

## 中国指挥与控制学会团体标准

T/CICC 31006—2025

### 智能兵棋推演系统设计要求

Requirements for the design of an intelligent wargame system

2025-11-27 发布

2025-11-27 实施

中国指挥与控制学会 发布



## 目 次

前言 .....	III
1 范围 .....	1
2 规范性引用文件 .....	1
3 术语和定义 .....	1
4 智能兵棋推演系统总体架构 .....	3
4.1 总体架构 .....	3
4.2 业务逻辑层（服务层） .....	4
4.2.1 对抗推演引擎 .....	4
4.2.2 业务管理 .....	4
4.3 应用层（客户端） .....	5
4.3.1 管理员及导演客户端 .....	5
4.3.2 观摩客户端 .....	5
4.3.3 推演交互客户端 .....	5
4.3.4 兵棋智能体框架（兵棋智能体客户端） .....	5
4.3.5 复盘观看客户端 .....	6
4.4 数据层 .....	6
4.4.1 决策总线 .....	6
4.4.2 复盘数据存储、回放 .....	7
5 智能兵棋推演系统接口 .....	7
5.1 推演环境接口 .....	7
5.1.1 用户授权验证接口 .....	7
5.1.2 数据回传系列接口 .....	7
5.2 兵棋训练环境接口 .....	7
5.3 兵棋智能体接口 .....	8
5.4 兵棋智能体开发SDK .....	8
5.5 用户UI规范 .....	9
5.5.1 算子和编成区 .....	9
5.5.2 态势区 .....	9
5.5.3 数据区 .....	9
5.5.4 指令下达区 .....	9
6 智能兵棋推演系统主要流程 .....	9
6.1 智能体本地高速训练流程 .....	9
6.2 网络对抗流程 .....	11
附录A 示例 .....	13
A.1 推演环境接口相关示例 .....	13
A.1.1 用户授权验证接口示例 .....	13
A.1.2 数据回传系列接口示例 .....	14
A.2 兵棋训练环境接口相关示例 .....	15

A.3 兵棋智能体接口相关示例 ..... 17

A.4 兵棋智能体开发SDK示例 ..... 19

全国团体标准信息平台

## 前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文充分考虑通用智能体、大模型等技术的近期发展趋势，通过扩展接口、优化决策总线架构、完善开发工具链等方式，为未来异构智能实体的接入预留适配空间，确保系统具备技术前瞻性与扩展性。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中国科学院自动化研究所提出。由中国指挥与控制学会归口。

本文件起草参与单位：

组长单位：中国科学院自动化研究所；

副组长单位：南开大学人工智能学院、北京理工大学机电学院；

成员单位：中国电子科技集团有限公司电子科学研究院、中国人民解放军国防科技大学试验训练基地、紫光恒越技术有限公司。

本文件主要起草人：倪晚成、于海涛、黄凯奇、张雪波、赵铭慧、杨成伟、杨宇、凌艳香、支烽耀、杨俊强、林晖、郭庆浪、刘建勋、李伟、许晓龙、何进、刘凤鸣、汤再江、徐享忠、杨海波

（注：排名不分先后）。







# 智能兵棋推演系统设计要求

## 1 范围

本文件规定了智能兵棋推演系统设计要求，明确了智能兵棋推演系统的总体架构、智能兵棋接口、智能兵棋推演分布式训练及业务服务流程。通过制定智能决策总线架构和统一智能体开发和接入接口，达成异构系统间的互操作，为通用智能体的研发及多环境协同推演提供支撑

本文件适用于智能兵棋推演系统在国防教育、应急救援演练、商业博弈模拟等场景的设计、开发、测试及部署，同时为通用智能体、大模型等新兴智能实体的接入提供标准化支撑。

## 2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

- GB/T 5271.6—2006 数据处理词汇
- T/CICC 31001—2025 兵棋推演名词术语
- T/CICC 31008—2025 智能兵棋推演算法模型规范

## 3 术语和定义

下列术语和定义适用于本文件。

### 3.1

#### 智能兵棋推演系统 **intelligent wargame system**

智能兵棋推演系统是指符合人工智能博弈框架的计算机兵棋系统，其在计算机兵棋推演基础功能上具备智能体开放接口（包含人机，机机，人机混合），支持第三方智能体接入并实现态势信息与推演指令的交互以实施在线推演，同时可为学习型智能体提供训练环境并支持高速推演数据生成与模型训练。

### 3.2

#### 兵棋智能体 **wargaming agent / wargaming AI**

兵棋智能体是指能在兵棋推演中自主感知环境、判断态势、作出决策并执行行动的计算机程序，其与智能兵棋推演系统集成运行并进行信息交互，且运行和决策具备自主性。

### 3.3

#### 兵棋推演环境 **wargame ENV**

兵棋推演环境是指依据推演方（含推演人员、兵棋 AI、导演人员等）的输入，按照兵棋规则的推进流程、交互逻辑及裁决计算等，模拟生成战争情境的计算机程序。

### 3.4

#### 兵棋训练环境 **wargame train ENV**

由总线控制器通过 mGBC 消息授权具备总线调度能力的 RT 终端。

### 3.5

#### 接口 **interface**

接口是指在兵棋智能体、兵棋推演环境等不同模块间实现数据交换的通道及其数据内容与形式。

### 3.6

#### 通用智能体接口 **general agent interface**

用智能体接口指用于实现通用智能体（具备跨场景决策能力的智能实体）与智能兵棋推演系统集成的标准化接口，支持多模态输入（态势文本描述、图像化态势等）与灵活输出（结构化指令、自然语言指令转换），兼容通用智能体的异步交互特性。

### 3.7

#### 大模型接口适配 **open-source LLM interface adaptation**

大模型接口是指为大模型（如通用大语言模型、多模态大模型）接入智能兵棋推演系统提供的适配层机制，包括数据格式转换（态势数据→模型输入格式）、指令解析（模型输出自然语言→结构化动作指令）、性能优化（批量推理、缓存机制）等功能。

### 3.8

#### 兵棋智能体接口 **wargaming AI interface**

兵棋智能体接口是指用于实现兵棋智能体、通用智能体、大模型等与智能兵棋推演系统集成及推演训练交互的接口，包含基础交互规范及扩展适配层，支持同步/异步交互、单步/批量决策等模式。

### 3.9

#### 总线 **bus**

总线是指系统中标准化、模块化的信息交互与调度协调机制，是各功能模块间实现数据传输与通信的中枢架构。

[来源：GB/T 5271.6—2006, 3.12]

