接口说明

一、智能体开发框架介绍

该训练框架主要由客户端和服务端组成。客户端由虚拟环境、模型与环境交互类和强化学习模型/基于规则的模型组成。虚拟环境执行环境/智能体的初始化设置、将动作序列发送到仿真系统并获取态势信息，并且根据仿真系统发送的事件（单位被摧毁/推演结束/完成任务等）计算智能体奖励等。模型与环境交互类将虚拟环境接收的态势信息作为输入，并且使用学习模型为智能体生成动作。此外，模型与环境交互类还执行模型训练、保存和加载任务。

服务端根据客户端初始化信息在场景中创建红蓝双方智能体（比如士兵、坦克和无人机等）并开始城市对战推演，发送实时态势信息到客户端虚拟环境。此外，服务端具有实时态势显示功能。

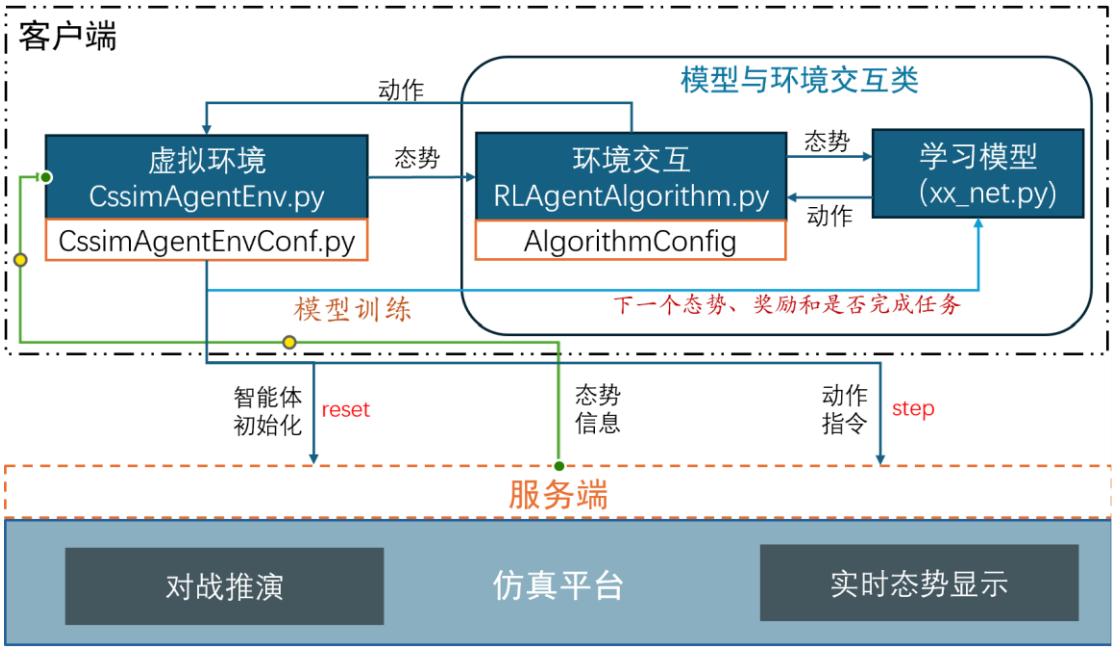

图 1. 训练框架示意图

二、项目目录结构

项目解决方案包括算法、任务配置、训练日志和工具类文件夹。

算法目录（Algorithm）：存放用户设计的决策模型、模型训练和模型、环境交互类和虚拟环境类（比赛用户需要在虚拟环境类中设计自己的奖励机制）。每个比赛用户在该目录下单独创建一个文件夹保存上述类文件。

任务配置目录（TaskConfig）：保存用户的全局配置文件（jsonc）。该配置文件中包含全局配置、算法配置和场景配置信息。

训练日志目录（TrainLogs）：保存训练好的模型及模型备份。程序启动后，会默认在该目录中加载保存的模型。

工具类目录（Utils）：保存了深度学习张量操作相关的类，比如自动根据全局配置文件，将张量放置的对应的训练设备上。

三、全局配置文件

该配置文件由全局配置（GlobalConfig）、场景配置（ScenarioConfig）和算法配置（AlgorithmConfig）三部分组成。

全局配置主要包括训练设备、线程/进程数量、最大训练次数和模型保存相关的配置信息。

表 1. 全局配置

序号	配置项	功能
1	team_name	保存模型文件夹名称，建议命名为队伍名称
2	num_worker_threads	线程数量，多个线程同时进行推演
3	test_start_interval	测试间隔，每模拟多少次测试一次
4	test_epoch	测试次数，测试结果为其平均值
5	seed	随机数种子
6	device	设置训练设备 cpu 或 cuda，默认为 cpu
7	max_episode	最大模拟次数
8	AlgorithmCofig	算法配置类路径->类名称

