Video Summarization

with Flexible Multi-Agent Reinforcement Learning

Allen (Haomiao Tang)
tanghm183@gmail.com

Bob (Bo Hu)
1294730262@qq.com

Derek (Tianhao Dai)
thdai2000@163.com

Farnante (Facheng Yu)
yufacheng@whu.edu.cn

Content

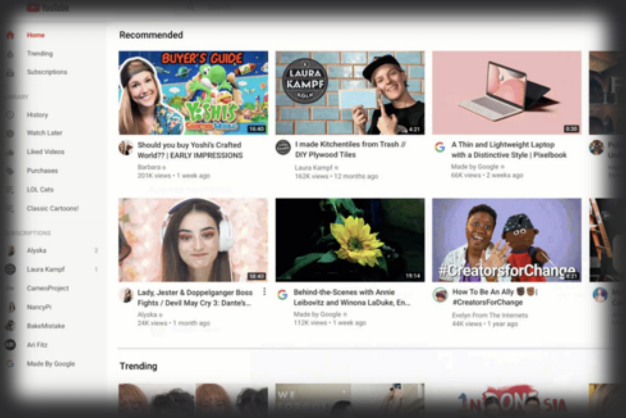
Presentation

1

Background

Why does the topic appeal to us.

Video Summarization is very helpful



YouTube Front Page

Quick glance at videos

Ball Games

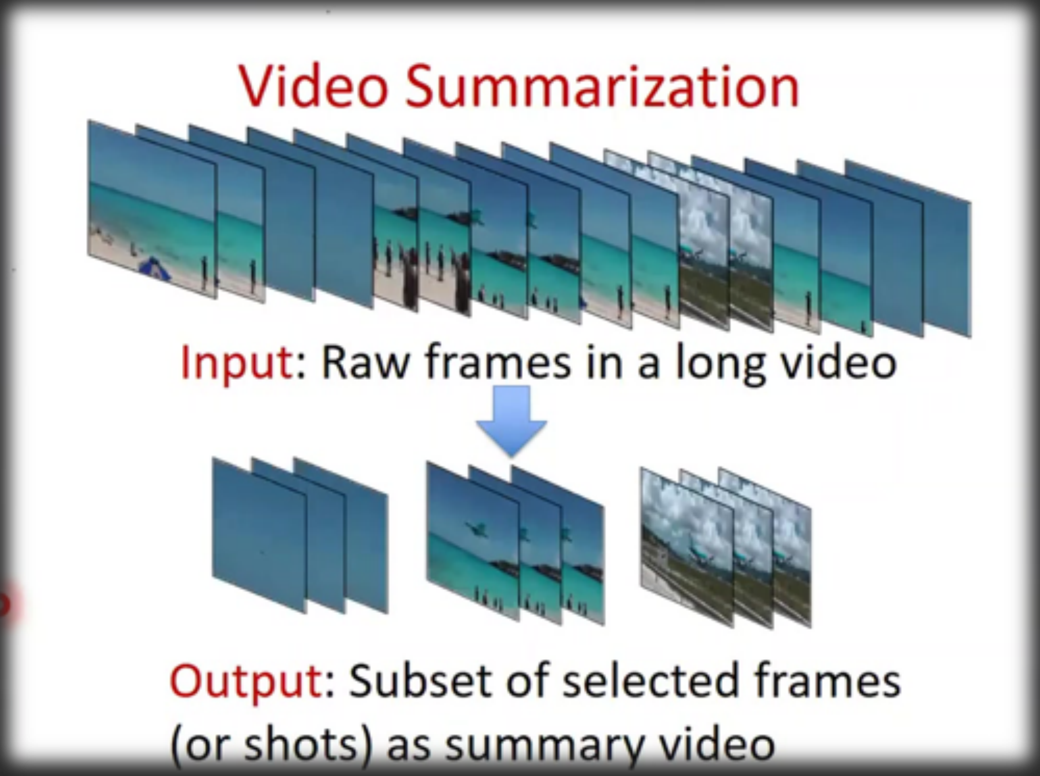
Mark highlight moments

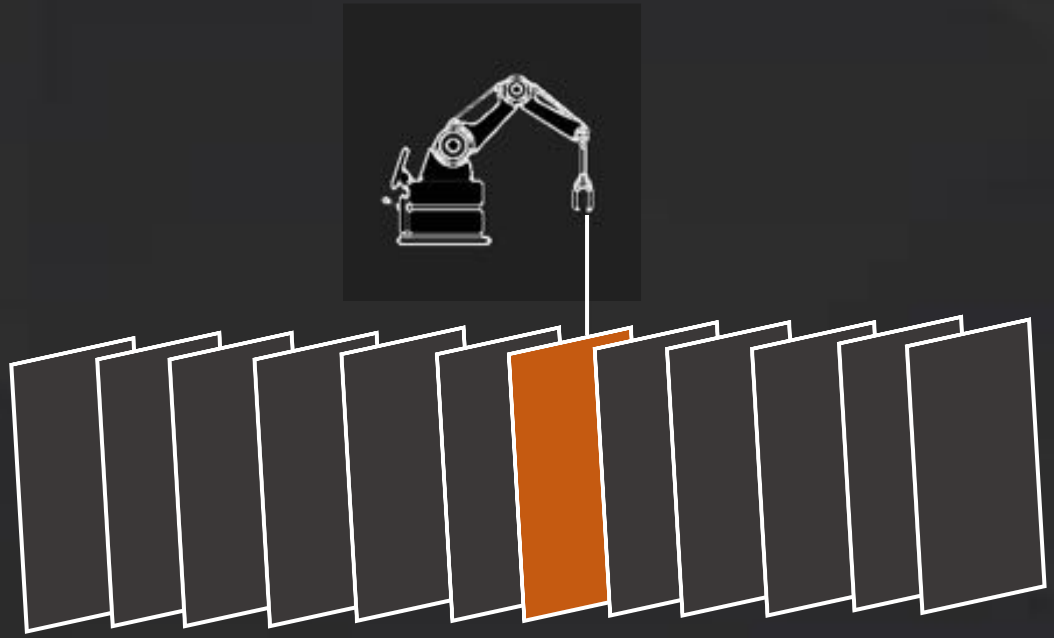
Cameras in the Wild

Track animal occurrences

Reinforcement Learning works well on Video Summarization

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





Agent picking out important frames from video

2

Relevant Works

Important papers that inspired us.

Important Papers: DSN (2018)

✓ **Single Agent Model**

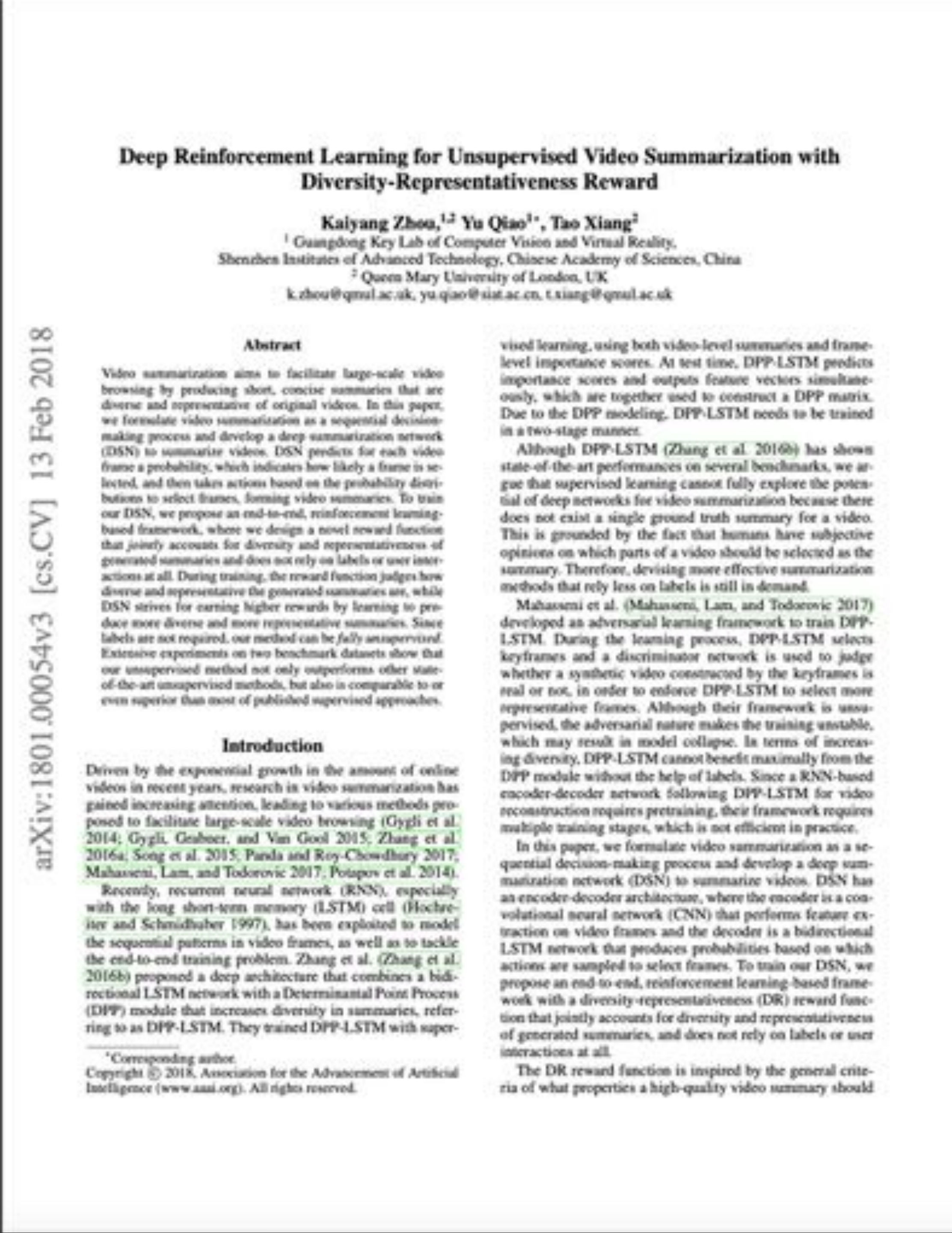
➢ **Representativeness:**

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

Kaiyang Zhou,[1,2] Yu Qiao[1*], Tao Xiang[2]
[1] Guangdong Key Lab of Computer Vision and Virtual Reality,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[2] 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, Grabner, 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 supervised 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

*Corresponding author.