### 3.10

#### 决策总线 **decision-making bus**

决策总线是指智能兵棋推演系统中实现推演人员、智能体、兵棋推演环境等模块信息交互并支持智能体训练的标准化、模块化中枢信息基础架构，其分为消息总线 and 数据总线，前者用于完成实时信息交互，后者主要用于完成持久化数据传输。

### 3.11

#### 调度 **scheduling**

调度是指智能兵棋推演系统中管理智能体与模拟战争环境之间交互流程和资源分配的控制机制。

### 3.12

#### 软件开发工具包 SDK **software development kit**

软件开发工具包 SDK 指为开发者提供编写应用程序所需各类资源与工具的集合，本规范中特指为开发兵棋智能体提供的资源与工具，包含智能体应遵循的开发环境及接口。

### 3.13

#### JSON

JSON 是指一种轻量级的数据交换格式，以纯文本形式存储和传输数据，具有人类可读、机器易解析的特点，广泛应用于 Web 开发、API 数据交互、配置文件等场景。

### 3.14

#### 数据字典 **dict**

数据字典是指一种无序、可变且可哈希键的集合，以键值对（key-value）形式存储数据，其中每个键必须唯一，值可以是数字、字符串、列表、元组或另一个字典等任意数据类型。

### 3.15

#### 想定 **scenario**

想定是指包含多种战场场景和任务类型的数据库。

### 3.16

#### 态势 situation

态势是指在一定时间和空间范围内，涉及各方力量对比、环境条件、行动意图等因素的综合数据。

### 3.17

#### 动作 action

动作是指智能体在推演过程中根据当前态势采取的具体行动或决策，用于影响推演进程和结果。

### 3.18

#### 指令 command

指令是指由操作者或智能体发出，用于指导兵棋推演中角色或单位执行特定行动的命令。

## 4 智能兵棋推演系统总体架构

### 4.1 总体架构

智能兵棋推演系统总体架构采用应用层、业务逻辑层、数据服务层的分层架构，具体对应关系：

应用层：包括管理员及导演客户端、观摩客户端、推演交互客户端、复盘观看客户端和兵棋智能体框架；

业务逻辑层：包括服务端的对抗推演引擎（含推演房间、调度模块）、业务管理；

数据服务层：包括应用层和业务逻辑层之间的决策总线（消息总线+数据总线）实现状态信息与数据信息的感知和传输，以及复盘数据存储模块。

智能兵棋推演系统总体架构如图 1 所示。

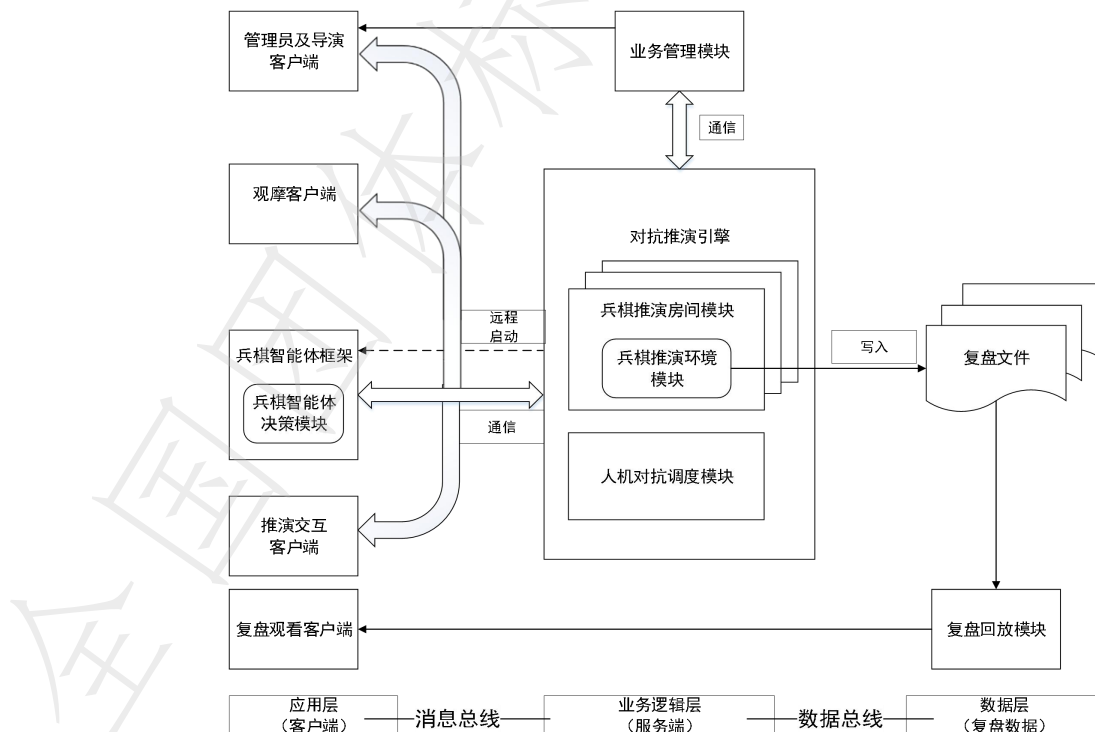


图 1 智能兵棋推演系统总体架构

## 4.2 业务逻辑层（服务层）

### 4.2.1 对抗推演引擎

对抗推演引擎程序负责调度系统资源、控制兵棋推演环境的启动与停止，并完成推演数据的持久化记录，其由兵棋推演房间模块、推演环境模块、人机对抗调度模块构成。

#### a) 兵棋推演环境模块

兵棋推演环境模块实现某一兵棋推演棋盘、算子、规则的计算机程序化并提供环境初始化、启动、停止、推进控制接口及复盘数据记录等功能，包括：

- 1) 兵棋规则实现：实现兵棋推演规则并判断指令动作能否执行，再根据可执行指令执行规则计算以获得新态势；
- 2) 单步推进：在推演过程中按推进时间步长，基于所有推演参与方的指令 / 动作调用兵棋规则计算并更新态势信息；
- 3) 态势初始化：在推演开始前依据推演参数初始化态势信息，包括加载想定数据、配置内部状态等；
- 4) 复盘记录：同步单步推进过程持续记录推演中的所有态势和指令动作以形成复盘数据；
- 5) 状态重置：在推演结束后清空内部状态，为下一场推演做准备。

