

# Video Summarization

## with Flexible Multi-Agent Reinforcement Learning

Allen (Haomiao Tang)  
[tanghm183@gmail.com](mailto:tanghm183@gmail.com)

Bob (Bo Hu)  
[1294730262@qq.com](mailto:1294730262@qq.com)

Derek (Tianhao Dai)  
[thdai2000@163.com](mailto:thdai2000@163.com)

Farnante (Facheng Yu)  
[yufacheng@whu.edu.cn](mailto:yufacheng@whu.edu.cn)

---

Content

Presentation

- 1
- 2
- 3
- 4
- 5

## Background

Why does the topic appeal to us.

## Relevant Works

Important papers that inspired us.

## Our Practices & Progress

Our own implementation and result analysis.

## Our Ideas & Future Plans

More advanced optimization methods, and our plans to carry them out.

## Credits

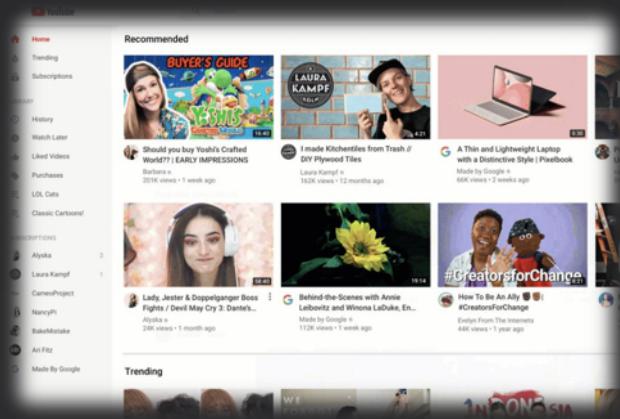
Team members & contributions.



# Background

Why does the topic appeal to us.

## Video Summarization is very helpful



YouTube Front Page

Quick glance at videos



Ball Games

Mark highlight moments

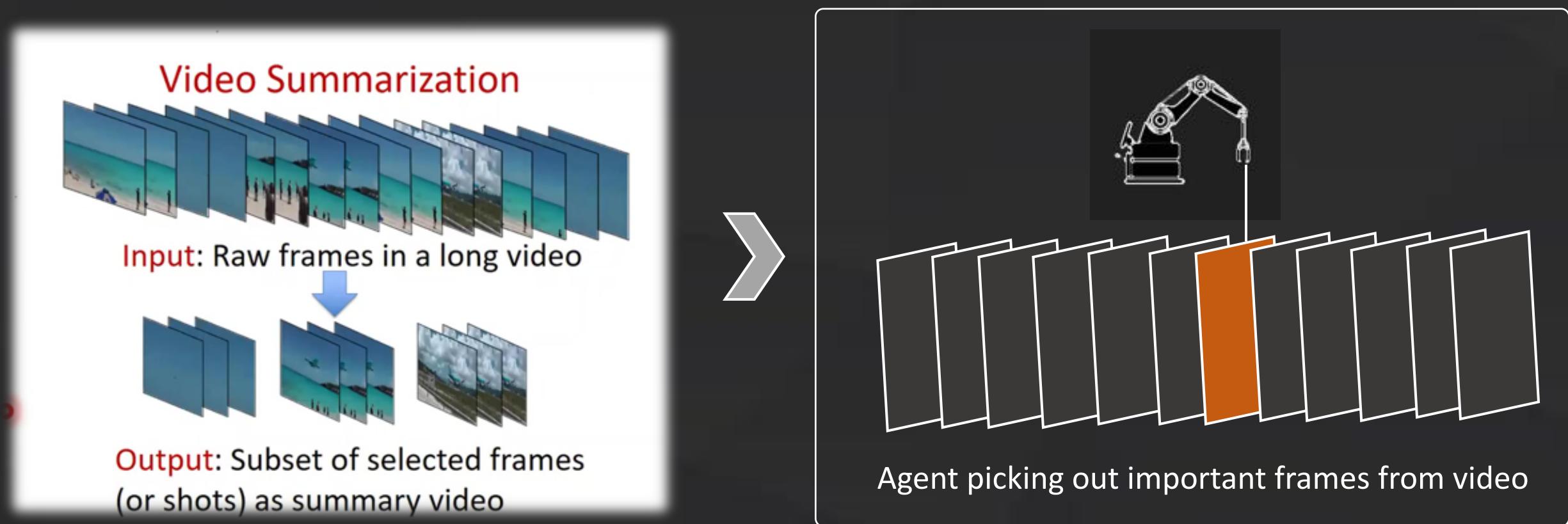


Cameras in the Wild

Track animal occurrences

# Reinforcement Learning works well on Video Summarization

Reinforcement Learning represents the process of video summarization in a smooth way.





# 2

## Relevant Works

Important papers that inspired us.

# Important Papers: DSN (2018)

✓ Single Agent Model

➤ Representativeness:

Each chosen clip should be **as similar as possible** to all of its neighboring clips in the video (as we want to choose clips that can well represent the whole area).

➤ Diversity:

The chosen clips should be **as different as possible** with other chosen clips (as we want to avoid redundancy).

**Note that diversity & representativeness are rewards of different dimension. Representative reward is calculated between the chosen clip and other common clips in the video, while diversity reward is calculated between different chosen clips.**

## Deep Reinforcement Learning for Unsupervised Video Summarization with Diversity-Representativeness Reward

Kalyang Zhou,<sup>1,2</sup> Yu Qiao<sup>1\*</sup>, Tao Xiang<sup>2</sup>

<sup>1</sup> Guangdong Key Lab of Computer Vision and Virtual Reality,

Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China

<sup>2</sup> Queen Mary University of London, UK

k.zhou@qmul.ac.uk, yu.qiao@siat.ac.cn, t.xiang@qmul.ac.uk

### Abstract

Video summarization aims to facilitate large-scale video browsing by producing short, concise summaries that are diverse and representative of original videos. In this paper, we formulate video summarization as a sequential decision-making process and develop a deep summarization network (DSN) to summarize videos. DSN predicts for each video frame a probability, which indicates how likely a frame is selected, and then takes actions based on the probability distributions to select frames, forming video summaries. To train our DSN, we propose an end-to-end, reinforcement learning-based framework, where we design a novel reward function that jointly accounts for diversity and representativeness of generated summaries and does not rely on labels or user interactions at all. During training, the reward function judges how diverse and representative the generated summaries are, while DSN strives for earning higher rewards by learning to produce more diverse and more representative summaries. Since labels are not required, our method can be fully unsupervised. Extensive experiments on two benchmark datasets show that our unsupervised method not only outperforms other state-of-the-art unsupervised methods, but also is comparable to or even superior than most of published supervised approaches.

### Introduction

Driven by the exponential growth in the amount of online videos in recent years, research in video summarization has gained increasing attention, leading to various methods proposed to facilitate large-scale video browsing (Gygli et al. 2014; Gygli, Geiger, and Van Gool 2015; Zhang et al. 2016a; Song et al. 2015; Panda and Roy-Chowdhury 2017; Mahasseni, Lam, and Todorovic 2017; Potapov et al. 2014).

Recently, recurrent neural network (RNN), especially with the long short-term memory (LSTM) cell (Hochreiter and Schmidhuber 1997), has been exploited to model the sequential patterns in video frames, as well as to tackle the end-to-end training problem. Zhang et al. (Zhang et al. 2016b) proposed a deep architecture that combines a bidirectional LSTM network with a Determinantal Point Process (DPP) module that increases diversity in summaries, referring to as DPP-LSTM. They trained DPP-LSTM with super-

\*Corresponding author.

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

vised learning, using both video-level summaries and frame-level importance scores. At test time, DPP-LSTM predicts importance scores and outputs feature vectors simultaneously, which are together used to construct a DPP matrix. Due to the DPP modeling, DPP-LSTM needs to be trained in a two-stage manner.

Although DPP-LSTM (Zhang et al. 2016b) has shown state-of-the-art performances on several benchmarks, we argue that supervised learning cannot fully explore the potential of deep networks for video summarization because there does not exist a single ground truth summary for a video. This is grounded by the fact that humans have subjective opinions on which parts of a video should be selected as the summary. Therefore, devising more effective summarization methods that rely less on labels is still in demand.

Mahasseni et al. (Mahasseni, Lam, and Todorovic 2017) developed an adversarial learning framework to train DPP-LSTM. During the learning process, DPP-LSTM selects keyframes and a discriminator network is used to judge whether a synthetic video constructed by the keyframes is real or not, in order to enforce DPP-LSTM to select more representative frames. Although their framework is unsupervised, the adversarial nature makes the training unstable, which may result in model collapse. In terms of increasing diversity, DPP-LSTM cannot benefit maximally from the DPP module without the help of labels. Since a RNN-based encoder-decoder network following DPP-LSTM for video reconstruction requires pretraining, their framework requires multiple training stages, which is not efficient in practice.

In this paper, we formulate video summarization as a sequential decision-making process and develop a deep summarization network (DSN) to summarize videos. DSN has an encoder-decoder architecture, where the encoder is a convolutional neural network (CNN) that performs feature extraction on video frames and the decoder is a bidirectional LSTM network that produces probabilities based on which actions are sampled to select frames. To train our DSN, we propose an end-to-end, reinforcement learning-based framework with a diversity-representativeness (DR) reward function that jointly accounts for diversity and representativeness of generated summaries, and does not rely on labels or user interactions at all.

The DR reward function is inspired by the general criteria of what properties a high-quality video summary should

arXiv:1801.00054v3 [cs.CV] 13 Feb 2018

# Important Papers: CoSNet (2020)

- ✓ **Multi-Agent Model**
- Formulates the process of video summarization as **comparing and selecting frames**: CoSNet.
- Agents are represented by **LSTM (Long-Short Term Memory) networks**.
- Also uses a kind of **diversity-representativeness reward**.

## Compare and Select: Video Summarization with Multi-Agent Reinforcement Learning

Tianyu Liu  
Peking University  
Beijing, China  
lutyb@gmail.com

### ABSTRACT

Video summarization aims at generating concise video summaries from the lengthy videos, to achieve better user watching experience. Due to the subjectivity, purely supervised methods for video summarization may bring the inherent errors from the annotations. To solve the subjectivity problem, we study the general user summarization process. General users usually watch the whole video, compare interesting clips and select some clips to form a final summary. Inspired by the general user behaviours, we formulate the summarization process as multiple sequential decision-making processes, and propose Comparison-Selection Network (CoSNet) based on multi-agent reinforcement learning. Each agent focuses on a video clip and constantly changes its focus during the iterations, and the final focus clips of all agents form the summary. The comparison network provides the agent with the visual feature from clips and the chronological feature from the past round, while the selection network of the agent makes decisions on the change of its focus clip. The specially designed unsupervised reward and supervised reward together contribute to the policy advancement, each containing local and global paths. Extensive experiments on two benchmark datasets show that CoSNet outperforms state-of-the-art unsupervised methods with the unsupervised reward and surpasses most supervised methods with the complete reward.

### CCS CONCEPTS

• Computing methodologies → Video summarization; Multi-agent reinforcement learning.

### KEYWORDS

video summarization, multi-agent reinforcement learning

### 1 INTRODUCTION

Gigantic amounts of videos are produced by mobile phones, wearable devices and surveillance cameras. The lengthy raw videos with sparse information make it difficult for viewing, browsing and retrieving, resulting in the decline of user experience. In the meantime, video summaries can shorten the viewing time, provide dense information and save the storage space. To alleviate the problems of raw videos, we need video summarization to transform the lengthy raw videos into concise video summaries.