场景配置中常用的配置项有：最大模拟步长、使用计算方式或渲染方式进行模拟以及各个队伍使用的决策算法。

表 2. 场景配置

序号	配置项	功能
1	MaxStepsPerEpisode	每次模拟中智能体与环境交互的最大次数
2	render	是否渲染，开启后自动打开可视化界面
3	Enable_Reward_Sharing	每个队伍的智能体共享集体奖励（True），或者每个队伍的智能体都独享个体奖励（False）

4	SubTaskSelection	配置用户设计的虚拟环境类的路径
5	CssimServerPath	测试次数，测试结果为其平均值
6	TimeDilation	推演倍速
7	Algorithm_Names	配置红蓝双方决策算法 (算法路径->算法名称)。蓝方配置为预留，目前没有用到。

算法配置完成决策算法相关的配置，主要为模型训练/加载和用户设计的算法相关的参数设置。其余配置参数的含义见案例代码中的注释。

表 3. 算法配置

序号	配置项	功能
1	num_training_trajectories	训练所需轨迹数量(模拟一次为一条轨迹)
2	use_batch_norm	是否使用批归一化
3	load_trained_model	是否在模拟开始前载入保存的模型
4	gamma	折扣因子
5	reward_forwarding	是否开启奖励前移
6	reward_forwarding_gamma	奖励前移折扣因子
7	memory_safety_check	开启批量训练，防止显存溢出
8	learning_rate	学习率
9	train_epoch	每次训练迭代的轮数
10	skip_test	是否忽略测试环节

四、决策模型和虚拟环境

本节介绍模型与环境交互模块 RLAgentAlgorithm.py 和虚拟环境模块 CssimAgentEnv.py。模型与环境交互模块包含两个类，分别是算法配置类 AlgorithmConfig 和智能算法类 RLAgentAlgorithm。虚拟环境类主要实现用户兹定于的奖励机制设计。主要功能说明如下，详细参数含义见代码注释。

1、算法配置

AlgorithmConfig 为用户设计的决策算法和模型训练相关参数的配置类。用户可根据实际需要自行添加配置项，并在全局配置文件的 AlgorithmConfig 部分添加对应的配置项并对参数进行设置。

表 4 . 类 AlgorithmConfig 及其方法

参数	类型	说明
gamma	float	折扣因子

tau	float	使用 GAE 处理奖励值时使用的奖励平滑参数
num_training_trajectories	int	训练所需轨迹数量
use_batch_norm	bool	是否使用归一化
load_trained_model	bool	是否读取保存的模型
specific_model_path	string	用户自定义的读取模型路径
train_epoch	int	训练次数
num_batches	int	将训练集分为几批
max_grad_norm	float	最大梯度范数
learning_rate	float	学习率
memory_safety_check	bool	是否避免批量过大导致显存溢出
reward_forwarding	bool	是否奖励前移
reward_forwarding_gamma	float	奖励前移折扣因子
hidden_dim	float	网络隐藏层维度
num_agent	int	设置队伍中智能体的数量
action_filter_enabled	bool	默认为 True，是否过滤模型生成的动作
skip_test	bool	是否忽略模型测试过程

2、决策方法

RLAgentAlgorithm 模块是模型与环境的交互类。该模块从环境接收态势信息并使用决策模型生成动作。此外，该模块完成模型加载、训练、保存和训练轨迹保存功能。详细说明见案例代码注释。

表 5 . 类 RLAgentAlgorithm 及其方法

函数	功能	参数说明		
		参数	类型	说明
init()	构造类实例	num_agent	int	当前队伍智能体数量
		num_thread	int	线程数量
		space	dictionary	动作空间维度以及观测空间维度
		team	int	当前队伍编号

making_decision()	生成动作序列，返回动作列表和当前状态	state	dictionary	包含观测空间信息、可用动作信息以及智能体相关信息等
		test_mode	bool	是否测试
train()	训练模型	/	/	/
save_model()	保存模型参数及其它相关信息	update_count	int	模型更新次数
		info	dictionary	需要记录的其它信息，如胜利队伍和胜率等
load_model()	加载模型	/	/	从算法配置中获取模型保存路径

3、智能体环境

本节介绍多智能体训练环境文件 CssimAgentEnv.py。该类从仿真环境中获取态势信息，并使用智能体模型生成的动作序列控制仿真环境中的作战单元执行动作。主要功能说明如下，详细参数含义见代码中注释。