#### b) 兵棋推演房间模块

兵棋推演房间模块创建并封装兵棋推演环境，处理推演中各方通信、流程控制及异常管理等系统业务，从兵棋推演环境获取复盘数据并形成持久化存储，且单个推演房间仅创建并管理一个兵棋推演环境，其服务器端可并行运行多个推演房间进程以实现并发推演，其应包含以下功能：

- 1) 创建、控制、停止兵棋推演环境：创建并控制兵棋推演环境的启动与停止；
- 2) 通信建立：与参与推演各方建立可靠通信方式以确保数据实时准确传输；
- 3) 时间控制：推动推演时间步进并按预设规则控制时间节奏；
- 4) 态势信息广播：广播推演环境模块生成的态势信息以确保所有参与者同步知晓当前态势；
- 5) 决策收集：收集参与方决策产生的动作集合作为推演环境模块单步推进的参数；
- 6) 异常处理：在推演环境出现异常时执行相应处置逻辑以保障推演的连续性与稳定性；
- 7) 信息流向控制：控制态势信息流向以实现红、蓝、导演各方分别获取各自态势并防止信息泄露；
- 8) 推演方身份认证：标记认证推演方身份以防止非法控制。

#### c) 人机对抗调度模块

人机对抗调度模块负责调度推演资源以确保推演顺利进行，其应包含以下功能：

- 1) 资源分配：依据推演需求合理分配计算资源、存储资源等推演资源；
- 2) 流程控制：控制推演各阶段流程以确保按既定规则进行；
- 3) 状态监控：实时监控推演状态并及时发现处理异常情况；
- 4) 结果统计：收集推演结果并进行统计分析以为后续推演提供参考；
- 5) 日志记录：记录推演过程中的关键信息以便复盘和分析。

### 4.2.2 业务管理

业务管理负责支撑推演的用户身份、权限、席位和业务流程管理，其应包含以下功能：

- a) 用户管理：管理用户信息，包括用户注册、登录、权限分配等；
- b) 推演配置：配置推演规则、态势设置、推演时间等推演参数；
- c) 权限控制：依据用户角色和权限控制其对推演系统的访问与操作；
- d) 日志管理：记录业务操作日志以便审计和问题追踪；

- e) 数据备份与恢复：定期备份业务数据以确保数据安全。

#### 4.3 应用层（客户端）

##### 4.3.1 管理员及导演客户端

管理员及导演客户端是提供给推演管理者使用的操作界面，允许管理者对推演过程进行监控、控制和配置，其应包含以下功能：

- a) 推演监控：实时监控推演进程，包括态势变化、参与者动作等；
- b) 推演控制：启动、暂停、结束推演及调整推演参数；
- c) 态势配置：设置包含地图、兵力部署等的初始态势；
- d) 参与者管理：添加、移除或调整推演参与者；
- e) 权限管理：分配和管理参与者权限；
- f) 日志查看：查看包括态势变化、参与者动作等的推演日志。

##### 4.3.2 观摩客户端

观摩客户端是在兵棋推演过程中为观摩者提供实时观察推演过程并从不同视角获取态势信息的操作界面，其应包含以下功能：

- a) 态势信息获取：通过网络接口实时获取推演态势信息；
- b) 态势可视化：将态势信息解析为3D模型和图像并以可视化形式展示；
- c) 视角切换：提供预设视角切换功能以便观摩者从不同角度观察推演；
- d) 信息展示：展示兵力状态、动作记录等推演相关详细信息；
- e) 倍速播放：支持推演过程的倍速播放以便快速回顾；
- f) 进度条调整：允许观摩者调整推演进度并定位到特定时刻观察。

##### 4.3.3 推演交互客户端

推演交互客户端是为推演人员提供的用于接收态势信息、发送动作指令及进行人机交互的图形化操作界面核心模块，其应包含以下功能：

- a) 态势信息可视化：将兵棋推演房间模块发送的态势信息以图形化形式展示给用户；
- b) 动作指令发送：允许用户通过鼠标点击、键盘输入等方式向兵棋推演房间模块发送动作指令；
- c) 人机交互：提供支持用户与系统交互的用户界面；
- d) 实时监控：实时显示推演过程中的态势变化和动作执行情况。

##### 4.3.4 兵棋智能体框架（兵棋智能体客户端）

兵棋智能体客户端是用于替换兵棋智能体决策模块的兵棋智能体框架，该客户端用于对接第三方智能体代码，实现智能体接入兵棋推演环境实施机机对抗推演、人机对抗推演甚至人机混合编组对抗推演，其框架代码完成通信、流程控制以及异常处理，具体用于将兵棋推演房间模块发送的态势信息传入兵棋智能体决策模块，并将兵棋智能体决策模块生成的动作集合发送至兵棋推演房间模块，其应包含以下功能：

- a) 接入推演房间：提供接入正确的推演房间功能，接收本方态势信息，并执行智能体模块中的单步决策算法；
- b) 决策函数调用：在训练过程中主进程调用智能体的决策函数，在网络对抗中收到新态势后调用决策函数；
- c) 流程控制与异常处理：控制单场推演中智能体的执行流程，集成异常处理机制，捕获决策函数中的错误，维持智能体运行；
- d) 动作集合生成与发送：生成动作集合并向推演房间发送，等待下一轮态势信息，循环执行直至推演结束；

- e) 集成与兼容性：通过定义标准接口使不同智能体无需修改即可接入平台，实现异构智能体的开放接入；
- f) 生命周期管理：对智能体运行的生命周期进行管控，确保智能体在推演过程中的稳定性和可靠性。
- g) 通用智能体接口：支持通用智能体的接入协议（如 RESTful、gRPC），提供态势数据标准化转换模块（将兵棋结构化态势转换为通用智能体可理解的文本 / 多模态格式）。
- h) 大模型接口适配：集成大模型交互适配层，支持大模型的异步决策调用（适应大模型推理延迟特性），提供自然语言指令到兵棋动作指令的解析引擎。
- i) 多模型协同接口：支持多智能体（含通用智能体、大模型）的协同决策接入，提供决策结果融合与冲突消解机制。

#### 4.3.5 复盘观看客户端

复盘观看客户端是用于可视化回放复盘文件并可扩展推演数据统计与分析功能的工具，其应包含以下功能：

- a) 回放引擎：负责播放推演过程中的态势信息、动作指令等数据；
- b) 态势可视化模块：将推演数据转换为可视化态势图和 3D 模型；
- c) 播放控制：提供包含暂停、播放拖动、时间跳转和交互式查看界面数据等功能的用户交互界面。