Video summarization is a relatively subjective task. In the process of creating datasets, different annotators may produce largely different annotations for the same video. Therefore, the annotation of video summarization datasets requires more annotators than other tasks, to ensure the maximum annotation accuracy. In the analysis of the widely used benchmark datasets, Sunble [39] and

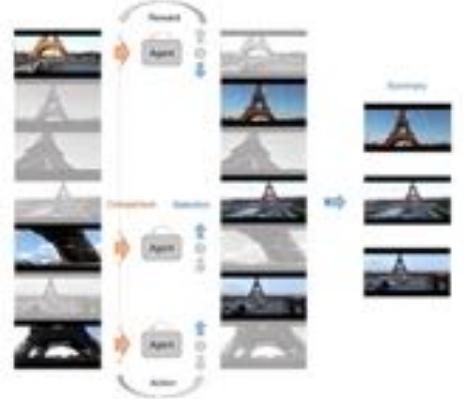


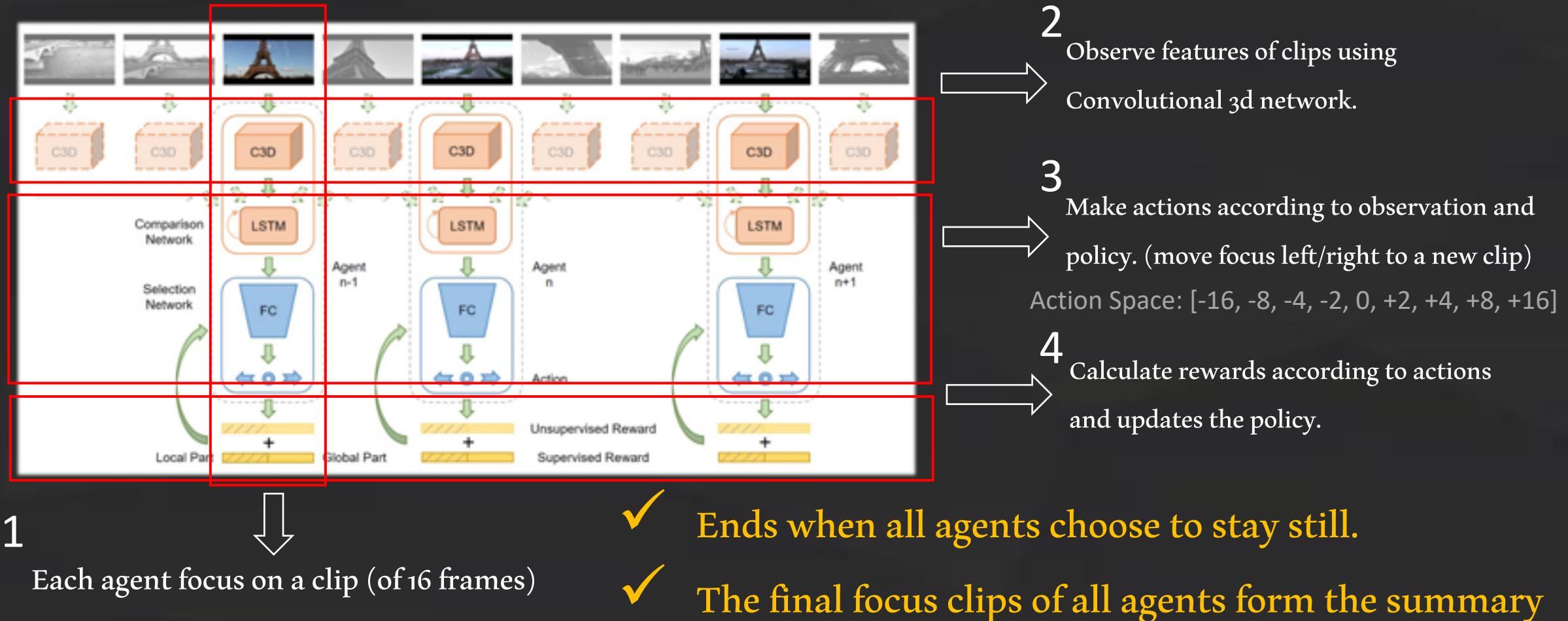
Figure 3: Agents compare the clips by taking the visual features and chronological features as input, then make decisions on which clips to select in the next round.

TVSum [46], the former dataset has 15 to 18 annotations for each video, while the latter dataset has 20 annotations for each video. However, the annotations may still suffer from subjectivity due to the irreconcilable differences among different annotations.

In order to solve the subjectivity problem, we specially conduct a survey on the methods that can relieve the problem. Unsupervised methods [12, 15, 27, 39, 62] can work without annotations, but may lose some effective information in annotations. Among the various methods proposed in the literature, some methods originate from the inherent characteristics of the videos, and some other methods are from the inspiration of video user behaviours. In [46], Song et al. proposed a method that selects frames which are most relevant to the video titles to form the summaries. Like video titles, Panda et al. [38] used video-level category annotations. The category annotations contain less information than frame-level annotations, but are actually more "accurate", because the opinions about the categories are consistent among almost all annotators. In [38], Rechan and Wang used the idea that unpaired videos and summaries can reduce the subjectivity caused by the interdependence of paired ones. In [34], Xiong et al. applied the thought of "less is more" to the video summarization task, which indicates that shorter videos are more informative than longer ones. The above-mentioned methods

# CoSNet: How it works

Agents move from clip to clip and the final position of agents form the summary.



# Our Practices & Progress

# Survey

## Understanding CosNet



@Allen



# Our Practices & Progress

Our own implementation and result analysis.

# Code Implementation

# Coding the Model

- ✓ 500+ Lines of Code
  - ✓ Gradient Decent Algorithm: Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning.  
Machine learning 8, 3-4 (1992), 229–256.

@Farnante

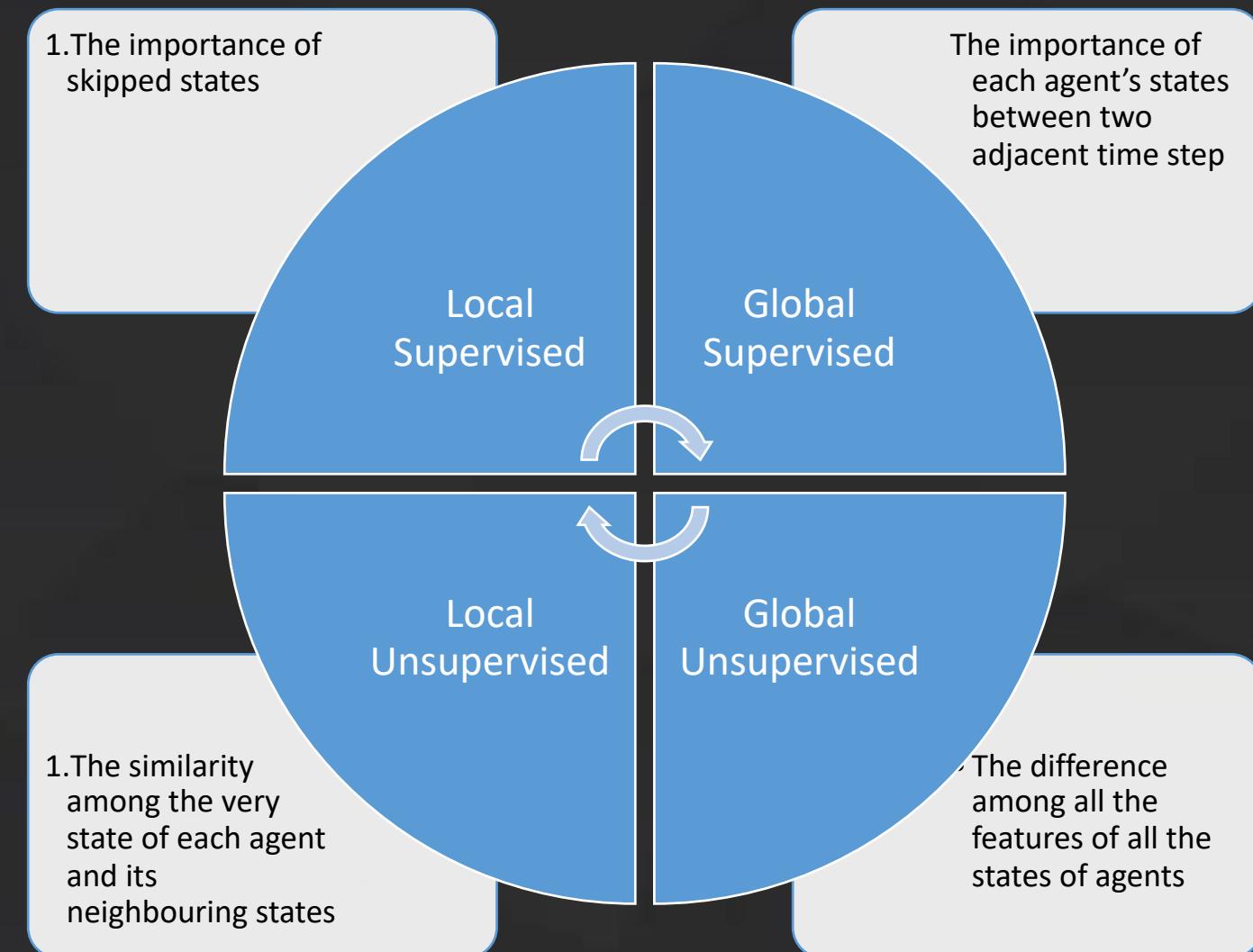
# The Design of 4 kinds of Rewards

## Local Reward:

Allow agents to find out whether there are more important states in the neighbor.

## Global Reward:

Assess the global importance of all the current states choose by agents



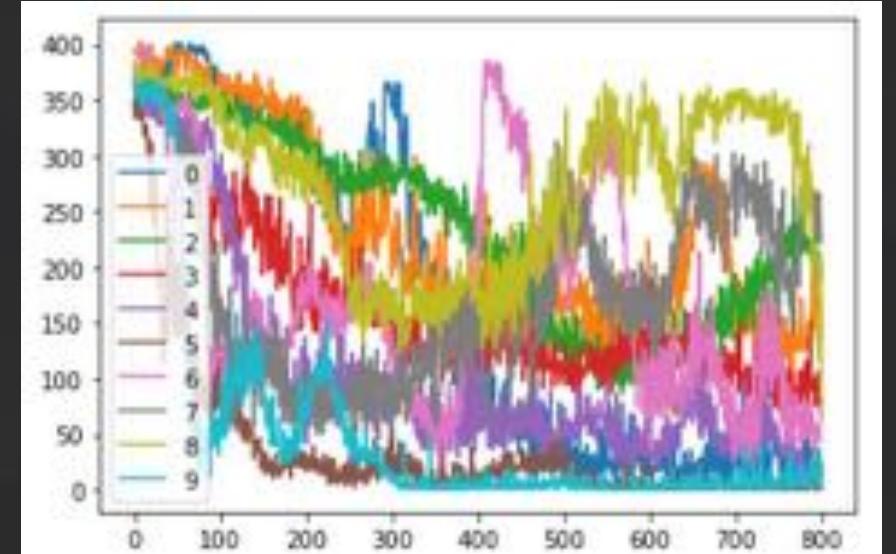
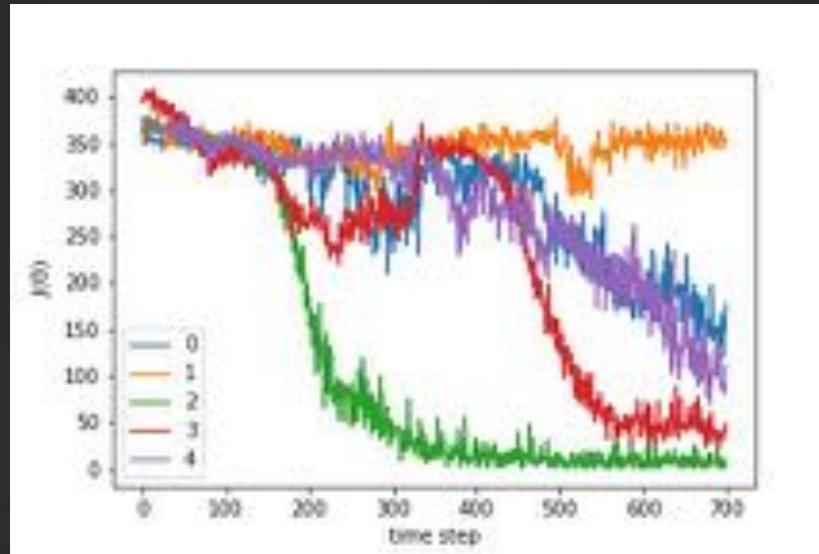
# Our Agents Share the Same Goal and Policy

