Computer Science and Application 计算机科学与应用, 2025, 15(11), 269-281
Published Online November 2025 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2025.1511303

Hans汉斯

智能时尚穿搭系统的设计与实现

张师瑜，王辉扬，王若愉*

仲恺农业工程学院人工智能学院，广东 广州

收稿日期：2025年10月17日；录用日期：2025年11月17日；发布日期：2025年11月27日

摘　要

在当前时尚穿搭社区智能化功能不足、用户个性化需求难以满足的背景下，本文设计并实现了一款基于人工智能的时尚穿搭社区系统。系统采用Vue前端框架与Flask后端架构，结合SQLAlchemy ORM与RESTful API设计，并集成FLUX.1模型以实现智能穿搭生成。该系统支持用户进行AI穿搭生成、图文分享与浏览，同时提供管理员后台管理功能。论文从系统设计、关键技术实现与功能验证等方面展开论述，设计合理的测试方案，性能压力测试表明，在100并发用户持续访问下，系统平均响应时间为285 ms，吞吐量稳定在125 req/s，且无服务宕机，证实了其良好的稳定性，图像生成效率良好，为智能化穿搭社区的构建提供了有效参考，具备较高的应用价值。

关键词

智能穿搭生成，AI生成，FLUX，Vue，Flask

Design and Implementation of Intelligent Fashion Outfit System

Shiyu Zhang, Huiyang Wang, Ruoyu Wang*

College of Artificial Intelligence, Zhongkai University of Agriculture and Engineering, Guangzhou Guangdong

Received: October 17, 2025; accepted: November 17, 2025; published: November 27, 2025

Abstract

Against the backdrop of insufficient intelligent features in current fashion styling communities and the difficulty in meeting users' personalized needs, this paper designs and implements an AI-powered fashion styling community system. The system adopts a Vue.js frontend framework and a Flask backend architecture, utilizes SQLAlchemy ORM and RESTful API design, and integrates the FLUX.1 model for intelligent outfit generation. It supports users in AI-generated outfit creation, image-text sharing, and

*通讯作者。

文章引用: 张师瑜, 王辉扬, 王若愉. 智能时尚穿搭系统的设计与实现[J]. 计算机科学与应用, 2025, 15(11): 269-281.
DOI: 10.12677/csa.2025.1511303

browsing, while also providing administrator backend management functionalities. The paper elaborates on the system design, key technology implementation, and functional verification. A well-designed testing scheme was implemented, with performance stress tests indicating that under a sustained load of 100 concurrent users, the system maintains an average response time of 285 ms and a stable throughput of 125 req/s without service downtime, confirming its robust stability. The image generation efficiency is demonstrated to be effective. This work provides a valuable reference for constructing intelligent fashion communities and possesses significant practical application value.

Keywords

Intelligent Outfit Generation, AI Generation, FLUX, Vue, Flask

1. 引言

经济发展与时尚消费审美的关系是一个动态演变的进程[1]。目前，人类的消费审美化进程越来越快，范围越来越广，审美文化产业的崛起表明了消费审美化所达到的新阶段，以及审美经济在现代的重要地位[1]。随着审美经济的崛起，时尚消费产品日益多样化，人们在日常生活中的穿衣搭配需求也愈来愈加复杂多样化，与此同时，面对琳琅满目的时尚单品，人们常常面临搭配方面的困扰。在面对众多服装单品时，人们常因缺乏搭配知识和灵感，花大量时间仍难以选出满意组合；购物时，面对众多款式，难以判断是否适合自己，且线上购物无法直观看到上身效果，容易买回不合适服装，造成浪费。

2. 研究背景

而过去五年间，穿搭类平台虽快速发展，但仍存在明显局限。尽管综合电商与直播电商渠道保持稳健增长，但传统穿搭推荐方式存在两大缺陷：一是用户生成内容质量参差不齐；二是平台推荐算法缺乏个性化深度[2]。这导致用户在获取穿搭建议时面临信息过载和匹配度低的双重困境。

而今随着大数据和人工智能的快速发展，时尚产业正迎来一场数字变革。从设计服饰、个性化推荐到虚拟试衣间，AI 正逐步改变人们选择服装搭配的方式。因此，未来的线上穿搭分享平台的趋势必定是集成 AI，通过用户与 AI 的交互实现对用户的穿衣搭配进行个性化推荐，满足用户日常生活中复杂的个性化穿搭需求[3]。本文通过设计与实现基于 Flux.1 模型的时尚穿搭社区系统，解决上述传统穿搭分享平台存在的滞后性、个性化推荐不足及用户决策效率低下等问题。通过技术整合与功能创新，本次研究与系统开发构建了一个整合 AIGC 技术具有个性化推荐、社交熟悉的智能穿搭平台，为用户提供“灵感获取－穿搭生成－社区分享”的一站式服务。

3. 需求分析

需求分析阶段，需统筹兼顾用户端与管理端的功能需求，以构建一个高效智能、交互流畅的时尚穿搭社区平台。

面向用户端，系统应重点支持以下功能：提供图文穿搭内容的浏览、搜索、筛选、收藏与评论功能，满足用户的内容消费与社区互动需求；构建基于兴趣标签的个性化推荐机制，提升内容分发的精准性与用户粘性；持用户访问他人主页、关注感兴趣用户并查看其动态，增强社区社交属性和活跃度；实现智