### 4.4 数据层

#### 4.4.1 决策总线

决策总线由消息总线 and 数据总线组成，借助高效通信机制与资源管理技术实现多智能体、多环境、多任务的协同调度，以保障高并发场景下的实时性与稳定性。

智能兵棋推演系统通过消息总线 and 数据总线作为连接异构对抗环境（仿真进程）与不同决策智能体（含 AI 和人）的通道，形成开放式系统整体架构。

##### a) 消息总线

消息总线是智能兵棋推演系统中负责实时信息传递、分发与路由并确保对抗推演过程中系统信息准确及时传递的核心组件，其主要功能应包括：

- 1) 消息传递：负责不同系统组件间的消息传递，支持多 Agent 系统中的数据交换，实现人 / 机多智能体的接入与协同工作；
- 2) 消息分发：按特定路由规则将消息分发至目标系统或组件；
- 3) 消息路由：确保消息按预定路径到达目的地；
- 4) 扩展性：在多智能体和多仿真平台环境下保障系统的灵活扩展与适应；
- 5) 支持动态消息格式适配：兼容 JSON、Protocol Buffers 等数据格式，满足大模型对高效二进制数据传输的需求；
- 6) 新增批量消息处理机制：支持通用智能体批量态势请求的并发处理。

##### b) 数据总线

数据总线是智能兵棋推演系统中负责持久化数据传输、同步与共享并支撑系统正常运行及数据驱动决策过程的重要基础设施，其主要功能应包括：

- 1) 连接对抗推演引擎：实现与对抗推演引擎的数据交互；
- 2) 支持多 Agent 系统中的数据交换：实现人 / 机智能体接入与协同工作；
- 3) 为 AI 训练计算框架提供数据支持：确保 AI 模型获取训练和推理所需的数据；

- 4) 扩展持久化数据类型：支持存储大模型训练所需的历史态势序列、多模态复盘数据，提供高效的批量数据查询接口；
- 5) 集成数据压缩与分片传输：针对大模型输入的大规模历史数据，提供自动分片与压缩功能，降低传输带宽占用。

#### 4.4.2 复盘数据存储、回放

##### 4.4.2.1 复盘数据

智能兵棋推演系统中的复盘数据，指特定推演实例执行过程中产生的、用于完整记录推演过程并支持精确重现的结构化信息集合，其中智能体数据为核心要素。其核心内容包括：

- a) 初始态势数据（战场环境配置、参演实体部署、规则集及胜利条件设定）；
- b) 动态推演过程数据（各智能体及其他参演方的指令、系统裁决事件及结果、智能体与其他实体的状态演化、环境变更、关键交互事件及其精确时间戳）；
- c) 必要的元数据（推演标识、智能体属性信息、参与者信息、时间范围、最终结果）。

##### 4.4.2.2 复盘数据存储

复盘数据存储需以人类操作数据集与智能体数据为双核心，采用可靠高效的结构化方案，确保两类数据及关联信息的持久化、安全性与访问效率。

数据内容组织需系统区分：

- a) 基础静态数据（含人类操作员初始配置、智能体初始属性、初始态势及元数据）；
- b) 动态过程数据流（人类操作指令序列、行为轨迹日志，智能体决策指令、状态演化记录，以及两类主体交互事件与环境关联数据）。

##### 4.4.2.3 复盘数据回放

智能兵棋推演系统的复盘数据回放功能，旨在基于存储的人类操作复盘数据与智能体数据，通过一体化机制精确、可控地重现推演全过程，为多维度分析提供融合视角。其核心实现依赖于双主体协同的确定性状态重建引擎：该引擎加载含人类与智能体初始状态的初始态势，严格按照原始时序和逻辑，同步重放人类操作指令与智能体决策指令，联动执行两类主体关联的裁决事件，确保输出状态与原推演中双主体的行为交互完全一致。

回放需提供基础控制（播放、暂停、变速、跳转）与深度融合的高级分析能力。

## 5 智能兵棋推演系统接口

### 5.1 推演环境接口

#### 5.1.1 用户授权验证接口

用户授权验证系列接口用于实现智能化环境平台的单点登录，应包含登录口令获取接口（含 Web 系统口令获取接口、客户端系统口令获取接口）和登录验证接口。

接口实例详见《附录 A 示例 A.1 用户授权验证接口示例》。

#### 5.1.2 数据回传系列接口

数据回传接口用于各训练子系统在完成对抗后的对抗成绩记录（包括对抗双方、对抗 ID、对抗成绩、对抗时间信息等）回传给平台，并将对应的复盘文件传递至平台，该接口应包含对抗成绩回传和对抗复盘文件回传两个接口。

为了保证复盘文件传输和解析的建议，复盘文件大小应控制在一定大小（如 $\leq 100\text{MB}$ ）。

接口实例详见《附录 A 示例 A.1.2 数据回传系列接口示例》。

### 5.2 兵棋训练环境接口

- a) 功能定义

兵棋训练环境接口为智能体提供包含战场空间拓扑、兵力部署、对抗规则等核心要素的实时态势感知与推演裁决反馈的接口。

b) 接口

兵棋训练环境接口要求通过定义'TrainEnv'作为所有 AI 智能体获取态势感知与推演裁决反馈的基类以实现对抗系统的推演裁决逻辑，且开发者在开发时必须继承'TrainEnv'基类并实现

`__init__`、`setup`、`step`、`reset`、`get_multimodal_situation` 五个核心方法，方法功能如下所示：

- `__init__`：构建环境实例，返回环境实例；
- `setup`：配置推演过程中推演环境参数；
- `step`：正常推进推演，向环境输入各方动作的集合；
- `reset`：重置环境，清空环境中的变量，释放环境占用的资源；
- `get_multimodal_situation`：获取包含文本描述、简化态势图像等多模态数据。

方法接口实例详见《附录 A 示例 A.2.1 `__init__` 方法》，《附录 A 示例 A.2.2 `setup` 方法》，《附录 A 示例 A.2.3 `step` 方法》，《附录 A 示例 A.2.4 `reset` 方法》，《附录 A 示例 A.2.5 `get_multimodal_situation` 方法》。