---

- Our agents are try to find out all the important states and move towards to it.
- Our agents share the same policy so it 's possible to compute the policy at the beginning of each update epoch.
- The sharing allow our model running fast.
- We set in each update epoch:
  - 10 episodes, each with a length of 20 steps.
  - About 6-10s to run 10 epochs

# Our Model is learning

- We set in each update epoch:
  - At least 500 update epochs
  - 10 episodes in each epoch
  - 20 steps in every episode
- About **6-10s** to run **10 epochs**
- For most videos, the policy **began to converge** after 200 - 400 update epochs

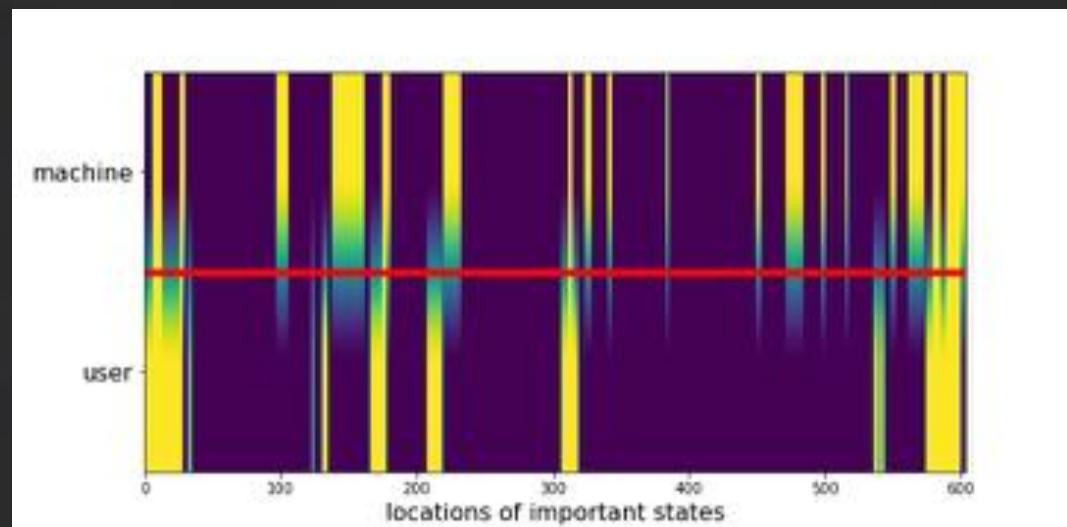
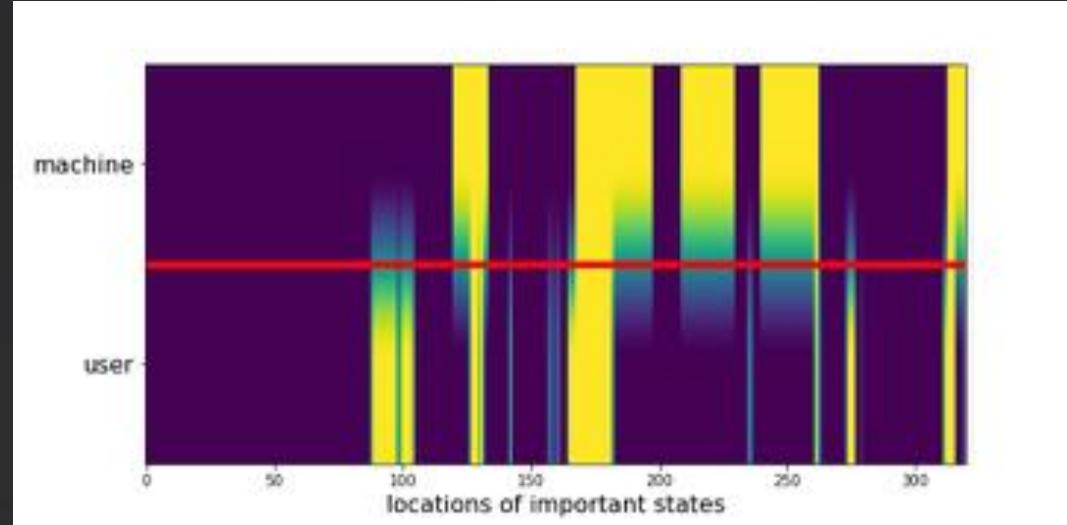


# The Goal States Our Model learned

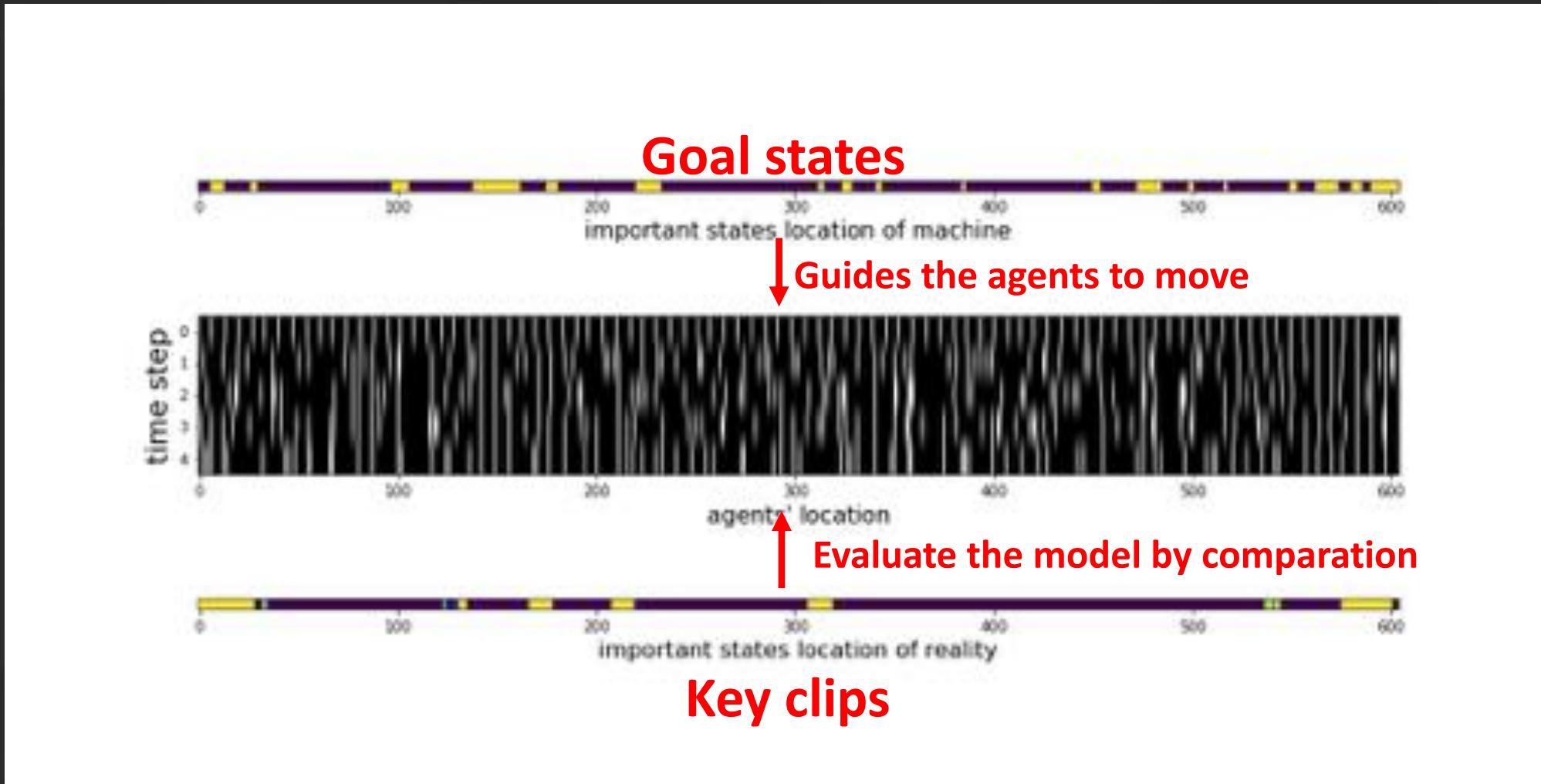
A **goal state** means a state with a high no-move action possibility.

In reality, a goal state correspond to a **key clip** that summarizes certain part of a video.

RHS is two graphs in two videos, each of which describes **the contrast** of machine goal states and user's key clips.

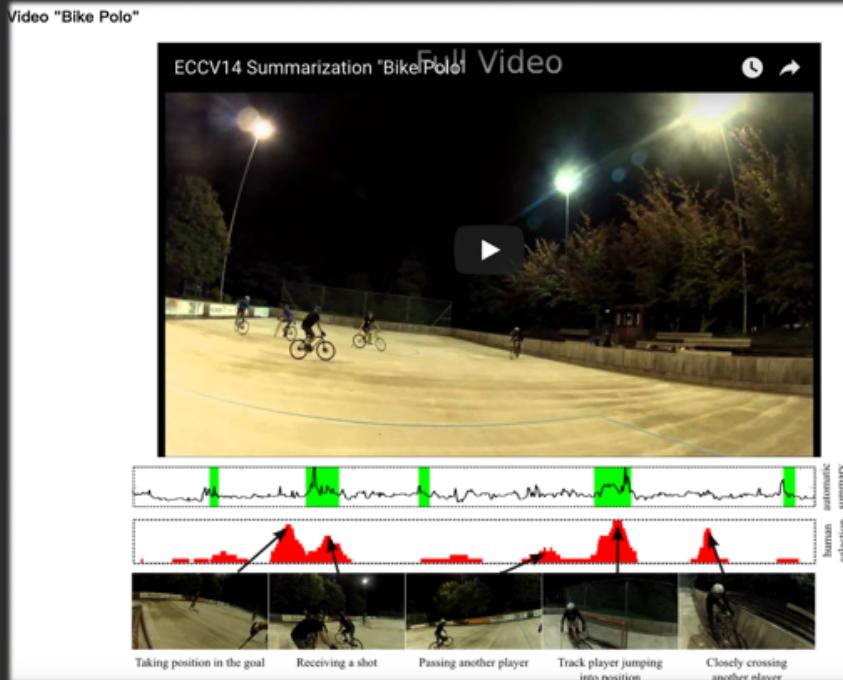


# Agents Try to Move Following the Trained Policy



# Evaluation

## Popular benchmarks



**25 VIDEOS**

Each video 1~6 minutes

Various topics such as holidays and sports

**15 – 18 ANNOTATORS PER VIDEO**

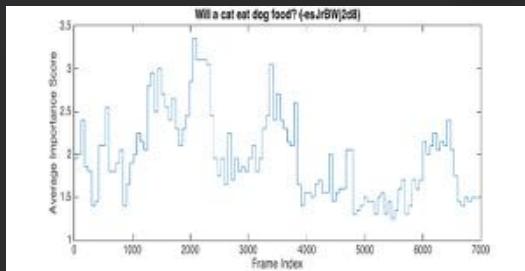
390 in total

**SumMe**

<https://gyglim.github.io/me/vsum/index.html>

# Evaluation

# Popular benchmarks



# 50 VIDEOS

Each video 2~10 minutes

Collected from YouTube and vary in topic

# 20 ANNOTATORS PER VIDEO

## Frame-level importance score

# TVSum

# Evaluation

## 1. Divide video into clips



16 frames  
per clip



clip 1



clip 2

.....



