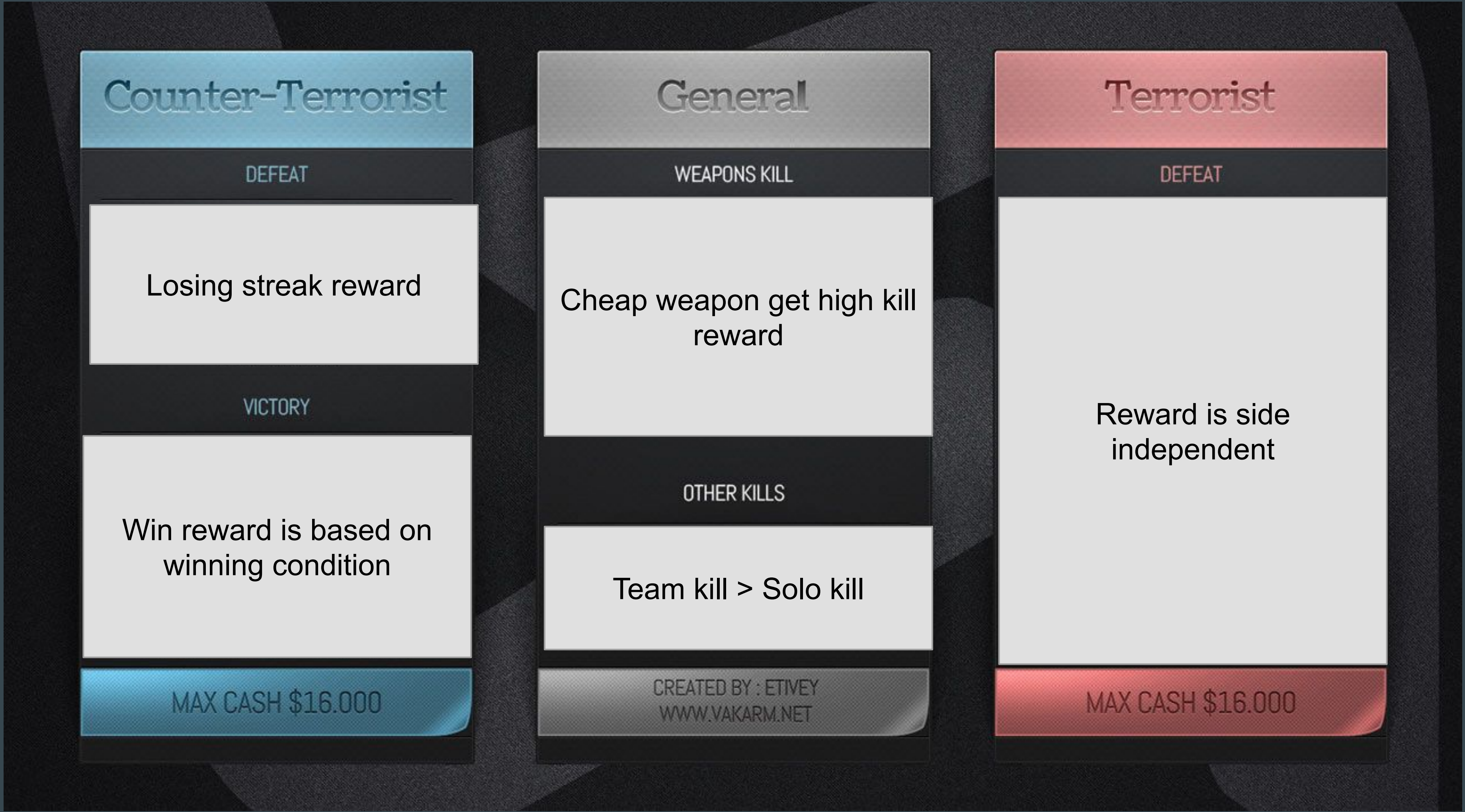
Into the CS:GO Economy Strategy

Deren Lei, Beichen Li, Yang Yang, Sijie Duan, Gangrong Jiang, Qi Luo
Mentor: Yilei Zeng

Game Side

CS:GO Information

1. 30-round game. Each team play maximum of 15 round of terrorist and counter-terrorist
2. Each team has 5 player.
3. Exists different type of weapons: Pistols, shotguns, rifles, SMG, LMG, etc
4. Other equipments: grenades, flashbang, helmet, etc
5. Players buy their weapons at the beginning of each round.



Counter-Terrorist

DEFEAT

Losing streak reward

VICTORY

Win reward is based on winning condition

MAX CASH $16.000

General

WEAPONS KILL

Cheap weapon get high kill reward

OTHER KILLS

Team kill > Solo kill

CREATED BY : ETIVEY
WWW.VAKARM.NET

Terrorist

DEFEAT

Reward is side independent

MAX CASH $16.000

Managing Economy

Basic options based on game progress:

1. **Eco**: Save as much money for future rounds
2. **Anti-Eco**: Opposition is about to perform an Eco. Win and maximize kill award

3. **Force Buying**: Poor economy, fully utilize the money.
4. **Full Buying** : Maximise their chances of winning the round

Machine Learning Side

(Technical detail alert)

Why new algorithm/model architecture

Most prior work on multi-agent reinforcement learning (MARL)

- Train in an interactive environment and receiving a sparse reward after long-term consecutive actions.

Our study

- Agent receives dense reward in a multi-round scenario and each round is dependent on each other. How to use the global round information?

