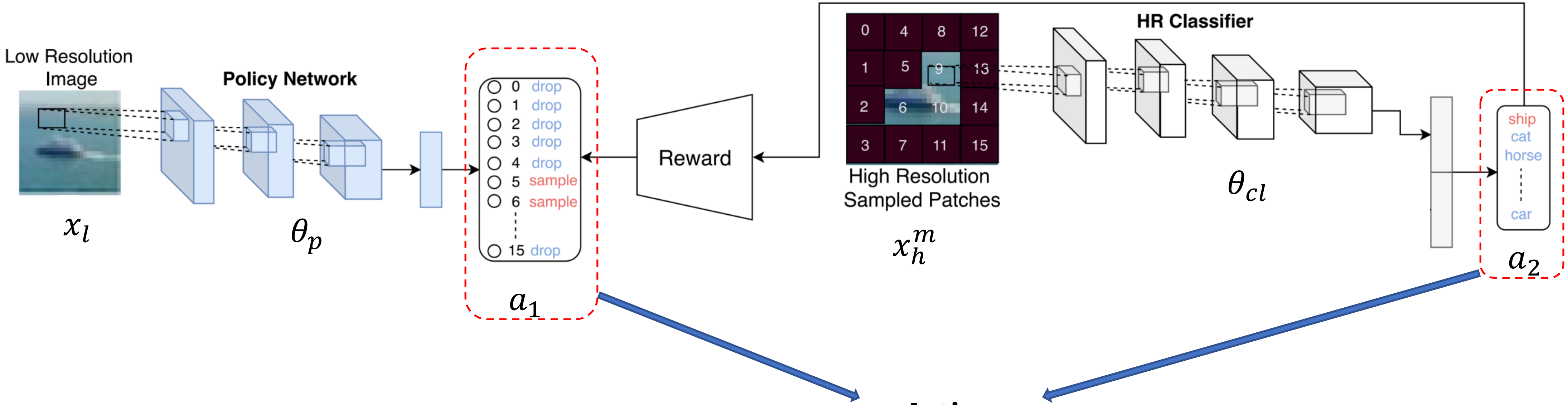
SOLUTION BY DEEP REINFORCEMENT LEARNING



Agent: The part of application that do the action (Policy Network and HR Classifier)

Environment: The NN architecture that performs optimization to get the best performance and ground true data.

SOLUTION BY DEEP REINFORCEMENT LEARNING



Action

$\boldsymbol{a_1} \in \{0,1\}^P$

$\boldsymbol{a_1} \in \{0, 1, \dots N-1\}$

Actions:

Actions in step 1 (a_1): Choose the patches in low resolution image to sampled high-resolution image. *(discrete action)*

Example: $a_1 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$ ➔ drop all patches

