

AI 智能 · 学习搭子

解决方案文档



声像科技

Sheng Xiang Ke Ji

发包方：数字马力

承接方：声像科技

日期：2025 年 10 月

目标与服务模型

本项目立足数字马力“从校园到职场”的人才培养使命，针对当代学习者在知识获取效率低、个性化支持缺失、学习动力不足、校园与就业脱节等核心痛点，提出“AI 智能·学习搭子”整体解决方案。

方案以“陪伴-知识-评估-激励”为服务主线，构建四大协同系统：通过数字人伴学系统提供情感化、低延迟的实时交互，缓解学习孤独感；依托多智能体-知识库协作系统实现动态知识整合与主动学习规划；借助多学科客制化评估系统精准诊断高校学生能力短板，对接就业技能图谱；利用游戏化学习平台激发 K12 及语言学习者的内在动机，实现“玩中学”。

需求分析

本项目旨在解决当代学习者在两大核心场景中面临的痛点：

一阶段（校园考试）：知识吸收效率低、缺乏个性化指导、学习过程孤独、难以精准评估薄弱环节。

二阶段（就业技能）：高校教学与市场需求脱节、缺乏对标岗位的能力评估、缺少持续的学习动力和规划。

基于此，我们设计了“AI 智能·学习搭子”解决方案，并设计以下的三大模块、四大系统：

可视化交互模块：数字人伴学系统

核心需求分析：

情感陪伴与激励：学习过程孤独是普遍痛点，用户需要一个能提供情感支持、缓解压力、保持学习热情的“学习搭子”。

自然、沉浸的交互体验：传统的文字问答缺乏温度。用户期望通过自然语言（尤其是语音）与学习伙伴进行流畅、低延迟的对话，获得接近真人交流的体验。

个性化形象与认同感：一个固定、无特色的形象难以建立长期陪伴关系。用户希望拥有可定制、甚至能与自己形象关联的专属学习伙伴，增强认同感和使用粘性。

解决方案：

端到端实时语音交互：采用 SenseVoice ASR + Qwen3-4B LLM + CosyVoice 2.0 TTS 的技术栈，构建平均延迟约 3 秒的实时对话流，确保交互的流畅性。

情感化智能陪伴：系统不仅理解语义，还通过 ASR 的情感识别（SER）和 LLM 的情感分析，驱动 TTS 输出不同情感语调（鼓励、耐心等），并计划联动 LAM Avatar 引擎实现表情同步，打造有温度的陪伴体验。

个性化形象生成：利用 LAM 引擎，未来支持学生上传照片生成专属 3D 数字人形象，将“学习搭子”从工具转变为有情感连接的伙伴。

后端服务模块：多智能体-知识库协作系统

核心需求分析：

知识的动态性与准确性：静态知识库无法应对知识的快速迭代。用户需要一个能整合多源信息（教材、论文、行业报告）、并能动态更新的“活”知识库。

复杂问题的深度解答：单一问答模型难以处理跨领域、深层次的问题。用户期望系统能像一个由“图书管理员”、“内容编辑”、“学业顾问”组成的专家团队，协同提供精准、易懂的答案和规划。

从“被动问答”到“主动规划”：用户不仅需要答疑，更需要系统能根据其目标和薄弱点，主动规划学习路径，实现从“授人以鱼”到“授人以渔”的转变。

解决方案：

基于 AgentUniverse 的多智能体融合架构：构建四大核心智能体——知识库检索智能体、内容生成与摘要智能体、学习规划智能体、反思与进化智能体。它们遵循 MCP/A2A 协议，通过 Service Mesh 微服务架构协同工作，形成一个可推理、可协作、可进化的动态知识网络。

活知识库与自我进化：“反思与进化智能体”通过分析交互日志和用户反馈，自动发现知识盲区并触发更新流程，确保系统“越用越懂你”。

深度协同与规划：该系统作为整个方案的“大脑”，为数字人伴学提供知识支持，为评估系统提供规划依据，真正实现了知识服务的智能化和主动化。

评估系统模块：多学科客制化评估系统 & 游戏化学习平台及评估系统

核心需求分析：

分层评估需求：

入门层（K12/兴趣导向）：用户需要无压力、高趣味性的评估方式，将学习与游戏结合，通过即时正向反馈激发内在动机。

进阶层（初高中/高校/就业导向）：用户需要精准、多维度、对标目标的评估，不仅能判断对错，更能诊断薄弱环节，并与学习目标（考试大纲/岗位技能）挂钩。

评估即学习：用户反感孤立的考试。期望评估过程本身就是学习过程的一部分，能提供即时、具体的反馈和指导，实现“以评促学”。

学习路径的无缝衔接：两大评估系统需要打通，实现从“兴趣激发”到“目标导向”的平滑过渡，覆盖用户完整的成长路径。

解决方案：

游戏化学习平台（入门层）：以 Ant Design 为 UI 基座，设计角色成长、即时奖励、挑战任务等游戏化机制。评估完全嵌入式于游戏流程中（如通过口语对话闯关），利用 ASR/NLU 技术实时评估，让学习者在“玩”中无感完成能力诊断。

多学科客制化评估系统（进阶层）：针对不同学科特性定制评估方式。例如，计算机学科基于 OpenSumi IDE 框架，提供云端编程环境和多维度代码评测；数学学科利用

Qwen-VL 识别手写公式；化学学科通过 Three.js 构建 3D 分子模型进行可视化评估。系统可对接相应行业的技能图谱，为就业导向用户提供对标企业标准的能力雷达图。

双系统协同闭环：当游戏化平台用户达到预设水平，系统自动引导其进入客制化评估系统；客制化评估的诊断结果则会触发数字人伴学系统进行主动关怀和针对性辅导，形成“学-练-测-评-优”的完整闭环。

通过四大系统协同工作，声像科技团队构建了一个覆盖“兴趣激发-知识学习-能力评估-职业规划”全链路的智能学习生态。方案深度融合蚂蚁集团技术生态，确保系统的先进性、安全性和可扩展性。

业务流程

声像科技团队就“AI 智能·学习搭子”的项目目标，设计了三种分学段循序渐进的业务流程规划：

业务流程一：小学（K12）（面向兴趣激发与入门学习）

核心目标：激发学习兴趣，培养学习习惯，为后续进阶学习打下坚实基础。

游戏化入口与角色创建：小学生用户进入“游戏化学习平台”，创建专属虚拟角色（Avatar），并可选择与蚂蚁庄园等 IP 联动的皮肤，快速建立情感连接。

玩中学的沉浸式体验：学习过程完全融入游戏。例如，在“单词冒险岛”关卡中，学生需通过口语对话（ASR/NLU 评估）与 NPC 互动来完成任务；在“句法对对碰”游戏中，通过匹配正确词组来消除方块。

即时反馈与正向激励：每完成一个微小任务，系统立即给予动画、音效等多感官即时奖励，形成强大的正向激励循环，有效解决低龄用户注意力难以集中的问题。

数据驱动的个性化难度：平台后台记录学生的发音准确度、错误类型、反应时间等数据，并动态调整后续关卡的难度，确保挑战始终与学生能力匹配，维持其学习兴趣；同时将相应数据发送给家长及教师，更好地掌握学生的学习情况。

平滑过渡至进阶学习：当学生在游戏化平台中达成预设里程碑（如完成小学英语全部课程），系统会自动推荐并引导其进入“多学科客制化评估系统”的更高阶学习路径，实现从“兴趣驱动”到“目标导向”的无缝衔接，覆盖完整的成长轨迹。

业务流程二：初高中（面向“一阶段”校园考试导向）

核心目标：解决传统教学“千人一面”的问题，提供精准的薄弱点诊断与个性化辅导，提升应试效率。

日常学习与嵌入式诊断：学生在日常学习中使用“多学科客制化评估系统”完成作业或单元测试。系统利用 Qwen-VL 识别手写数学公式，或通过 Three.js 进行化学分子结构可视化评估，实现无感化、过程性的学情诊断。

精准定位薄弱环节：系统不仅能判断对错，更能通过 LLM 智能分析，精准定位知识盲区（如“三角函数图像变换理解不清”），并生成详细的错题报告和知识点溯源。

主动干预与伴学辅导：诊断结果实时触发两大系统：

数字人伴学系统：数字人主动发起对话，以温和耐心的语气引导学生回顾错题，并进行针对性讲解。

多智能体-知识库协作系统：学习规划智能体自动调整学习计划，增加相关知识点的练习和复习任务。

情感化陪伴与动力维持：面对升学压力，数字人通过情感化语音（CosyVoice 2.0）和未来将实现的表情同步（LAM），为学生提供持续的情感支持和正向激励，有效缓解学习孤独感和焦虑情绪。

效果追踪与目标达成：系统持续追踪学生在薄弱知识点上的进步情况，形成动态的学习效果报告，帮助学生和教师清晰把握备考进度，直至顺利通过考试。

业务流程三：高校（面向“二阶段”就业技能导向）

核心目标：弥合校园学习与就业需求的鸿沟，提供技能评估与职业规划指导。

入口与目标设定：大学生用户注册后，明确学习目标（如“成为 Java 后端开发工程师”），系统自动关联对应相关企业的行业技能图谱。

客制化能力评估：用户进入“多学科客制化评估系统”，完成对标岗位的实战任务（如在客制化系统的 IDE 中完成一个微服务模块开发）。系统将从代码正确性、算法效率、工程规范等多维度进行精细化评估，生成能力雷达图。

智能规划与知识协同：评估结果同步至“多智能体-知识库协作系统”。学习规划智能体结合岗位要求与用户薄弱点（如“数据库设计能力不足”），生成个性化学习路径；知识库检索智能体与内容生成智能体协同，推送精选学习资源和实战项目。

数字人伴学与情感激励：在整个学习过程中，数字人伴学系统作为“学业伙伴”提供 7x24 小时答疑。当用户在评估中受挫时，系统会主动触发数字人进行情感关怀和错因分析，缓解学习压力。

闭环与就业衔接：用户完成学习路径后，可再次进行评估，验证能力提升效果。系统最终输出一份对标企业标准的综合能力报告，并提供用于求职的简历提升指导，实现从“校园”到“职场”的无缝衔接。