### 5.3 兵棋智能体接口

a) 功能定义

兵棋智能体接口用于将推演环境产生的对抗态势以合适数据结构发送给智能体，并将智能体决策产生的对抗操作指令（或称“动作”）在推演中做出响应。

b) 接口

- 兵棋智能体接口遵循强化学习 OODA 环模型，要求通过定义 'BaseAgent' 作为所有 AI 智能体的基类，该基类包含 `setup`、`step`、`reset` 等核心方法及 `deploy`、`command` 等额外接口，开发者开发 AI 智能体时必须继承 'BaseAgent' 基类且必须实现 `setup`、`step`、`reset`、`load_llm_model` 和 `parse_natural_language_command` 五个核心方法，方法功能如下所示：`setup`：初始化智能体参数（如阵营、想定 ID），加载预训练模型或规则库；
- `step`：接收环境态势，返回动作指令队列；
- `reset`：重置智能体内部状态，释放推演缓存；
- `load_llm_model`：加载大模型并配置大模型模参数；
- `parse_natural_language_command`：自然语言决策指令解析为兵棋系统可执行的结构化动作指令

方法接口实例详见《附录 A 示例 A.3.1 `setup` 方法》，《附录 A 示例 A.3.2 `step` 方法》，《附录 A 示例 A.3.3 `reset` 方法》，《附录 A 示例 A.3.4 `load_llm_model` 方法》，《附录 A 示例 A.3.5 `parse_natural_language_command` 方法》。

### 5.4 兵棋智能体开发 SDK

兵棋智能体开发 SDK 是基于 5.3 兵棋智能体接口为智能体（Agent）开发提供的工具集合，其将创建、测试和部署智能体的必要组件及接口封装为程序框架形式供兵棋智能体开发者使用。

兵棋智能体开发 SDK 主要包含以下功能：

- a) 核心环境模块搭建
- b) 想定数据管理
- c) 大模型适配
- d) 多模态数据处理
- e) 大模型微调
- f) 智能体标准的接入结构

g) 标准的开发文档

具体的兵棋智能体开发 SDK 结构示例参见《附录 A 示例 A.4 兵棋智能体开发 SDK 示例》。

## 5.5 用户 UI 规范

### 5.5.1 算子和编成区

算子和编成区在于提供一个集成化、直观化的操作平台，以实现兵棋推演中棋子单位的展示、编队管理、策略规划和实时交互，从而提升推演效率和用户体验，确保推演过程更加高效和真实，该区域应包含以下功能：

- a) 算子显示：用于展示参与推演的各个棋子或单位，如士兵、坦克、飞机等；
- b) 信息显示：显示算子的属性信息，如位置、生命值、行动点等；
- c) 编队管理：用户创建和管理算子的编队；
- d) 资源分配：为编队分配资源，如兵力、装备、补给等。

### 5.5.2 态势区

态势区在于提供一个直观、实时反映战场状况的视觉界面，帮助推演者快速理解当前战场的态势，便于做出决策和战略规划，该区域应包含以下功能：

- a) 态势展示：实时展示战场上的所有棋子、单位及其状态，包括位置、生命值、行动范围等；
- b) 动态更新：随着推演的进行，态势区会动态更新战场信息，反映推演的实时进展；
- c) 交互反馈：允许推演者与态势信息进行交互，如放大查看特定区域、标记关键点等；
- d) 操作指令下达：为推演者提供算子选择，机动，攻击等可视化操作；
- e) 辅助工具可视化：支持如测量等辅助工具的可视化操作和反馈；
- f) 视角修改：支持地图的放大和缩小；
- g) 2D 和 3D 地形切换：支持 2D 和 3D 地形的一键切换。

### 5.5.3 数据区

兵棋推演系统中推演界面的数据区是为了集中展示和管理推演过程中产生的各类数据信息，确保推演者能够快速获取、分析和利用这些数据，以支持推演方快速获取、分析数据及决策制定与推演评估，该区域应包含以下功能：

- a) 信息集中展示：将推演相关的关键数据，如单位属性、行动日志、得分统计（如夺控制点得分，剩余算子得分，战斗得分，净胜分以及总分）等，集中在一个区域展示；
- b) 数据实时更新：随着推演的进行，数据区实时更新，反映当前推演状态。

### 5.5.4 指令下达区

指令下达区在于提供一个用户友好的界面，使得推演者能够方便、高效地观察已经下达的指令序列以及指令的执行过程，确保推演的流程顺畅，同时模拟真实战场中的指挥与控制过程，该区域应包含以下功能：

- a) 指令反馈：在指令下达后提供即时反馈，如显示单位移动路径、攻击效果等；
- b) 指令浏览：记录和观察已经下达的指令列表，为复盘分析提供数据。

## 6 智能兵棋推演系统主要流程

### 6.1 智能体本地高速训练流程

智能体本地高速训练流程是将系统的兵棋智能体决策模块和兵棋推演环境模块采用单进程串行模式实现的无界面内存运行高速推演，该流程通过避免进程和网络间通信实现高运算速度并支持断点测试。

智能体本地高速训练流程如图 2 所示，其流程如下：

步骤 T1: 是否加载大模型，如是则执行步骤 T2; 否则执行步骤 T3;

步骤 T2: **大模型加载**, 加载并初始化大模型;

步骤 T3: **实例化推演环境和智能体**, 确定推演想定、局数等参数, 依次实例化推演环境模块及红蓝方智能体决策模块;

步骤 T4: **初始化推演环境生成初始态势**, 执行推演环境模块初始化函数得到初始态势, 并执行所有推演方兵棋智能体决策模块初始化函数;

步骤 T5: **推演方 1 智能体决策得到动作集合**, 执行推演方 1 兵棋智能体决策模块的单步决策函数, 传入态势信息, 得到推演方 1 动作集合;

步骤 T6: **推演方 2 智能体决策得到动作集合**, 执行推演方 2 兵棋智能体决策模块的单步决策函数, 传入态势信息, 得到推演方 2 动作集合;

步骤 T7: **推演方 n 智能体决策得到动作集合**, 顺序执行推演方 n 兵棋智能体决策模块的单步决策函数, 传入态势信息, 得到推演方 n 动作集合;

步骤 T8: **推演环境单步推进生成新的态势**, 执行推演环境模块的单步推进函数, 传入多方的动作集合得到新的态势;

步骤 T9: **推演是否结束判断**, 判断推演是否结束, 若是, 则结束对抗, 执行步骤 T10, 否则, 返回步骤 T5, 循环执行步骤 T5~T8。

步骤 T10: **推演环境保存复盘**, 执行推演环境模块的保存复盘函数, 生成复盘文件;