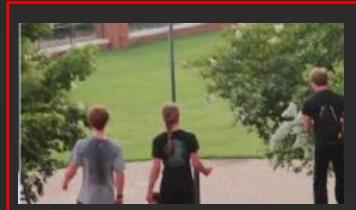
clip n

# Evaluation

## 2. Select clips through trained model



clip 1



clip 2

.....



clip n

✓

Save as  
binary  
vector

[0, 0, ..., 0, **1, 1, ..., 1**, ..., 0, 0, ..., 0]

# Evaluation

## 3. Generate human-friendly summary



✗



Video clips with fixed length  
(in our case is 16 frames)



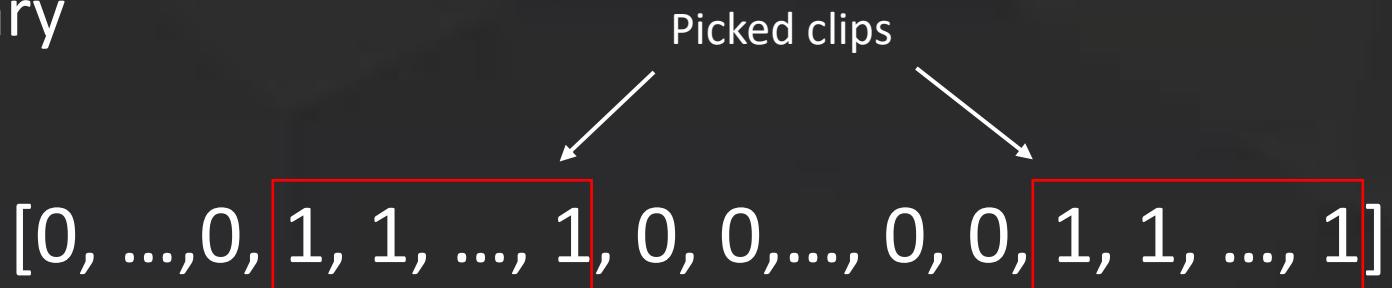
Video clips with flexible length.  
Each represents a unique scenario.

# Evaluation

## 3. Generate human-friendly summary

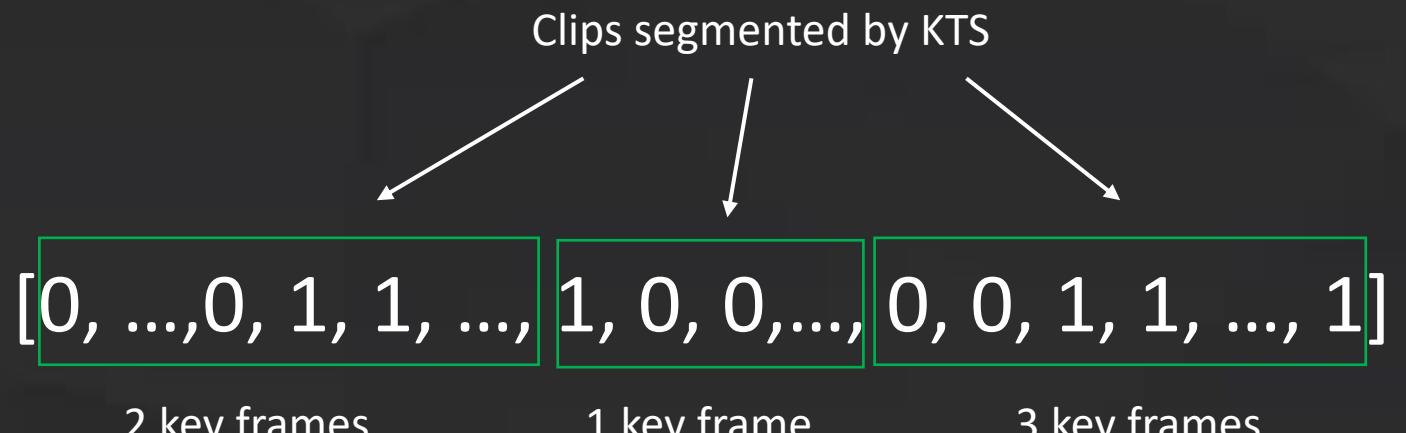


(Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. 2014. Category-specific video summarization. In *European conference on computer vision*. Springer, 540–555.)



# Evaluation

## 3. Generate human-friendly summary



Let our summary be less than  
15% in duration of original video

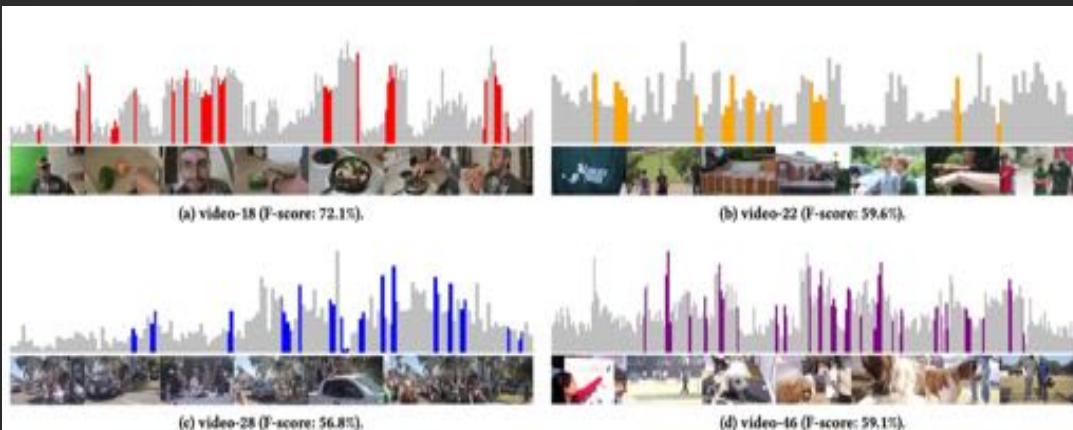


Knapsack problem/ranking problem

Final summary = [0, ..., 0, 0, 0, ...], [1, 1, 1, ...], [0, 0, 0, 0, ...]

# Evaluation

## 4. Calculate the F-score



### What is F-Score:

A percentage score that depicts the similarity between the ground truth summaries and the generated summaries.

Calculated by precision P and recall R. The bigger the F-Score, the better is the quality of summarization.

$$F = \frac{2PR}{P + R} \times 100\%$$

### How to calculate it in practice:

$A$  = vector of machine summary

$B$  = vector of user summary

$$P = \frac{\text{overlapped duration of } A \text{ and } B}{\text{duration of } A}$$

[0, 0, 1, 1, ..., 0, 1, 0]  
\* [0, 1, 1, 1, ..., 1, 1, 0]

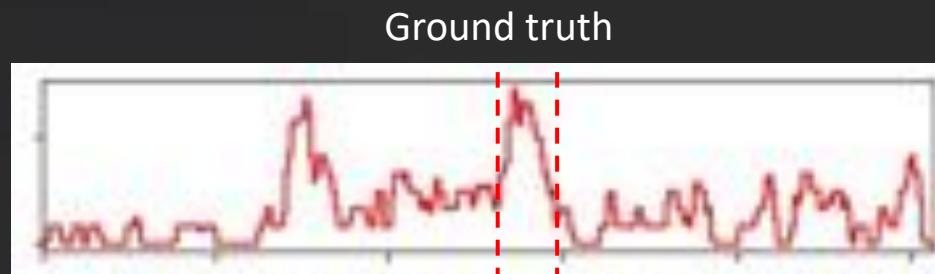
$$R = \frac{\text{overlapped duration of } A \text{ and } B}{\text{duration of } B}$$

$$F = \frac{2PR}{P + R} \times 100\%$$

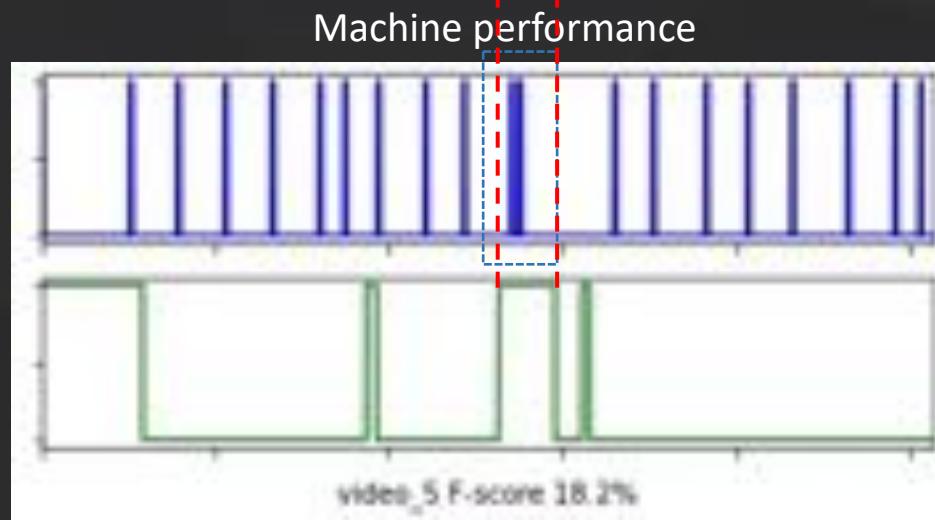
# Evaluation

## 5. Result visualization

Ground truth  
importance  
score of clips



Clips agents  
finally choose



Final summary

$$\text{softmax } [0, 1, 2, 3, \dots, 2, 1, 0] = \begin{cases} \text{User 1: } [0, 0, 1, 1, \dots, 1, 0, 0] \\ + \\ \text{User 2: } [0, 1, 1, 1, \dots, 0, 0, 0] \\ + \\ \vdots \\ \text{User n: } [0, 0, 0, 1, \dots, 1, 1, 0] \end{cases}$$