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
liutyb@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 parts. 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, SumMe [10] and 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 difference 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, 59, 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. [30] 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], Rochan and Wang used the idea that unpaired videos and summaries can reduce the subjectivity caused by the interdependence of paired ones. In [54], 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

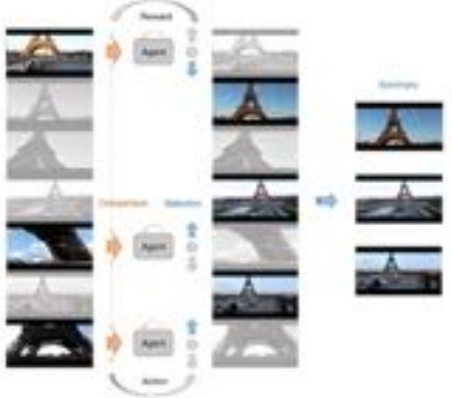
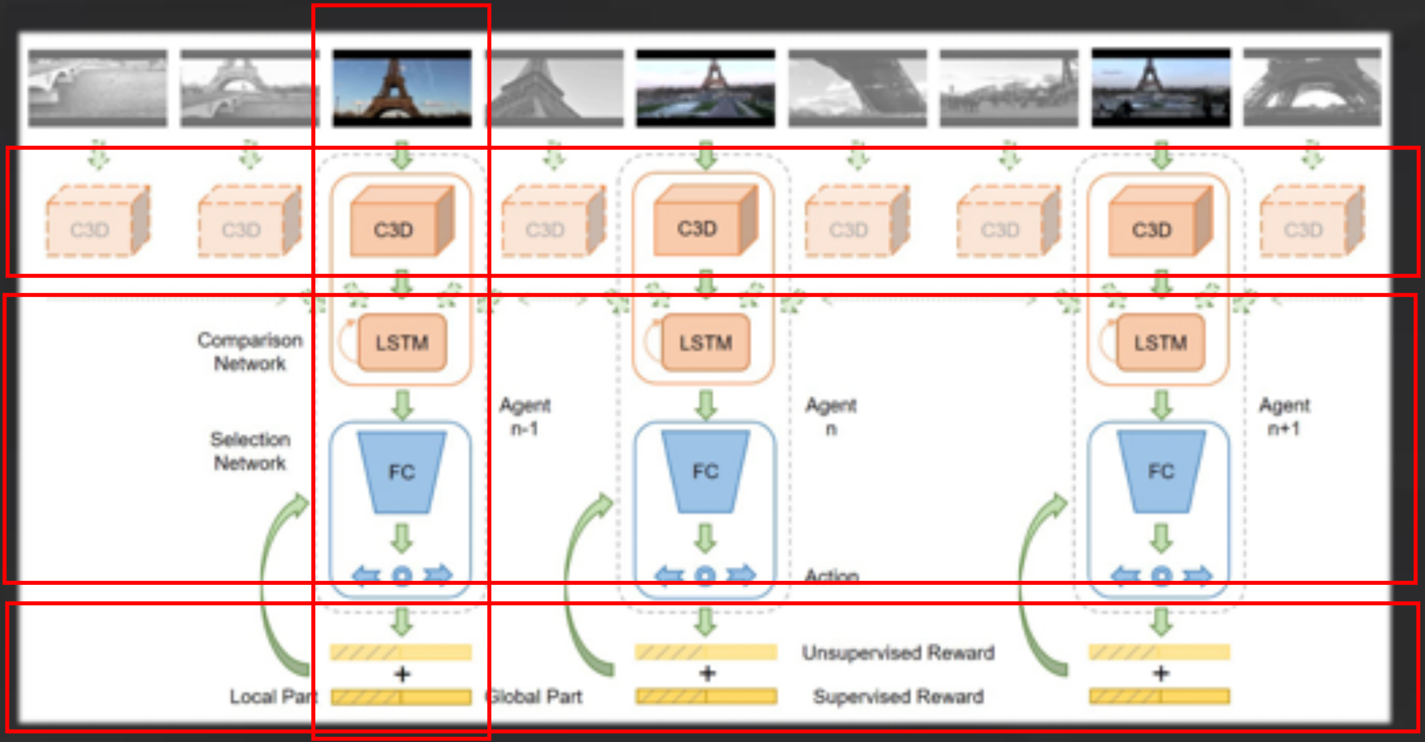


Figure 1: Agents compare the clips by taking the visual features and chronological features as input, then make decisions on which clips to select in the next round.

State of the Field

CoSNet: How it works

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



2
Observe features of clips using Convolutional 3d network.

3
Make actions according to observation and policy. (move focus left/right to a new clip)

Action Space: [-16, -8, -4, -2, 0, +2, +4, +8, +16]

4
Calculate rewards according to actions and updates the policy.

1
Each agent focus on a clip (of 16 frames)

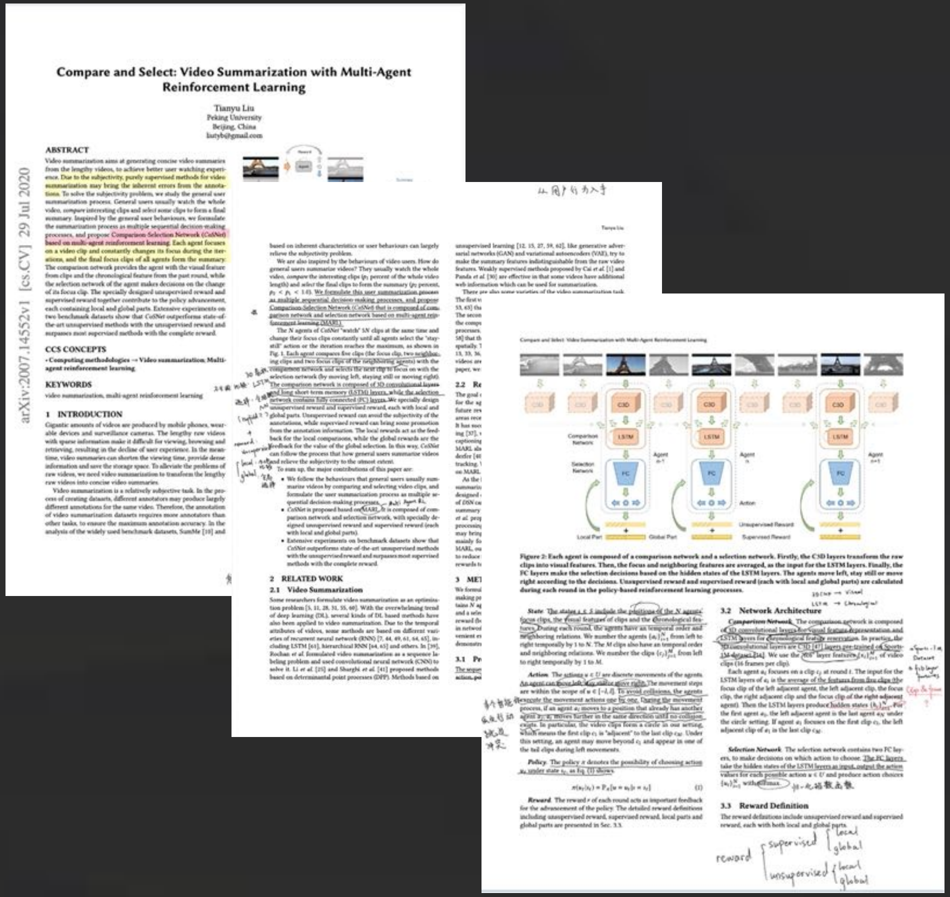
✓ Ends when all agents choose to stay still.

✓ The final focus clips of all agents form the summary

Our Practices & Progress

Survey

Understanding CosNet



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

Compare and Selec
with Multi-Agent R

Abstract

概述

主要成果

网络结构

策略改进

效果测试

网络结构

我们关心的问题

Unsupervised Reward

Local

Global

unsupervised reward

Supervised Reward

Method Comparison

@Allen

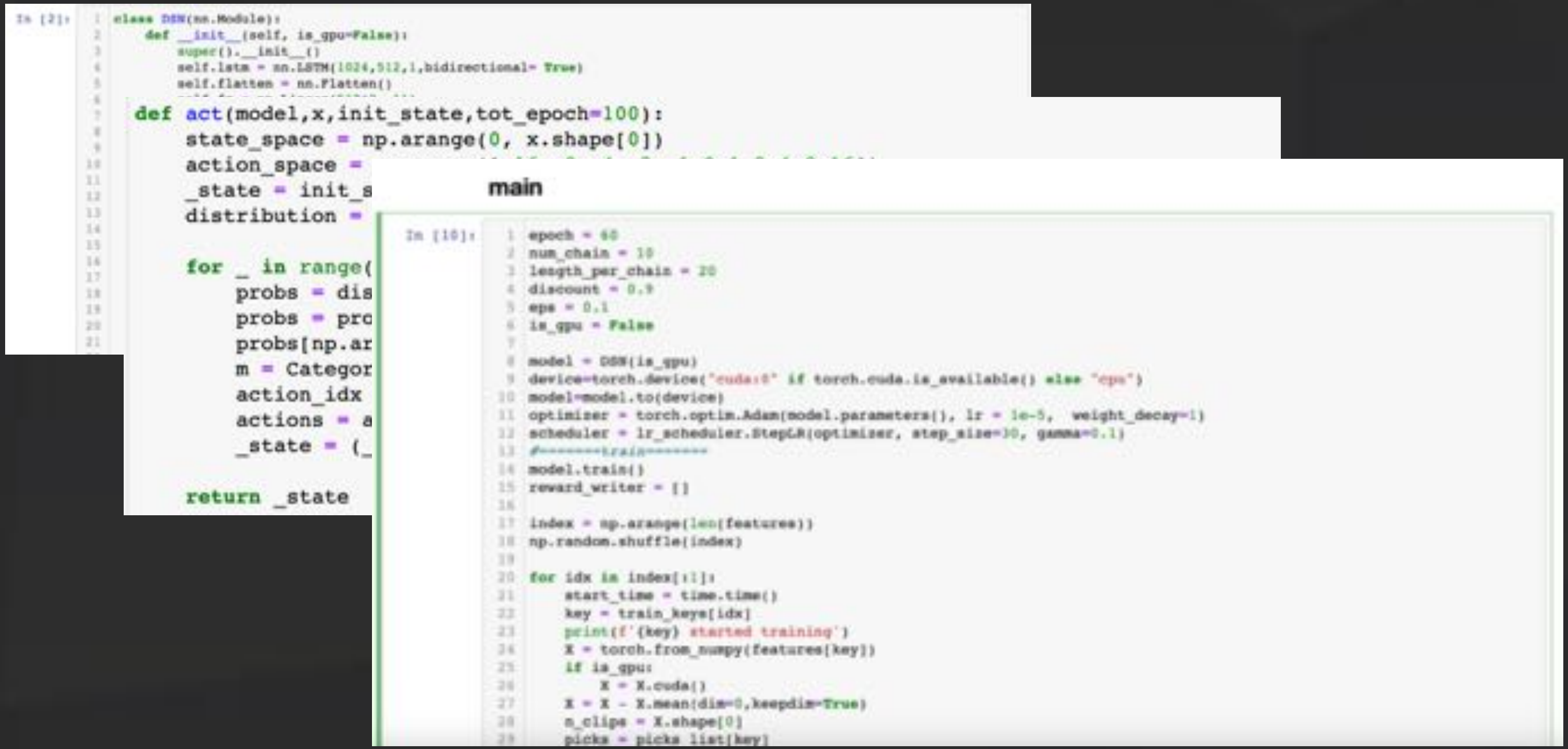
3

Our Practices & Progress

Our own implementation and result analysis.