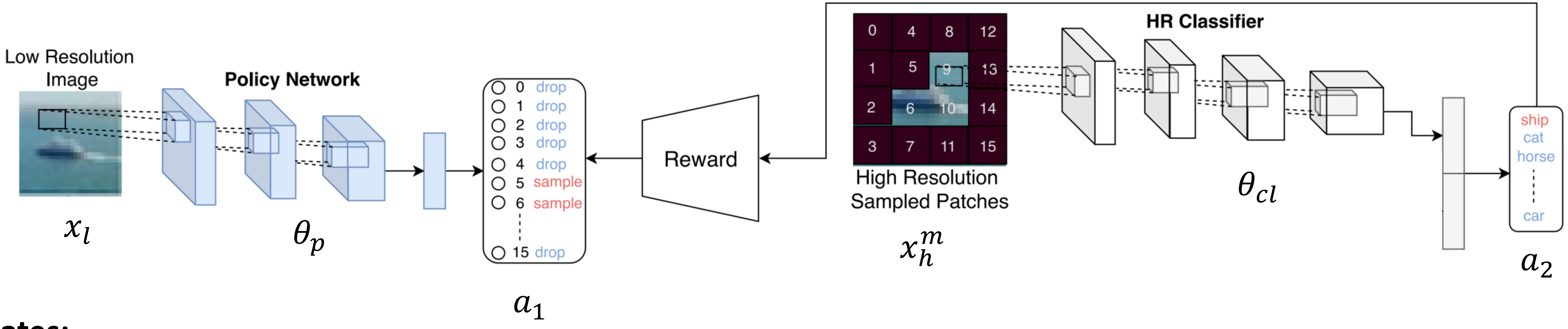
➔Number of action a_1: **2^{16}**

Actions in step 2 (a_2): Choose the label of classification. *(discrete action)*

Example: $a_2 = 0$ ➔ The choosing label is Ship

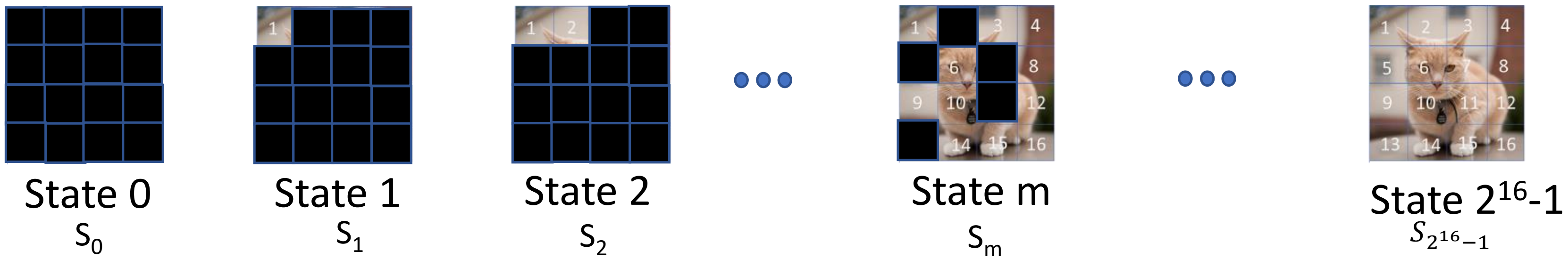
➔Number of action a_2: N (N is the number of class)

SOLUTION BY DEEP REINFORCEMENT LEARNING

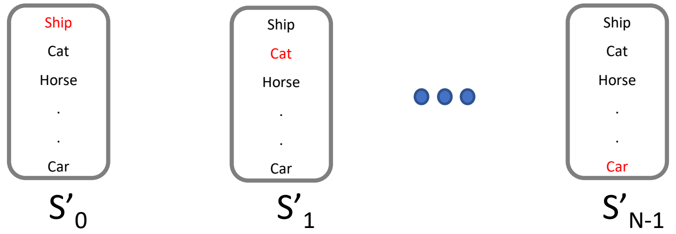


States:

States in step 1: 2^{16} cases of choosing patch of image to generate high resolution.



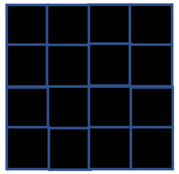
States in step 2: Numbers of classification (N).



SOLUTION BY DEEP REINFORCEMENT LEARNING

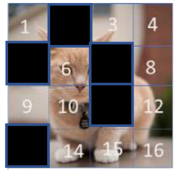
step 1:

$a_1 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$ →



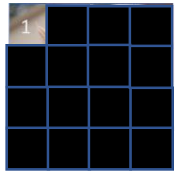
S_0
State 0

$a_1 = [0,1,0,0,0,1,0,1,0,0,1,0,1,0,0,0]$ →



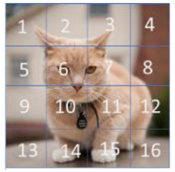
S_m
State m

$a_1 = [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$ →



S_1
State 1

$a_1 = [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]$ →



$S_{2^{16}-1}$
State 2^{16}-1

step 2:

a_2 =0 →

Ship
Cat
Horse
.
.
Car

S'_0

a_2 =1 →

Ship
Cat
Horse
.
.
Car

S'_1

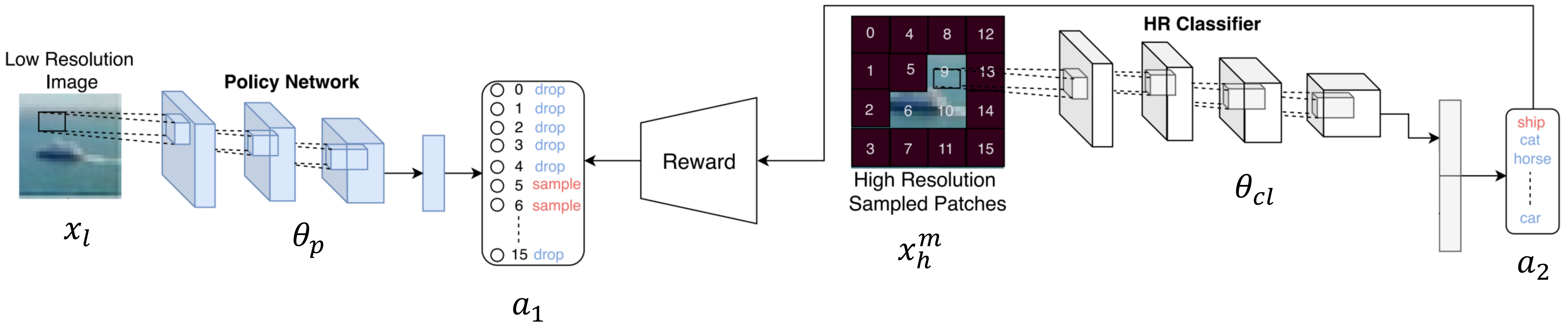
...