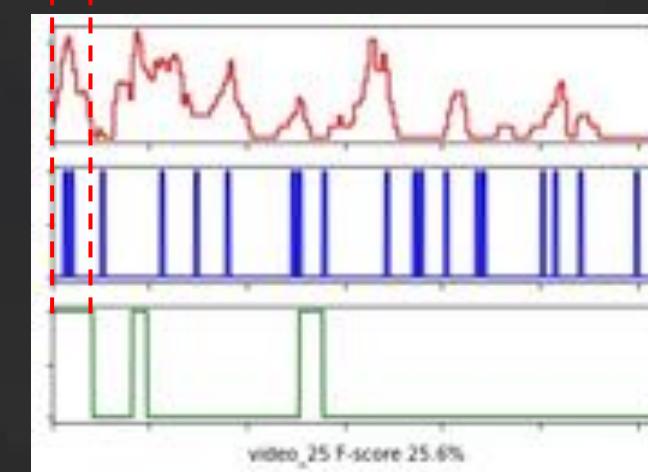
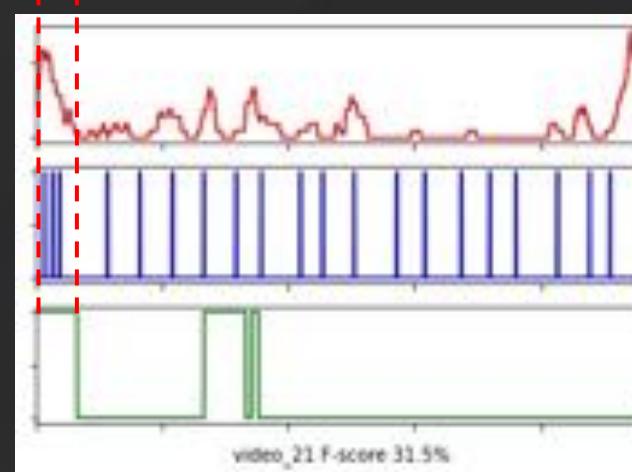
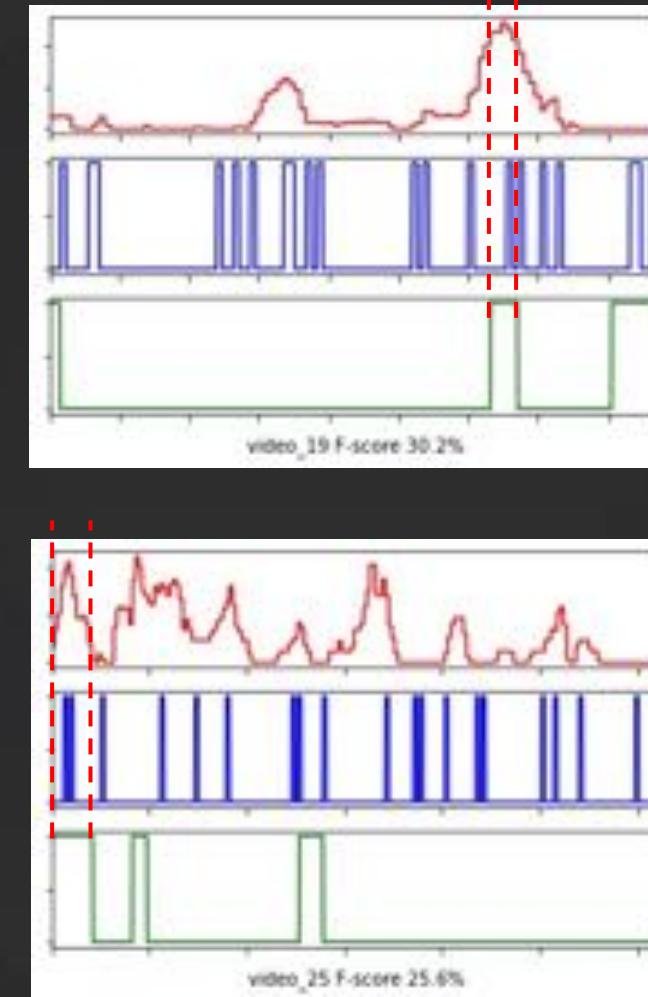
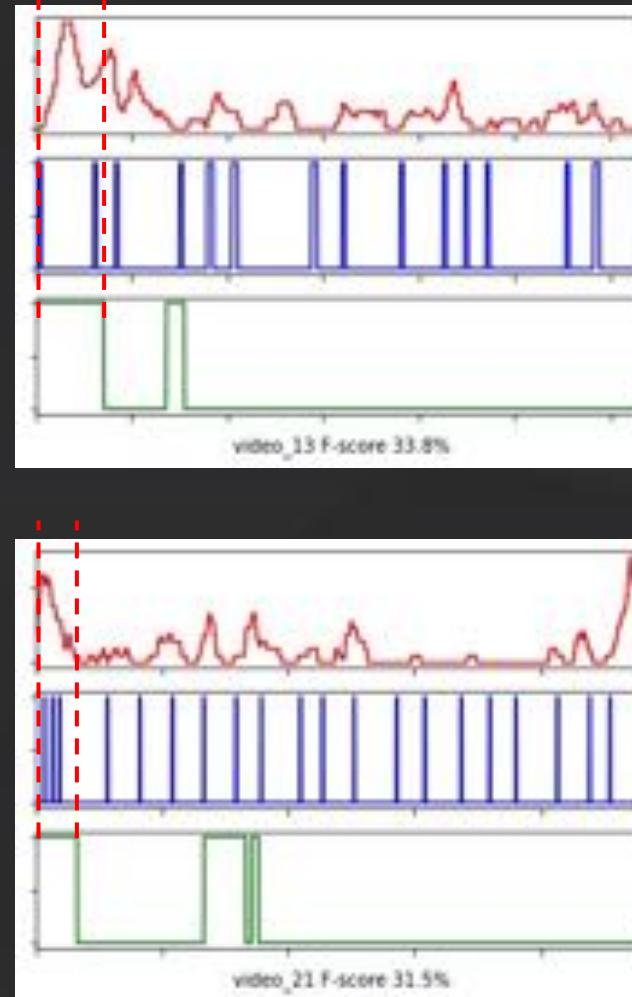
The more dense blue bars are, the more important model thinks the clips are.

The most important segments model chooses.  
The closer they are to high ground truth importance parts,  
the higher F-score is.

# Evaluation

## 6. Good results

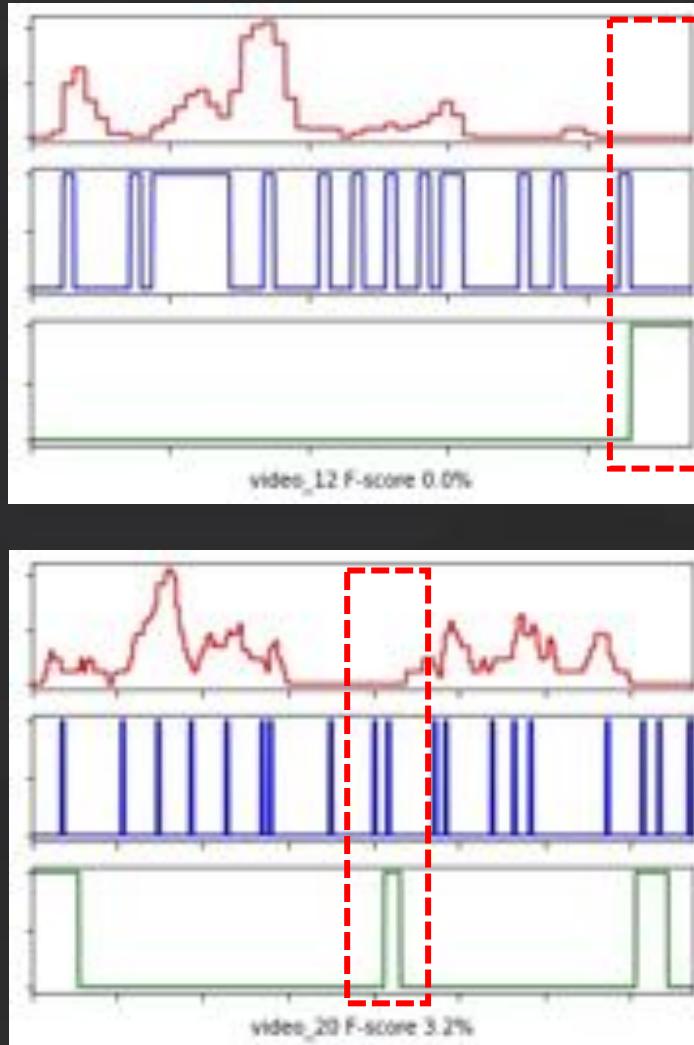
No.	Video	F-score
1	video_1	12.8%
2	video_10	21.7%
3	video_11	5.4%
4	video_12	0.0%
5	video_13	33.8%
6	video_14	21.1%
7	video_15	20.1%
8	video_16	18.0%
9	video_17	19.7%
10	video_18	18.7%
11	video_19	30.2%
12	video_2	9.4%
13	video_20	3.2%
14	video_21	31.5%
15	video_22	8.1%
16	video_23	7.2%
17	video_24	9.9%
18	video_25	25.6%
19	video_3	15.8%
20	video_4	4.9%
21	video_5	6.0%
22	video_6	18.8%
23	video_7	5.9%
24	video_8	11.1%
25	video_9	20.6%
<hr/>		
Average F-score 15.2%		



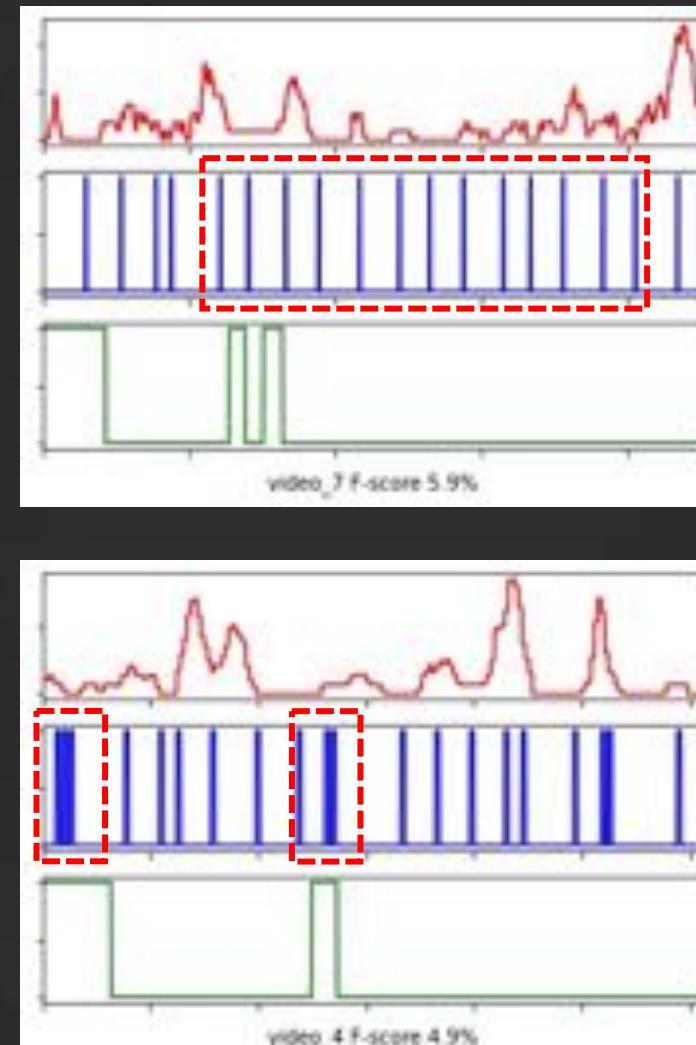
# Evaluation

## 6. Poor results

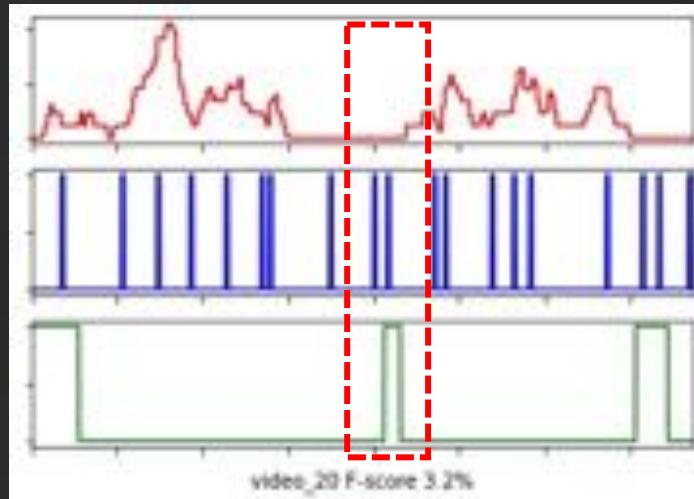
No.	Video	F-score
1	video_1	12.8%
2	video_10	21.7%
3	video_11	5.4%
4	video_12	0.0%
5	video_13	33.8%
6	video_14	21.1%
7	video_15	20.1%
8	video_16	18.0%
9	video_17	19.7%
10	video_18	18.7%
11	video_19	30.2%
12	video_2	9.4%
13	video_20	3.2%
14	video_21	31.5%
15	video_22	8.1%
16	video_23	7.2%
17	video_24	9.9%
18	video_25	25.6%
19	video_3	15.8%
20	video_4	4.9%
21	video_5	6.0%
22	video_6	18.8%
23	video_7	5.9%
24	video_8	11.1%
25	video_9	20.6%
<hr/>		
Average F-score 15.2%		