能穿搭生成功能，支持用户通过输入基础信息、服装描述与搭配偏好，生成个性化穿搭效果图像；设立个人中心模块，便于用户查看和管理个人数据。

面向管理端，系统需提供完善的后台管理支持：用户管理模块应支持查询、修改与删除等操作，保障社区成员管理的规范性与灵活性；社区内容管理模块需实现对帖子内容与风格标签的增删改查，并提供数据分析功能，辅助运营决策；提供用户生成内容审查功能，支持管理员查看穿搭生成记录、实施内容监管，确保平台内容质量与合规性。

系统功能需全面覆盖内容服务、智能生成、个性化管理与平台监管等多维度需求，为打造一个融合分享、互动与智能服务的时尚穿搭社区提供坚实支撑。综合上述需求，系统功能结构图如图 1 所示。

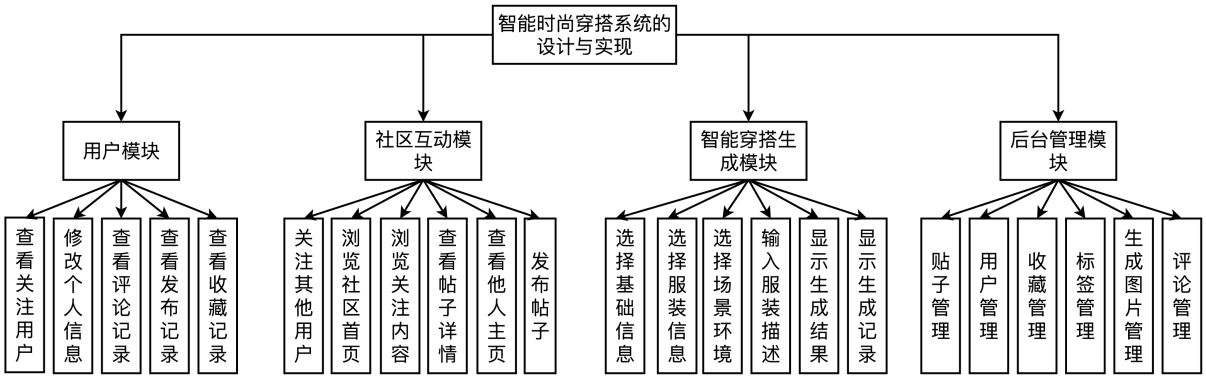

Figure 1. System functional structure diagram
图 1. 系统功能结构图

4. 系统设计

4.1. 系统架构设计

本系统采用了前后端分离的技术架构，其中前端基于 Vue2 框架结合 ElementUI 组件库进行开发，用以构建清晰美观的用户交互界面。而用户在页面的操作，通过 JavaScript 编写的业务逻辑以及 Axios 向后端暴露的 RESTful 接口发送请求，实现如用户注册登录、帖子操作、评论管理、生成穿搭图像等功能的数据交互。

后端基于 Flask 框架开发，负责处理来自前端的各类请求，实现系统业务逻辑的处理，并结合 SQLAlchemy 实现对数据库中各类结构化信息的高效操作，提升数据管理的灵活性与效率[4]。

在数据存储方面，系统选用 MySQL 作为关系型数据库，用于存储用户信息、帖子内容、评论记录等结构化数据；同时，通过 Flask 的 request 全局对象中的 g 变量实现用户 token 的临时存储与状态管理。

在智能穿搭生成模块中，系统将 Flux.1 模型部署于 Featurize 云端平台，并通过 Forge 开源项目对模型进行运行管理。在云端服务器将 Forge 的推理端口映射至公网，调用其开放的 API 实现文本生成穿搭图像的功能。Flask 后端服务充当中间层，接收前端提交的图像生成请求，转发至云端运行的模型 API，模型在云端服务器上完成推理过程，返回结果至 Flask 后端，最后再将生成图像反馈给前端展示模块，从而实现穿搭效果图的自动化生成。本系统架构如图 2 所示。

4.2. 数据库设计

为了更直观地呈现系统数据模型的结构，本文使用实体－关系图(E-R 图)来展示各数据表之间的关联关系及其关键字段。如图 3 所示，该 E-R 图详细描绘了系统中各实体的属性定义以及它们之间的逻辑关联，为后续数据库设计提供了清晰的参考依据。