Problems for training a strategy

1. **Non-static environment**: Each data has different team (different strategy), each team has different players (different skills and gun preferences). Essentially facing 10000 different tasks, not 1 task that contains 10000 data.

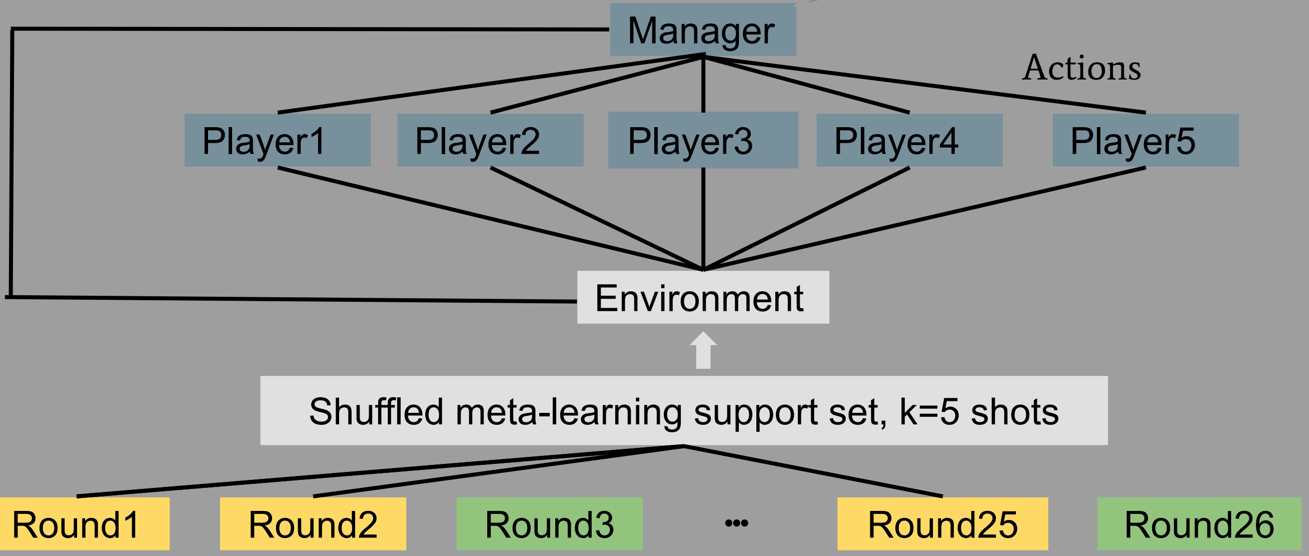
2. **Label definition**: Each player need to buy multiple weapons. How to design a model to generate such complex output?

3. **State definition**: Each game contains 30 dependent rounds. What information do the agent need to make a decision for each round?

Non-static environment solution

1. Design hierarchical model: manager-workers
2. Do few-shot learning

I'm a new coach to this team (current data). Based on my limited observations (few shots), I think player1 is our only sniper and player3 is on fire! I'll do the following assignments for the other rounds (target set).



Manager

Actions

State,
Reward

Player1

Player2

Player3

Player4

Player5

Environment

Shuffled meta-learning support set, k=5 shots

Round1

Round2

Round3

...

Round25

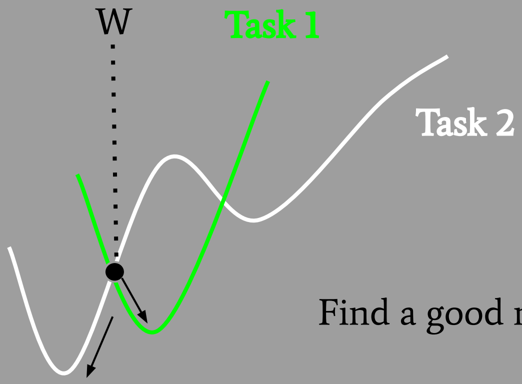
Round26

Note: each round is considered as an independent data

Our modified reptile meta-learning algorithm (model agnostic)

```
Initialize model parameters W
for meta_iteration = 1,2,...do
    Sample single data (with repetition)
    for shot = 1,2,..k do
        Sample single round and compute loss L
        W' = Adam(L, W)
    end for
    Compute target set loss L
    W' := Adam(L, W')
    Update W:= W + epsilon*(W'-W)
end for
```

Vanilla SGD to update meta model after each task



Find a good model initialization

Label definition solution

1. **Make the label sequential**. Use RNN to generate sequence of output. Since the label order doesn't matter, use precision, recall, F1 as evaluation metrics.

   Example output: <Buy AK-47>, <Buy AK-47 and drop it for teammate2>, <Buy Flashbang>, <Buy Grenade>, <End>