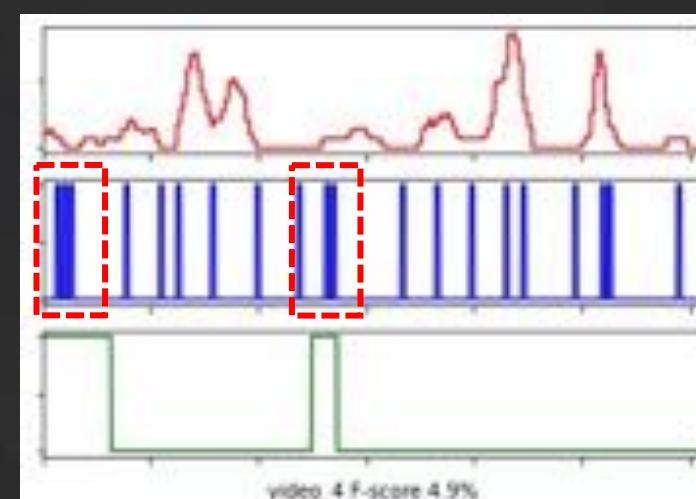
Wrong  
Segment  
picking



Evenly  
distributed



Wrong  
Segment  
picking



Mistakenly  
populated

# Evaluation

## 6. Results analysis

### Why good?

1. Chosen segments are close to most important parts.
2. Agents gather correctly.

### Why poor?

1. KTS-generated segments are too big, so they easily reach the 15% limitation.
2. Agents did not learn the distribution very well.

### Solution:

1. Use other methods to do video segmentation.
2. Redesign the rules to pick segments.
3. Train the model with more diverse datasets.



## Our Ideas & Future Plans

More advanced optimization methods.  
And our plans to carry them out.

# Survey

## Finding Methods & Modelling

@Bob

arXiv:2001.05864v2 [cs.CV] 29 Feb 2020

**Weakly Supervised Video Summarization by Hierarchical Reinforcement Learning**

Yiyan Chen, Li Tao, Xuetong Wang, Toshihiko Yamamoto  
The University of Tokyo  
Tokyo, Japan  
{chenyiyan,taoli,xt\_wang,yamamoto}@hal.t.u-tokyo.ac.jp

**ABSTRACT**  
Conventional video summarization approaches based on reinforcement learning have the problem that the reward can only be received after the whole summary is generated. Such kind of reward is sparse and it makes reinforcement learning hard to converge. Another problem is that the global reward is often too large or too small, which usually prohibits the construction of large-scale datasets. To solve these problems, we propose a weakly supervised hierarchical reinforcement learning framework, which decomposes the whole task into several subtasks and divides the total reward into several parts, which usually prohibits the construction of large-scale datasets. To solve these problems, we propose a weakly supervised hierarchical reinforcement learning framework, which decomposes the whole task into several subtasks and divides the total reward into several parts, which usually prohibits the construction of large-scale datasets. This framework consists of a manager network and a worker network. For each subtask, the manager is trained to set a subgoal only by a task-level binary label, which requires much fewer labels than conventional approaches. A video frame is divided into several subtasks, and each subtask processes on several video frames. A Manager is designed to set a subgoal for each subtask, and a Worker is guided by it to predict the importance score for each frame.

**CCS CONCEPTS**  
 • Computing methodologies → Video summarization • Theory of computation → Reinforcement learning

**KEYWORDS**  
 video summarization, hierarchical reinforcement learning, subreward

**ACM Reference Format:**  
 Yiyan Chen, Li Tao, Xuetong Wang, Toshihiko Yamamoto. 2019. Weakly Supervised Video Summarization by Hierarchical Reinforcement Learning. In *ACM Multimedia Asia (MMAsia '19)*, December 19–18, 2019, Beijing, China. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3350533.3346831>

**1 INTRODUCTION**  
 Nowadays, video data are increasing explosively on the Internet. Video processing has attracted much attention from researchers. However, considering the temporal information, a video for a long duration is still hard to process. A video can be processed as a sequence of frames. Usually, one can downsample the sequence of frames but it is still too long and it is a big problem for video tasks. Video summarization, which shortens an original video to a compact summarization offer as a short and representative sequence of frames.

Recently, deep neural networks have been applied to video summarization tasks to learn to extract features from raw video and to infer a global summary [11][22]. These supervised methods require a large number of frame-level labels for each video. Consequently, collecting a large scale of annotated videos costs a lot of time and money. Moreover, some recent work proposes unsupervised methods such as [23]. They propose an unsupervised reinforcement learning method and train the model by policy gradient with a diversity-representativeness global reward to evaluate the selected frames. This kind of global reward is too sparse to generate a good summary. This kind of global reward is too sparse to evaluate a long series of actions. Here, each action is defined as whether to select a frame or not.

Therefore, we propose a new weakly supervised video summarization framework featuring hierarchical reinforcement learning, which requires only a small number of annotations and avoids the sparse reward problem. The main idea is to divide the whole task into several subtasks. While each subtask includes several annotations, the Manager is able to divide the whole task into subtasks in order. In this way, only task-level annotations (detail in Sec. 3.3), i.e., wendy

**Weakly Supervised Video Summarization by Hierarchical Reinforcement Learning: Inspired the thought of manager-worker agent model**

arXiv:2003.11778v2 [cs.AI] 6 Jul 2020

**Too many cooks: Bayesian inference for coordinating multi-agent collaboration**

Rose E. Wang<sup>\*</sup> MIT revang@mit.edu Sarah A. Wu<sup>\*</sup> MIT sarahwu@mit.edu James A. Evans UChicago jevans@chicago.edu  
 Joshua B. Tenenbaum MIT jbt@mit.edu David C. Parkes Harvard parkes@seas.harvard.edu Max Kleiman-Weiner Harvard, MIT & Diffo maxkw@mit.edu

**Abstract**  
 Collaboration requires agents to coordinate their behavior on the fly, sometimes cooperating to solve a single task together and other times dividing it up into sub-tasks to work on in parallel. Understanding the human ability to collaborate is theoretically important, and also critical for building systems that let others act. Here, we develop Bayesian Delegation, a decentralized multi-agent learning mechanism with these abilities. Bayesian Delegation enables agents to rapidly infer the hidden intentions of others by inverse planning. We test Bayesian Delegation in a multi-agent learning task and a social planning task, both involving complex problems. On these tasks, agents with Bayesian Delegation coordinate both their high-level plans (e.g. what sub-task they should plan) and their low-level actions (e.g. avoiding getting in each other's way). In a self-play evaluation, Bayesian Delegation outperforms alternative algorithms. Bayesian Delegation is also a capable tool for distributed reinforcement learning, allowing agents to learn to coordinate in the absence of prior experience. Finally, in a behavioral experiment, we show that Bayesian Delegation makes inferences similar to human observers about the intent of others. Together, these results demonstrate the power of Bayesian Delegation for decentralized multi-agent collaboration.

**Keywords:** coordination; social learning; inverse planning; Bayesian inference; multi-agent reinforcement learning

**1 Introduction**  
 Working together enables a group of agents to achieve together what no individual could achieve on their own [37, 20]. However, collaboration is challenging as it requires agents to coordinate their behavior. In the absence of prior experience, socially intelligent agents need to infer the intentions of other agents in order to coordinate their own actions. When we are writing a scientific manuscript with collaborators or preparing a meal with friends, core questions we and ourselves ask are: how can I help out the group? What should I work on next, and with whom should I do it with? Figure 1 illustrates how flexibly coordinating a collaborative endeavor is a fundamental challenge for agents in a multi-agent system.

Central to this challenge is that agents' reasoning about what they should do in a multi-agent context depends on the future actions and intentions of others. When agents, like people, make independent decisions, these intentions are unobserved. Actions can reveal information about intentions, but

<sup>\*</sup>Indicates equal contribution.

Proprietary. Under review.

**Too many cooks: Bayesian Inference for coordinating multi-agent collaboration: Inspired the thought of agent collaboration.**

# Survey

## Understanding PettingZoo

**PettingZoo**

官方开发者文档

从 raw\_env 基础上做封装，能够提供各种各样的逻辑抽象。

# 需要的基本

```
1 from gym import spaces
2 import numpy as np
3 from pettingzoo import pettingzoo
4 from pettingzoo import pettingzoo
```

是常用的两种 Space: Discrete&Box

space固有方法

```
1 Sample(): 随机取样
2 # Discrete
3 # List define
4 players = ['player_0', 'at']
5 players = ['player_0', 'at']
6 # List iteration
7 oldlist = [1, 2, 4]
8 newlist = [k+2 for k in
9             range(len(oldlist)-1)]
10 newlist = [1, 4, 10]
```

# raw\_env

原始环境，为自己的

metadata

存放一些全局环境参数

render\_modes

name: 实际实现的进程中

# \_\_init\_\_

# gym.space

space类

space可以自定义，有

space中的方法

1. np.random
2. seed: 设置随机数种子

# 类的初始化:

```
1 # 通过重写 __init__ 方法
2     def __init__(self):
3         pass
```

# 可以用来自身类名和self，同时父类要继承object

```
1 # 重写父类的 __init__ 方法
2     def __init__(self, *args, **kwargs):
3         super().__init__(*args, **kwargs)
```

# 调用父类

```
1 class MyClass:
2     pass
```

Python 2.x: 重写入自身类名和self，同时父类要继承object

```
1 #!/usr/bin/python
2 #-- coding: utf-8 --
3
4 class Object(): # Python2.x 定义继承 object
5     def __init__(self, *args):
6         self.args = args
7         print(args)
8
9 class B(A):
10     def __init__(self, *args):
11         super(B, self).__init__(*args)
12         self.args = args
13         print(self.args)
```

Python 3.x: 直接用super()

# 疑问

most方法为什么要有 cumulative\_reward? Reward有什么区别?

action是什么类型的数组?

gymofspaces怎么使用?