Code Implementation

Coding the Model



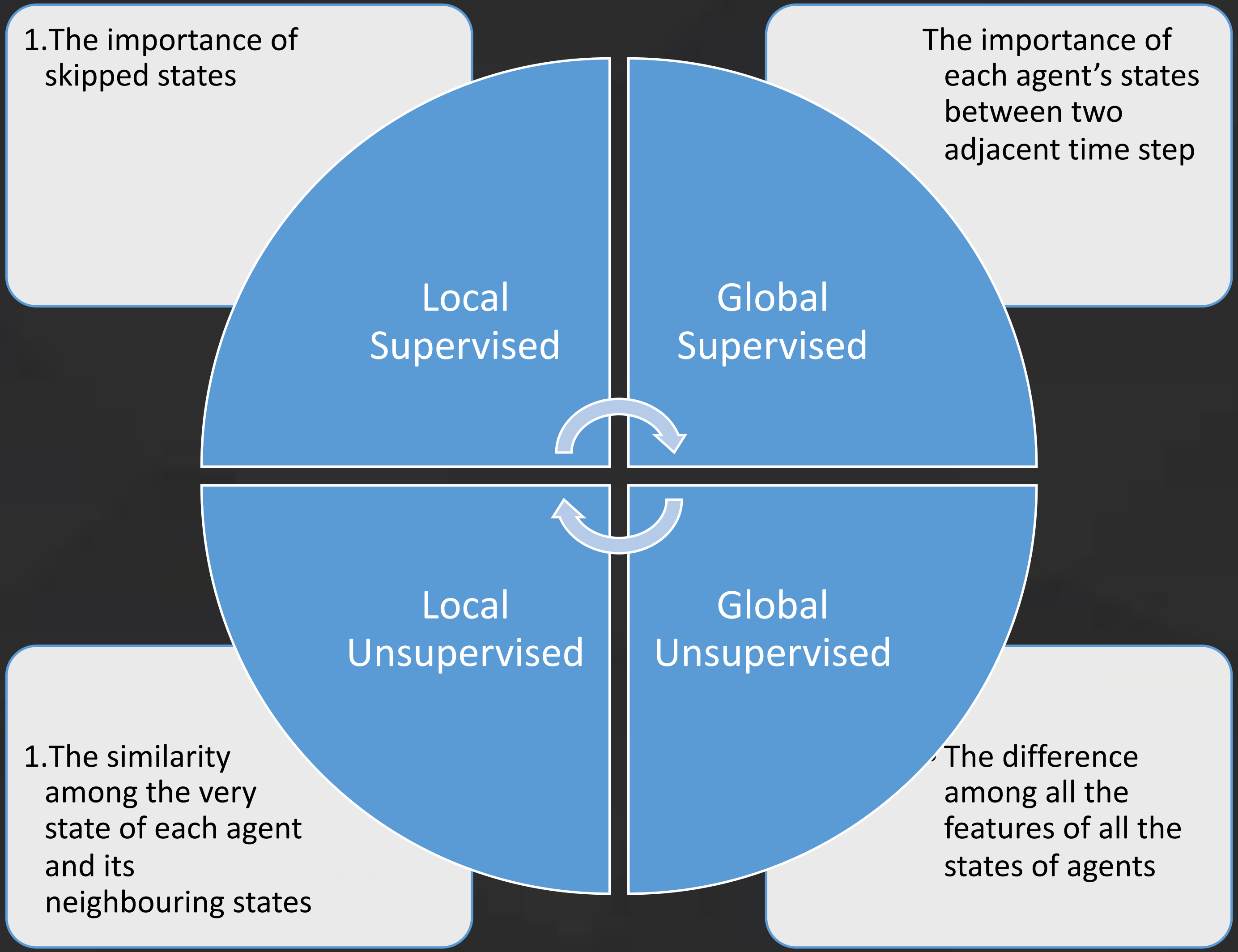
- ✓ 500+ Lines of Code
- ✓ Gradient Decent Algorithm: Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning 8, 3-4 (1992), 229–256.

@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.

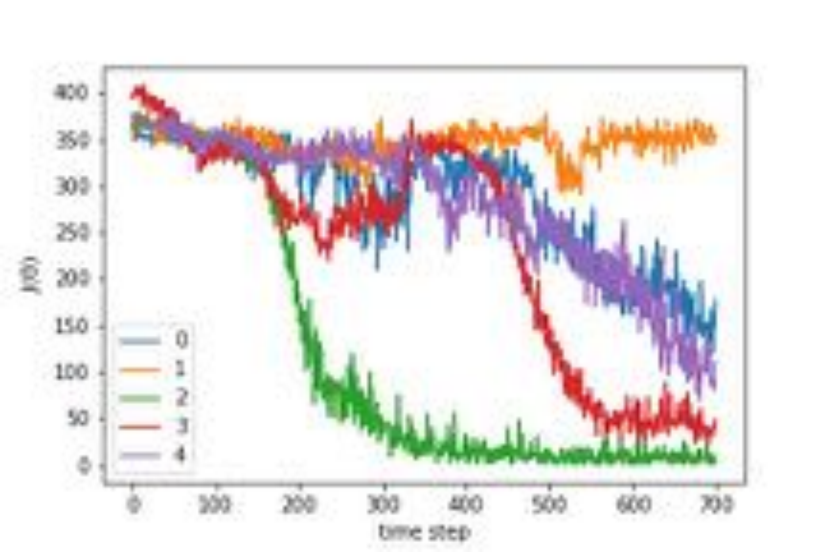
- Our agents share the same policy so it ‘s possible to compute the policy at the beginning of each update epoch.

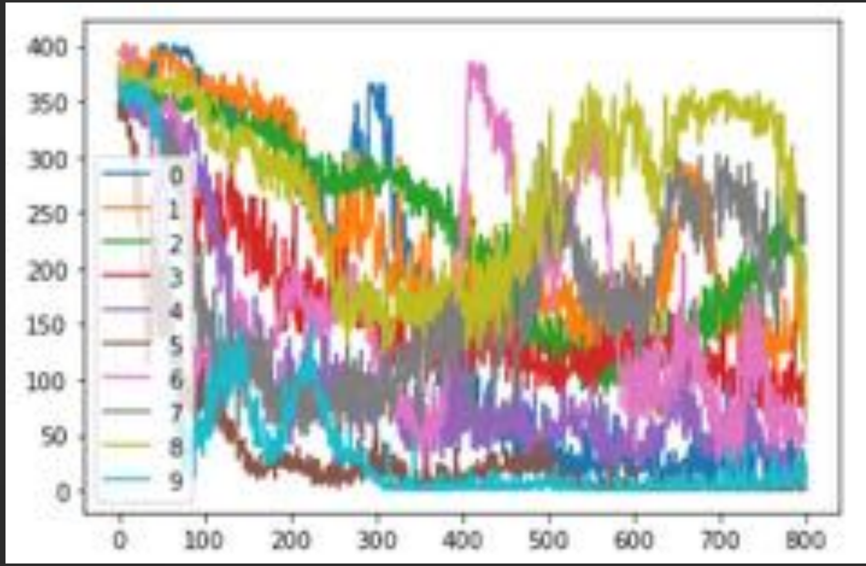
- The sharing allow our model running fast.