步骤 T11: **重置智能体和推演环境**, 执行各智能体和推演环境模块的重置函数完成环境重置。

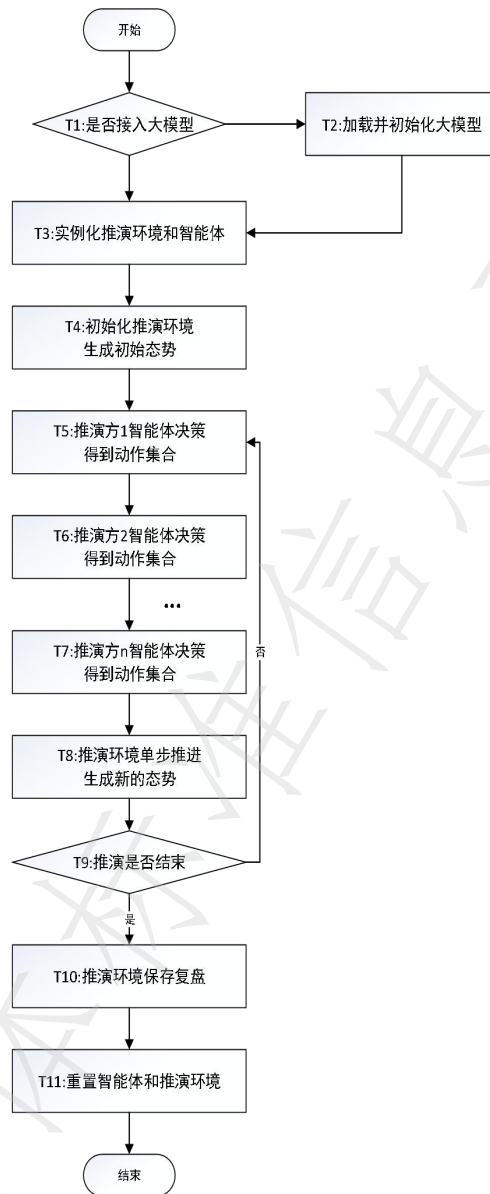


图 2 兵棋智能体本地训练流程

## 6.2 网络对抗流程

在经过本地训练，完成兵棋智能体决策模块逻辑的基础上，将兵棋智能体决策模块按照接口接入智能兵棋推演系统，在网络通信和兵棋引擎的消息调度管理下实现兵棋智能体客户端、兵棋推演交互客户端与服务器端房间模块的态势和指令信息交互，实现人机对抗推演。

网络对抗流程如图 3 所示，其流程如下：

步骤 S1：**建立网络连接实例化智能体**，兵棋智能体框架模块与兵棋推演房间模块建立连接，创建参与推演的智能体决策模块；

步骤 S2：**接口模式协商**，兵棋智能体框架与推演房间模块协商交互模式（同步 / 异步）、数据格式（JSON/Protocol Buffers），若为大模型则自动启用批量消息处理通道。

步骤 S3：**初始化智能体**，按照推演参数初始化兵棋智能体决策模块；

步骤 S4：**监听从推演房间发来的态势**，监听从兵棋推演房间模块发来的态势信息；

步骤 S5: **收到态势后执行智能体单步决策**, 收到态势信息后, 执行兵棋智能体决策模块的单步决策函数, 传入当前态势, 得到动作集合;

步骤 S6: **向推演房间发送动作集合**, 向兵棋推演房间模块发送动作集合;

步骤 S7: **推演是否结束**, 判断推演是否结束, 若是, 则结束对抗, 执行步骤 S8, 否则, 返回步骤 S4, 循环执行步骤 S4~S7;

步骤 S8: **重置智能体**, 执行兵棋智能体决策模块的重置函数。

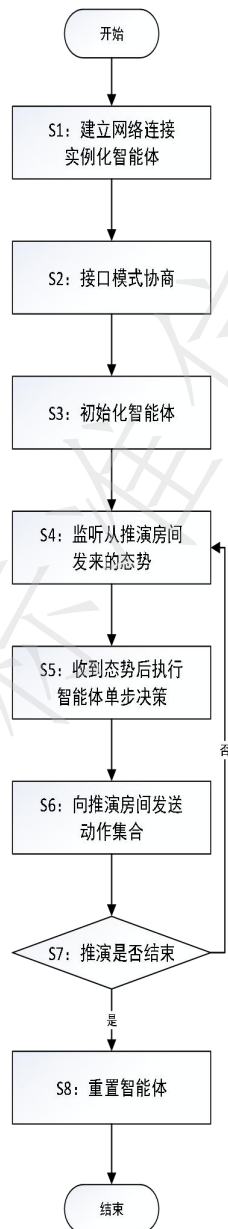


图 3 兵棋智能体网络对抗过程

## 附录 A 示 例

### A.1 推演环境接口相关示例

#### A.1.1 用户授权验证接口示例

##### A.1.1.1 Web系统口令获取接口

- a) 通信协议：HTTP，HTTPS；
- b) 请求参数（具体参数见表 A.1）；

表 A.1 Web 系统口令获取请求参数

参数名	类型	描述
userId	string	用户 ID
timestamp	string	时间戳
platformId	string	平台 ID

\*时间戳为当前时间戳，子系统请验证时间戳为 10 分钟以内的为合法口令。

- c) 响应参数（具体参数见表 A.2）

表 A.2 Web 系统口令获取响应参数

参数名	类型	描述
userToken	string	系统生成的登录口令
expireTime	string	登录口令的过期时间

##### A.1.1.2 客户端系统口令获取接口

- a) 通信协议：HTTP，HTTPS；
- b) 请求参数（具体参数见表 A.3）；

表 A.3 客户端系统口令获取请求参数

参数名	类型	描述
userId	string	用户 ID
timestamp	string	时间戳
clientId	string	客户端 ID

\*时间戳为当前时间戳，子系统请验证时间戳为 10 分钟以内的为合法口令。

- c) 响应参数（具体参数见表 A.4）；

表 A.4 客户端系统口令获取响应参数

参数名	类型	描述
userToken	string	系统生成的登录口令
expireTime	string	登录口令的过期时间

## d) 响应参数

用户从页面手工复制口令后开启本地客户端，在客户端入口输入从门户网站拷贝的验证口令进行登录。

## A.1.1.3 登录验证接口

- a) 通信协议：采用 HTTP，HTTPS 协议的 POST 方法进行数据传输；
- b) 请求参数（具体参数见表 A.5）；

表 A.5 登录验证接口-请求参数

Key	类型	描述
Authorization	string	授权信息
userId	string	用户 ID
userToken	string	时间戳
clientId	string	客户端 ID