@Allen

Pettingzoo 环境搭配

Monday, August 16, 2021

AEC Environment

- AECAgent Environments
  - 说，首先某个（玩家）后另一个（玩家）一些
  - AEC 具体的可选
- 论文另说明了三个类别
  - 1. For every POSG
  - 2. Every AEC Game
  - 3. Every AEC Game
- 因此pettingzoo只提供

# def observe

Returns 1

# def render

Displays

Alternate

which net

classic, and "ans"

# def state

State ret

centralized

Returns the obse

dictionary,

where each dictic

# close

Closes the other resource

that shos

# reset(self, agent\_name)

resets the environ

agent name

# def \_\_init\_\_(self, see

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1

# Methods

---

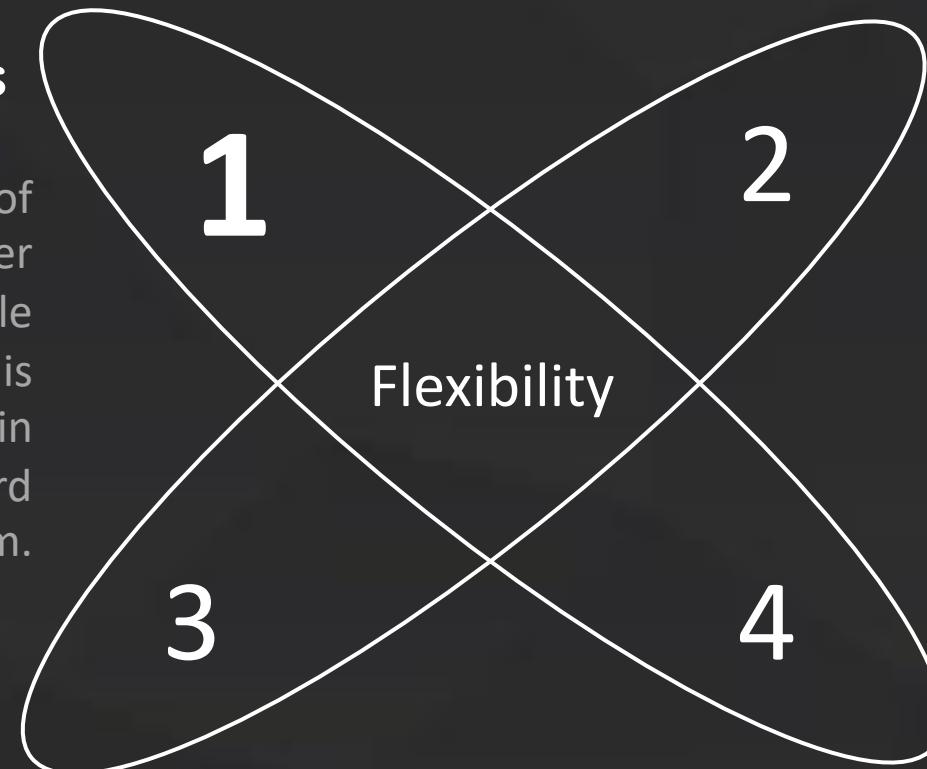
## Hierarchical Agents

Worker agents work under the range of a manager agent. Different manager agents repel with each other while worker agents attract each other. This can prevent agents from gathering in one place and cope with sparse reward problem.

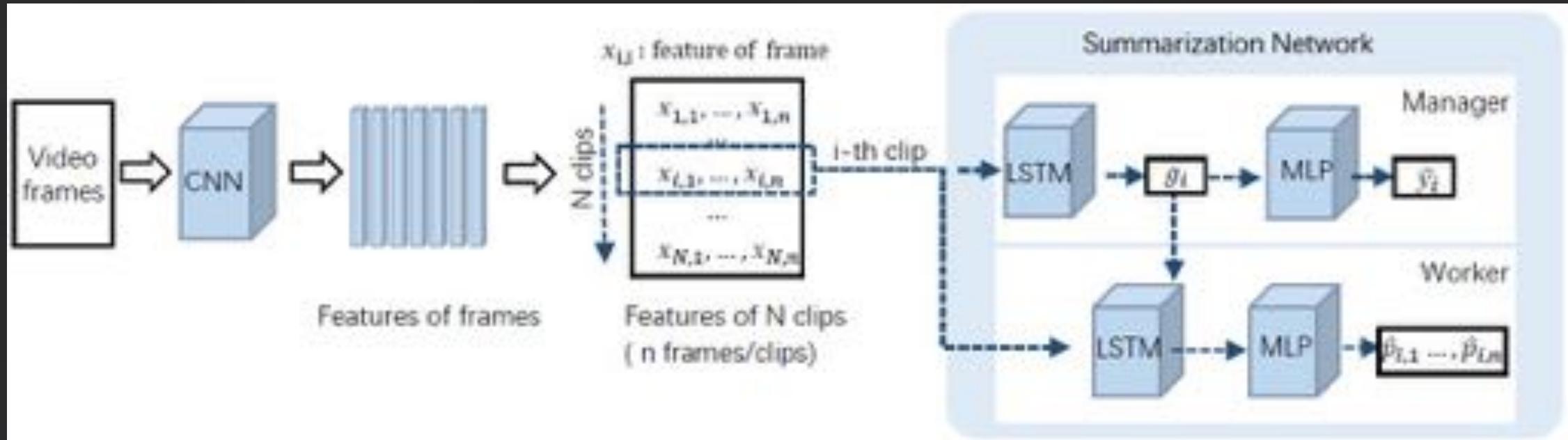
## Agent Cooperation

## Dynamic Number of Agents

## Using PettingZoo Library



# Hierarchical Agents



## Manager Network

Input features clips into manager network. Get the hidden state of Manager's LSTM which contains the global feature of video.

## Worker Network

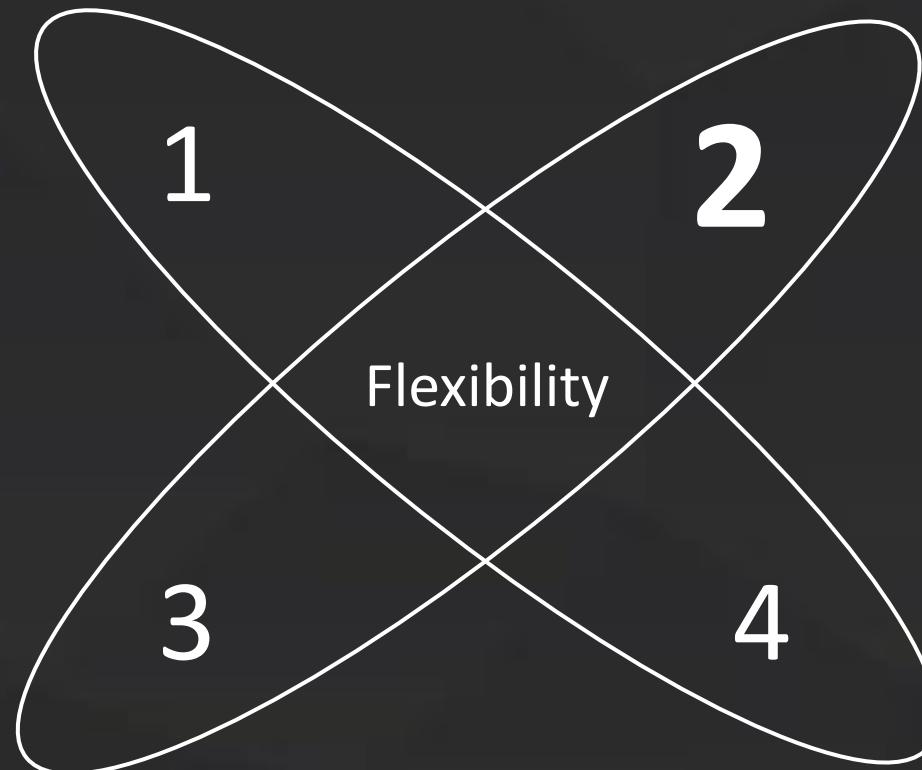
Due to getting both features from manager's LSTM and the clips, the worker agent can both focus on local feature and the global feature

# Methods

---

Hierarchical Agents

Agent Cooperation



**Dynamic Number of Agents**

Agents can be terminated if their selected clips are too sparse. Applying dynamic number of agents allow us to provide appropriate summarization according to the richness of information in a certain video.