   Precision = correctly generated / output length

   Recall = correctly generated / label length

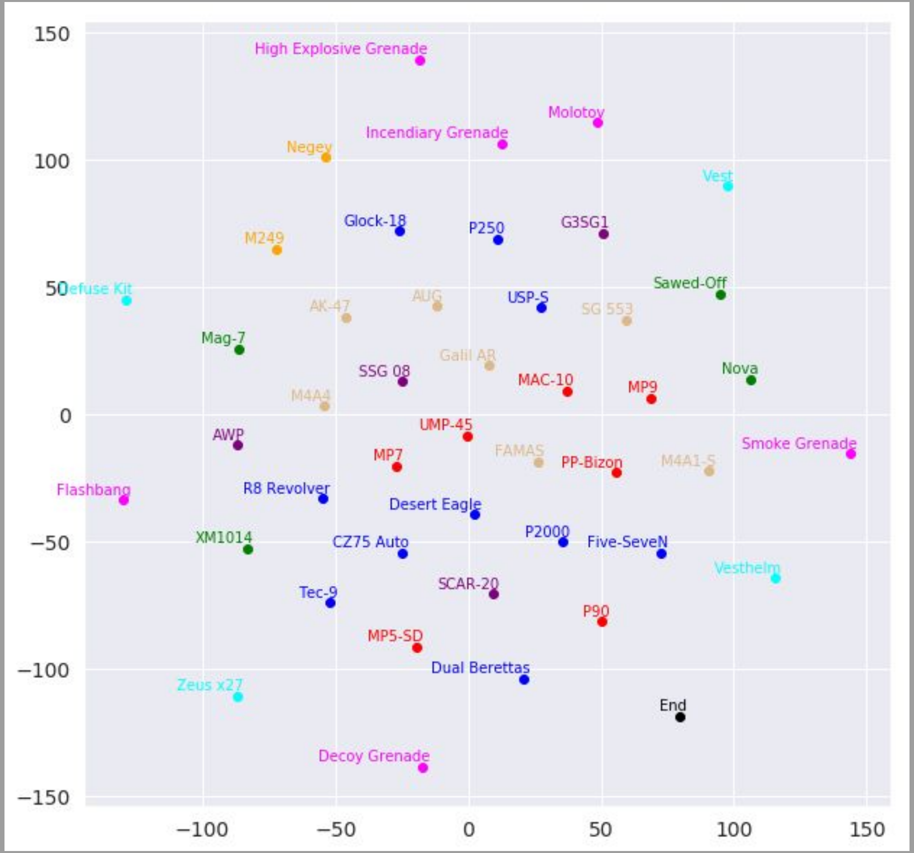
   F1 = 2 * precision * recall / (precision + recall) We use F1 as reward function

Label definition solution

2. **Action representation**: Use weapon attributes as action embedding. For end token and other equipment like helmet, append one-hot features to the embedding.

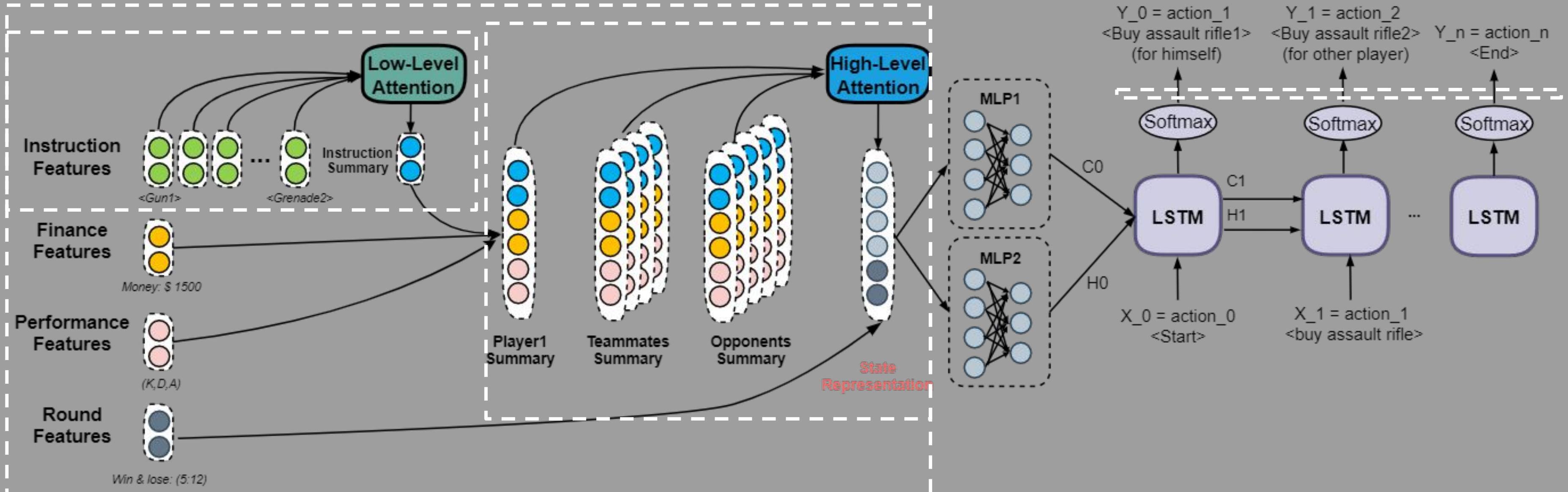
	Attribute1	Attribute2	Attribute3	Attribute4	one-hot	one-hot
Gun1	float	float	float	float	0	0
Gun2	float	float	float	float	0	0
Grenade1	float	float	float	NA	0	0
helmet	NA	NA	NA	NA	1	0
End	NA	NA	NA	NA	0	1

t-SNE visualization of action embedding



Show detail in demo...