- c) 响应参数（具体参数见表 A.6）。

表 A.6 登录验证接口-响应参数

Key	类型	描述
status	string	状态
msg	string	消息
data	string	数据

## A.1.2 数据回传系列接口示例

## A.1.2.1 对抗成绩回传接口

对抗成绩回传接口用于将训练子系统推演完成的单场对抗成绩发送并保存至数据库。

- a) 通信协议：HTTP，HTTPS，SFTP；
- b) 请求参数（具体参数见表 A.7）；

表 A.7 对抗成绩回传接口-请求参数

Key	类型	描述
Authorization	string	授权信息
clientId	string	系统平台 ID
competitionId	string	赛事 ID

表 A.7 对抗成绩回传接口-请求参数 (续)

Key	类型	描述
scheduleId	string	赛程 ID
matchId	string	比赛识别码
startTime	string	比赛开始时间
endTime	string	比赛结束时间

c) 响应消息格式 (具体参数见表 A.8)。

表 A.8 对抗成绩回传接口-响应参数

Key	类型	描述
status	string	状态
msg	string	消息
data	string	数据

## A.2 兵棋训练环境接口相关示例

采用基类 TrainEnv 实现对抗系统的推演裁决逻辑, 智能体是与 TrainEnv 互动, 并产生结果。

### A.2.1 \_\_init\_\_ 方法

- 功能: 构建环境实例, 返回环境实例;
- 形式: TrainEnv::\_\_init\_\_() -> TrainEnv;
- 输入参数: 无;
- 输出参数 (具体参数见表 A.9)。

表 A.9 兵棋训练环境接口-init 输出参数

参数名	类型	描述
trainEnv	TrainEnv	环境实例

### A.2.2 s 方法

- 功能: 配置推演过程中推演环境参数;
- 形式: TrainEnv::setup(setup\_info:Dict)->List[Dict];
- 输入参数 (具体参数见表 A.10)

表 A.10 兵棋训练环境接口-setup 函数输入参数

参数名	类型	描述
setup_info	Dict	环境参数 Dict 合集

setup\_info: Dict 类型, 包含整个环境参数, 包含以下信息:

- setup\_info['scenario']: int 类型, 为环境的想定 id, 必传参数;
- setup\_info['save\_replay\_flag']: Boolean 类型, 表示是否保存复盘, 默认认为是, 非必传参数;

- setup\_info['save\_path']: string 类型，表示复盘文件的地址，非必传参数；
  - setup\_info['seed']: int 或 string 类型，表示环境的随机数生成器的种子，非必传参数；
  - setup\_info['annual\_version']: string 类型，表示用于选择环境的规则版本，非必要参数。
- d) 输出参数（具体参数见表 A.11）。

表 A.11 兵棋训练环境接口-setup 函数输出参数

参数名	类型	描述
confrontposture	List[Dict]	返回参与对抗方的态势队列

### A.2.3 step 方法

- a) 功能：正常推进推演，向环境输入各方动作的集合；
- b) 形式：TrainEnv::step(actions:List[Dict])-(List[Dict],Boolean)；
- c) 输入参数（具体参数见表 A.12）；

表 A.12 兵棋训练环境接口-step 函数输入参数

参数名	类型	描述
action_msgs	list[Dict]	动作列表；可根据具体推演层级定义机动，打击，登车，下车，夺控，移除压制，间瞄射击，引导射击，电子干扰、布雷、扫雷等

- d) 输出参数（具体参数见表 A.13）。

表 A.13 兵棋训练环境接口-step 函数输出参数

参数名	类型	描述
confrontposture	List[Dict]	返回参与对抗方的态势队列
endStatus	Boolean	推演是否结束，True: 推演结束；False: 推演未结束

### A.2.4 reset 方法

- a) 功能：重置环境，清空环境中的变量，释放环境占用的资源；
- b) 形式：TrainEnv::reset()->Boolean；
- c) 输入参数：无；
- d) 输出参数（具体参数见表 A.14）。

表 A.14 兵棋训练环境接口-reset 函数输出参数

参数名	类型	描述
resetStatus	Boolean	是否成功重置环境

### A.2.5 get\_multimodal\_situation 方法

- a) 功能：获取包含文本描述、简化态势图像等多模态数据，为大模型的训练与推理提供适配的输入数据，支持通用智能体对态势信息的多维度理解；
- b) 形式：TrainEnv::get\_multimodal\_situation() -> Dict；
- c) 输入参数：无；

d) 输出参数（具体参数见表 A.15）。

表 A.15 兵棋训练环境接口-get\_multimodal\_situationp 函数输出参数

参数名	类型	描述
multiModal_data	Dict	多模态数据

multiModal\_data 包含文本描述、简化态势图像等多模态数据。

- multiModal\_data['text\_description']: string 类型，当前态势的自然语言文本描述，包含各方力量部署、关键行动状态、环境要素等核心信息；
- multiModal\_data['situation\_image']: string 类型，简化态势图像的 Base64 编码字符串，呈现战场空间拓扑、兵力分布等可视化信息；
- multiModal\_data['timestamp']: string 类型，态势数据生成的时间戳；
- multiModal\_data['situation\_id']: string 类型，当前态势的唯一标识符，用于关联历史态势数据。

### A.3 兵棋智能体接口相关示例

#### A.3.1 setup 方法

- a) 功能：初始化智能体参数（如阵营、想定 ID），加载预训练模型或规则库；
- b) 形式：BaseAgent::setup(setup\_info:Dict)->None；
- c) 输入参数（具体参数见表 A.16）；

表 A.16 兵棋智能体接口-setup 函数输入参数

参数名	类型	描述
setup_info	Dict	此场对局相关的详细信息

setup\_info: Dict 类型，包含整个环境参数，包含以下信息：

- setup\_info['scenario']: int 类型，为环境的想定 id；
- setup\_info['seat']: int 类型，表示本场对局的席位数；
- setup\_info['faction']: int 类型，表示阵营 id，0 为红，1 为蓝；
- setup\_info['role']: int 类型，表示角色，可根据对抗级别进行设定。

d) 输出参数：无。

#### A.3.2 step 方法

- a) 功能：接收环境态势，返回动作指令队列；
- b) 形式：BaseAgent::step(observation:Dict)->List[Dict]；
- c) 输入参数（具体参数见表 A.17）；

表 A.17 兵棋智能体接口-step 函数输入参数

参数名	类型	描述
observation	Dict	态势信息

d) 输出参数（具体参数见表 A.18）。

表 A.18 兵棋智能体接口-step 函数输出参数

参数名	类型	描述
actionDict	List[Dict]	动作指令队列（如机动路径、打击动作等）

## A.3.3 reset 方法

- 功能：重置智能体内部状态，释放推演缓存；
- 形式：BaseAgent::reset()->Boolean；
- 输入参数：无；
- 输出参数（具体参数见表 A.19）。