用户交互原型

用户交互原型：小学（K12）——游戏化学习与评估系统

设计目标：打造一个色彩鲜艳、交互直观、充满乐趣的沉浸式学习入口，让低龄用户“玩中学”。

界面风格：采用 Ant Design 的基础组件并进行高度定制化的设计，配色方案以明亮、活泼的糖果色系为主，图标采用圆润、卡通化的设计，符合儿童审美。

核心交互流程：

角色创建页：用户进入后，首先看到一个 3D 角色编辑器。用户可以选择基础形象，并通过“金币”（初始赠送）兑换与蚂蚁集团 IP 联动的皮肤（如小鸡帽子、树叶披风）。此页面利用 Ant Design 的 Modal 和 Tabs 组件组织选项。

主游戏大厅：以岛屿地图形式呈现，每个岛屿代表一个学习主题（如“单词冒险岛”、“句法对对碰”）。岛屿图标采用 Ant Design 的 Avatar 组件，并通过 Badge 徽章组件显示任务完成度。点击岛屿进入关卡选择。

“单词冒险岛”关卡：界面分为三部分：顶部显示任务目标（如“和船长对话，拿到宝藏地图”），中部为 3D 场景（使用 Three.js），底部为麦克风按钮。学生点击麦克风，说出指定句子（如“Where is the map?”），并利用 ASR/NLU 实时评估。成功后，场景中播放庆祝动画（宝箱打开），并弹出 Ant Design 的 Notification 组件，显示“+10 XP!”。

数据看板（家长/教师端）：在家长端 App 中，通过 Ant Design 的 Progress 进度条和 Chart 图表组件，以可视化方式展示孩子的学习时长、发音准确率趋势、高频错误词等数据，便于家长掌握学习情况。



用户交互原型：初高中——多学科客制化评估系统

设计目标：提供专业、清晰、高效的评估与诊断界面，帮助学生和教师聚焦核心问题。

界面风格：采用基于 Ant Design 的主题设计，布局简洁、信息密度高，强调内容的可读性和操作的便捷性。整体风格与蚂蚁系产品保持一致，降低用户学习成本。

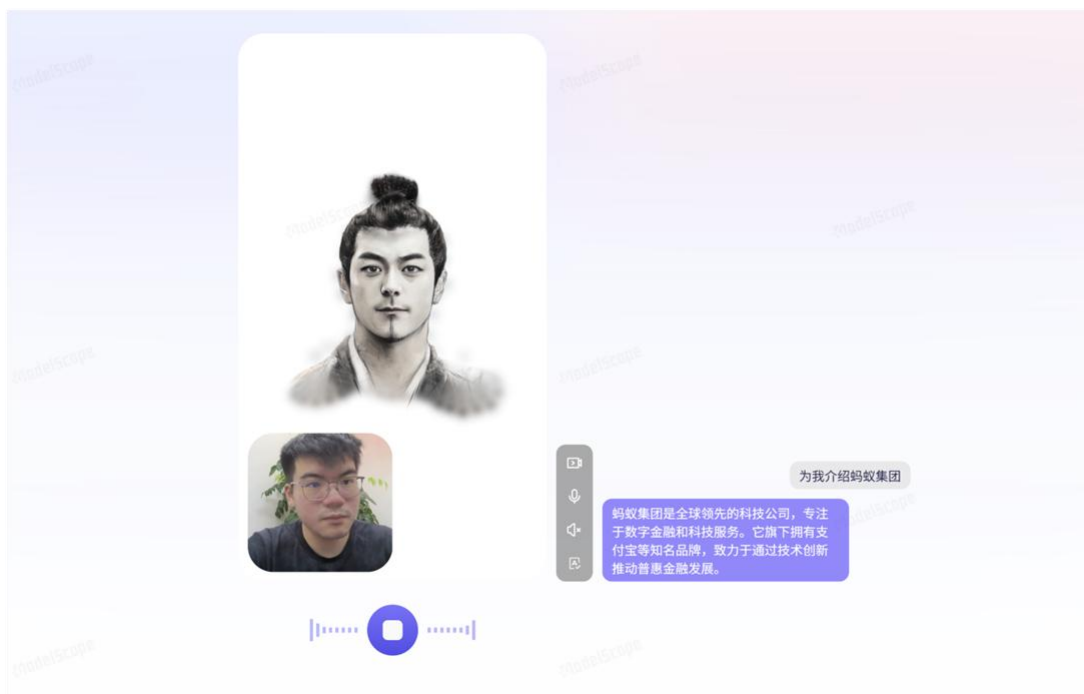
核心交互流程：

学科选择与任务面板：主界面左侧为学科导航栏（数学、化学等），右侧为任务列表。每个任务卡片（Card）清晰标注任务类型（作业/单元测试）和截止时间。

数学手写评估界面：学生点击数学任务后，进入一个类似在线答题卡的界面。学生可在画布上手写公式。提交后，系统调用 Qwen-VL 进行识别和逻辑分析。评估结果页分为三栏：左侧原题与学生手写稿（高亮错误步骤），中间为 LLM 生成的详细错因分析（如“此处应使用诱导公式，而非直接代入”），右侧为关联知识点卡片（可点击跳转复习）。

化学 3D 分子评估界面：在化学任务中，学生需在 Three.js 构建的 3D 空间中，拖拽原子构建指定分子。系统实时评估键角、键长是否符合标准。评估完成后，界面会展示标准模型与学生模型的对比，并用 Ant Design 的 Steps 组件引导学生回顾构建过程中的关键步骤。

错题本与数字人介入：所有错题自动归集到“错题本”。当学生查看错题本时，页面右下角会弹出数字人头像，并主动发起对话：“这道题看起来有点难，需要我帮你再讲一遍吗？”，实现无缝的伴学衔接。



AI 智能·学习搭子：解决方案

用户交互原型：高校——就业导向的客制化评估与规划系统

设计目标：模拟真实工作环境，提供对标产业标准的专业评估和职业发展指导。

界面风格：采用 Ant Design 的中台/后台解决方案，界面专业、沉稳，功能模块划分清晰，体现客制化应用的专业感。

核心交互流程：

职业目标设定页：新用户注册后，首先进入目标设定向导。通过 Ant Design 的 Cascader 组件，用户选择目标岗位（如“后端开发工程师”→“Java 方向”）。系统自动加载相关关联企业要求的行业技能图谱。

客制化 IDE 评估环境：用户点击“开始评估”后，直接进入基于 OpenSumi 框架构建的云端 IDE。界面包含标准的代码编辑区、终端、调试面板等。任务描述将以 Ant Design 的 Descriptions 组件清晰列出需求，用户在此环境中完成编码任务。

多维度评估报告：提交代码后，系统在定制 IDE 的侧边栏生成评估报告。报告采用 Ant Design 的 Tabs 和 Table 组件，分页展示用户的代码的单元测试，代码质量及能力雷达图。

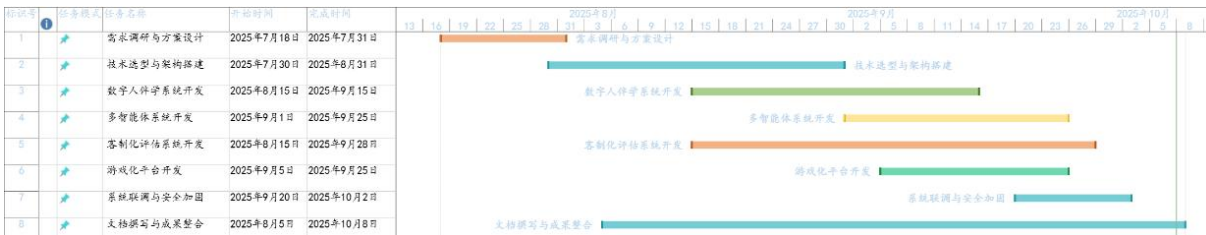
个性化学习路径：报告页底部提供“下一步行动”建议。点击后，跳转至学习路径规划页。该页面以 Ant Design 的 Timeline 组件展示动态生成的学习计划，包含推荐的课程、项目实战和复习资料，均由“多智能体-知识库协作系统”进行智能推送。



组织管理与业务分析方案

项目团队协作与开发管理

本项目由声像科技团队共 5 名成员协同完成，涵盖队长及客户关系专员、技术架构师、前端开发工程师、后端开发工程师、产品经理。为确保四大子系统（数字人伴学系统、多智能体-知识库协作系统、多学科客制化评估系统、游戏化学习平台）并行高效推进，团队采用敏捷开发模式，结合甘特图进行任务分解与进度追踪。



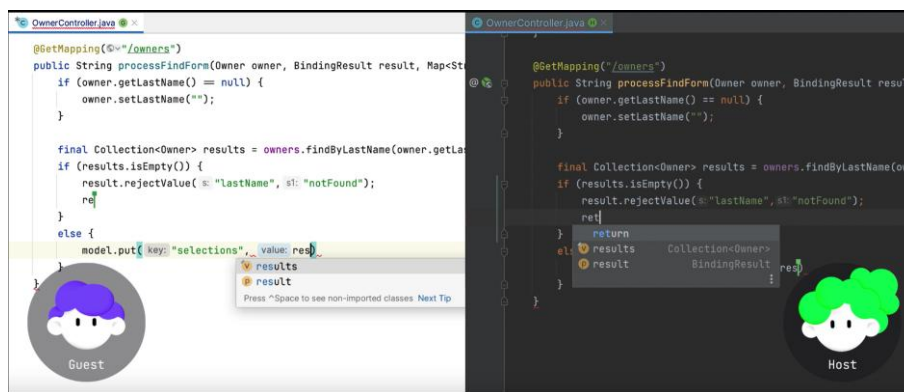
代码协作与版本管理

GitHub 代码仓库管理：团队在 GitHub 上建立组织仓库 SoundTech，按四大子系统划分子模块仓库，严格执行 PR（Pull Request）代码审查机制。



JetBrains Code-With-Me 实时协同编程：针对复杂算法模块（如 SenseVoice 伪流式改造、LAM 表情驱动联动），团队使用 JetBrains Code-With-Me 插件进行多人实时协同编码。该工具支持语音通话、权限分级、断点调试共享，显著提升跨地域协作效率，尤其在优化端到端延迟过程中发挥关键作用。





