<RoboTrain>

前景文档

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 25/06/2025 | 1.0 | 基于迭代一要求和项目方案的初始草案 | 徐丁杰 |

目录

1. 简介 5

1.1 目的 5

1.2 范围 5

1.3 定义、首字母缩写词和缩略语 5

2. 定位 5

2.1 商机 5

2.2 问题说明 5

2.3 产品定位说明 6

3. 涉众和用户说明 6

4. 产品概述 6

4.1 产品总体效果 6

4.2 功能摘要 7

4.3 假设与依赖关系 7

4.3.1 硬件依赖 7

4.3.2 软件依赖 7

5. 产品特性 7

5.1 用户认证与权限管理 7

5.2 项目工作空间管理 7

5.3 标准化数据注入 7

5.4 自定义数据转换服务 7

5.5 轨迹数据可视化回放 8

5.6 自动化训练任务调度 8

5.7 实时训练过程监控 8

5.8 模型结构自定义 8

5.9 模型资产管理 8

5.10 跨项目资产关联 8

6. 约束 8

6.1 硬件资源约束 8

6.2 并发性能约束 8

6.3 数据量约束 8

7. 质量属性 8

7.1 易用性 8

7.2 可靠性 8

7.3 时间性能 8

8. 优先级 9

8.1 第一优先级 (MVP核心功能) 9

8.2 第二优先级 (MVP后完善) 9

8.3 远期规划 (进阶需求) 9

前景

# 简介

## 本章旨在对RoboTrain项目前景文档进行总体概述，包括阐述此文档的目的与范围，并提供后续章节中使用的定义、缩写和参考资料。

## 目的

## 本文档的目的是为“RoboTrain：具身智能云端训练与推理平台”项目，定义其高层次的业务目标、用户需求和核心特性。它旨在成为项目所有参与者——包括开发团队、项目管理者及其他关键涉众——的共同纲领和沟通基石。通过清晰阐述项目的最终愿景和第一阶段（MVP）的目标，本文档将为后续的需求分析、技术选型、开发规划和项目决策提供统一的、高层次的指导和评判依据

## 范围

## 本文档的范围严格限定在“RoboTrain：具身智能云端训练与推理平台”这个相关项目之内。

## 定义、首字母缩写词和缩略语

|  |  |
| --- | --- |
| VLM | (Vision-Language Model) 视觉语言模型，一种能够同时理解图像和文本信息的多模态大模型 |
| Lerobot | (Huggingface Lerobot) Huggingface推出的一个专为机器人学习设计的开源库，本项目将其作为核心数据格式规范 |
| MinIO | 一个开源的、兼容S3协议的对象存储服务，用于本项目中存储数据集和模型权重。 |
| Accelerate | （Huggingface Accelerate) Huggingface推出的一个旨在简化PyTorch分布式训练的库 |
| DeepSpeed | 由微软开发的一个深度学习优化库，用于高效地进行大规模模型训练，本项目用其解决单机多卡训练的显存瓶颈问题 |
| ML | (Machine Learning) 机器学习，人工智能的一个领域，专注于构建能从数据中学习和改进的系统。在本项目中特指用于模型微调的核心技术。 |

# 定位

## 商机

## 本项目致力于解决当前机器人（具身智能）领域普遍存在的一大痛点：研究人员和开发者在进行视觉语言大模型（VLM）的微调与部署时，面临着流程繁琐、资源门槛高、缺乏统一工具链的困境。通过提供一个集数据管理、模型训练、云端推理于一体的Web平台，RoboTrain旨在抓住这一商机，大幅降低机器人大模型应用的门槛，加速研发迭代周期。