- 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



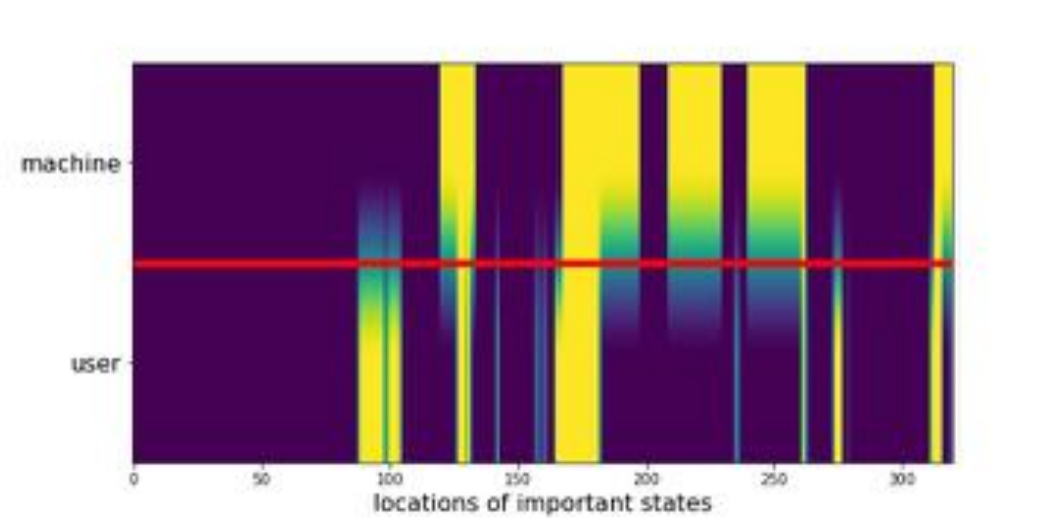


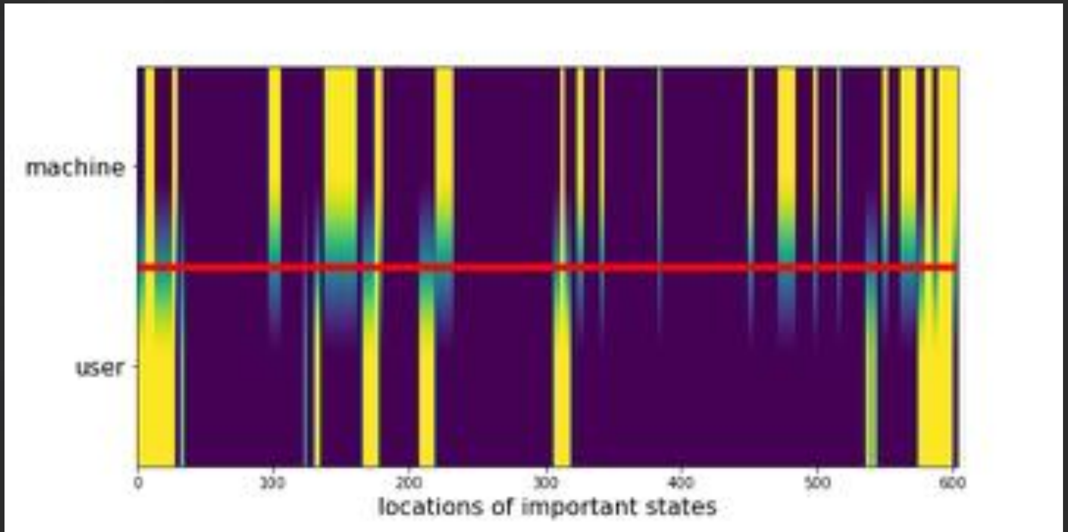
The Goal States Our Model learned

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

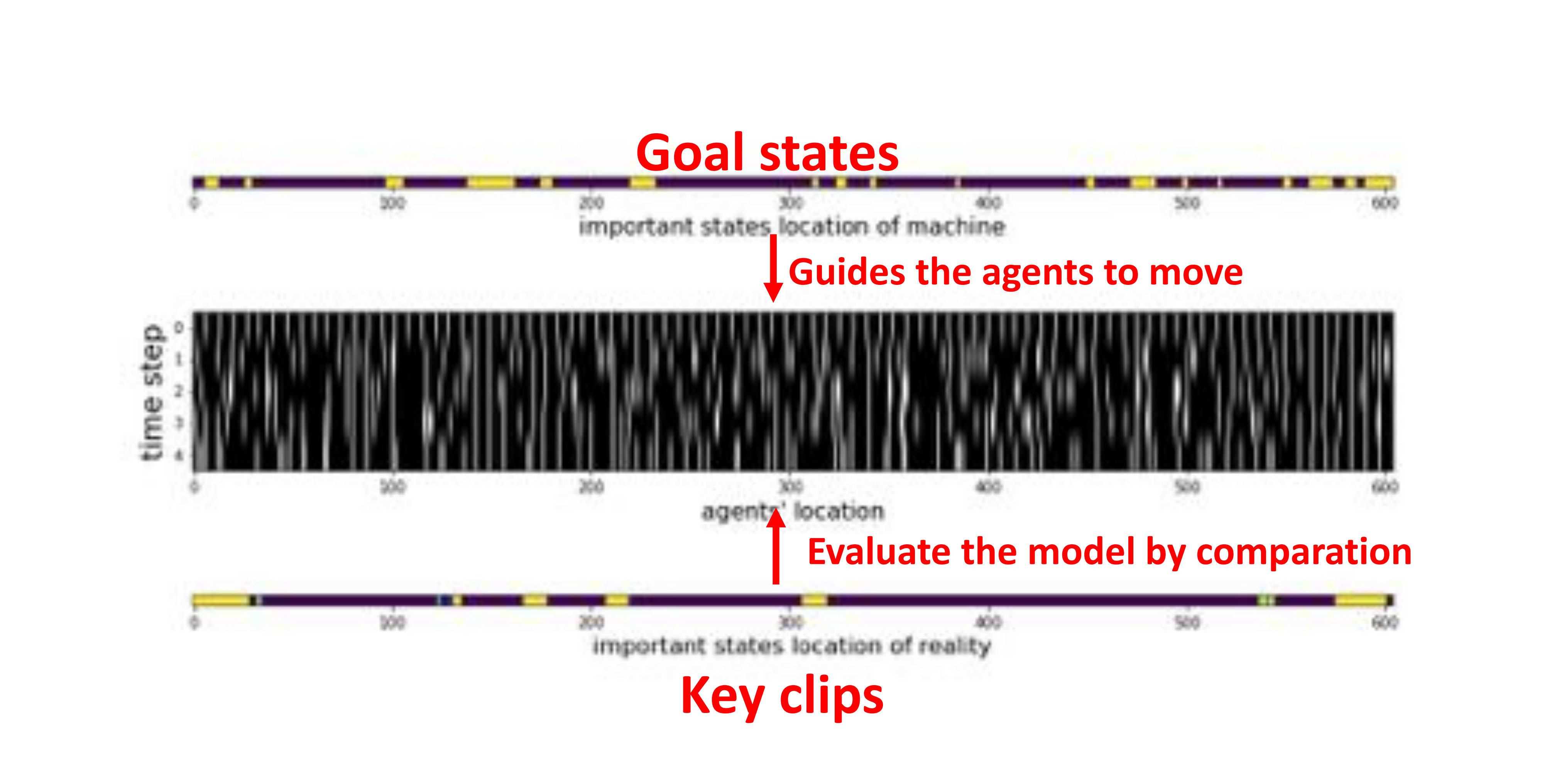
In reality, a goal state correspond to a **key clip** that summaries certain part of a video.

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



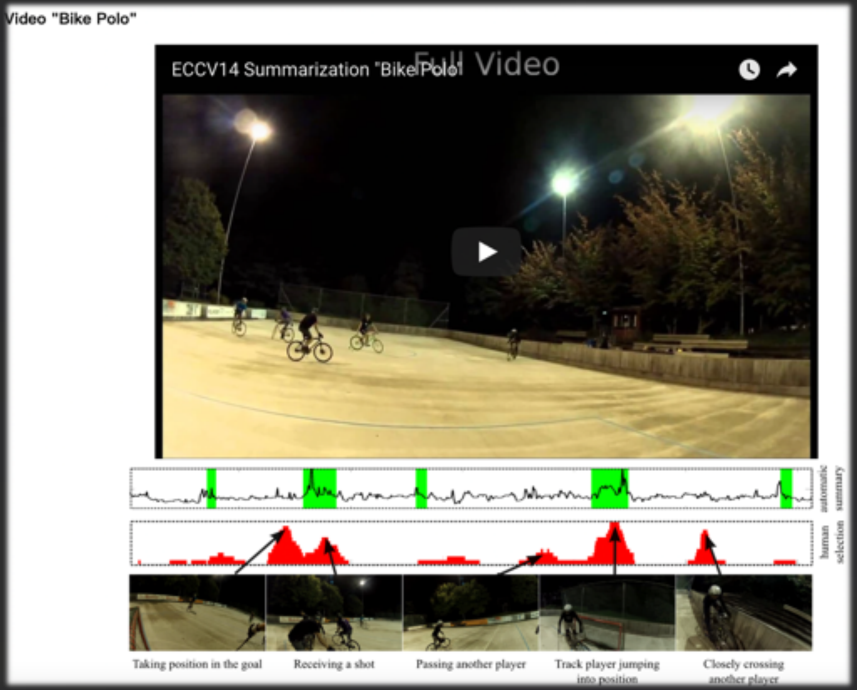


Agents Try to Move Following the Trained Policy



Evaluation

Popular benchmarks



SumMe

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

25 VIDEOS

Each video 1~6 minutes

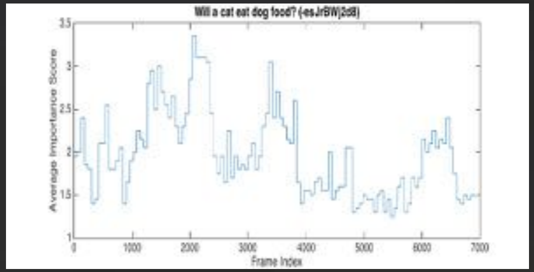
Various topics such as holidays and sports

15 – 18 ANNOTATORS PER VIDEO

390 in total

Evaluation

Popular benchmarks



TVSum

https://github.com/yalesong/tvsum/blob/master/

50 VIDEOS

Each video 2~10 minutes

Collected from YouTube and vary in topic

20 ANNOTATORS PER VIDEO

Frame-level importance score

Evaluation

1. Divide video into clips

16 frames per clip

clip 1

clip 2

......