a_2 =N-1 →

Ship
Cat
Horse
.
.
Car

S'_{N-1}

SOLUTION BY DEEP REINFORCEMENT LEARNING



Policy Network **Classifier**

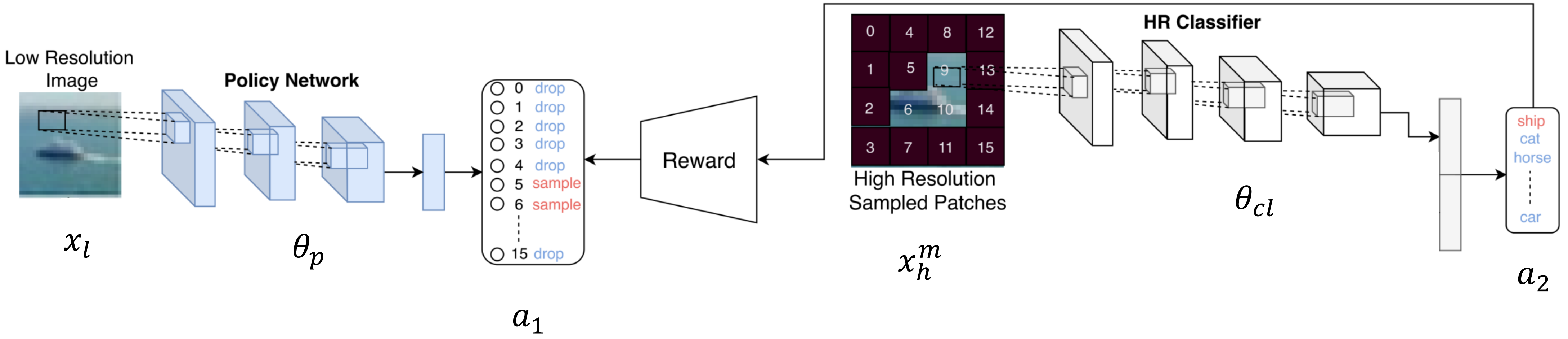
Actions $\boldsymbol{a_1} \in \{0,1\}^P$ $\boldsymbol{a_1} \in \{0, 1, \ldots N-1\}$

Rewards

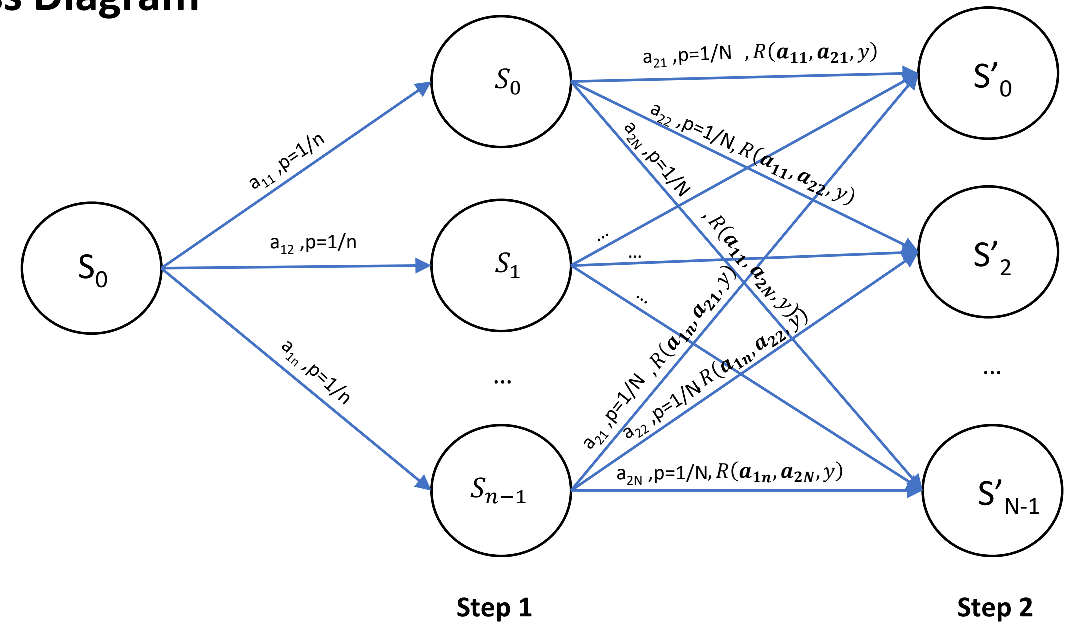
$$R(\boldsymbol{a_1}, \boldsymbol{a_2}, y) = \begin{cases} 1 - \left(\dfrac{|\boldsymbol{a_1}|}{P}\right)^2 & if\ y = \hat{y}\ (\boldsymbol{a_2}) \\ -\sigma & Otherwise \end{cases}$$