四大系统业务成本与开发分析

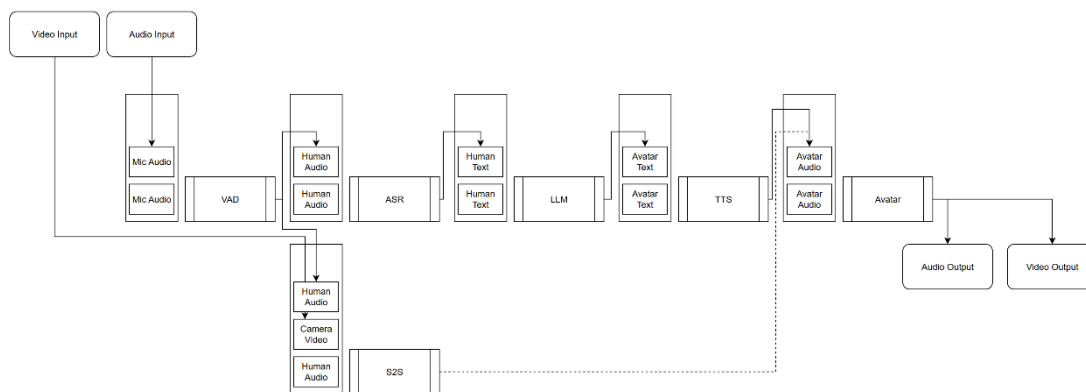
名称	开发成本	硬件资源成本	应用环境要求	经济效益分析
数字人伴学系统	2 人月	GPU 应用节点+ 轻量应用服务器 +API 调用 成本约¥600/月	客户端：PC/ 现代浏览器+ 摄像头+麦克 风	可复用于高校心理驿 站、远程家教等场景， 边际成本低
多智能体-知识 库协作系统	1 人月	向量数据库 +Redis 缓存集群 成本约¥200/月	客户端：PC/ 现代浏览器	作为“知识中枢”可独立 输出为 SaaS 服务，高 复用价值支持多租户
多学科 客制化 评估系 统	0.5 人月	GPU 应用节点+ 轻量应用服务器 +API 调用 成本约¥400/月	服务器： Docker 客户端：PC/ 现代浏览器	直接对接数字马力产教 融合项目，可向合作高 校按学科模块收费
游戏化 学习平 台与评 估系统	0.5 人月	轻量应用服务器 +API 调用 成本约¥100/月	客户端：现代 浏览器	适合公益推广与 B2B 模式，可通过蚂蚁公益 平台触达百万级用户

技术路线及实现方案

本项目采用“模块解耦、生态融合、安全可控、持续进化”的技术路线，四大系统分别基于蚂蚁集团与阿里通义实验室的成熟开源技术栈进行深度定制，确保先进性、可维护性与落地可行性。整体架构遵循企业文档提出的微服务 + Service Mesh + 多租户 SaaS 范式，并全面融入 OAuth2/JWT + HTTPS 安全体系。

数字人伴学系统

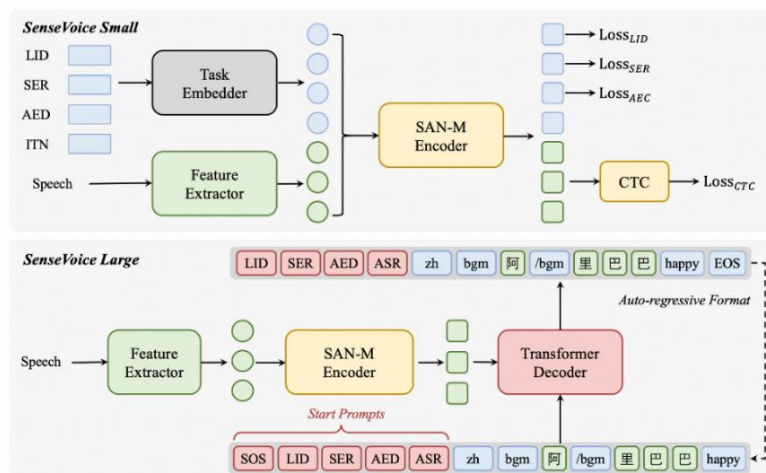
本系统以阿里巴巴通义实验室开源的 OpenAvatarChat 为基座，构建了一套端到端的实时数字人对话工作流。该工作流采用模块化设计，将语音识别（ASR）、大语言模型（LLM）、语音合成（TTS）和数字人驱动（Avatar）四大核心引擎解耦，便于独立优化和替换。



在本地部署测试环境中（Ultra7-265K CPU + Nvidia RTX 5080 GPU），我们选用以下技术栈，并在局域网 HTTPS（由 OpenSSL 提供 SSL 证书）环境下进行推流，实现了约 3 秒的平均端到端延迟。以下是对四大功能引擎的选型介绍和优化方案：

ASR 语音识别引擎：SenseVoice

SenseVoice 是阿里达摩院推出的工业级语音识别基础模型，不仅支持高精度的自动语音识别（ASR），还集成了语种识别（LID）、情感识别（SER）等多模态能力，以便为后续的个性化交互提供丰富的上下文信息。



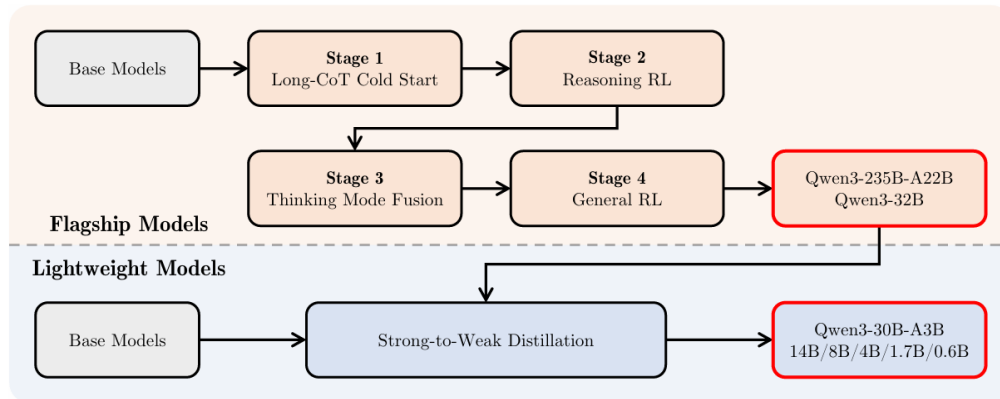
为了降低交互延迟，我们对 SenseVoice 进行伪流式处理改造。通过将用户的连续语音流按固定时间窗口（chunk）进行分块，并采用截断注意力机制（truncated attention）对每个语音块进行独立推理。这种方式在保证识别准确率的同时，显著减少了从用户说完话后到系统开始响应的等待时间，提升了对话的流畅感。

核心代码示例如下：

```
1 class StreamingSenseVoice:
2     def inference(self, speech):
3         speech = speech[None, :, :]
4         speech_lengths = torch.tensor([speech.shape[1]])
5         speech = speech.to(self.device)
6         speech_lengths = speech_lengths.to(self.device)
7         speech = torch.cat((self.query, speech), dim=1)
8         speech_lengths += 4
9         encoder_out, _ = self.model.encoder(speech, speech_lengths)
10        return self.model.ctc.log_softmax(encoder_out)[0, 4:]
11
12    def decode(self, times, tokens):
13        times_ms = []
14        for step, token in zip(times, tokens):
15            if len(self.tokenizer.decode(token).strip()) == 0:
16                continue
17            times_ms.append(step * 60)
18        return times_ms, self.tokenizer.decode(tokens)
19
20    def streaming_inference(self, audio, is_last):
21        self.fbanks.accept_waveform(audio, is_last)
22        features = self.fbanks.get_lfr_frames(
23            neg_mean=self.neg_mean, inv_stddev=self.inv_stddev
24        )
25        if is_last and len(features) == 0:
26            features = self.zeros
27        for idx, feature in enumerate(torch.unbind(torch.tensor(features), dim=0)):
28            is_last = is_last and idx == features.shape[0] - 1
29            self.caches = torch.roll(self.caches, -1, dims=0)
30            self.caches[-1, :] = feature
31            self.cur_idx += 1
32            cur_size = self.get_size()
33            if cur_size != self.chunk_size and not is_last:
34                continue
35            probs = self.inference(self.caches)[self.padding :]
36            if cur_size != self.chunk_size:
37                probs = probs[self.chunk_size - cur_size :]
38            if not is_last:
39                probs = probs[: self.chunk_size]
40            if self.beam_size > 1:
41                res = self.decoder.ctc_prefix_beam_search(
42                    probs, beam_size=self.beam_size, is_last=is_last
43                )
44                times_ms, text = self.decode(res["times"][0], res["tokens"][0])
45            else:
46                res = self.decoder.ctc_greedy_search(probs, is_last=is_last)
47                times_ms, text = self.decode(res["times"], res["tokens"])
48            yield {"timestamps": times_ms, "text": text}
```


LLM 大语言模型引擎：Qwen3-4B + AgentUniverse

Qwen3-4B 是阿里巴巴最新一代的大语言模型，以其卓越的性能效率和强大的多语言、工具调用能力著称。其性能可与上一代更大参数量的模型相媲美，非常适合在本地有限算力下进行部署。



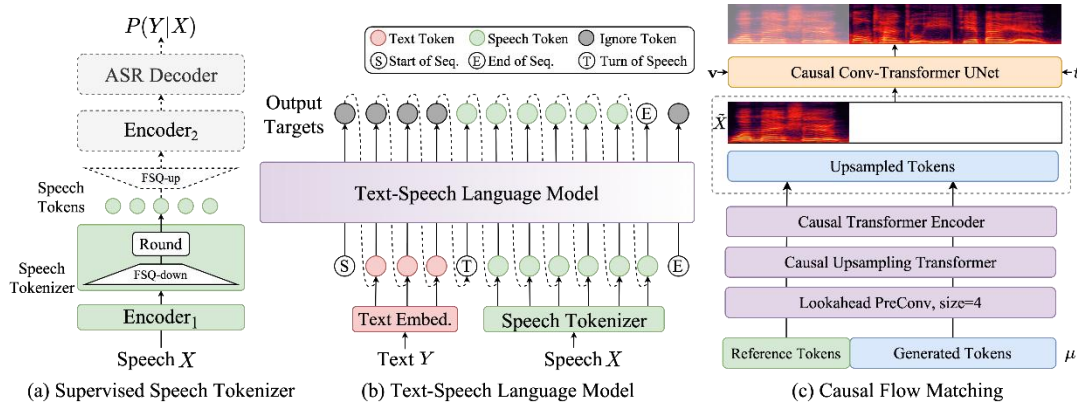
我们将 Qwen3-4B 深度集成到 AgentUniverse 多智能体框架中。数字人不再是一个孤立的对话模型，而是作为一个“伴学智能体”（Companion Agent），能够与其他智能体（如“知识库检索智能体”、“学习规划智能体”）协同工作。例如，当学生提出一个复杂问题时，伴学智能体会调用知识库智能体去检索最新资料，并将结果整合后以更易懂的方式回答，实现了知识的动态更新和深度整合。