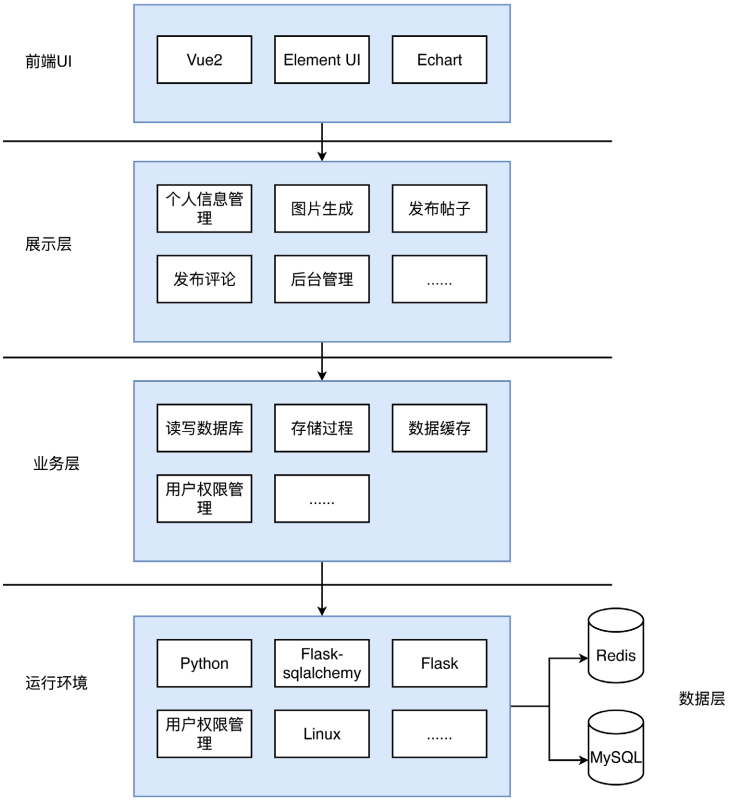

Figure 2. System architecture diagram
图 2. 系系统架构图

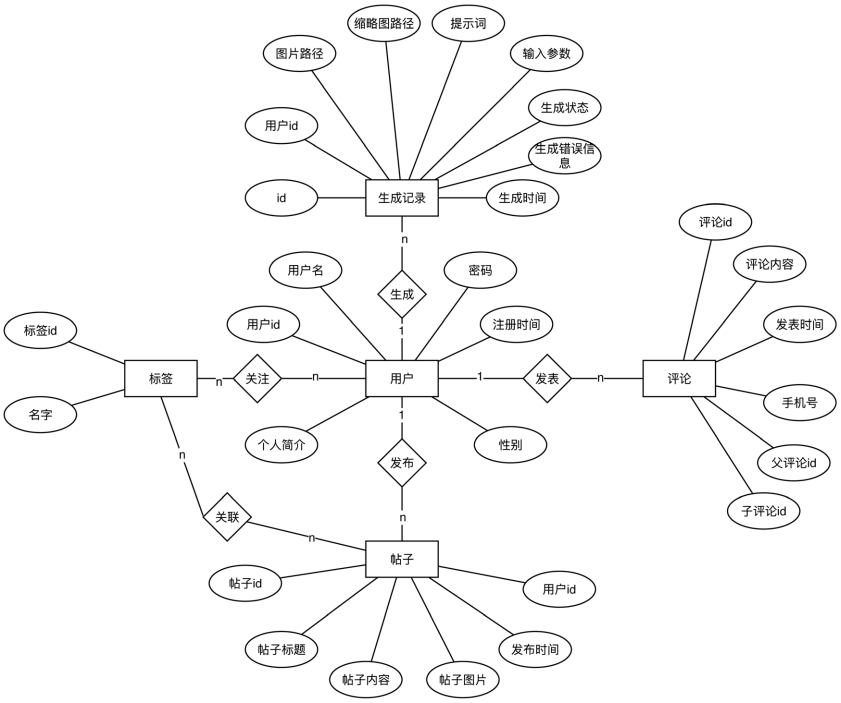

Figure 3. Entity-relationship diagram
图 3. 数据库 ER 图

4.3. 智能穿搭生成模块设计

实现高质量的智能穿搭生成面临三大挑战：1) 语义鸿沟：如何将用户抽象的时尚偏好(如“复古风”)转化为具体的视觉特征；2) 搭配兼容性：如何建模不同服饰单品(如上衣、裤子)之间复杂的风格、颜色、材质兼容关系；3) 个性化多样性：如何在保证符合用户口味的同时，避免推荐结果过于单一。

为应对上述挑战，我们基于一个 majicFlus_v1 (Flux.1 DEV 经过微调后专注于高质量人像生成的模型)模型，使用的公开时尚数据集如 DeepFashion 进行微调。通过用户优化用户使用的 prompt，进行个性化输出。后续将开发用户使用自定义数据库，进行个性化数据微调，达到进一步的个性化图片生成的目的。

4.3.1. 模块架构概述

模型推理服务作为智能穿搭生成模块的核心服务组件，采用分层架构设计，实现 Flux.1 模型在 Featurize 云平台的稳定部署和高效推理服务。系统将 Flux.1 模型集成至 Stable Diffusion WebUI Forge 构建的推理服务中，部署于 Featurize 云平台。本模块通过接收 Flask 中间层的请求转发参数，执行 Flux.1 模型的推理任务，负责根据用户提供的个性化穿搭参数生成符合场景与风格要求的穿搭效果图，构成系统 AI 能力的核心支撑部分。模块架构如图 4 所示。