penalizes the agent for selecting a large number of high-resolution patches

SOLUTION BY DEEP REINFORCEMENT LEARNING



Markov Decision Process Diagram



n $=2^{16}$

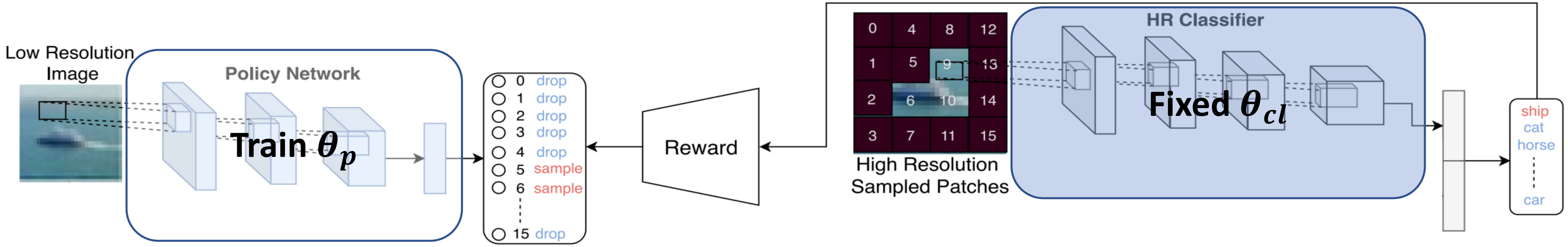
N :number of class

$$R(\boldsymbol{a_1}, \boldsymbol{a_2}, y) = \begin{cases} 1 - \left(\frac{|\boldsymbol{a_1}|}{P}\right)^2 & if\ y = \hat{y}\ (\boldsymbol{a_2}) \\ -\sigma & Otherwise \end{cases}$$