clip n

Our Practices & Progress

Evaluation

2. Select clips through trained model

clip 1

clip 2

……

clip n

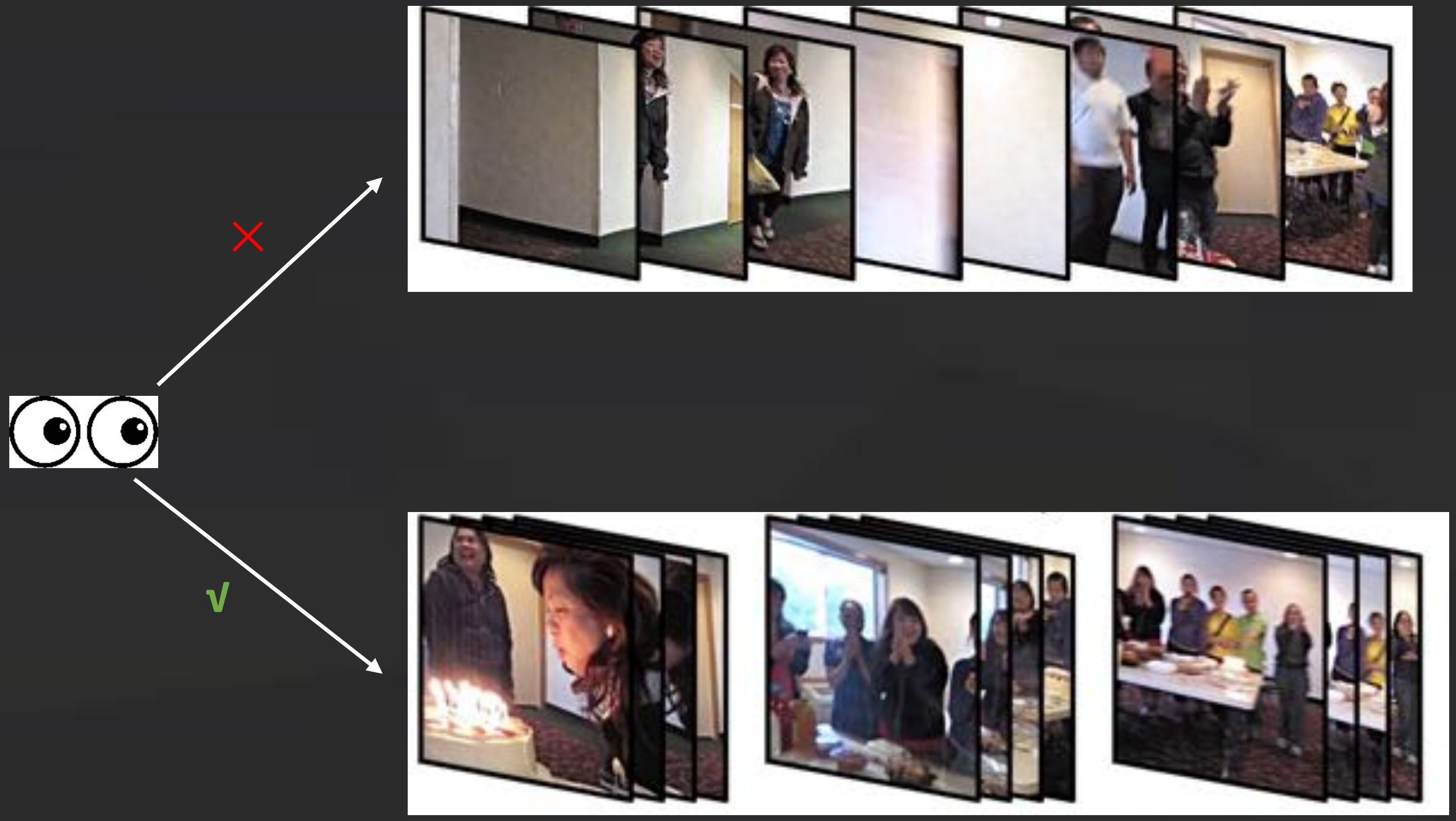
√

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

KTS method

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

Picked clips

[0, …,0, 1, 1, …, 1, 0, 0,…, 0, 0, 1, 1, …, 1]

Clips segmented by KTS

[0, …,0, 1, 1, …, 1, 0, 0,…, 0, 0, 1, 1, …, 1]

Evaluation

3. Generate human-friendly summary

Clips segmented by KTS

[0, …,0, 1, 1, …, 1, 0, 0,…, 0, 0, 1, 1, …, 1]

2 key frames   1 key frame   3 key frames

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

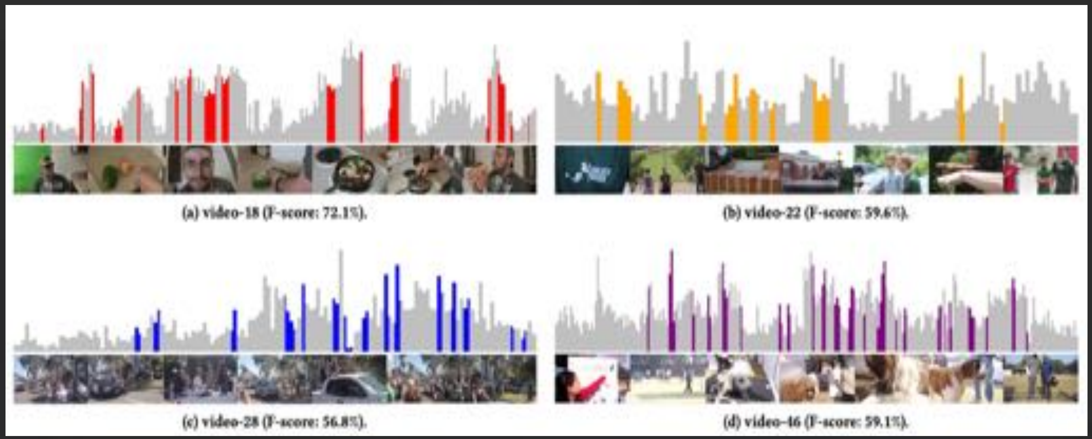
Knapsack problem/ranking problem

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

Our Practices & Progress

Evaluation

4. Calculate the F-score



(a) video-18 (F-score: 72.1%). (b) video-22 (F-score: 59.6%).

(c) video-28 (F-score: 56.8%). (d) video-46 (F-score: 59.1%).

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:

$$A = vector\ of\ machine\ summary$$

$$B = vector\ of\ user\ summary$$

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

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

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

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

Evaluation

5. Result visualization

Ground truth

Ground truth importance score of clips

softmax

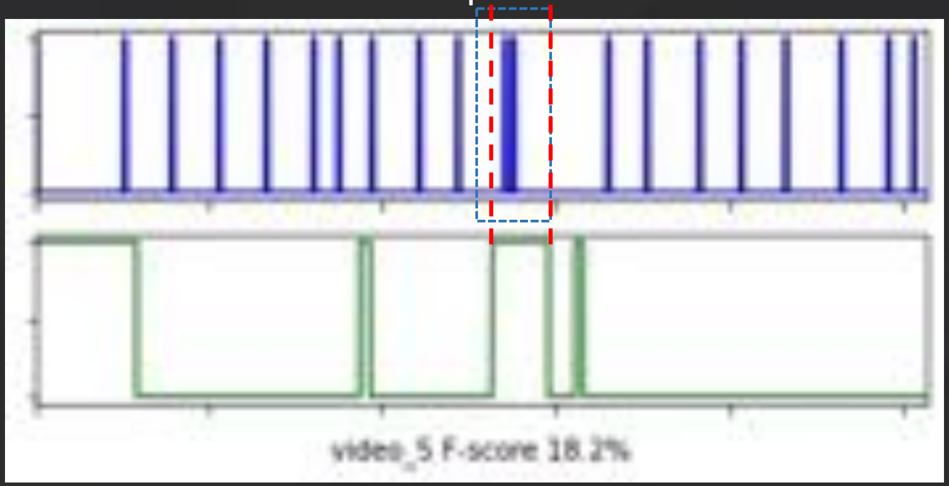
[0, 1, 2, 3, …, 2, 1, 0] =

User 1: [0, 0, 1, 1, …, 1, 0, 0]
+
User 2: [0, 1, 1, 1, …, 0, 0, 0]
+
User n: [0, 0, 0, 1, …, 1, 1, 0]

Machine performance

Clips agents finally choose

Final summary



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

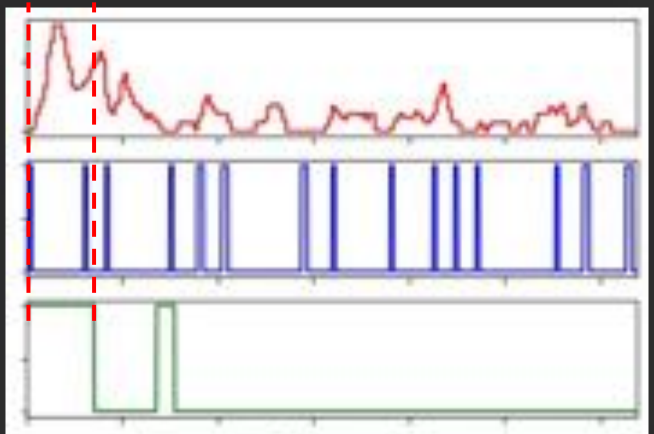
The most important segments model chooses.
The closer they are to high ground truth importance parts, the higher F-score is.