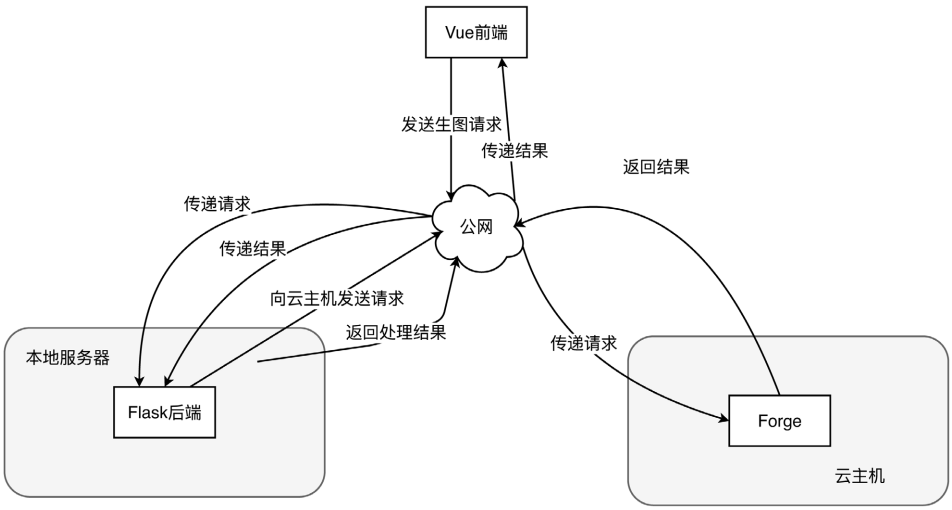

Figure 4. Inference service architecture diagram
图 4. 模型推理服务架构图

4.3.2. Stable Diffusion WebUI Forge 模型加载技术概述

Stable Diffusion 是当前主流的扩散式生成模型之一，具有高质量图像生成能力与结构可控性。为提高模型的部署效率与使用灵活性，系统采用其高性能衍生平台——Stable Diffusion WebUI Forge，支持通过 RESTful 接口远程推理服务调用，并具备动态切换模型、插件集成等扩展特性[5]。

Stable Diffusion WebUI Forge 在继承 Automatic1111 的 WebUI 架构的基础上进行了性能优化与插件增强，尤其在模型加载和推理控制方面构建了模块化、可热切换的机制。其模型加载过程主要包含两个核心部分，分别为主模型的加载与辅助模块的注入。

主模型的加载过程在 Forge 的模块 sd_models.py 中实现，系统通过配置模型名称字段，由 load_model_from_config 方法读取模型文件并完成初始化。开发者可在 Forge 启动时通过配置文件预设模型路径，也可通过 Web 界面或 RESTful API 实现模型热切换，从而支持动态的模型管理和扩展。

4.3.3. 模型加载与推理流程

本系统加载的 Flux.1 模型为 majicFlus_v1，是 LiblibAI 社区的作者麦橘 MERJIC 基于 Flux.1 DEV 经过微调后专注于高质量人像生成的模型，结合了深度语义嵌入与人像映射能力，为穿搭效果图生成提供内容与结构基础。

系统通过 Flask 后端转发前端请求至 Forge 文生图服务接口，Forge 则从请求中获取提示词与参数设置，将提示词传入 CLIP 模块，将提示词编码为语义向量，并与图像潜空间进行对齐[6]，接着根据参数设置加载 Flux.1 模型及相应的 VAE 模块，完成提示词编码以及主模型加载。

在主模型加载完成后，Forge 根据推理请求中的参数设置获取加载相应的 LoRA 模型。LoRA 是一种轻量化微调方法，通过注入额外的低秩权重矩阵，有效增强模型对特定风格的响应能力[7]。Forge 对 LoRA 的支持由 lora.py 模块提供，在模型推理前通过调用 load_loras 方法加载 LoRA 文件，并在推理过程中将其融合到 UNet 的正向传播结构中。LoRA 模型一般存放于 Forge 项目指定目录下，可以通过推理请求中的 alwayson_scripts 参数动态注入。例如，用户可指定[“fashion-lora”, 0.75]表示注入名称为 fashion-lora 的模型，注入强度为 0.75。该过程不改变主模型结构，并可随任务切换灵活加载与卸载。