表 6 . 类 CssimAgentEnv 及其方法

函数	功能	参数说明		
		参数	类型	说明
init()	初始化智能体训练环境实例	thread	int	线程编号（自动设置）
reset()	智能体训练开始前重置环境	/	/	无需用户添加其他执行逻辑
step()	1.将智能体模型传来的动作序列转换为作战单元实际执行的动作指令； 2.将动作指令发送给仿真系统并执行相应的动作； 3.执行完动作后，更新环境状态；	act	array	模型生成的动作（无需用户添加其他执行逻辑）

	4.返回下一个状态、执行完动作后的奖励，本轮任务是否完成等信息。			
reward_and_done()	设计智能体的奖励。可设置：每个队伍中所有智能体共享一个奖励值或为每个智能体设置单独的奖励	state	字典	包含全部智能体的状态信息和每个队伍的信息（用户需设计自己的奖励机制）

五、模型训练

本节介绍模型训练文件 train.py，包括构建轨迹池、计算模型损失以及模型训练等功能，主要功能说明如下，详细参数含义见代码中注释。

表 7 . 类 Trainer 及其方法

函数	功能	参数说明		
		参数	类型	说明
init()	初始化模型训练实例	model	类	需要训练的模型
		config	类	配置信息
train_model_on_trajectories()	创建多 GPU 训练环境	trajectories	类	轨迹池
		task	string	训练标志
learn_from_trajectories()	在轨迹池中采样数据并执行模型训练	trajectories	类	轨迹池
		task	string	训练标志
record_training_metrics	记录训练结果	dictionary	dictionary	需要记录的训练结果
print_training_summary()	训练结果处理	output	bool	是否在终端输出训练结果
run_training()	从采样的批数据中获取模型训练数据，执行决策模型中的训练函	flag	string	训练标志
		sample	dictionary	模型训练所需的输入信息，

	数，输出训练损失。			如当前状态、奖励、动作等

六、动作集合

动作指令分为警戒、移动和攻击三种类型。每个动作指令的格式为：（主动作指令，子动作指令，x，y，z，uid，team_id，unit_index）。警戒动作执行静态警戒，即保持静止状态。移动指令实现了按指定方向移动（子动作指令指定移动方向）或移动到指定位置（子命令为‘N/A’，设置动作指令的 x,y 和 z 参数的值）。攻击动作包括普通攻击和手雷攻击。手雷攻击的动作需要设置投掷手雷的方向。普通攻击动作的数量等于敌方单位的数量，即每条普通攻击指令通过设置攻击对象的 team_id（队伍编号）和 unit_index（队伍中智能体的编号）来指定该条命令攻击的敌方单位。

表 8 . 智能体动作集合

动作编号	主动作指令（string）	子动作指令（string）	功能
0	N/A	N/A	空动作
1	Alert	DynamicAlert	动态警戒
2	Moving	+X	+X 方向移动
3		+Y	+Y 方向移动
4		-X	-X 方向移动
5		-Y	-Y 方向移动
6		+X+Y	+X+Y 方向移动
7		-X+Y	-X+Y 方向移动
8		-X-Y	-X-Y 方向移动
9		+X-Y	+X-Y 方向移动

10	GrenadeAttacking	+X	+X 方向投掷手雷
11		+Y	+Y 方向投掷手雷
12		-X	-X 方向投掷手雷
13		-Y	-Y 方向投掷手雷
14		+X+Y	+X+Y 方向投掷手雷
15		-X+Y	-X+Y 方向投掷手雷
16		-X-Y	-X-Y 方向投掷手雷
17		+X-Y	+X-Y 方向投掷手雷
…	NormalAttacking	‘N/A’	攻击指定编号的敌方单位 （每个敌方单位对应一条该攻击命令）

对于基于模型的决策方法，智能体的作动由模型生成，并按照上述动作编号编码为具体执行的动作指令，因此无需用户为智能体指定具体的动作。对于基于规则的决策方法，用户需自行为每个智能体指定具体执行的动作。

七、提交文件

各参赛队伍在 Algorithm 文件夹中创建队伍名称命名的子文件夹，提交的文件都需要保存在该文件夹中。提交的文件如下：

1、 决策方法类文件（.py）：该文件为用户设计的决策模型，生成动作指令。案例代码中对应 attention_net 文件夹中的 Attention_Net.py

2、 模型与环境交互类文件（.py）：该文件接受态势信息输出决策方法生成的动作序列、保存推演轨迹、执行模型加载、训练和保存等功能。对应案例文件中的 RLAgentAlgorithm.py。

3、 虚拟环境类文件（.py）:该文件包含虚拟环境类。对应案例代码中的 CssimAgentEnv.py。

4、 模型训练类（.py）:该文件实现模型训练功能，包括训练好的智能体模型。对应案例代码中的 train.py。

5、 算法说明文档。