Model Design



Intuition for hierarchical attention design:

1. Each data may have different number of instructions: use low-level attention
2. Other players are not in order. Don't know who is who: use high-level attention

Learning Objective

Common approach: Teacher forcing (Williams & Zipser, 1989),

Suffers exposure bias for long sequence

$$L_{ml} = -\sum_{t=1}^{n'} \log p(y_t^* | y_1^*, \ldots, y_{t-1}^*, x)$$

We use: Self-critical policy gradient training algorithm (Rennie et al., 2016).

```
L = (reward(sampled path) - reward(greedy path)) * log probability(sampled path)
```

Prediction time: use beam search for final generation

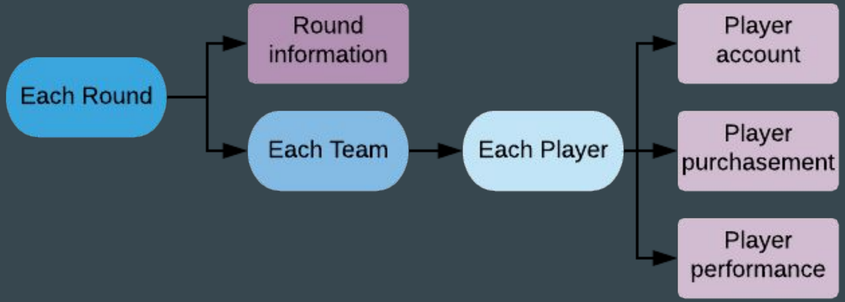
To train a model

Machine Learning is all about data!

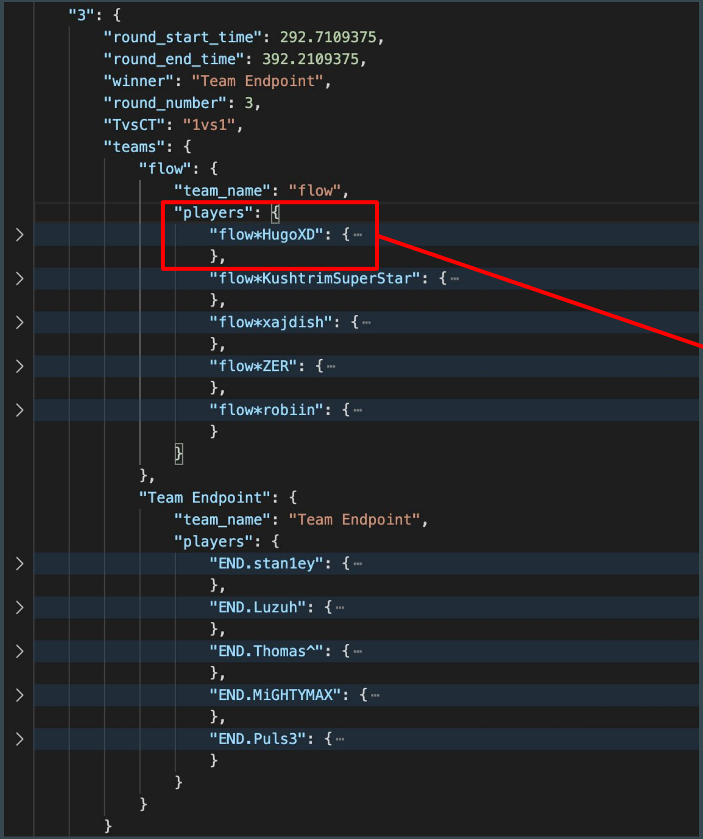
Non-benchmark data: Size, quality, attributes, feature engineering,..

Data preprocessing

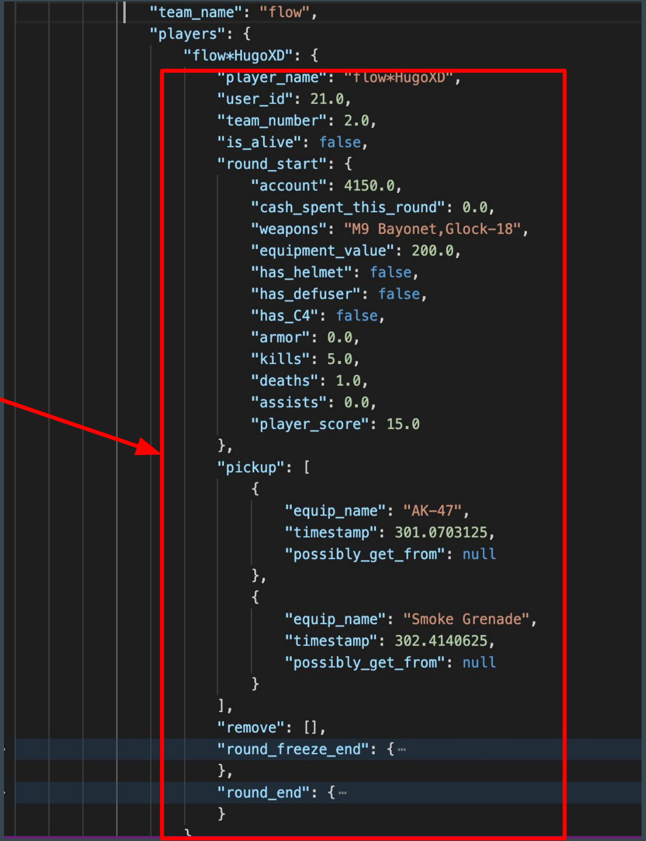
1. Raw data: 10,000 CSGO game replay .dem files.
2. Extract game events such as "round_start", "item_pickup" etc. with detailed player information and match information, and record them into .csv files by timeline.
3. Convert .csv timeline game information into organized json structure and do validation checks.