模型加载推理流程如图 5 所示。

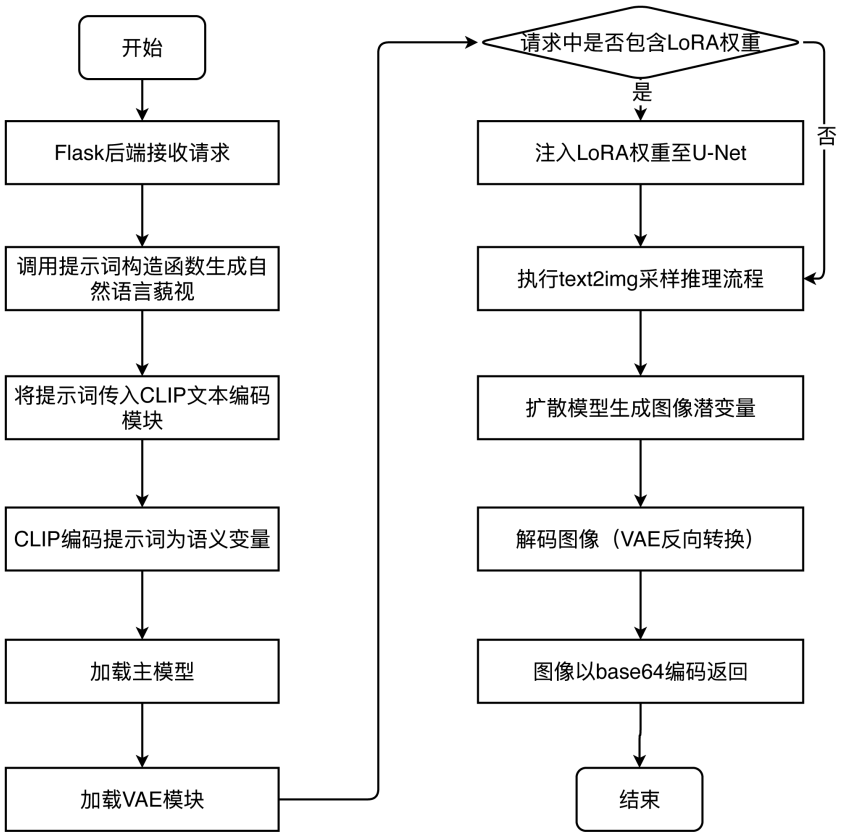

Figure 5. Model loading and inference flowchart
图 5. 模型加载推理流程图

4.3.4. 提示词构造函数设计

构造函数的核心功能是将用户在前端页面中填写的穿搭偏好信息，转化为符合 Flux.1 模型生成逻辑的自然语言提示词。并以模块化方式构建实现路径。构造函数流程如图 6 所示。

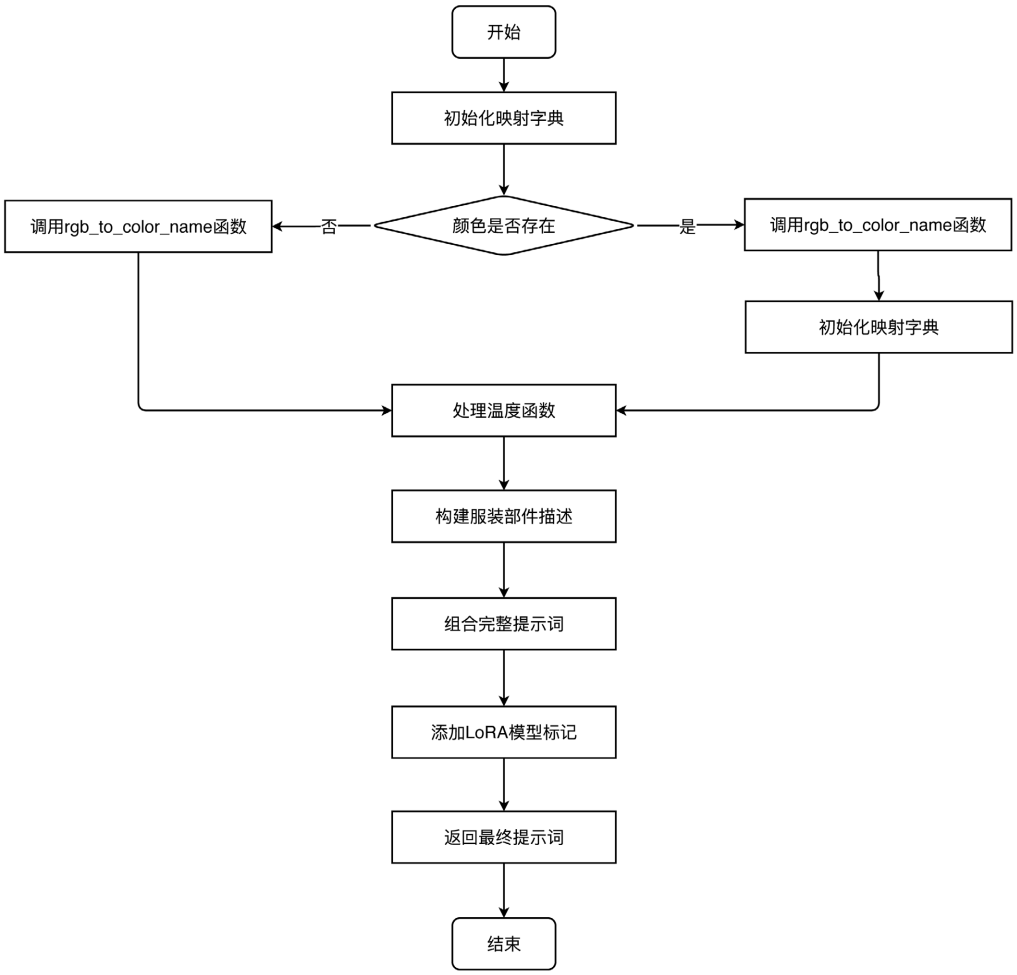

Figure 6. Prompt design flowchart
图 6. 提示词设计流程图

在处理用户输入的初始阶段，系统会调用一系列映射字典，将编码形式的属性值转化为语义化的英文表达。例如，用户选择的性别“男”被转换为“male”，季节“summer”则保持原样，但在组合中将与其他描述项一并构成完整句式。该转换逻辑覆盖性别、年龄区间、季节、穿衣风格及温度等多个维度，为构造高质量的语言提示提供了语义支撑。