核心代码示例如下：

```
1 class QWenOpenAISTyleLLM(OpenAISTyleLLM):
2     api_key: Optional[str] = Field(default_factory=lambda:
3     get_from_env("DASHSCOPE_API_KEY"))
4     api_base: Optional[str] = Field(default_factory=lambda: get_from_env(
5     "DASHSCOPE_API_BASE") or "https://dashscope.aliyuncs.com/compatible-mode/v1")
6     proxy: Optional[str] = Field(default_factory=lambda: get_from_env("DASHSCOPE_PROXY"))
7     organization: Optional[str] = Field(default_factory=lambda:
8     get_from_env("DASHSCOPE_ORGANIZATION"))
9
10    def _call(self, messages: list, **kwargs: Any) -> Union[LLMOutput,
11    Iterator[LLMOutput]]:
12        return super()._call(messages, **kwargs)
13
14    async def _acall(self, messages: list, **kwargs: Any) -> Union[LLMOutput,
15    AsyncIterator[LLMOutput]]:
16        return await super()._acall(messages, **kwargs)
17
18    def max_context_length(self) -> int:
19        if super().max_context_length():
20            return super().max_context_length()
21        return QWen_Max_CONTEXT_LENGTH.get(self.model_name, 8000)
22
23    def get_num_tokens(self, text: str) -> int:
24        tokenizer = get_tokenizer(self.model_name)
25        return len(tokenizer.encode(text))
```

TTS 语音合成引擎：CosyVoice 2.0

CosyVoice 2.0 在发音准确性、语音自然度和响应速度上均有显著提升，其发音错误率相较于 1.0 版本降低了 30%-50%。更重要的是，它支持通过自然语言指令控制语音的情感 and 韵律，这对于营造有温度的伴学体验至关重要。



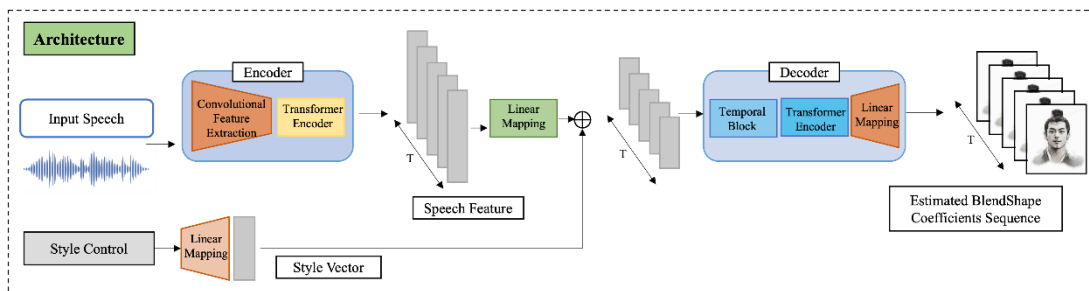
我们利用其情感控制能力，根据 LLM 对对话上下文的情感分析结果，动态调整 TTS 输出的语调。例如，在学生回答正确时，数字人会用更欢快、鼓励的语气；在学生遇到困难时，则会切换到更温和、耐心的语气，实现真正的情感化陪伴。

核心代码示例如下：

```
1 class HandlerTTS(HandlerBase, ABC):
2     def start_context(self, session_context, context: HandlerContext):
3         context = cast(TTSTContext, context)
4         output_definition = self.get_handler_detail(session_context,
5             context).outputs.get(ChatDataType.AVATAR_AUDIO).definition
6
7     def task_consumer(task_inner_queue: deque, callback: callable):
8         while True:
9             if len(task_inner_queue) == 0:
10                 time.sleep(0.03)
11                 continue
12             task = task_inner_queue[0]
13             task = cast(HandlerTask, task)
14             if task is None:
15                 break
16             try:
17                 audio = task.result_queue.get(timeout=1)
18                 if audio is not None:
19                     output = DataBundle(output_definition)
20                     output.set_main_data(audio)
21                     output.add_meta("avatar_speech_end",
22                         False if not task.speech_end else True)
23                     output.add_meta("speech_id", task.speech_id)
24                     callback(output)
25                     if context.dump_audio:
26                         dump_audio = audio
27                         context.audio_dump_file.write(dump_audio.tobytes())
28                 else:
29                     task_inner_queue.popleft()
30             except Exception as e:
31                 logger.debug(e)
32
33     context.task_consume_thread = threading.Thread(target=task_consumer, args=
34         [context.task_queue, context.submit_data])
35     context.task_consume_thread.start()
36     self.task_queue_map[context.session_id] = context.task_queue
```

Avatar 数字人驱动引擎：LAM

LAM 是一种创新的大型数字人模型，能够根据单张图片快速生成超写实的 3D 高斯头像，并支持跨平台实时驱动和渲染。这极大地简化了数字人形象的创建流程，未来可支持学生上传自己的照片生成专属学习伙伴。



核心代码示例如下：

```
1 class HandlerAvatarLAM(HandlerBase):
2     def handle(self, context: HandlerContext, inputs: ChatData,
3               output_definitions: Dict[ChatDataType, HandlerDataInfo]):
4         ...
5         audio = inputs.data.get_main_data()
6         audio_segments = queue.Queue()
7         for audio_segment in slice_data(context.input_slice_context, audio.squeeze()):
8             audio_segments.put_nowait(audio_segment)
9         if speech_end:
10            end_segment = context.input_slice_context.flush()
11            if end_segment is not None:
12                audio_segments.put_nowait(end_segment)
13        if audio_segments.empty() and speech_end:
14            audio_segments.put_nowait(np.zeros([50], dtype=np.float32))
15        while not audio_segments.empty():
16            t_start = time.monotonic()
17            audio_segment = audio_segments.get_nowait()
18            result, context_update = self.infer.infer_streaming_audio(
19                audio=audio_segment,
20                ssr=context.config.audio_sample_rate,
21                context=context.inference_context,
22            )
23            context.inference_context = context_update
24            need_flush = speech_end and audio_segments.empty()
25            output = DataBundle(output_definition)
26            arkit_data = result.get("expression")
27            start_of_stream = speech_id != context.last_speech_id
28            output.set_main_data(arkit_data.astype(np.float32))
29            output.set_data("avatar_audio", audio_segment[np.newaxis, ...])
30            output.add_meta("speech_id", speech_id)
31            output.add_meta("avatar_speech_end", need_flush)
32            output.start_of_stream = start_of_stream
33            output.end_of_stream = need_flush
34            if speech_text is not None:
35                output.add_meta("avatar_speech_text", speech_text)
36            dur_inference = time.monotonic() - t_start
37            context.submit_data(output)
38            context.last_speech_id = speech_id
```


企事业单位的生态融合共建

团队就“数字人伴学系统”赴浙江联通开展了实地调研，重点围绕数字人技术在教育场景中的落地可行性、算力基础设施支持、以及应用数据的安全合规性等方面进行了深入交流。调研主要内容包括：

技术落地场景探讨：

在与浙江联通行业解决方案团队沟通后，了解到数字人教学场景在 K12 及高等教育中的实际应用案例，同时明确了其在智慧教室、远程教学、个性化辅导等场景中，对低延迟、高并发、多模态交互的数字人技术需求。



算力与网络基础设施支持：

浙江联通介绍了其在“算网融合”方面的布局，包括边缘计算节点、5G 专网和教育云平台，可为数字人系统提供稳定、低延迟的运行环境。团队据此验证了将数字人伴学系统进行本地部署与云边协同部署两种模式的可行性。

数据安全与合规建议：

针对学生隐私保护和教育数据安全，浙江联通强调了教育行业数据规范的重要性，建议在设计系统中强化身份认证、数据加密和访问审计机制，这与企业文档中提出的 OAuth2、JWT、HTTPS 等安全要求高度一致；同时，就数字人系统在实际场景中的应用规范，团队也与浙江联通展开了相关探讨。

团队系统将严格遵循企业文档中建议的安全规范，采用 OAuth2 和 JWT 进行用户认证授权，并通过 HTTPS 加密所有数据传输，确保学生隐私数据的安全；前端界面也将采用蚂蚁集团的 Ant Design 组件库进行开发，与蚂蚁系产品的 UI/UX 风格一致，确保团队产品良好的前端可交互性、降低用户学习成本。

除此之外，在数字人系统的应用安全层面上，声像科技团队以“守正创新”为核心理念，通过多模态内容分析、实时数字人交互和深度伪造检测技术，构建起一套安全、智能、高效的数字人伴学系统。融合语音、图像、视频的多模态感知能力，为教育行业提供从内容生成到安全鉴伪的全产业链服务，助力教学方式的数字化转型。

声像科技团队同时也研发了一套数字人深度伪造鉴别系统，助力生态共建的相关企事业单位实现从系统到应用层面的全方位安全防护；结合自主研发的多模态声纹特征提取与筛选神经网络、离散余弦引导与动态特征选择等技术，可以鉴别合成音视频的真伪，确保数字人合成内容的安全合规且可控，响应国家对 AI 伦理的监管要求。

与蚂蚁集团生态的融合

SenseVoice：多模态语音识别，理解更全面

SenseVoice 是达摩院推出的工业级语音基础模型，集 ASR、语种识别（LID）、情感识别（SER）于一体。其融合优势在于：

伪流式处理优化：团队采用分块+截断注意力机制，在保证识别准确率的同时，将语音输入到系统响应的延迟压缩至可接受范围（约 1 秒），提升对话流畅性。

情感与语境感知：不仅可以识别“说了什么”，还能感知“以什么情绪说的”。该情感信息可同步至 LAM（驱动表情）和 CosyVoice（调整回应语气），形成“听-感-说-演”的多模态闭环。