Evaluation

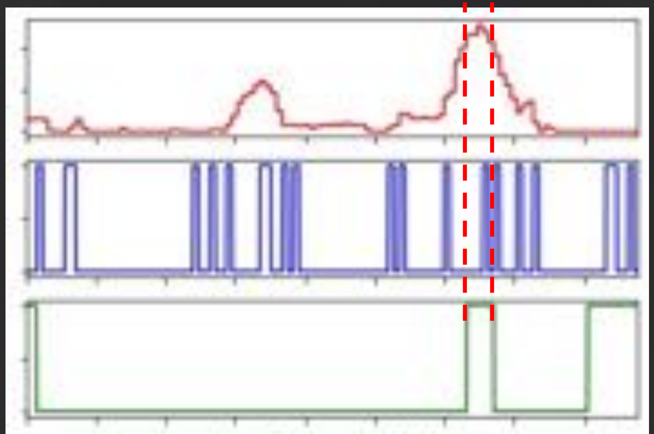
6. Good results

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%
---	--------	-------

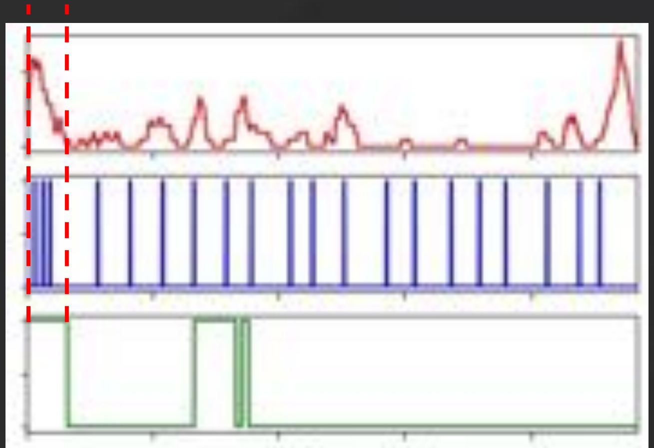
Average F-score 15.2%



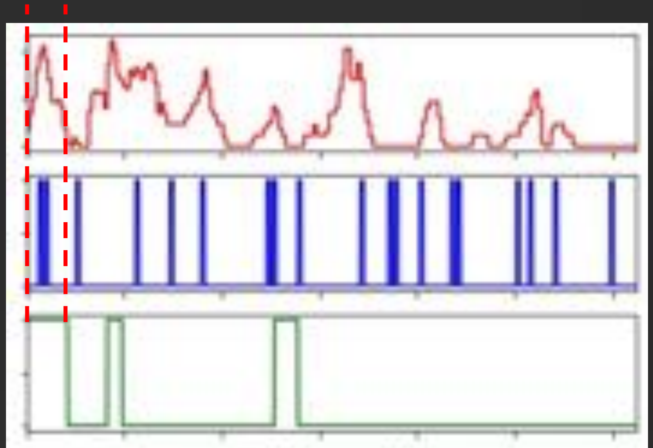
video_13 F-score 33.8%



video_19 F-score 30.2%



video_21 F-score 31.5%



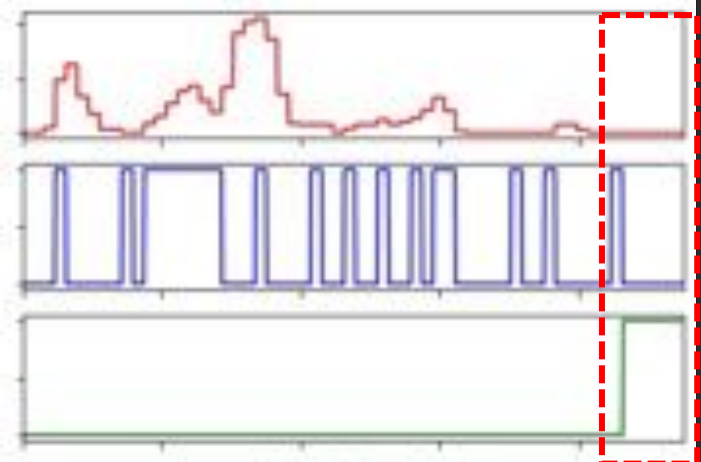
video_25 F-score 25.6%

Evaluation

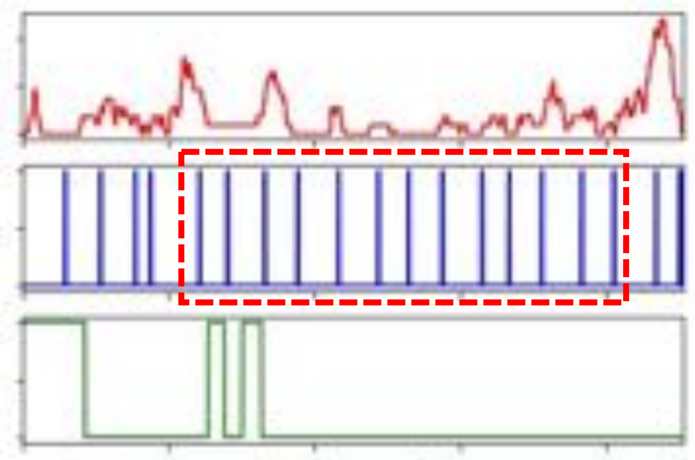
6. Poor results

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%

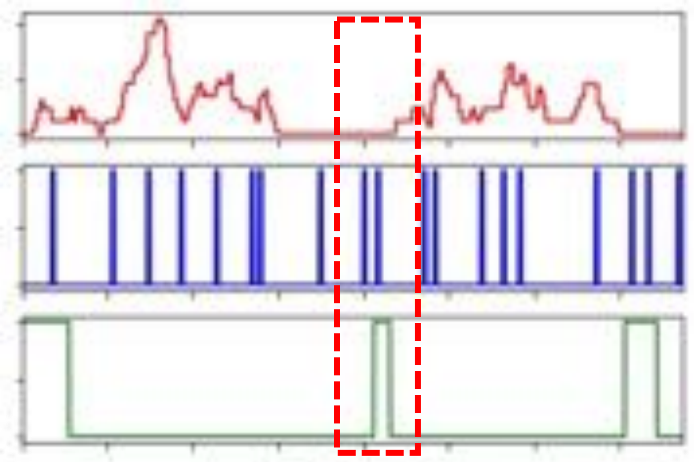
Average F-score 15.2%


video_12 F-score 0.0%

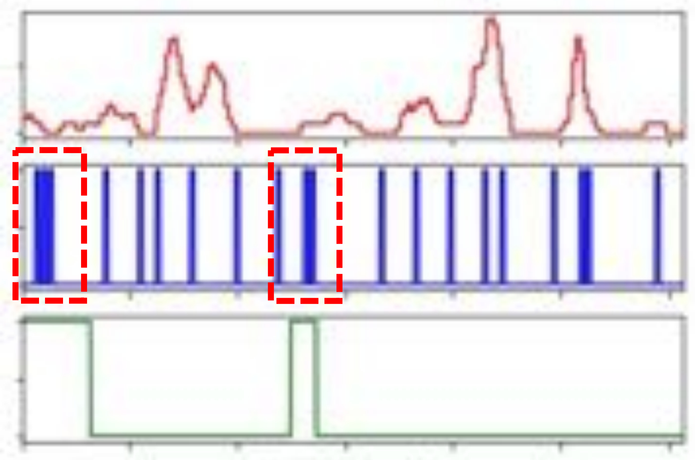
Wrong Segment picking


video_7 F-score 5.9%

Evenly distributed


video_20 F-score 3.2%

Wrong Segment picking


video_4 F-score 4.9%

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.

4

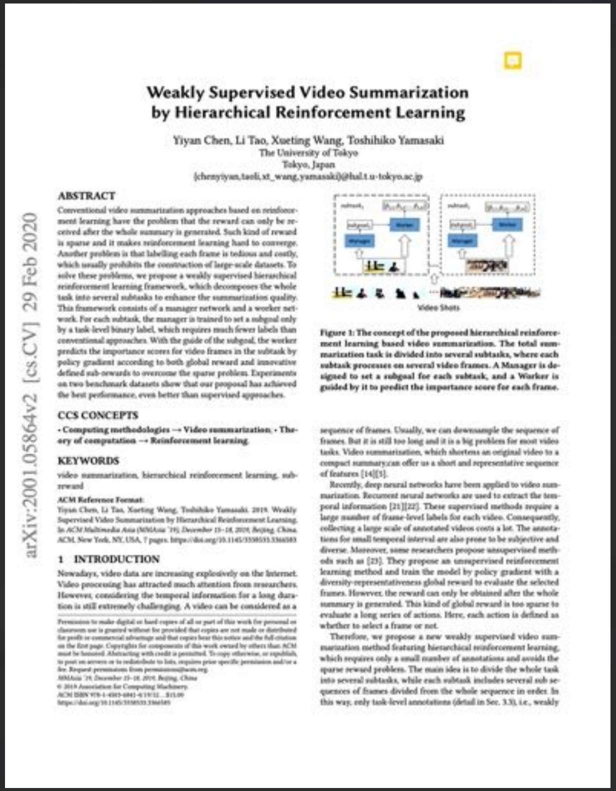
Our Ideas & Future Plans

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

Survey

Finding Methods & Modelling

@Bob



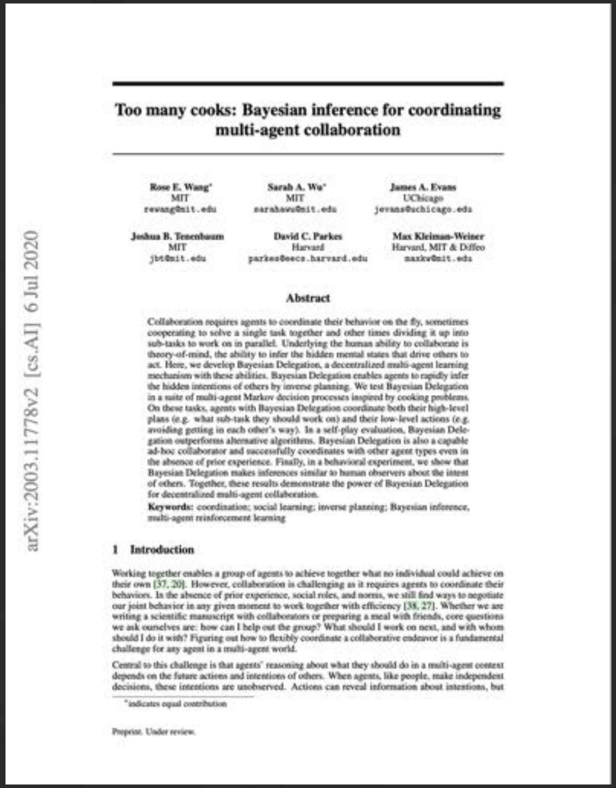
Weakly Supervised Video Summarization by Hierarchical Reinforcement Learning

Yiyan Chen, Li Tao, Xueting Wang, Toshihiko Yamasaki
The University of Tokyo
Tokyo, Japan

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

Weakly Supervised Video Summarization by Hierarchical Reinforcement Learning:

Inspired the thought of manager-worker agent model



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

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

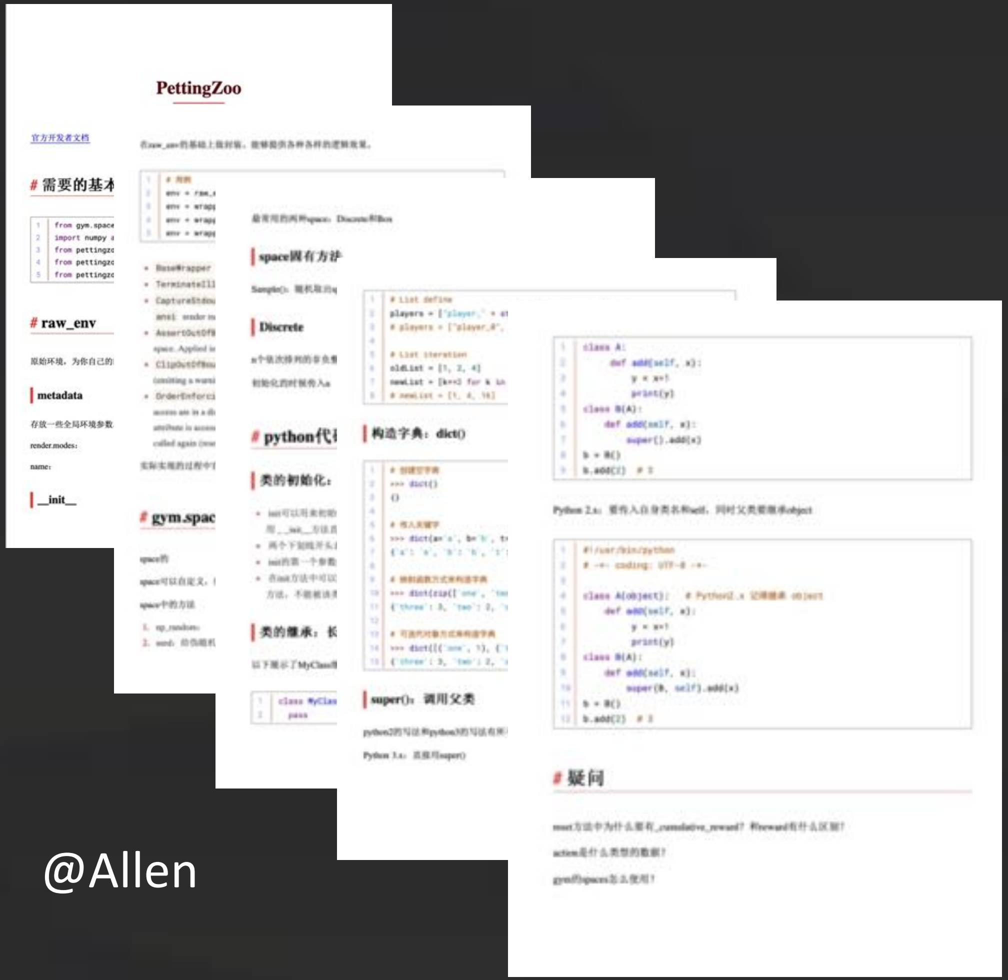
Too many cooks: Bayesian Inference for coordinating multi-agent collaboration:

Inspired the thought of agent collaboration.