针对颜色信息的处理，函数引入了 rgb_to_color_name 辅助模块，专门用于将十六进制颜色值映射为自然语言中的颜色名称。该模块通过内部维护的颜色区间表对输入颜色进行精确识别和命名，从而生成如“featuring a main color of red and a secondary color of blue”的描述性短语，使生成图像更具配色指导性和现实感。

在生成穿搭部件的文本内容方面，函数会将用户勾选的服装选项(如上装、下装、鞋履等)由数组形式拼接为语义连贯的短语，例如[“short sleeves”, “hoodie”]被格式化为“short sleeves, hoodie”；而对于未选择的项目，则统一生成如“no hat”的补充描述以确保结构完整。温度数值也会结合预设区间转换为相应的气候词汇，如“warm”或“cool”，从而增强提示词对场景的适应性。

在组合整体提示词时，函数采用双重语义结构进行组织：一部分描述人物穿搭效果，另一部分则以平铺视角呈现相同的服饰单品，形成展示性与识别性兼具的图像生成引导语。与此同时，提示词中还嵌

入 LoRA 模型权重控制字符，以细化生成输出的图像风格与细节控制。

4.3.5. 部署与应用

为了提升部署效率与可复用性，本模块通过编写 Shell 脚本在 Featurize 云主机上自动部署 Stable Diffusion WebUI Forge，并完成必要依赖的安装、模型文件加载及端口暴露设置。该方式降低了人工配置成本，提升了部署的可维护性与一致性。部署脚本的主要功能包括：创建 Python 虚拟环境、安装依赖项、克隆 Forge 项目、放置主模型与 LoRA 模型、启动服务并监听公网端口。

执行脚本后，Forge 会在 Featurize 云主机上以 Web 服务形式运行，并监听 7860 端口。通过 Featurize 提供的端口映射管理功能，可将该端口暴露至公网，从而使系统本地后端服务可以通过该地址访问远程模型推理接口。后端 Flask 服务通过 POST 请求将用户的穿搭描述及控制参数传送至 Forge 的接口，完成图像的生成与返回。

4.4. 系统核心功能时序图设计

用户生成穿搭效果图时序图

用户在生成穿搭效果图时，在前端页面输入穿搭需求信息，前端调用 translatePrompt 方法对用户提示词进行翻译优化，随后，调用 generateOutfit 方法向 txt2img_bp 发送生成请求，后者通过 handle_generation 方法调用 Forge 的接口进行图像生成，并将结果返回。生成完成后，系统调用 add_generation 方法使用 SQLAlchemy 将生成记录保存至数据库。最终，前端接收生成图像并展示给用户。该时序图如图 7 所示。