多语种支持：天然支持中、英、日等多语种识别，为英语、日语等语言学科的游戏化学习平台提供底层语音交互能力。

Qwen 大模型：智能内核，源于阿里、服务教育

Qwen 系列大模型由阿里巴巴通义实验室研发，是蚂蚁集团 AI 技术生态的重要组成部分。在数字人伴学系统中，我们采用 Qwen3-4B 作为核心语言理解与生成引擎，其融合体现在：

能力先进性：Qwen3 在参数效率、多语言支持、工具调用（Function Calling）等方面表现卓越，能在有限本地算力下提供接近更大模型的对话质量，契合高校学生对知识广度与深度的需求。

与 AgentUniverse 协同：Qwen 作为“伴学智能体”的大脑，深度集成于蚂蚁集团开源的 AgentUniverse 多智能体框架中，可主动调用知识库检索、学习规划等其他智能体，实现从“被动问答”到“主动引导”的跃迁。

安全可控：依托蚂蚁集团在大模型内容安全、价值观对齐方面的成熟机制，确保数字人输出内容符合教育伦理，杜绝有害或误导性信息。

CosyVoice 2.0：情感化语音合成，让声音有温度

CosyVoice 是通义实验室推出的语音大模型，其 2.0 版本在自然度、可控性和多语言支持上显著提升。在系统中的融合体现为：

情感韵律控制：通过自然语言指令（如“用鼓励的语气”、“用耐心的语调”），动态调节语音的情感色彩，使数字人不仅是“会说话”，更是“懂情绪”的学习伙伴。

低错误率与高自然度：发音错误率较 1.0 版本降低 30%-50%，确保专业术语（如化学式、数学符号）的准确朗读，提升学习体验的专业性。

与 Qwen 联动：LLM 在生成文本时即可预置情感标签，TTS 引擎据此自动匹配语音风格，实现端到端的情感化语音输出。

LAM 数字人驱动引擎：超写实形象，快速生成，沉浸交互

LAM (Large Avatar Model) 是通义实验室推出的大型数字人模型，代表了蚂蚁-阿里生态在 AIGC 与虚拟人领域的前沿探索。其融合价值在于：

形象个性化：LAM 支持从单张图片快速生成高保真 3D 数字人形象，未来可让学生上传照片生成“专属学习搭子”，极大提升情感连接与使用黏性。

实时驱动与渲染：LAM 具备跨平台实时驱动能力，结合情感识别结果，可实现口型、表情、眼神的自然同步，营造沉浸式陪伴感。

轻量化部署：LAM 优化了推理效率，可在配备消费级 GPU 乃至手机等移动设备端上流畅运行，降低高校或学生个人的使用门槛。

团队就数字人应用系统申请了两项软件著作权并取得授权：

1. 软件名称：《面向深度伪造音视频的智能检测软件》

证书号：软著登字第 16024611 号

授权日期：2025 年 7 月 25 日

2. 软件名称：《终身学伴-数字虚拟人合成平台》

证书号：软著登字第 16545571 号

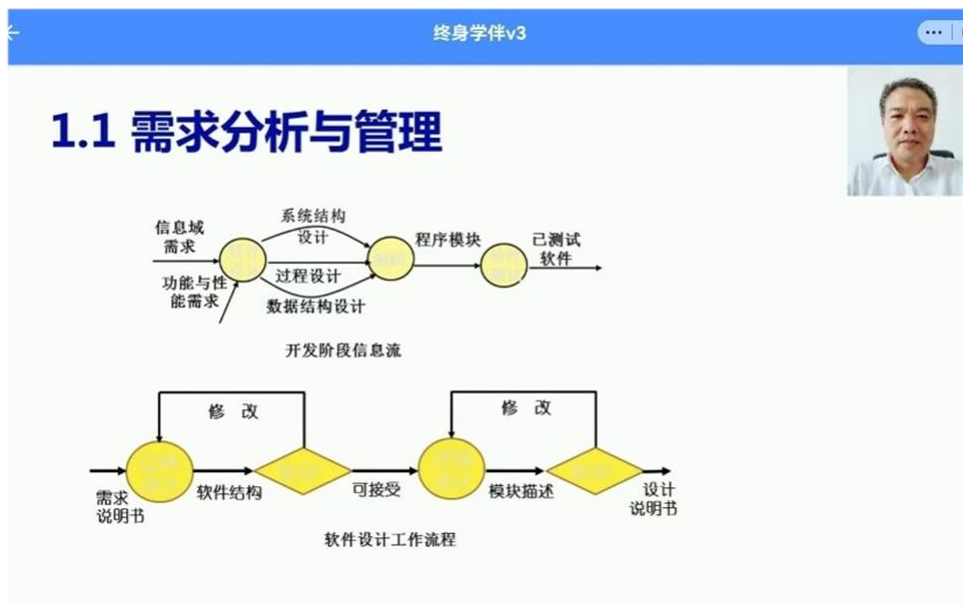
授权日期：2025 年 9 月 28 日



其中《面向深度伪造音视频的智能检测软件》主要面向数字人伴学系统的安全层面，为数字人的内容安全提供了相关保障与支持；《终身学伴-数字虚拟人合成平台》则面向数字人伴学系统的应用层面，搭建了一套高度模块化的数字人应用平台。

系统部署与应用：

声像科技团队所研发的数字人伴学系统在本校数字化办公室得到了实际部署和应用；以下是数字人伴学系统在实际产品中的应用画面：



产品迭代方向

V1.5: 实现 ASR 情感识别 (SER) 和 Avatar 数字人驱动引擎中表情驱动的联动效果，增强数字人情感交互的真实性。

V2.0: 打通与“多学科客制化评估系统”的数据接口。当评估系统发现学生知识薄弱点时，可自动触发数字人进行针对性的讲解和鼓励。

V3.0+: 探索与“AI 豆计划”等蚂蚁公益项目的结合，为特定群体（如乡村学生）提供定制化的数字人伴学服务，践行普惠教育理念。

多智能体-知识库协作系统

本系统是“AI 智能·学习搭子”解决方案的知识中枢和智能引擎。它并非一个孤立的问答机器人，而是依托蚂蚁集团在复杂金融业务场景中淬炼出的 AgentUniverse 多智能体框架，构建了一个由多个领域专家智能体协同工作的动态知识网络。其核心价值在于，能够将静态的知识库转化为可推理、可协作、可进化的活知识库，为数字人伴学系统和评估系统提供强大、精准、实时的智能支持。



AgentUniverse 是一个面向复杂业务场景设计的多智能体协作框架，其核心是一个“模式工厂（pattern factory）”，允许开发者对多智能体协作模式进行开发和定制。我们基于此框架，设计了以下关键智能体及其协作流程：

知识库检索智能体(Knowledge Retrieval Agent):

作为系统的“图书管理员”，负责监听来自数字人伴学系统或评估系统的查询请求。它能理解查询的语义，并在我们构建的多学科、多来源（教材、论文、开源项目、行业报告）的知识库中进行高效、精准的检索。

技术实现：采用向量数据库（如 Milvus、FAISS）存储知识嵌入，并结合关键词索引，实现混合检索，确保召回率和准确率。

核心代码示例如下：

```
1 class Knowledge(ComponentBase):
2     def update_knowledge(self, **kwargs) -> None:
3         document_list: List[Document] = self._load_data(**kwargs)
4         document_list = self._update_process(document_list)
5         futures = []
6         if "stores" in kwargs:
7             stores = kwargs["stores"]
8         else:
9             stores = self.stores
10        for _store_code in stores:
11            futures.append(
12                self.insert_executor.submit(
13                    StoreManager().get_instance_obj(_store_code).update_document,
14                    document_list))
15        wait(futures, return_when=ALL_COMPLETED)
16        for future in futures:
17            future.result()
```


内容生成与摘要智能体(Content Generation & Summarization Agent):

作为系统的“内容编辑”，接收检索智能体返回的原始资料。它利用大语言模型（如 Qwen3）的能力，对冗长、专业的原始信息进行提炼、总结、改写，生成适合学生当前认知水平和学习阶段的、易于理解的答案或学习材料。

技术实现：该智能体被配置为遵循特定的“写作规范”，确保输出内容的准确性、简洁性和教育性；同时在应用接口返回数据时被严格规范为使用 JSON 格式进行返回，这也有效保证了智能体对各个子系统的有效兼容。

```
1 [CORE_PACKAGE]
2 default = ['discussion_group_app.intelligence.agentic']
3 agent = ['discussion_group_app.intelligence.agentic.agent']
4 knowledge = ['discussion_group_app.intelligence.agentic.knowledge']
5 llm = ['discussion_group_app.intelligence.agentic.llm']
6 planner = []
7 tool = ['discussion_group_app.intelligence.agentic.tool']
8 memory = ['discussion_group_app.intelligence.agentic.memory']
9 service = ['discussion_group_app.intelligence.service.agent_service']
10 prompt = ['discussion_group_app.intelligence.agentic.prompt']
11 store = ['discussion_group_app.intelligence.agentic.knowledge.store']
12 rag_router = ['discussion_group_app.intelligence.agentic.knowledge.rag_router']
13 doc_processor = ['discussion_group_app.intelligence.agentic.knowledge.doc_processor']
14 query_paraphraser =
15     ['discussion_group_app.intelligence.agentic.knowledge.query_paraphraser']
16 memory_compressor = ['discussion_group_app.intelligence.agentic.memory.memory_compressor']
17
18 memory_storage = ['discussion_group_app.intelligence.agentic.memory.memory_storage']
```

学习规划智能体(Learning Planner Agent):

作为系统的“学业顾问”，它与“多学科客制化评估系统”深度联动。当评估系统诊断出学生的薄弱知识点后，该智能体会综合学生的学习目标（校园考试/就业技能）、历史学习数据和知识库内容，动态生成或调整个性化的学习路径和任务清单。