Using PettingZoo Library

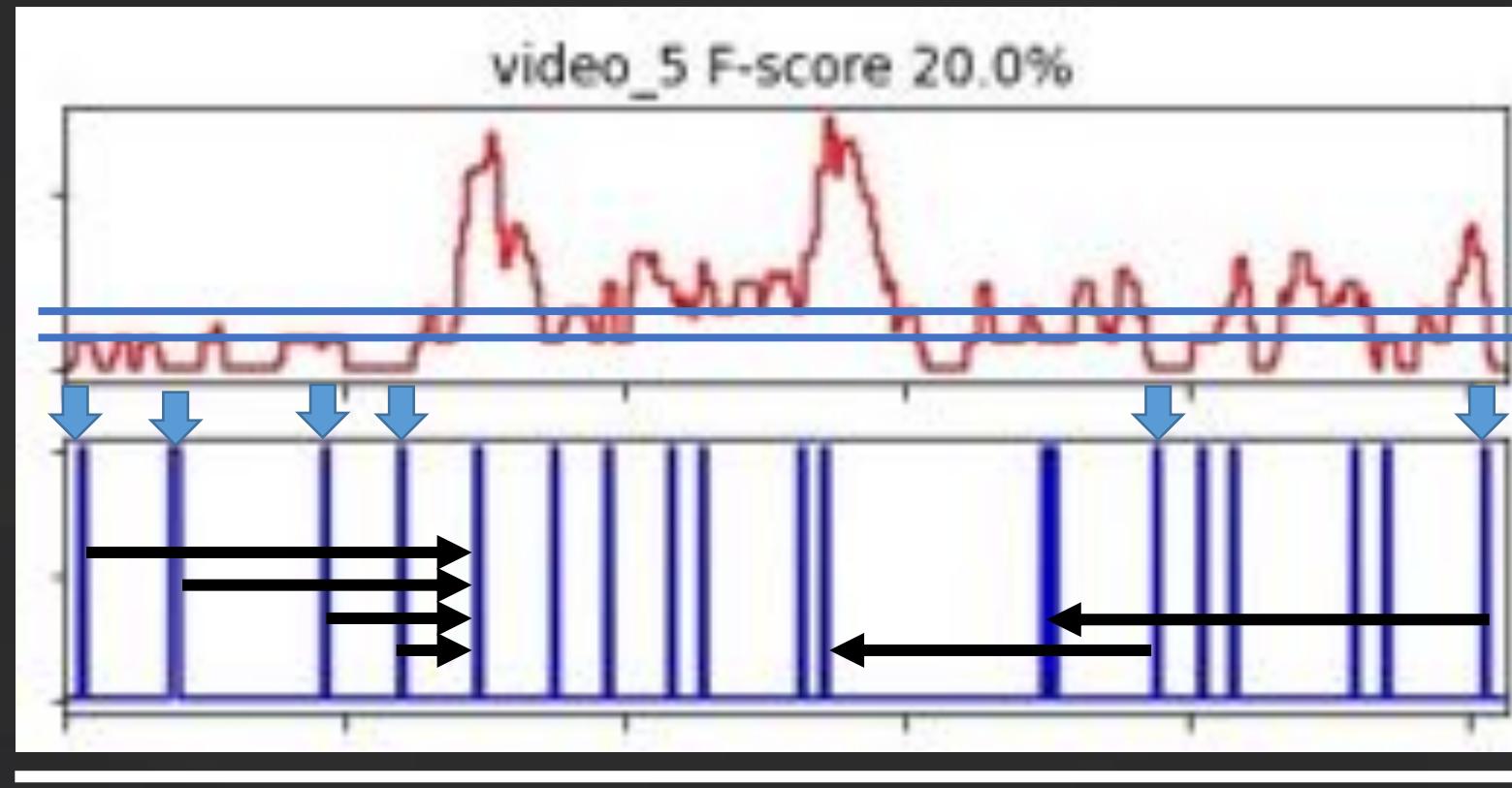
# Dynamic Number of Agents

Two types of problem

- The agents distribute sparsely
- The agents overlaps too much

Sparse

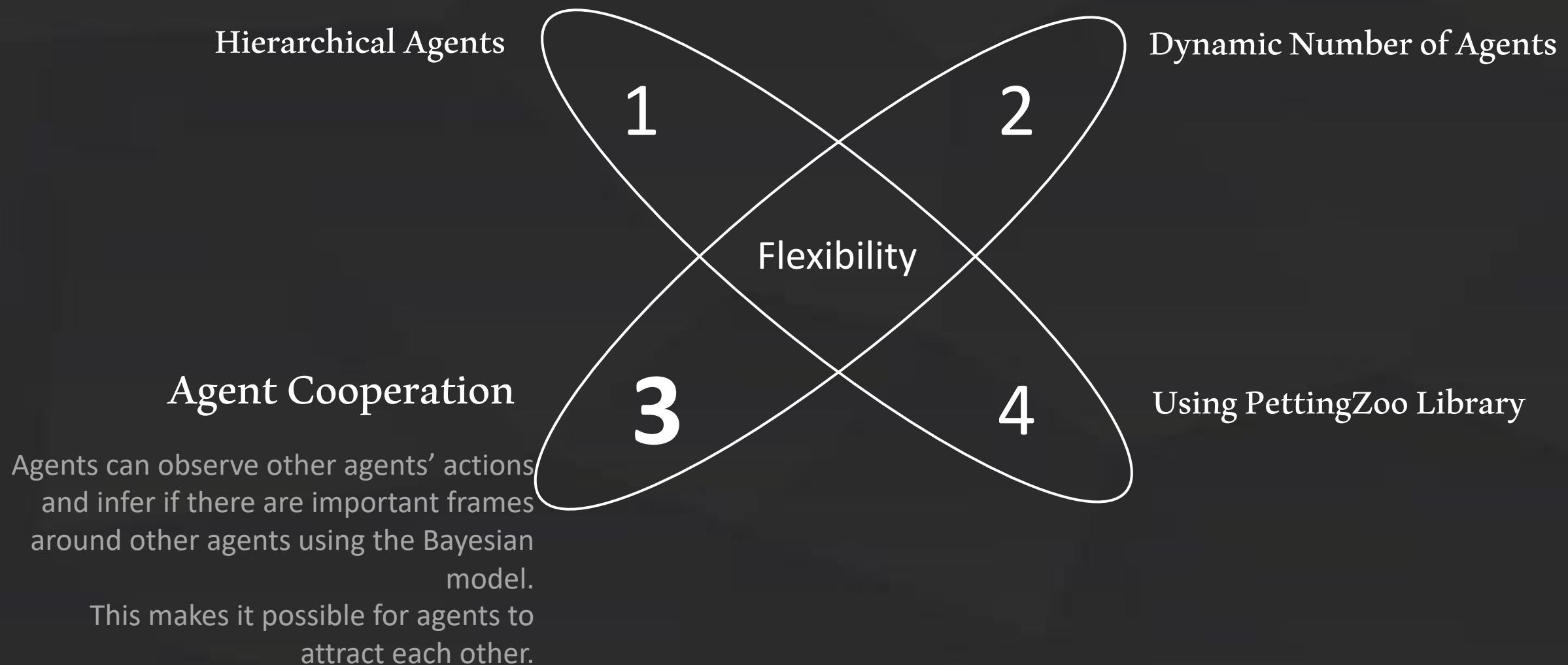
solution



- After the state of agents are stable, delete the agents stay alone from others which reward are below  $\beta \cdot reward_{av}$
- Distribute the deleted agents evenly on the middle of continuous agents

# Methods

---



---

# Agent Cooperation

---



- 1
- 2
- 3

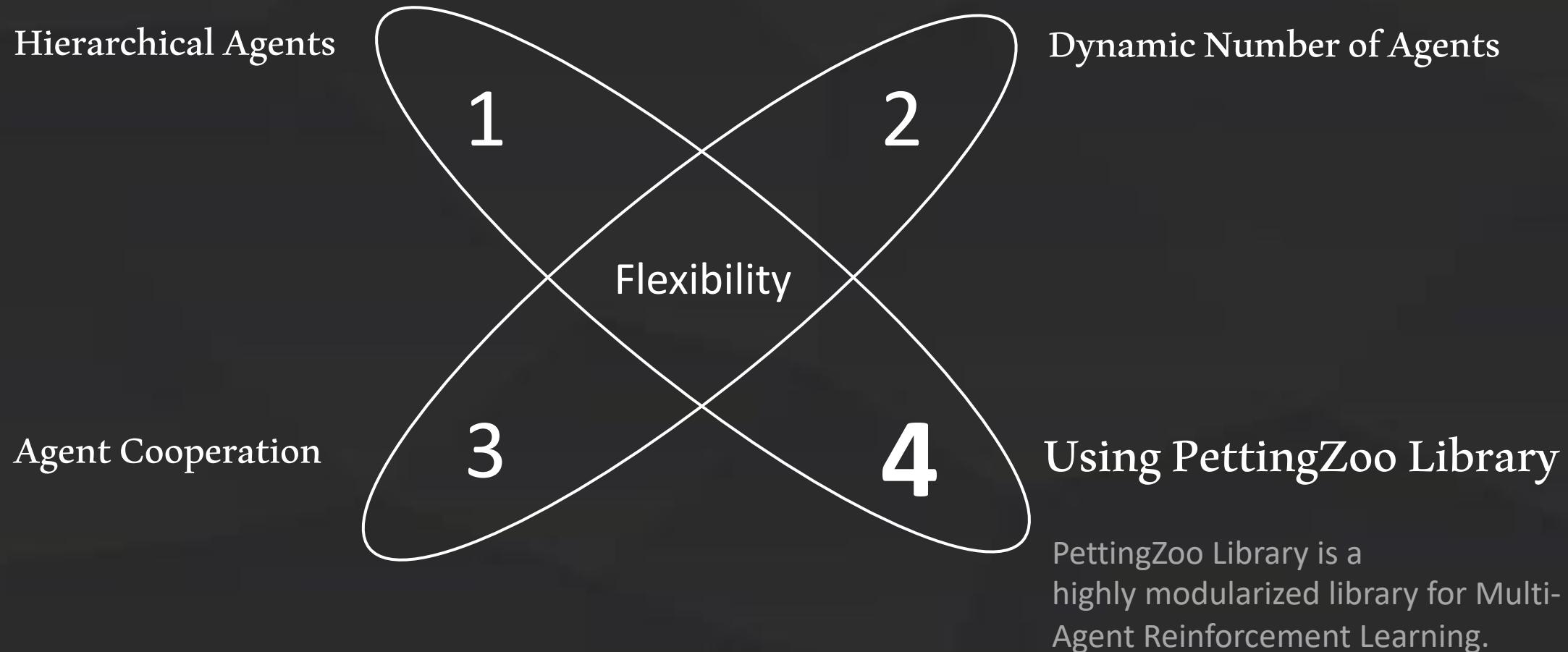
Using Bayes model based on the previous agents' action to get the features of other agents' state.

Deduce the value of other agents' state due to the features.

Then the agents can move forward to the more valuable state

# Methods

---



# Pettingzoo library

- PettingZoo can make the project easier to test and modify, as well as convenient for future comparison.
- Using AEC in pettingzoo can make agent update state one by one more quickly so that the agents can avoid overlap at all. It's difficult for parallel updating.



# Our Schedule

---

The Big Picture

1

Add flexibility to the  
multi-agent model

Agent cooperation, hierarchical  
agents, dynamic agent number, etc.

2

Replace C3D by GNN or  
transformer

They will have better performance on  
the feature extraction.

3

Build it with PettingZoo  
Using a library can make it easier to  
test and modify.

4

Test our model on more  
datasets. Build our own  
benchmark if necessary.  
Wild life cams and so on.



## Credits

Team information.  
Work allocation.

# Our Team



**Allen (Haomiao Tang)**

Wuhan University

Majoring in Information Security

[tanghm183@gmail.com](mailto:tanghm183@gmail.com)



**Bob (Bo Hu)**

Wuhan University

Majoring in Physics

[1294730262@qq.com](mailto:1294730262@qq.com)

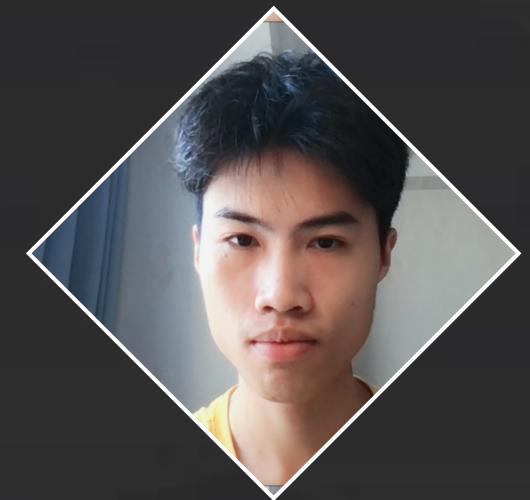


**Derek (Tianhao Dai)**

Wuhan University

Majoring in Cyberspace Security

[thdai2000@163.com](mailto:thdai2000@163.com)



**Farnante (Facheng Yu)**

Wuhan University

Majoring in Mathematics

[yufacheng@whu.edu.cn](mailto:yufacheng@whu.edu.cn)

# Allen Tang



## Work Summary:

I did some early surveys and helped with a little bit of everything in order to make sure everyone work consistently. I also make progress reports to the professor and TA.

Bob:

Allen is capable of organizing and promote project progress.

Derek:

Allen has great leading ability and sense of responsibility. He frequently initiated meetings in order to gather us together and discuss what we can do in the project. He spends a lot of time in learning to use PettingZoo library and finding for a feasible way agents cooperate. This nice presentation is also made by him.

Farnante:

Tang has a great ability to schedule things, thus he can do everything well in order and save time.

# Bob Hu



## Work Summary:

I did some job in survey, data preprocessing and code review. And I also discuss actively in group chat to share my ideas.

Allen:

Bob helped with many parts of our work, including surveys and evaluation. He had significant contribution on modelling as well.

Derek:

Bob joins many parts in the projects and puts forward many insightful ideas in the dataset representation and the cooperation of multi-agents. Before starting the project, he did an extensive survey in the video summarization and help us choose what architecture to use. He made the dataset for supervised learning, which is very useful.

Farnante:

Hu is willing to think, no matter in the theory or in the application.

# Derek Dai



## Work Summary:

Before starting our project, I did abundant survey of video summarization to know about different network architectures. During the project, I am responsible for the dataset processing and the model evaluation in the project. Together with Bob, I extracted important features from SumMe dataset, including transferring raw videos into visual and chronological features, picking key frames to help supervised learning, and then save them into the hdf5 file. I also modified existing codes to calculate the F-score for our project.

Allen:

Derek made great contributions to the visualization and analysis of our result. Without him we won't be able to draw so much useful conclusions.

Bob:

He is very helpful and is very kind to answer your questions.

Farnante:

Dai is enthusiastic and always helpful when doing experiments.

# Farnante Yu



## Work Summary:

I work in code implementation part. The modifying and optimising of the code is tiring but really cheering when it works. I am happy to work with members in our group.

Allen:

Farnante is very sharp on understanding models and diligent in implementing the code.

Bob:

He is doing a lot of coding work and is very patient to explain his code and ideas.

Derek:

Farnante, with his proficiency in mathematics, can quickly understand math in different papers and he is willing to help us get through those difficult parts. As for the coding, he has done the majority part of implementing the MARL in video summarization. He also wrote a simplified document of PettingZoo library to help us use it.

*Thank you  
For Watching!*