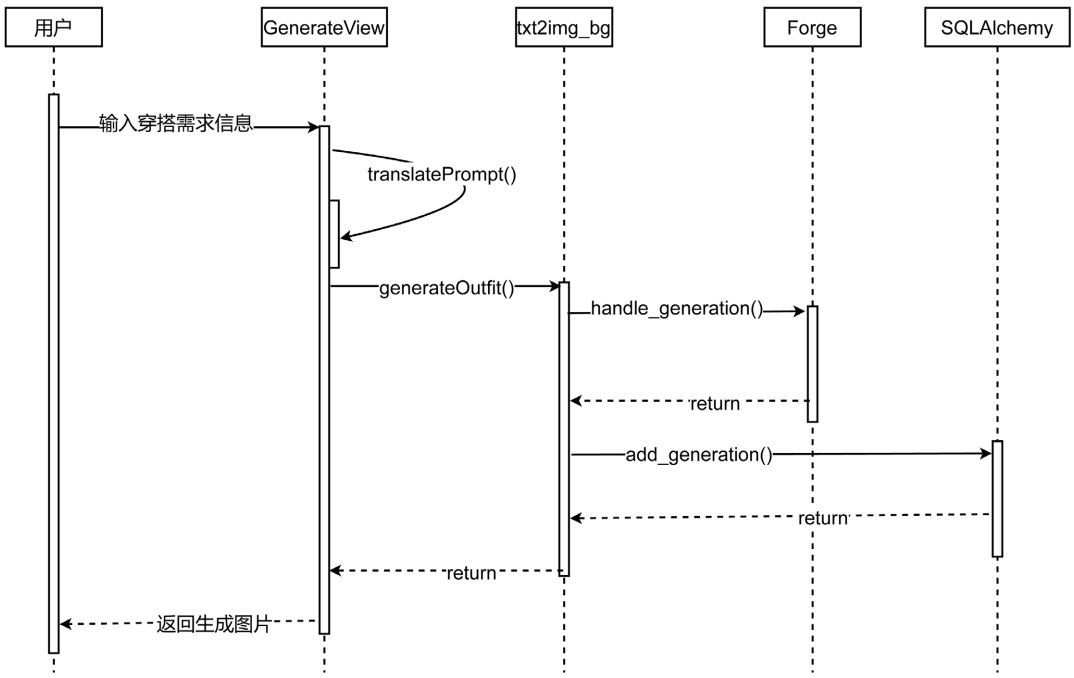

Figure 7. The time series diagram of outfit generation
图 7. 穿搭生成时序图

5. 系统实现

5.1. 系统核心开发技术

Flux.1 dev 是基于 Stable Diffusion 扩展优化的图像生成模型，属于典型的扩散式生成架构(Diffusion

Model)。该模型在继承原始 Stable Diffusion 强大图像生成能力的基础上，进一步强化了风格控制与提示词响应能力，特别适用于服饰、人物等细粒度领域的可控图像合成任务。Flux.1 dev 通常以.safetensors 格式发布，具备较高的模型压缩效率与运行稳定性，适合在 Forge 等轻量化服务平台上部署并远程调用。

其核心技术来源于 Latent Diffusion Model (LDM)，该方法通过将图像压缩至潜在空间再进行扩散采样，有效降低了高分辨率图像生成的资源消耗与时间复杂度[7]。

在本系统中，Flux.1 DEV 模型被加载至 Featurize 云端平台中的 Stable Diffusion WebUI Forge 环境中，并结合服装类 LoRA 微调模型进行图像推理任务。通过精心设计的提示词与参数配置，模型能够生成符合用户输入条件(如性别、季节、场景、服装颜色等)的高质量穿搭效果图，为用户提供高效直观的时尚参考与个性搭配方案。本系统前端采用 Vue.js 框架进行开发，并集成了 Element UI 组件库。本系统后端便采用 Flask 框架，实现轻服务轻应用开发。在本系统中，百度翻译 API 主要用于将用户输入的中文穿搭提示信息实时翻译为英文提示词，供后端构造英文自然语言 Prompt 以调用 Flux.1 模型进行图像生成。