表 A.19 兵棋智能体接口-reset 函数输出参数

参数名	类型	描述
resetStatus	Boolean	是否成功重置智能体状态

## A.3.4 load\_llm\_model 方法

- 功能：加载大模型，配置大模型模参数（如推理精度、最大上下文长度）；
- 形式：BaseAgent::load\_llm\_model(model\_path: str, config: Dict)->Boolean；
- 输入参数（具体参数见表 A.20）；

表 A.20 兵棋智能体接口-load\_llm\_model 函数输入参数

参数名	类型	描述
model_path	str	大模型 uri 访问地址或本地大模型地址
config	Dict	大模型参数

config: Dict 类型，包含加载的大模型参数，包含以下信息：

- config['inference\_precision']: string 类型，推理精度（如 "fp16"、"int8"）；
  - config['max\_context\_length']: int 类型，模型最大上下文长度；
  - config['device']: string 类型，运行设备（如 "cpu"、"cuda"）；
  - config['batch\_size']: setup\_info['seat']: int 类型，批量推理大小。
- 输出参数（具体参数见表 A.21）。

表 A.21 兵棋智能体接口-load\_llm\_model 函数输出参数

参数名	类型	描述
loadStatus	Boolean	是否成功加载大模型

## A.3.5 parse\_natural\_language\_command 方法

- 功能：将大模型输出的自然语言决策指令解析为兵棋系统可执行的结构化动作指令，实现自然语言到机器指令的转换；
- 形式：BaseAgent::parse\_natural\_language\_command(command: str) -> List[Dict]；
- 输入参数（具体参数见表 A.22）；

表 A.22 兵棋智能体接口-parse\_natural\_language\_command 函数输入参数

参数名	类型	描述
command	str	自然语言决策字符串

d) 输出参数（具体参数见表 A.23）。

表 A.23 兵棋智能体接口-parse\_natural\_language\_command 函数输出参数

参数名	类型	描述
actionList	List[Dict]	动作指令队列

ActionList:List[Dict] 结构化动作指令队列，每个指令字典包含以下信息：

- Dict['unit\_id']: string 类型，执行动作的兵棋单位 ID；
- Dict['action\_type']: string 类型，动作类型（如 "move"、"attack"）；
- Dict['target']: Dict 类型，动作目标（含坐标、单位 ID 等）；
- Dict['actionP']: Dict 类型，动作参数（如速度、火力强度等）。

#### A.4 兵棋智能体开发SDK示例

根据兵棋智能体开发 SDK 功能需求，给出以 python 为编程语言的兵棋开发 SDK 示例，其结构如图 A.1 所示。

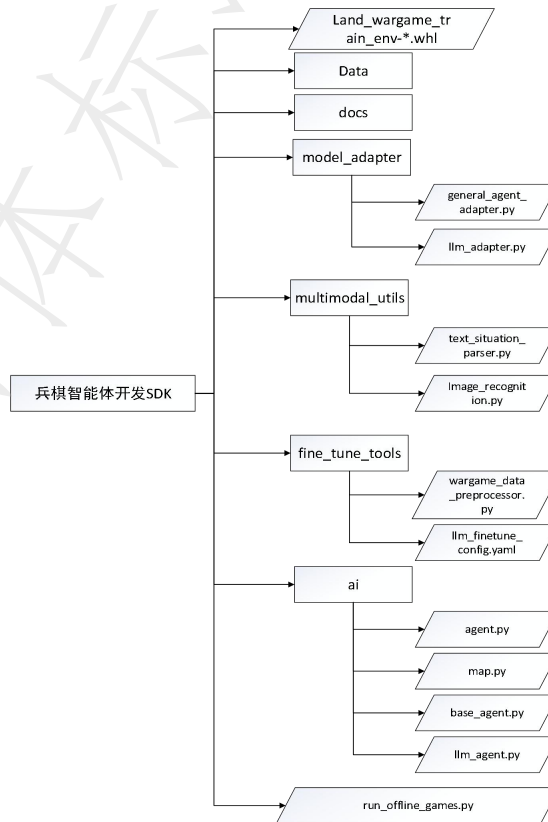


图 A.1 兵棋智能体开发 SDK 目录结构

- a) `land_wargame_train_env-*.whl`: SDK 提供的环境安装包，用于安装核心模块；
  - b) `Data`: 包含 SDK 可用的地图，想定数据；
  - c) `docs`: 开发参考文档；
  - d) `run_offline_games.py`: 推演启动入口程序，提供了环境和 AI 的使用示例；开发者通过修改本文件代码，可以导入自己开发的 `agent`，自由设置对抗的次数，逻辑等；
  - e) `model_adapter`: 模型适配模块，包含通用智能体协议转换工具（`general_agent_adapter.py`）、大模型接口封装（`llm_adapter.py`，支持 GPT、LLaMA 等主流模型）；
  - f) `multimodal_utils`: 多模态数据处理工具，提供态势图像生成（`situation_renderer.py`）、文本态势解析（`text_situation_parser.py`）等功能；
  - g) `fine_tune_tools` : 大模型微调工具，包含兵棋领域数据预处理脚本（`wargame_data_preprocessor.py`）、微调配置模板（`llm_finetune_config.yaml`）；
  - h) `ai`: 开发者需在 SDK 根目录下自己创建名为 `ai` 的开发包，并确保可以从此开发包中导入到名为 `Agent` 的类，此类将在入口程序中被实例化来启动 AI。公开的 SDK 已经为大家设置好正确的结构，且在 `agent.py` 中内置了 `DemoAI`。开发者可以参照公开的文件，类及结构，避免在线上上传 AI 时产生不必要的麻烦；
    - 1) `ai/base_agent.py`: 陆战兵棋 AI 基类。开发人员在编写自己的 AI 时，必须继承 `base_agent`，并实现其中的抽象方法；
    - 2) `ai/map.py`: 为方便 AI 开发，SDK 提供的地图相关基础函数工具（如寻路），供开发团队选用；
    - 3) `ai/agent.py`: SDK 中自带的 `DemoAI`，此 AI 会下达随机动作。开发者可以参考也可以完全重写；
    - 4) `ai/llm_agent.py`: 基于大模型的智能体示例，展示如何通过 `BaseAgent` 扩展方法实现大模型驱动的决策逻辑。
-