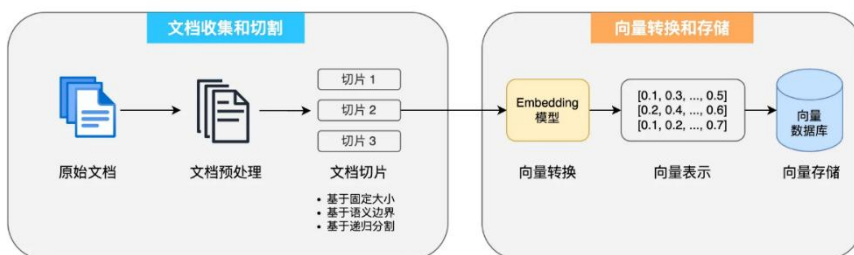
```
1 introduction: 你是一位专业的学业规划顾问，精通教育心理学和学习科学。
2 target: 基于多学科客制化评估系统的诊断结果，为学生制定个性化的学习路径和任务清单，帮助学生实现学习目标。
3 instruction: |
4     你需要遵守的规则是：
5     1. 深度分析评估系统提供的学生知识薄弱点诊断结果，准确理解学生的学习现状。
6     2. 综合考虑学生的学习目标类型（校园考试提升/就业技能培养），制定针对性的学习策略。
7     3. 基于学生的历史学习数据，识别学习偏好、学习节奏和有效的学习方式。
8     4. 利用知识库内容，为每个薄弱知识点匹配最适合的学习资源和练习材料。
9     5. 动态生成分阶段的学习路径，包括：基础巩固→能力提升→综合应用三个层次。
10    6. 制定具体可执行的任务清单，包括学习时间安排、里程碑检查点和评估标准。
11    7. 根据学生的学习进度反馈，实时调整和优化学习计划。
12    8. 提供学习方法指导，包括记忆技巧、理解策略和应用练习建议。
13    9. 设置激励机制和成就感培养，保持学生的学习动力。
14    10. 与评估系统保持联动，定期更新学习计划以适应学生能力的变化。
15    .....
16    学习计划周期: {planning_period}天
17    当前规划阶段: 第{current_phase}阶段（共{total_phases}个阶段）
18
19    请用中文回答，需要规划的学习需求是: {input}
20 metadata:
21     type: 'PROMPT'
22     version: academic_planner_agent.cn
23
```

反思与进化智能体(Reflection & Evolution Agent):

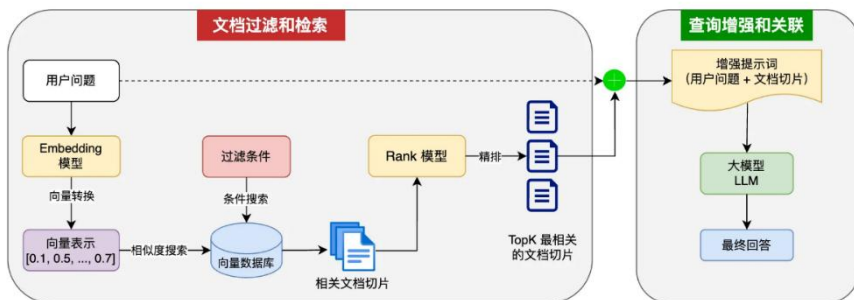
作为系统的“质量监督员”和“进化引擎”，它会定期分析所有智能体的交互日志和用户反馈。通过反思（Reflection）机制，它能发现知识库的缺失、答案的不准确或规划的不合理之处，并自动触发知识库的更新流程或向管理员发出优化建议，实现系统的持续自我进化，让系统“越用越懂你”。

RAG 检索增强生成工作流程

建立索引

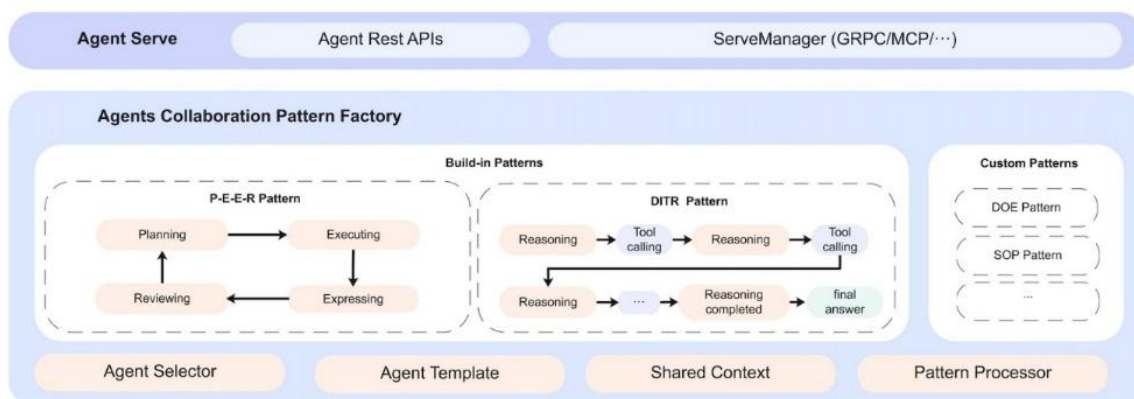


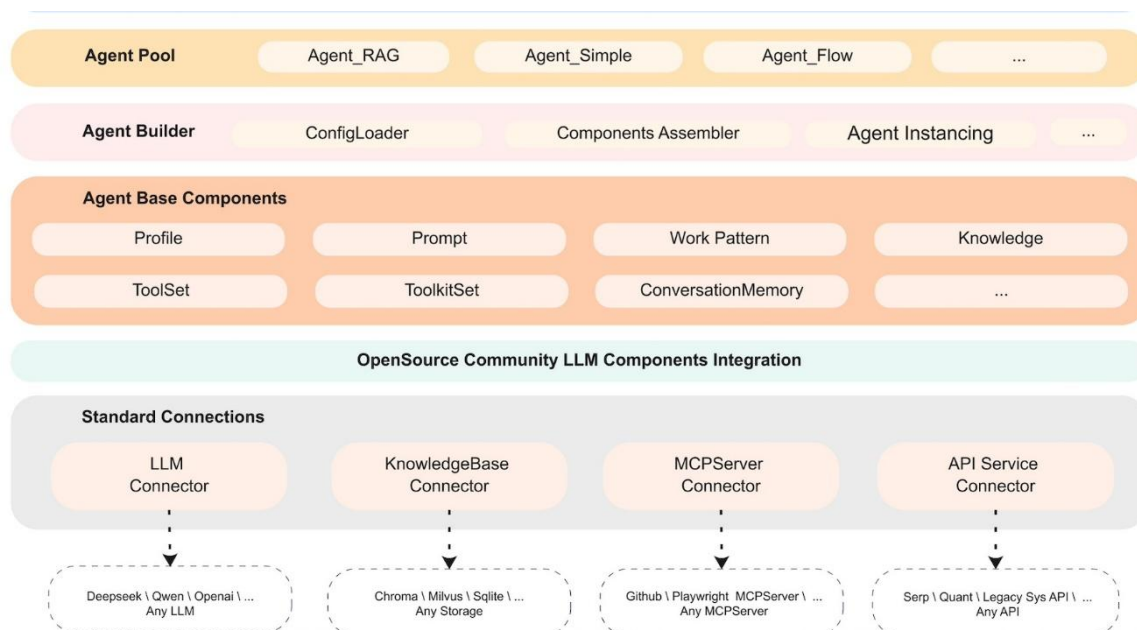
检索生成



与蚂蚁集团生态的融合

AgentUniverse 框架来源于蚂蚁集团探索大模型技术在金融等严谨产业场景中的最新成果，已在真实业务中得到验证。这为我们系统的稳定性、可靠性和处理复杂任务的能力提供了坚实基础。





除 AgentUniverse 框架外，系统在整体架构上全面融入蚂蚁技术生态：

所有智能体均以 Qwen 系列大模型（如 Qwen3-4B）为推理引擎，该模型由阿里通义实验室研发，是蚂蚁 AI 生态的核心组件。统一的模型底座确保了智能体间语义理解的一致性，并便于利用蚂蚁在模型安全、价值观对齐等方面的成熟机制；智能体间的通信部署在微服务+ Service Mesh 架构之上，通过蚂蚁推荐的 OAuth2 + JWT 机制进行认证授权，并采用 HTTPS 协议进行加密传输，确保知识交互过程中的数据安全与隐私合规，符合教育行业对数据安全的严苛要求。

系统中的“学习规划智能体”可直接对接数字马力在产教融合中积累的行业技能图谱与岗位能力模型。当学生进入“二阶段”（就业导向）学习时，智能体能基于真实岗位需求（如“Java 后端开发工程师”），动态生成包含技术栈、项目经验、面试题库在内的个性化学习路径，有效弥合高校教学与市场需求的鸿沟；并可作为核心能力模块，打包接入蚂蚁公益平台，以技术手段推动教育公平。

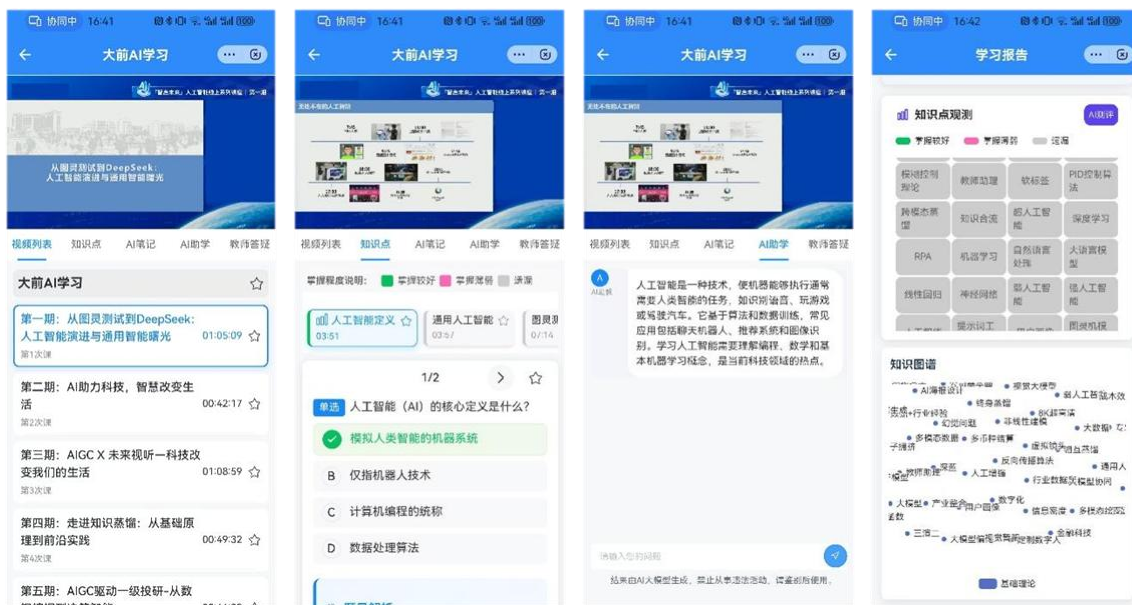
技术落地场景探讨：

团队赴云栖大会调研了多个基于多智能体架构的行业解决方案，如智能客服、科研文献分析助手、企业知识管理平台等。这些案例为团队在“知识检索—内容生成—学习规划”智能体协作流程的设计上提供了宝贵参考；团队特别关注了相关企业在云栖大会展示的教育科技解决方案和产教融合实践成果，并确认了将多智能体系统和数字马力已有的“助教、助学、助就业”能力体系深度融合的可行性。