JSON Structure

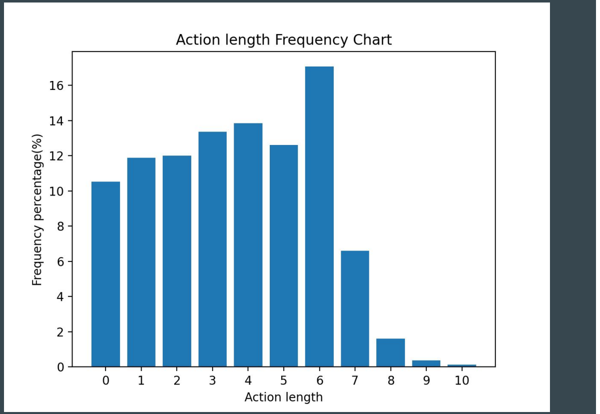


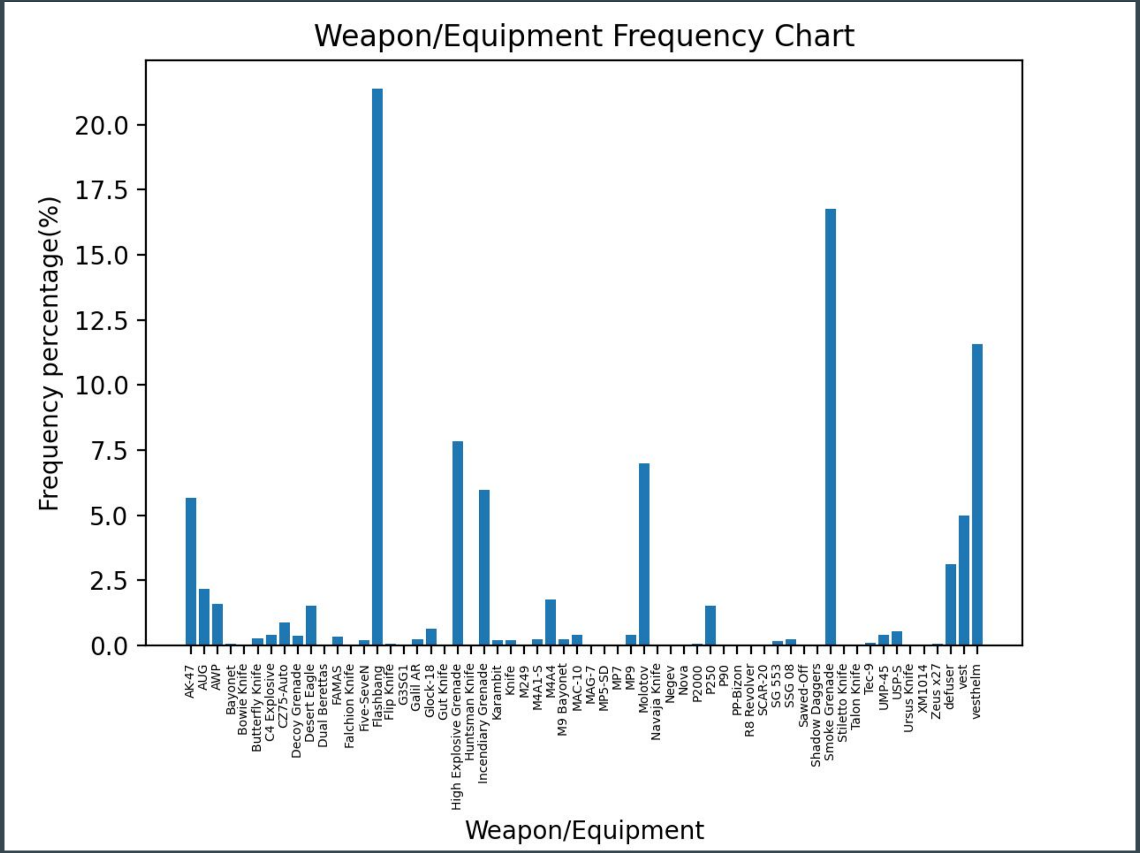
```
"3": {
    "round_start_time": 292.7109375,
    "round_end_time": 392.2109375,
    "winner": "Team Endpoint",
    "round_number": 3,
    "TvsCT": "1vs1",
    "teams": {
        "flow": {
            "team_name": "flow",
            "players": {
                "flow*HugoXD": {…
                },
                "flow*KushtrimSuperStar": {…
                },
                "flow*xajdish": {…
                },
                "flow*ZER": {…
                },
                "flow*robiin": {…
                }
            }
        },
        "Team Endpoint": {
            "team_name": "Team Endpoint",
            "players": {
                "END.stan1ey": {…
                },
                "END.Luzuh": {…
                },
                "END.Thomas^": {…
                },
                "END.MiGHTYMAX": {…
                },
                "END.Puls3": {…
                }
            }
        }
    }
```



```
"team_name": "flow",
"players": {
    "flow*HugoXD": {
        "player_name": "flow*HugoXD",
        "user_id": 21.0,
        "team_number": 2.0,
        "is_alive": false,
        "round_start": {
            "account": 4150.0,
            "cash_spent_this_round": 0.0,
            "weapons": "M9 Bayonet,Glock-18",
            "equipment_value": 200.0,
            "has_helmet": false,
            "has_defuser": false,
            "has_C4": false,
            "armor": 0.0,
            "kills": 5.0,
            "deaths": 1.0,
            "assists": 0.0,
            "player_score": 15.0
        },
        "pickup": [
            {
                "equip_name": "AK-47",
                "timestamp": 301.0703125,
                "possibly_get_from": null
            },
            {
                "equip_name": "Smoke Grenade",
                "timestamp": 302.4140625,
                "possibly_get_from": null
            }
        ],
        "remove": [],
        "round_freeze_end": {…
        },
        "round_end": {…
        }
```

Dataset visualization & analysis





Occam’s razor

Baseline: Greedy algorithm

Detail: For each round, always do full buying.