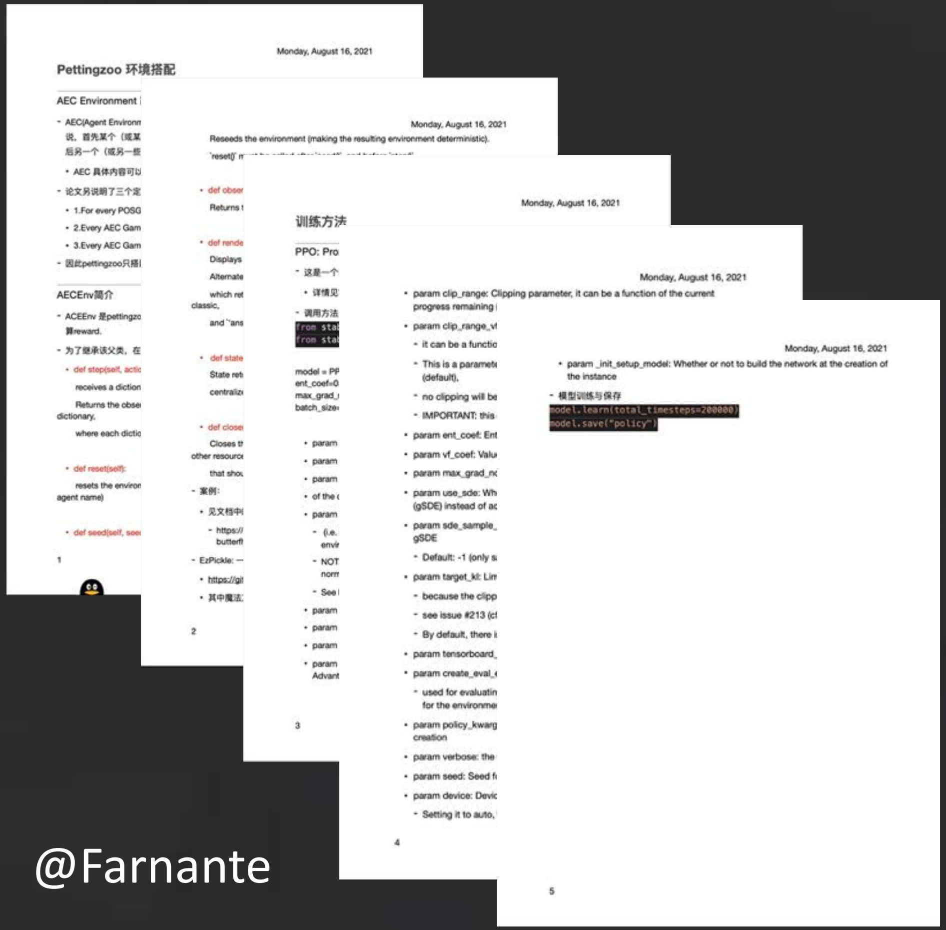
Our Ideas & Future Plans

Survey

Understanding PettingZoo



@Allen

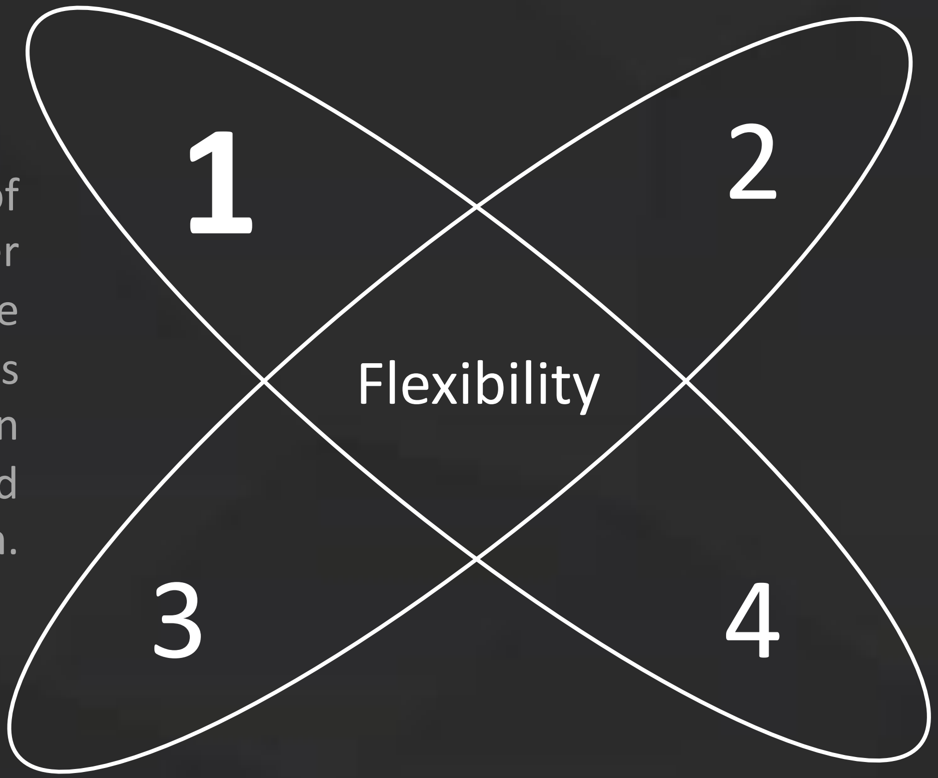


@Farnante

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.



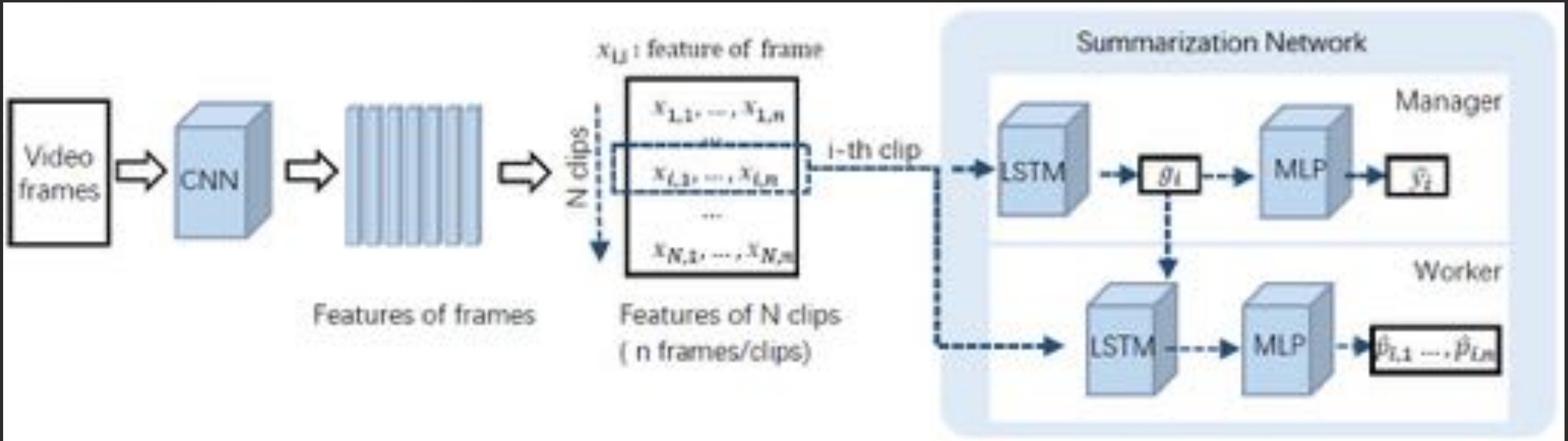
Dynamic Number of Agents

Agent Cooperation

Using PettingZoo Library

Modelling

Hierarchical Agents



Manager Network

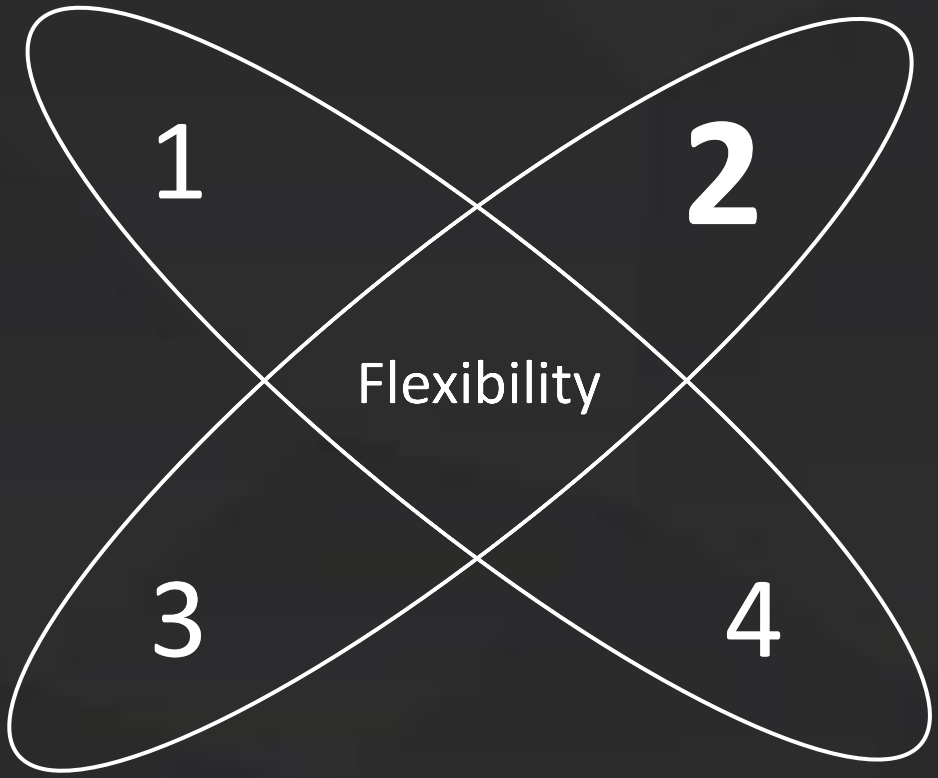
Input features clips into manager network. Get the hidden state of Manager's LSTM which contains the global feature of video.