系统部署与应用：

声像科技团队所研发的多智能体-知识库协作系统在本校数字化办公室得到了实际部署和应用；以下是多智能体-知识库协作系统在实际产品中的应用画面：



产品迭代方向

V1.5：实现“反思与进化智能体”的初步功能，建立基于用户反馈的半自动知识库更新闭环，实现多智能体-知识库协作系统“越用越懂你”的研发目标。

V2.0：引入更多垂直领域的专家智能体（如“编程导师智能体”、“金融分析师智能体”），以更好地支撑“二阶段学习场景”（就业导向）的需求，直接对接数字马力在产教融合中积累的行业技能图谱。

V3.0+：探索将整个多智能体协作网络开放为一个教育智能体市场，允许高校教师或合作教育机构贡献自己的教学智能体，共同丰富“AI 智能·学习搭子”的知识生态，构建一套完整的、面向未来的 MCP 智能体协作系统。

通过构建这样一个基于 AgentUniverse 的多智能体-知识库协作系统，团队不仅解决了知识问答的准确性问题，更构建了一个能够主动规划、持续进化、紧密贴合“从校园到职场”双阶段学习目标的动态智能学习引擎，有力强化了“AI 智能·学习搭子”的核心竞争力。

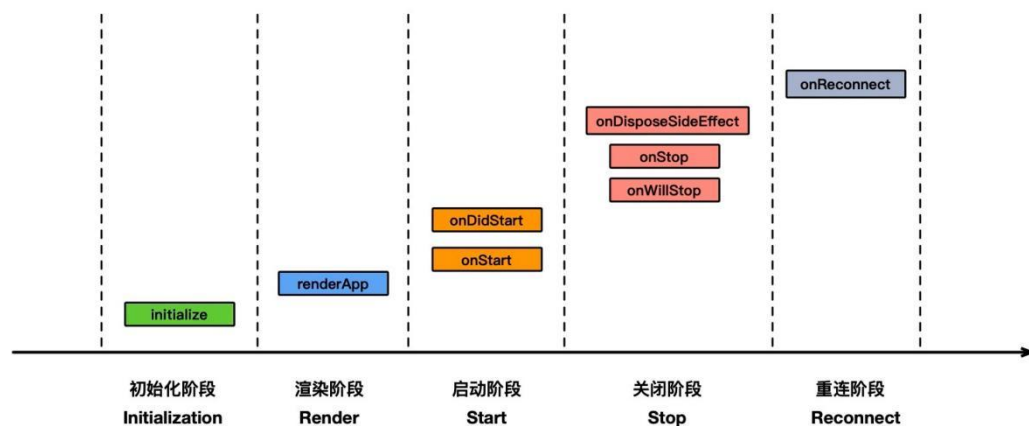
多学科客制化评估系统

本系统是“AI 智能·学习搭子”解决方案中实现精准诊断和效果闭环的关键一环。它并非一个通用的题库系统，而是针对不同理工科学科的独特知识结构和能力评估标准，提供高度客制化的测评和反馈服务。系统旨在解决“学得怎么样”和“哪里不会”的核心问题，为数字人伴学系统和学习规划提供数据驱动的决策依据，契合赛题中“一阶段（校园考试）”与“二阶段（就业技能）”的双场景需求。

我们以高校计算机学科的实际教学为切入点，构建了首个客制化评估模块，并为该系统在其他学科的快速扩展奠定了技术基础；其采用蚂蚁集团开源的 OpenSumi IDE 框架作为核心载体。OpenSumi 是一个高性能、高可定制的双端（Web/Electron）IDE 开发框架，能够快速构建出云端或桌面端的集成开发环境。

生命周期贡献点

ClientAppContribution



基于 OpenSumi 提供的前端 IDE 框架，团队集成了隔离的代码执行沙箱环境。学生可以直接在浏览器中编写、运行和调试代码，无需担心环境配置问题；同时，团队也开发了与项目深度集成的代码自动评测系统。该引擎不仅能判断代码的正确性（通过单元测试），还能从代码风格、算法效率、内存占用等多个维度进行综合评分，提供远超“对/错”的精细化反馈；同时，结合大语言模型（LLM），对学生的错误代码进行智能分析，不仅能指出错误原因，还能生成针对性的修复建议和相关知识点链接，实现“以评促学”，让编程学习不止停留在“纸上谈兵”。其核心代码示例如下：

```
1 import { AppRenderer, EditorRenderer } from '@codeblitzjs/ide-core';
2 function App() {
3   return <AppRenderer appConfig={{}} />
4 }
5 function Editor() {
6   return <EditorRenderer appConfig={{}} />
7 }
8
```

针对高校计算机学科在线评测系统（Online Judge, OJ）的实际需要，声像科技团队也搭建了对应的代码评测沙箱，其核心代码示例如下：

```
1 @Override
2 public ExecuteCodeResponse executeCode(ExecuteCodeRequest executeCodeRequest) {
3     List<String> inputList = executeCodeRequest.getInputList();
4     String code = executeCodeRequest.getCode();
5     String language = executeCodeRequest.getLanguage();
6     // 1. 把用户的代码保存为文件
7     File userCodeFile = saveCodeToFile(code);
8     // 2. 编译代码, 得到 class 文件
9     ExecuteMessage compileFileExecuteMessage = compileFile(userCodeFile);
10    System.out.println(compileFileExecuteMessage);
11    // 3. 执行代码, 得到输出结果
12    List<ExecuteMessage> executeMessageList = runFile(userCodeFile, inputList);
13    // 4. 收集整理输出结果
14    ExecuteCodeResponse outputResponse = getOutputResponse(executeMessageList);
15    // 5. 文件清理
16    boolean b = deleteFile(userCodeFile);
17    if (!b) {
18        log.error("deleteFile error, userCodeFilePath = {}",
19            userCodeFile.getAbsolutePath());
20    }
21    return outputResponse;
22 }
```

对用户提交的代码进行编译与执行, 其核心代码示例如下:

```
1 public List<ExecuteMessage> runFile(File userCodeFile, List<String> inputList) {
2     String userCodeParentPath = userCodeFile.getParentFile().getAbsolutePath();
3     List<ExecuteMessage> executeMessageList = new ArrayList<>();
4     for (String inputArgs : inputList) {
5         String runCmd = String.format("java -Xmx256m -Dfile.encoding=UTF-8 -cp %s Main %s",
6             userCodeParentPath, inputArgs);
7         try {
8             Process runProcess = Runtime.getRuntime().exec(runCmd);
9             new Thread(() -> {
10                 try {
11                     Thread.sleep(TIME_OUT);
12                     runProcess.destroy();
13                 } catch (InterruptedException e) {
14                     throw new RuntimeException(e);
15                 }
16             }).start();
17             ExecuteMessage executeMessage =
18                 ProcessUtils.runProcessAndGetMessage(runProcess, "运行");
19             System.out.println(executeMessage);
20             executeMessageList.add(executeMessage);
21         } catch (Exception e) {
22             throw new RuntimeException("执行错误", e);
23         }
24     }
25     return executeMessageList;
26 }
```

跨学科扩展框架：

针对其他学科的实际需要，声像科技团队也搭建了相关的学科题库，以更好服务于团队所研发的多智能体-知识库协同系统与实际教育教学过程中的客观需要。



面向部分学科，我们也搭建了相应的跨学科拓展框架。

数学：团队通过集成视觉大模型（Qwen-VL）实现数学公式识别和符号计算；学生可以手写或输入数学公式，系统能自动解析并评估其推导过程的逻辑性和正确性。

化学：团队基于 Three.js 的前端 3D 界面构建分子结构的可视化模型；使得教师能够在实际授课中全面展现分子结构，实现可视化的教学过程。

通用技术：团队结合教学过程中的授课 PPT、教科书 PDF 等多元信息，对 Qwen3 大模型进行微调（Fine-Tuning），实现题目对应相关知识的溯源过程。

与整体解决方案的深度协同

数据驱动学习规划：评估系统的诊断结果（如“动态规划算法掌握薄弱”、“有机化学反应机理理解不清”）会实时同步给“多智能体-知识库协作系统”中的学习规划智能体，后者据此动态调整学生的学习路径，推送针对性的学习资料和练习题。

赋能数字人伴学：当学生在评估中受挫时，评估结果会触发数字人伴学系统的主动关怀。数字人会以鼓励的语气，结合评估报告，与学生一起分析错因，并引导其进入复习流程，实现情感激励和知识巩固的结合。

衔接“二阶段”就业场景：针对“就业导向”的学习目标，评估系统可直接对接数字马力产教融合项目中定义的行业技能图谱。例如，对“后端开发工程师”岗位，系统可模拟真实的编码面试题或项目任务进行评估，并生成一份对标企业用人标准的能力雷达图，让学生清晰了解自身与岗位要求的差距。

系统部署与应用：

声像科技团队所研发的多学科客制化评估系统在浙江省温州中学和本校计算机科学与技术学院得到了实际部署和应用；以下是面面向前者的多学科客制化评估系统在实际产品中的应用画面与试用报告：