Performance

	F1
Greedy	0.260
Our RL Baseline	**0.318**
Improved novel model	TODO

Future study

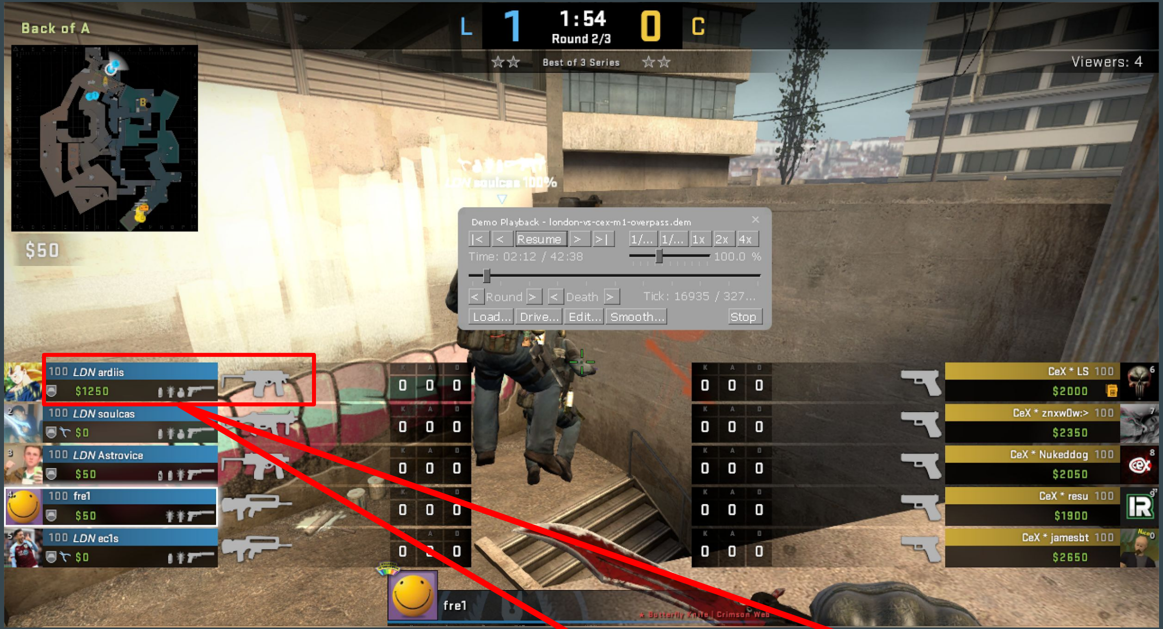
1. Improve action embedding
    a. Pre-train embedding by a variational auto-encoder
2. Switch back to complex multi-agent scenario
3. Improve decoder architecture:
    a. bi-LSTM
    b. Transformer
4. Integrate money into training policy? weight, reward

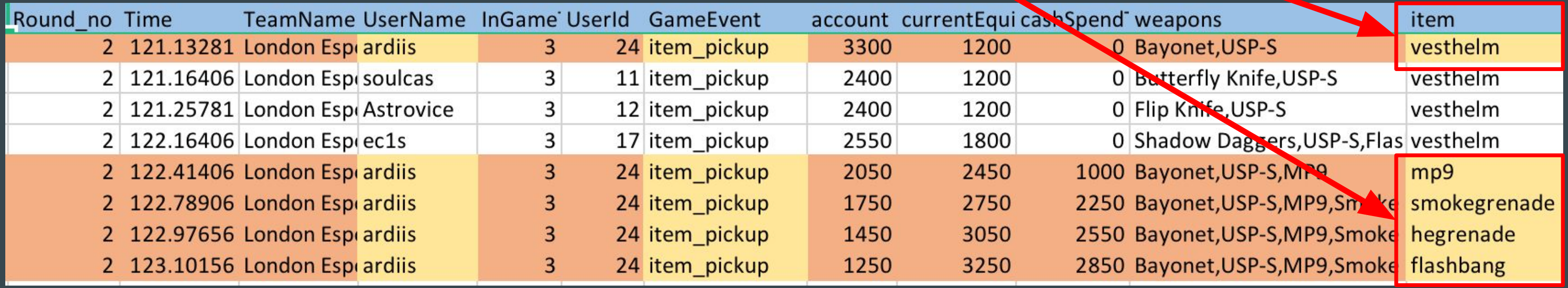
We open-sourced our model here:

https://github.com/derenlei/MAML

Demo

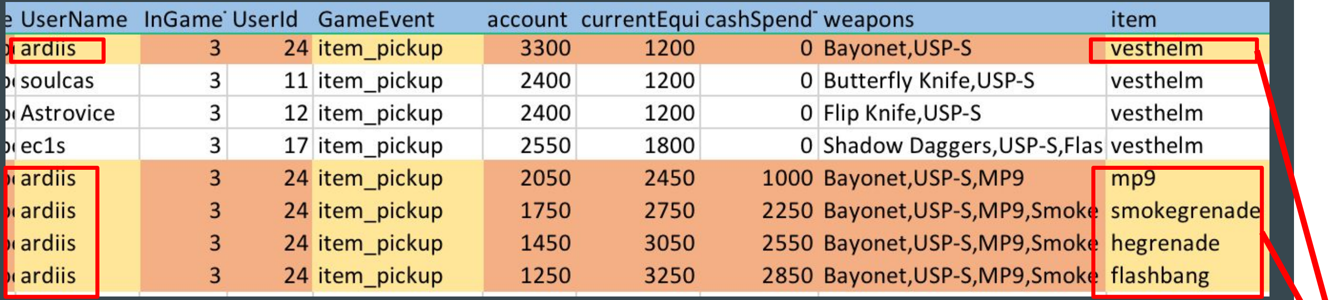
Demo file Convert to CSV Structure



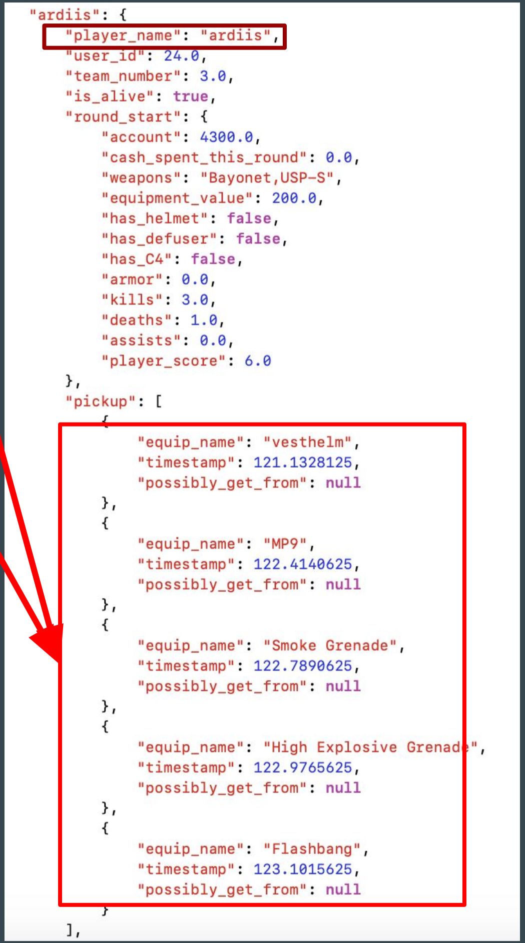


Round_no	Time	TeamName	UserName	InGame	UserId	GameEvent	account	currentEqui	cashSpend	weapons	item
2	121.13281	London Esp	ardiis	3	24	item_pickup	3300	1200	0	Bayonet,USP-S	vesthelm
2	121.16406	London Esp	soulcas	3	11	item_pickup	2400	1200	0	Butterfly Knife,USP-S	vesthelm
2	121.25781	London Esp	Astrovice	3	12	item_pickup	2400	1200	0	Flip Knife,USP-S	vesthelm
2	122.16406	London Esp	ec1s	3	17	item_pickup	2550	1800	0	Shadow Daggers,USP-S,Flas	vesthelm
2	122.41406	London Esp	ardiis	3	24	item_pickup	2050	2450	1000	Bayonet,USP-S,MP9	mp9
2	122.78906	London Esp	ardiis	3	24	item_pickup	1750	2750	2250	Bayonet,USP-S,MP9,Smoke	smokegrenade
2	122.97656	London Esp	ardiis	3	24	item_pickup	1450	3050	2550	Bayonet,USP-S,MP9,Smoke	hegrenade
2	123.10156	London Esp	ardiis	3	24	item_pickup	1250	3250	2850	Bayonet,USP-S,MP9,Smoke	flashbang

CSV data Convert to JSON structure

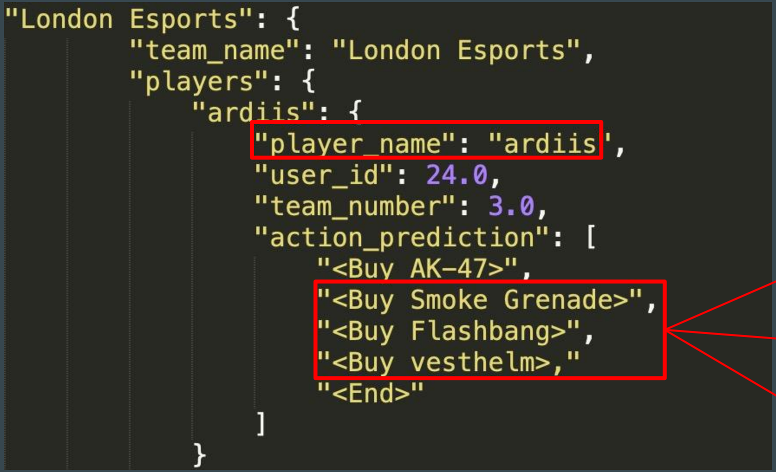


UserName	InGame	UserId	GameEvent	account	currentEqui	cashSpend	weapons	item
ardiis	3	24	item_pickup	3300	1200	0	Bayonet,USP-S	vesthelm
soulcas	3	11	item_pickup	2400	1200	0	Butterfly Knife,USP-S	vesthelm
Astrovice	3	12	item_pickup	2400	1200	0	Flip Knife,USP-S	vesthelm
ec1s	3	17	item_pickup	2550	1800	0	Shadow Daggers,USP-S,Flas	vesthelm
ardiis	3	24	item_pickup	2050	2450	1000	Bayonet,USP-S,MP9	mp9
ardiis	3	24	item_pickup	1750	2750	2250	Bayonet,USP-S,MP9,Smoke	smokegrenade
ardiis	3	24	item_pickup	1450	3050	2550	Bayonet,USP-S,MP9,Smoke	hegrenade
ardiis	3	24	item_pickup	1250	3250	2850	Bayonet,USP-S,MP9,Smoke	flashbang