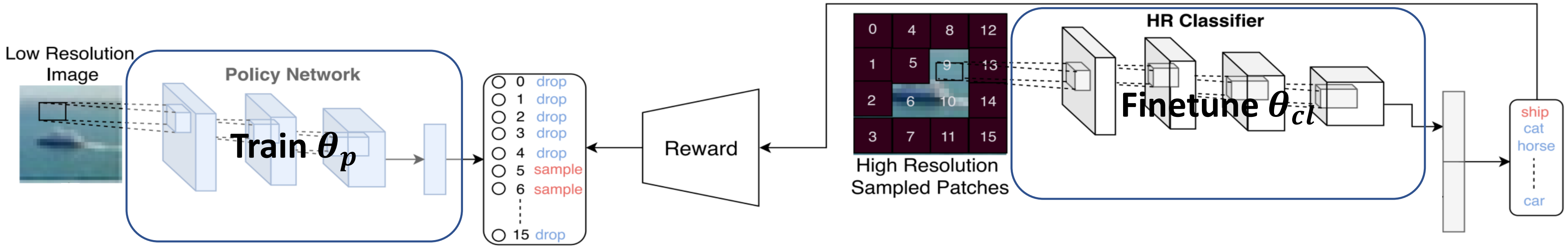
SOLUTION BY DEEP REINFORCEMENT LEARNING

TRAINING PROTOCOL: **POLICY GRADIENT**

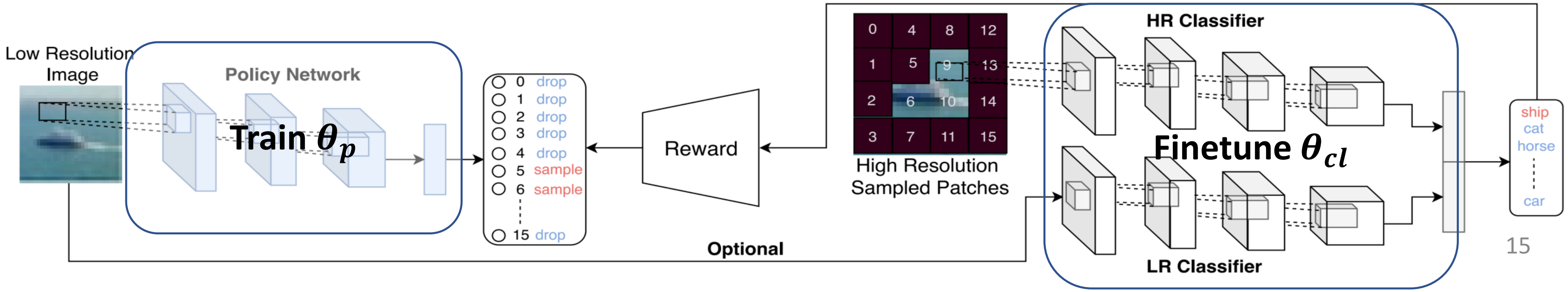
Pretraining



Finetuning-1



Finetuning-2



PERFORMANCE EVALUATION

- Use only **40% of full HR images** without any significant loss of accuracy in fMoW dataset➔ can save 100,000 dollars.

	Acc. (%) (Pt)	S	Acc. (%) (Ft-1)	S	Acc. (%) (Ft-2)	S
LR-CNN	61.4	0	61.4	0	61.4	0
SRGAN [19]	62.3	0	62.3	0	62.3	0
KD [37]	63.1	0	63.1	0	63.1	0
PCN [45]	63.5	0	63.5	0	63.5	0
HR-CNN	67.3	16	67.3	16	67.3	16
Fixed-H	47.7	7	63.3	6	64.9	6
Fixed-V	48.3	7	63.2	6	64.7	6
Stochastic	29.1	7	57.1	6	63.6	6
STN [31]	46.5	7	61.8	6	64.8	6
PatchDrop	**53.4**	**7**	**67.1**	**5.9**	**68.3**	**5.2**

Table 1: The performance of the proposed *PatchDrop* and baseline models on the fMoW dataset

PERFORMANCE EVALUATION

- Samples about **50% of HR images** on average with a minimal loss in the accuracy on ImageNet dataset.

	ImageNet			
	Acc. (%) (Pt)	Acc. (%) (Ft-1)	Acc. (%) (Ft-2)	S (Pt,Ft-1,Ft-2)
LR-CNN	58.1	58.1	58.1	0,0,0
SRGAN [19]	63.1	63.1	63.1	0,0,0
KD [37]	62.4	62.4	62.4	0,0,0
PCN [37]	63.9	63.9	63.9	0,0,0
HR-CNN	76.5	76.5	76.5	16,16,16
Fixed-H	48.8	68.6	70.4	10,9,8
Fixed-V	48.4	68.4	70.8	10,9,8
Stochastic	38.6	66.2	68.4	10,9,8
STN [31]	58.6	69.4	71.4	10,9,8
PatchDrop	**60.2**	**74.9**	**76.0**	**10.1,9.1,7.9**

Table 2: The results on ImageNet datasets.

PERFORMANCE EVALUATION

- Increase run-time performance in BagNet by **2 times**.

	Acc. (%) (Pt)	S	Acc. (%) (Ft-1)	S	Run-time. (%) (ms)
BagNet (No Patch Drop) [1]	85.6	16	85.6	16	192
CNN (No Patch Drop)	92.3	16	92.3	16	77
Fixed-H	67.7	10	86.3	9	98
Fixed-V	68.3	10	86.2	9	98
Stochastic	49.1	10	83.1	9	98
STN [19]	67.5	10	86.8	9	112
BagNet (PatchDrop)	**77.4**	**9.5**	**92.7**	**8.5**	98

Table : The performance of the PatchDrop and other models on improving BagNet on CIFAR10dataset

PERFORMANCE EVALUATION


Thank you
for listening.