## 问题说明

|  |  |
| --- | --- |
| 问题是 | 机器人领域的研究者和开发者在微调和部署大型视觉语言模型（VLM）时，缺乏一个统一、高效、低门槛的平台化解决方案 |
| 影响 | 机器人研发团队、算法工程师和高校研究人员 |
| 问题的后果 | 研发效率低下，硬件资源成本高昂，技术成果难以复现和快速转化，创新受阻 |
| 成功的解决方案 | 应该能提供一个自动化的工作流，让用户可以轻松上传自定义数据，通过Web界面配置并启动VLM微调任务，并最终获得可用于部署的模型，从而显著提升研发效率和鲁棒性 |

## 产品定位说明

|  |  |
| --- | --- |
| 针对于 | 需要利用自定义数据微调视觉语言模型（VLM）的机器人领域研究人员和算法工程师 |
| 谁 | 希望能摆脱繁琐的环境配置和脚本编写，专注于算法和应用本身，并需要一个能管理数据、训练和模型的统一平台 |
| RoboTrain | 是一个具身智能云端训练与推理平台 |
| 功能 | 提供标准化的数据接入、自动化的模型微调流程、训练过程监控，以及清晰的模型资产管理功能 |
| 不同于 | 零散的命令行脚本、通用的机器学习云平台或需要大量定制化开发的内部工具 |
| 我们的产品 | 提供一个专为机器人学习场景设计的、端到端的解决方案，通过对Lerobot等事实标准的兼容和对训练流程的深度优化，为用户带来开箱即用的便捷体验 |

# 涉众和用户说明

本项目的核心涉众与最终用户在MVP阶段高度统一，即由前端与项目统筹（P1）、后端与数据架构师（P2）、ML核心与算法工程师（P3）、以及ML系统与加速专家（P4）四类角色构成的内部开发团队。

# 产品概述

## 产品总体效果

RoboTrain是一个独立的Web应用程序，为用户提供一个从数据到模型的闭环管理和训练环境。其系统架构设计旨在将用户界面、后端服务和底层计算资源解耦，以实现模块化开发和未来扩展。



如上图所示，用户通过Web前端（React/Vue）与系统交互。所有请求被发送至FastAPI后端服务 。后端负责处理业务逻辑，将元数据存入PostgreSQL数据库，将大型文件（数据集、模型权重）存入MinIO对象存储。当用户提交训练任务时，后端服务将其放入一个简单的任务队列，并调用底层ML/DL栈（PyTorch, Accelerate, DeepSpeed）在GPU硬件上执行训练。

## 功能摘要

**客户支持系统**

|  |  |
| --- | --- |
| **客户利益** | **支持特性** |
| 数据标准化，易于管理 | 平台以Huggingface Lerobot为核心数据格式，并为自定义数据提供转换脚本接口和执行环境 。所有数据集在平台内有统一的列表展示和管理 。 |
| 训练流程自动化，降低门槛 | 用户只需在Web界面选择数据集和预设模型，平台即可自动启动训练任务，无需编写训练循环代码 。 |
| 有限资源下的高效微调 | 平台后端集成了DeepSpeed和Accelerate，自动处理多GPU并行和显存优化，使得在单张或多张4090上微调大型VLM成为可能 。 |
| 训练过程透明可控 | Web前端提供基础的训练日志输出和损失曲线显示，用户可以实时监控任务状态 。 |
| 资产集中管理，便于追溯 | 用户、项目、数据集、训练任务和生成的模型都作为资产在平台中进行统一管理和关联，方便追溯和复用 。 |

## 假设与依赖关系

### 硬件依赖

项目MVP阶段的运行和测试依赖于一台配备至少一张NVIDIA RTX 4090 GPU的服务器。

### 软件依赖

项目强依赖于Huggingface生态（Lerobot, Transformers, Accelerate）和PyTorch框架，这些库的API稳定性会影响项目。