与蚂蚁集团生态的融合

声像科技团队依据高校计算机学科的实际教学需要，结合蚂蚁集团的 OpenSumi 前端，开发了客制化的编程 IDE；默认集成 Ant Design 组件库，确保评估界面与蚂蚁系产品风格统一，操作逻辑符合用户习惯，降低高校师生的学习成本。

系统严格遵循企业文档提出的安全规范；通过 OAuth2 + JWT 实现用户认证，所有代码执行均在隔离的安全沙箱中进行，防止恶意代码注入；数据传输全程采用 HTTPS 加密，保障学生代码与评估结果的数据信息安全。

评估系统的评测引擎、题库管理、结果分析等模块可拆分为独立微服务，通过 Service Mesh 进行治理，与整体“AI 智能·学习搭子”解决方案的技术架构保持一致，实现 SaaS（Software as a Service）便于未来能力复用与弹性扩展。

评估系统可对接数字马力产教融合体系；可接入高校共建的课程标准与技能图谱，实现客制化服务定制。例如，针对“Java 后端开发”岗位，系统可自动生成包含 Spring Boot、MySQL、Redis 等技术栈的综合项目题，并依据企业用人标准进行多维度评分，实现“学即所用、评即所聘”。

数据驱动学习闭环；用户在使用评估系统时的结果（如“算法效率低”、“异常处理缺失”）将实时同步至多智能体-知识库系统，触发学习规划智能体生成针对性复习计划；同时，该数据也会反馈给数字人伴学系统，由数字人主动关怀并引导学生查漏补缺，形成“评估—规划—陪伴—再评估”的完整闭环。

通过以上所述的多学科客制化评估系统，声像科技团队不仅实现了技术上的工程化、标准化与可扩展性，更在业务层面深度融入了数字马力的产教融合实践与蚂蚁集团的开发者工具生态。这种融合将企业级的工程能力创造性地应用于教育评估领域，为“AI 智能·学习搭子”项目提供了真实、精准、可衡量的能力验证手段，有力支撑了“从校园到职场”的双阶段学习目标，更彰显着服务外包大赛“来源实际需求，鼓励创新应用”的核心原则。

产品迭代方向

V1.5: 完成并完善物理、生物等 1-2 个新学科的评估模块原型开发，验证跨学科扩展框架的通用性。

V2.0: 深化与“数字马力”就业技能体系的对接，开发面向热门岗位（如数据分析、前端开发）的专项技能评估套件。

V3.0+: 探索将评估能力封装为符合 MCP/A2A 协议的标准智能体服务，开放给高校教师或合作教育机构，允许他们基于我们的框架自定义和发布自己学科的评估内容，共建共享评估生态。

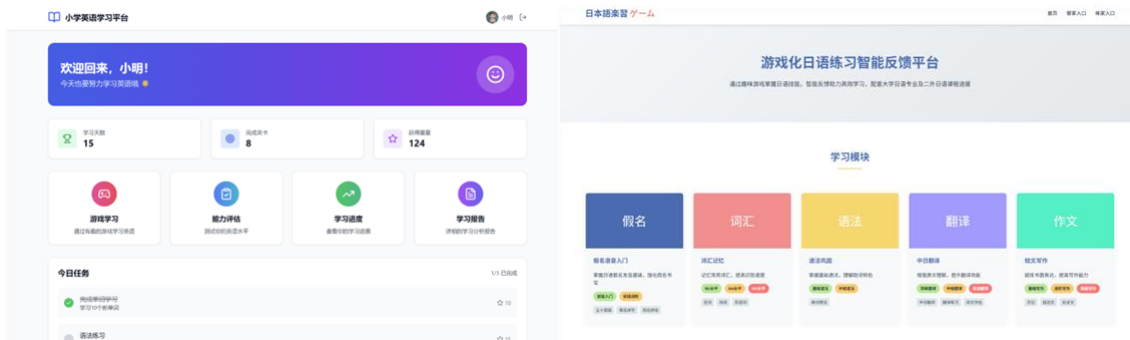
通过构建这样一个灵活、精准、多学科的客制化评估系统，我们不仅为学生提供了衡量学习效果的“标尺”，更打通了“学-练-测-评-优”的完整闭环，有力支撑了“AI 智能·学习搭子”从个性化陪伴到精准化提升的核心价值。

游戏化学习平台及评估系统

本系统是“AI 智能·学习搭子”解决方案中面向入门级学习者（如小学生）和兴趣驱动型学习场景（如语言学习）的核心模块。其核心价值在于，通过将游戏化（Gamification）设计理念深度融入学习过程，有效解决低龄或初学者“学习动力不足、注意力难以集中、缺乏即时正向反馈”的痛点，让学习变得像游戏一样有趣、有挑战、有成就感，从而激发内在学习动机，为后续的进阶学习打下坚实基础。

技术架构与游戏化设计

我们以小学英语为切入点，构建了首个游戏化学习模块，并为其他语言类学科（如日语）的快速扩展提供了范式。



技术基座：采用蚂蚁集团主导开发的 Ant Design 企业级 UI 设计系统作为前端开发框架。Ant Design 不仅提供了丰富、美观、一致的 UI 组件，其设计价值观中也蕴含着“创造快乐工作”的理念，这与我们的游戏化目标高度契合。我们利用其响应式布局能力，确保平台在平板和电脑上均能提供流畅、沉浸的用户体验。

核心游戏化机制：

角色与成长体系：学生创建自己的虚拟角色（Avatar），通过完成学习任务（如单词拼写、听力练习、口语对话）获得经验值（XP）和金币，用于解锁新装扮、新技能及进入更高级别的学习关卡。

即时反馈与奖励：团队借鉴了“百词斩”等成功产品的经验，系统在学生完成每个微小任务后，立即给予视觉（动画、徽章）和听觉（音效）的正向反馈，激励并强化用户学习行为，形成正向激励的学习循环。

挑战与任务系统：将学习目标分解为一系列由易到难的“每日挑战”和“主题任务”；任务设计紧密结合教学大纲，确保游戏性和教育性的统一，实现“以玩助学”。

智能评估与反馈：

嵌入式评估：评估不再是独立的考试，而是完全融入游戏流程中。例如，在“单

词句法对对碰”游戏中，玩家可以通过说出对应单词的词组匹配来消除对应的区块内容；在游戏过程中，系统会通过语音识别（ASR）和自然语言理解（NLU）技术，实时评估学生的发音准确度和句型使用是否恰当。

数据驱动个性化：平台会记录学生在游戏中的所有行为数据（如错误类型、反应时间、重试次数），并利用这些数据动态调整后续任务的难度，确保挑战始终与学生的能力相匹配，维持其“玩中学、学中玩”正向激励的学习循环。

与整体解决方案的深度协同

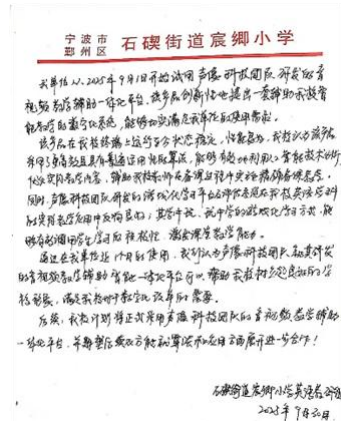
无缝衔接进阶学习：当学生在游戏化平台中达到一定水平时（如完成小学英语全部课程），系统会自动推荐并引导其进入“多学科客制化评估系统”中的更高阶英语学习路径，实现从“兴趣入门”到“目标导向”的平滑过渡。

赋能普惠教育：本系统适合通过蚂蚁公益平台进行推广。团队所研发的游戏化学习平台及评估系统可以被打包为一个公益项目，同“乡村儿童英语启蒙计划”等支付宝公益频道的“行为公益”（如用户捐步数）形成有机结合，为资源匮乏地区的学校和学生提供免费的、高质量的游戏化英语学习资源，践行数字马力的普惠教育使命。

融入教育机构合作：与数字马力已合作的高校及中小学合作，将本平台作为其英语教学的官方辅助工具，教师可以在后台查看班级整体和学生个体的学习数据报告，实现以玩助学、精准教学。

系统部署与应用：

声像科技团队所研发的游戏化学习平台及评估系统在宁波市海曙区宸卿小学与本校东方语言与哲学学院得到了实际部署与应用；以下是游戏化学习平台及评估系统在实际产品中的应用画面和试用报告：



产品迭代方向

V1.5：增加并完善日语、法语等第二外语的游戏化学习模块，并完善 AI 语音评测功能，从语调、语法等角度提供更专业的发音指导。

V2.0：深化与“数字人伴学系统”的结合，营造“元宇宙”式的游戏场景；在游戏中，数字人可以扮演 NPC（非玩家角色）和学生进行互动对话，营造身临其境的语言学习场景，提供更生动的沉浸式体验。

V3.0+：探索与蚂蚁集团 IP（如蚂蚁庄园）的联动，推出联名主题活动和虚拟道具，利用蚂蚁集团强大的用户触达能力，进一步扩大平台影响力。

成本模型及可行性分析

成本结构模型

本项目采用“轻资产+高复用”成本模型，核心成本集中在初期研发，后期以 SaaS 订阅与模块授权为主要收入来源。

初期投入成本

类别	明细	金额（人民币）
硬件设备	Tesla-T4 开发机 1 台（校内提供） /	¥8,500/
	Nvidia RTX 5080 开发机 1 台（校内提供）	¥16,000
云资源与 API	阿里云 GPU 实例、语音识别调用等	¥100
安全合规认证	SSL 证书、域名申请、软件著作权申请	¥400
合计		¥25,000

可行性分析

本项目在技术、经济、政策与生态四个维度均具备高度可行性。

技术可行性

所有核心技术（如 Qwen3-4B、SenseVoice、LAM、AgentUniverse 等）均为阿里-蚂蚁集团开源或内部验证组件，技术风险低；

团队已在校内完成端到端 Demo，功能闭环验证通过。

经济可行性

初期投入可控，首年即可实现正向现金流；

模块化设计支持按需采购，降低客户决策门槛。

社会与政策可行性

符合国家“教育数字化战略”与“AI 向善”导向；

公益属性契合蚂蚁“AI 豆计划”与数字马力普惠教育使命。

生态协同可行性

全面融入蚂蚁技术栈（Ant Design、OpenSumi、AgentUniverse、OAuth2）；

可无缝接入数字马力“助教-助学-助就业”体系，形成商业闭环。