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



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

Agent Cooperation

Modelling

Dynamic Number of Agents

Two types of problem

- The agents distribute sparsely
- The agents overlaps too much

Sparse

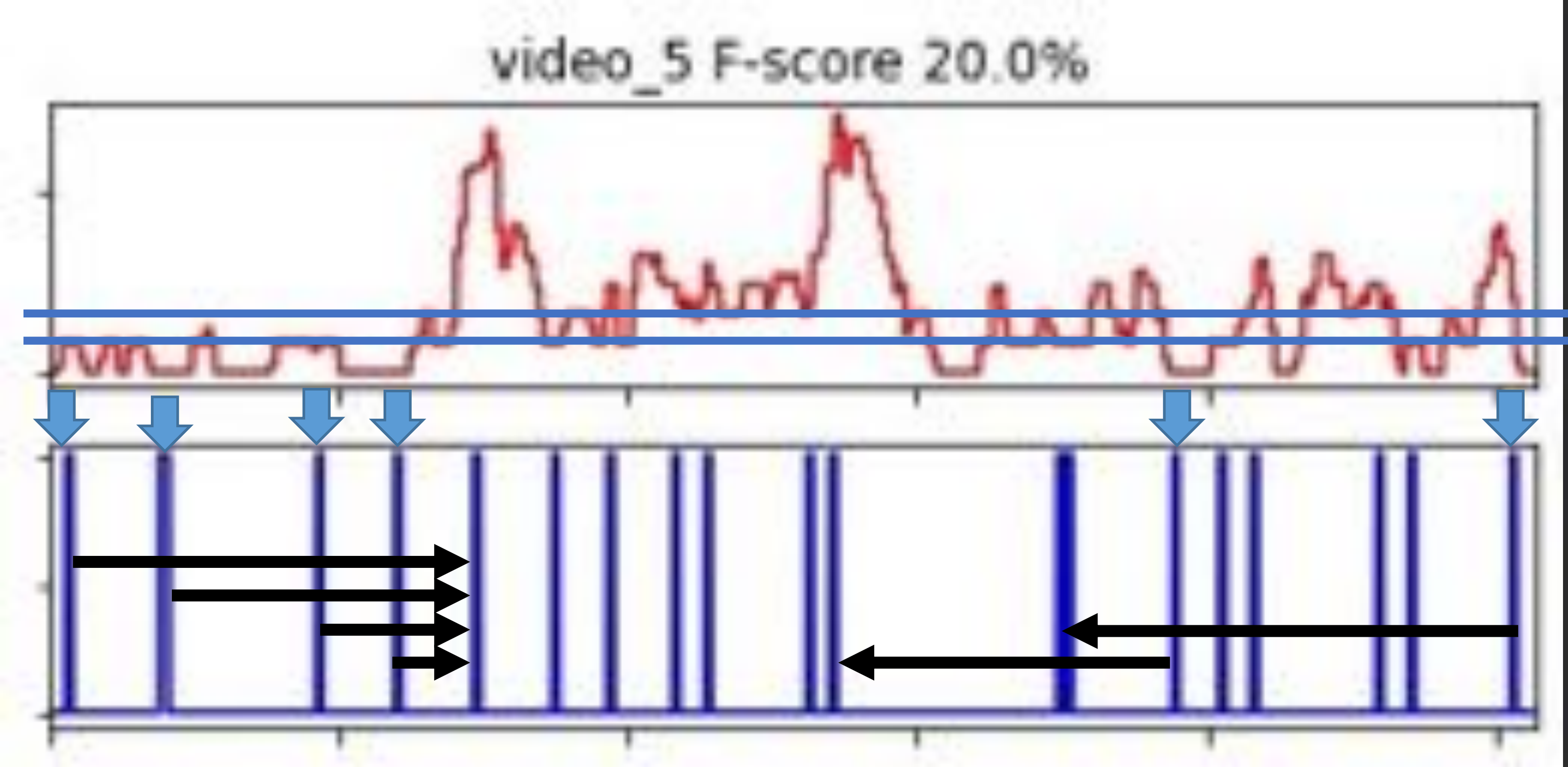
solution

Agent reward

$reward_{av}$

$\beta \cdot reward_{av}$

Agent final state



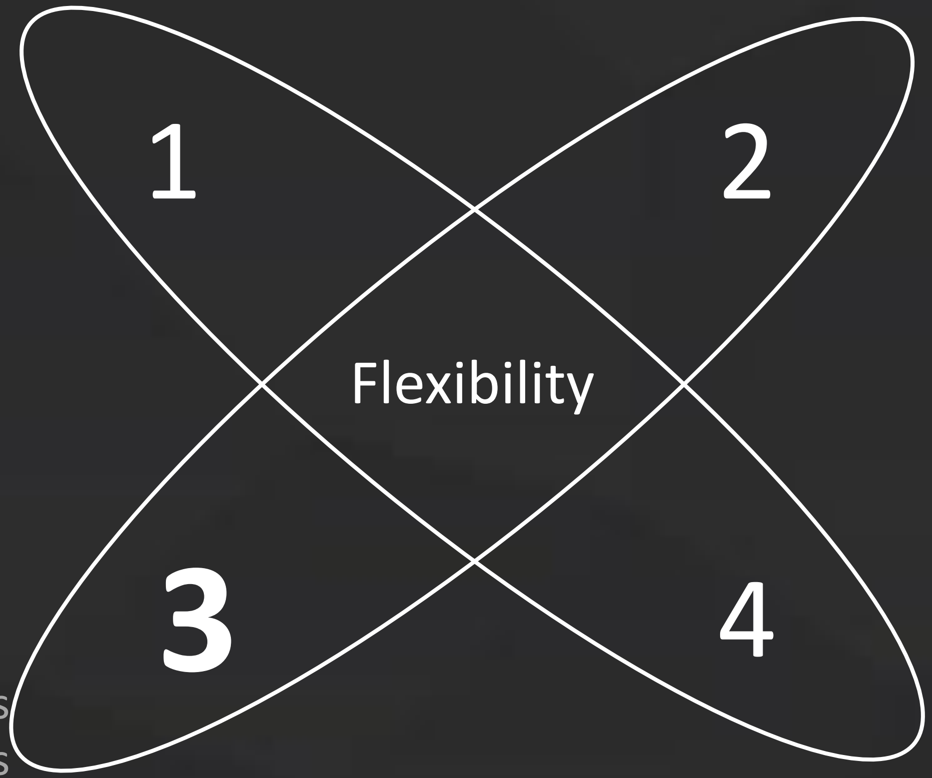
state

- 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

Hierarchical Agents

Dynamic Number of Agents



Agent Cooperation

Agents can observe other agents’ actions and infer if there are important frames around other agents using the Bayesian model.
This makes it possible for agents to attract each other.

Using PettingZoo Library

Modelling

Agent Cooperation

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

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

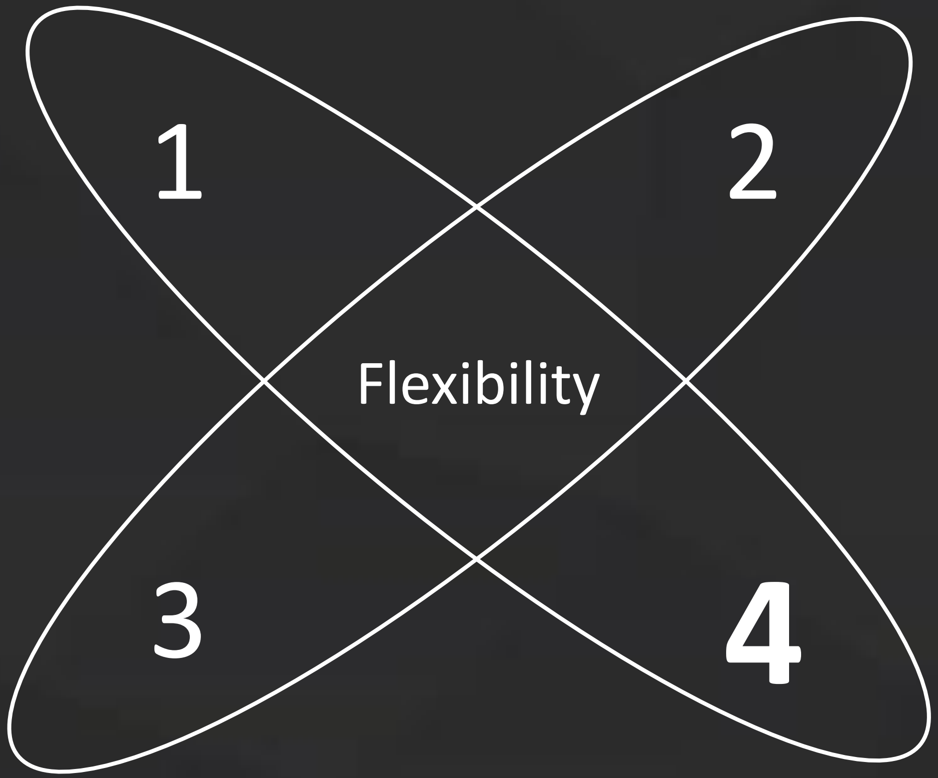
3. Then the agents can move forward to the more valuable state

Our Ideas & Future Plans

Methods

Hierarchical Agents

Dynamic Number of Agents



Agent Cooperation

Using PettingZoo Library

PettingZoo Library is a highly modularized library for Multi-Agent Reinforcement Learning.

Modelling

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.

5

Credits

Team information.
Work allocation.

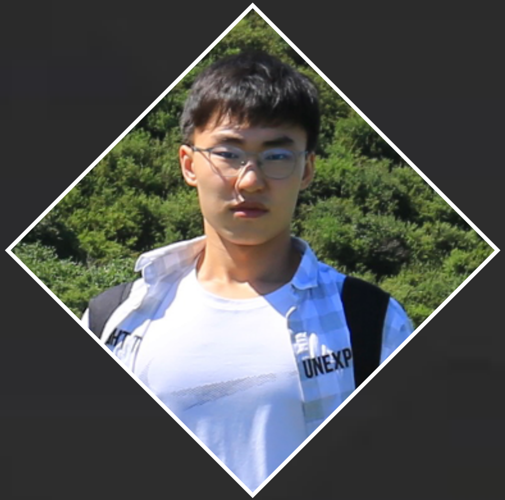
Our Team

Allen (Haomiao Tang)

Wuhan University

Majoring in Information Security

tanghm183@gmail.com



Bob (Bo Hu)

Wuhan University

Majoring in Physics

1294730262@qq.com

Derek (Tianhao Dai)

Wuhan University

Majoring in Cyberspace Security

thdai2000@163.com

Farnante (Facheng Yu)

Wuhan University

Majoring in Mathematics

yufacheng@whu.edu.cn

Credits

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.

Credits

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.

Credits

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.

Credits

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!

Presented By Allen Tang