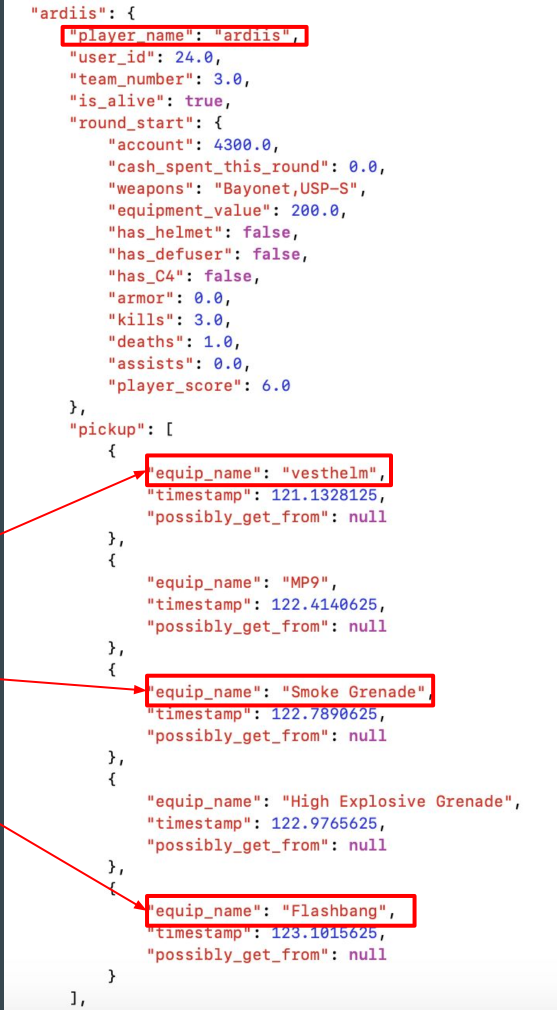


```
"ardiis": {
    "player_name": "ardiis",
    "user_id": 24.0,
    "team_number": 3.0,
    "is_alive": true,
    "round_start": {
        "account": 4300.0,
        "cash_spent_this_round": 0.0,
        "weapons": "Bayonet,USP-S",
        "equipment_value": 200.0,
        "has_helmet": false,
        "has_defuser": false,
        "has_C4": false,
        "armor": 0.0,
        "kills": 3.0,
        "deaths": 1.0,
        "assists": 0.0,
        "player_score": 6.0
    },
    "pickup": [
        {
            "equip_name": "vesthelm",
            "timestamp": 121.1328125,
            "possibly_get_from": null
        },
        {
            "equip_name": "MP9",
            "timestamp": 122.4140625,
            "possibly_get_from": null
        },
        {
            "equip_name": "Smoke Grenade",
            "timestamp": 122.7890625,
            "possibly_get_from": null
        },
        {
            "equip_name": "High Explosive Grenade",
            "timestamp": 122.9765625,
            "possibly_get_from": null
        },
        {
            "equip_name": "Flashbang",
            "timestamp": 123.1015625,
            "possibly_get_from": null
        }
    ],
```

Sample output generated by our model



```
"London Esports": {
        "team_name": "London Esports",
        "players": {
            "ardiis": {
                "player_name": "ardiis",
                "user_id": 24.0,
                "team_number": 3.0,
                "action_prediction": [
                    "<Buy AK-47>",
                    "<Buy Smoke Grenade>",
                    "<Buy Flashbang>",
                    "<Buy vesthelm>,"
                    "<End>"
                ]
            }
```



```
"ardiis": {
    "player_name": "ardiis",
    "user_id": 24.0,
    "team_number": 3.0,
    "is_alive": true,
    "round_start": {
        "account": 4300.0,
        "cash_spent_this_round": 0.0,
        "weapons": "Bayonet,USP-S",
        "equipment_value": 200.0,
        "has_helmet": false,
        "has_defuser": false,
        "has_C4": false,
        "armor": 0.0,
        "kills": 3.0,
        "deaths": 1.0,
        "assists": 0.0,
        "player_score": 6.0
    },
    "pickup": [
        {
            "equip_name": "vesthelm",
            "timestamp": 121.1328125,
            "possibly_get_from": null
        },
        {
            "equip_name": "MP9",
            "timestamp": 122.4140625,
            "possibly_get_from": null
        },
        {
            "equip_name": "Smoke Grenade",
            "timestamp": 122.7890625,
            "possibly_get_from": null
        },
        {
            "equip_name": "High Explosive Grenade",
            "timestamp": 122.9765625,
            "possibly_get_from": null
        },
        {
            "equip_name": "Flashbang",
            "timestamp": 123.1015625,
            "possibly_get_from": null
        }
    ],
```