# 产品特性

## 用户认证与权限管理 (User Authentication and Authorization)

## 系统应提供安全的用户注册与登录机制，并对通过身份验证的用户进行区分。用户的操作权限将与身份绑定，以保障项目和数据资产的隔离与安全。

## 项目工作空间管理 (Project Workspace Management)

系统应支持用户创建逻辑隔离的项目工作空间。每个项目作为独立的容器，用于聚合与该项目相关的数据集、训练任务和模型资产，从而实现多任务的有序管理和资产追溯。

## 标准化数据注入 (Standardized Data Ingestion)

## 系统必须支持对遵循Huggingface Lerobot数据格式规范的机器人轨迹数据的直接上传和解析。此特性是确保数据能够无缝流入平台后续处理与训练流程的基础。

## 自定义数据转换服务 (Custom Data Transformation Service)

## 为兼容多样化的用户数据，系统应提供一个可扩展的数据转换接口。用户可上传自定义格式的数据集，并附带一个符合规范的Python转换脚本，平台将在托管环境中调用该脚本，将数据自动转换为内部标准格式。

## 轨迹数据可视化回放 (Trajectory Data Visual Playback)

## 系统需提供一个基础的数据检视工具，允许用户对单个数据轨迹进行可视化回放。该工具应能同步展示关键的传感器数据（如图像序列）与对应的执行器指令，以供用户进行数据质量的直观校验。

## 自动化训练任务调度 (Automated Training Task Scheduling)

## 系统应能根据用户在Web界面提交的训练配置（包含指定的数据集和模型）自动生成并调度训练任务。平台内置一个单任务队列机制，用于管理对底层计算资源的访问冲突，确保任务在指定的硬件上（如单张RTX 4090）按序执行。

## 实时训练过程监控 (Real-time Training Process Monitoring)

## 对于正在执行的训练任务，系统必须提供一个实时的监控界面。该界面应向客户端流式传输关键的训练过程信息，至少包括底层的日志输出和损失函数（Loss）随时间变化的收敛曲线。

## 模型结构自定义 (Model Structure Customization)

## 系统应允许用户在启动训练前，对所选的基础模型架构进行结构上的微调。这包括通过参数化配置选择性地冻结模型的部分层级以保留其预训练知识，以及为模型添加一个简单的、用于任务适配的自定义头部（如MLP层）。

## 模型资产管理 (Model Asset Management)

## 训练完成后，系统需将产出的模型资产（包括模型权重文件和训练日志）进行持久化存储。用户可通过平台界面浏览自己训练生成的模型列表，并有权下载这些资产以用于本地评估或进一步的部署。

## 跨项目资产关联 (Cross-Project Asset Association)

## 系统需在数据库层面建立清晰的关联逻辑。每个训练任务都必须与其所使用的数据集、所生成的模型进行绑定，同时所有这些资产都需归属于一个明确的用户和项目，形成一个完整的、可追溯的实验链条。

# 约束

## 硬件资源约束

RTX 4090虽强，但训练大型VLM模型（如RT-2）仍然非常耗时，且无法支持多用户同时训练，这是项目的核心瓶颈。

## 并发性能约束

系统仅支持1个训练任务并发执行，其他提交的任务需要进入队列等待。

## 数据量约束

为适应单机存储，初期支持的数据集规模应控制在10k条轨迹以内，总数据量在几十GB范围内。

# 质量属性

## 易用性

Web界面应设计简洁直观，将复杂的机器学习参数进行抽象和简化，并提供合理的默认值，核心操作应符合用户直觉。

## 可靠性

后端服务应能稳定处理前端请求。训练任务的启停、失败、成功等状态应能被准确记录和反映。

## 时间性能

普通用户在Web界面的操作，响应时间应小于2秒。训练任务的提交请求，UI响应时间应小于1秒。

# 优先级

## 第一优先级 (MVP核心功能)

数据上传与管理（遵循Lerobot格式）；

**模型训练与微调**（核心功能）；

基础的用户与项目管理；

## 第二优先级 (MVP后完善)

更高的时间性能；

更丰富的数据可视化功能；

## 远期规划 (进阶需求)

所有与云端推理和“快慢系统”部署相关的功能；

支持点云、语音等多模态数据；

模型评估与排行榜、自动超参数优化；