5.2. 系统实现细节与展示

5.2.1. 用户登录

用户输入账号密码后，系统会跳转社区首页，进入社区互动模块页面，如图 8 所示。

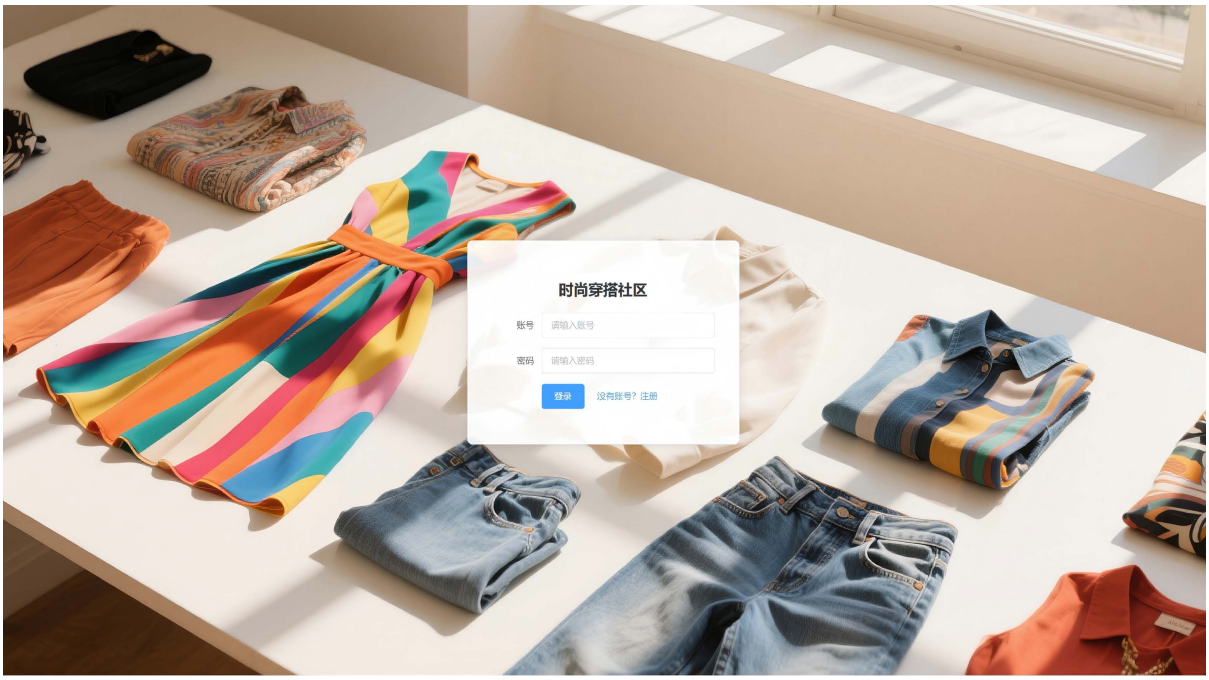

Figure 8. User login page diagram
图 8. 用户登录页面图

5.2.2. 社区首页

用户登录成功后进入社区首页，在首页可浏览社区的帖子和基于用户感兴趣标签推荐的帖子，支持搜索和风格标签筛选帖子，如图 9 所示。

5.2.3. 发布帖子页面

用户点击社区首页的“发布”按钮，系统可跳转帖子发布页面，用户可以输入标题、上传图片、输入穿搭描述、选择相关标签，然后发布帖子。页面如图 10 所示。

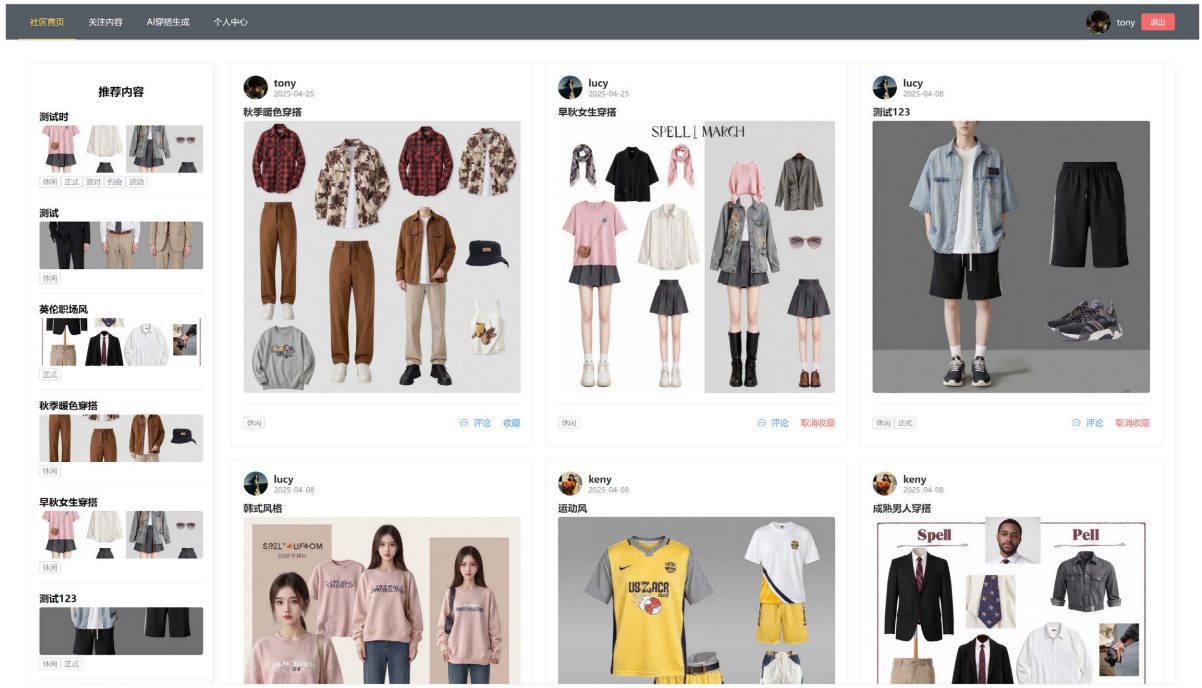

Figure 9. Community homepage diagram
图 9. 社区首页页面图

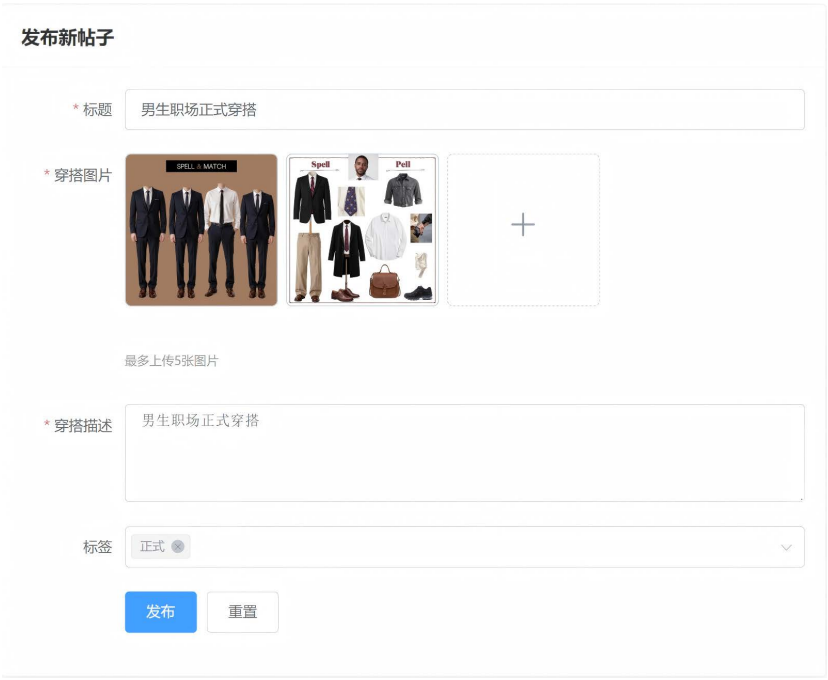

Figure 10. Post creation page diagram
图 10. 发布帖子页面图

5.2.4. 帖子详情页

用户点击某一帖子可以打开帖子详情页，查看穿搭帖子详细内容，并且可以在该帖子下发布评论，操作页面如图 11 所示。

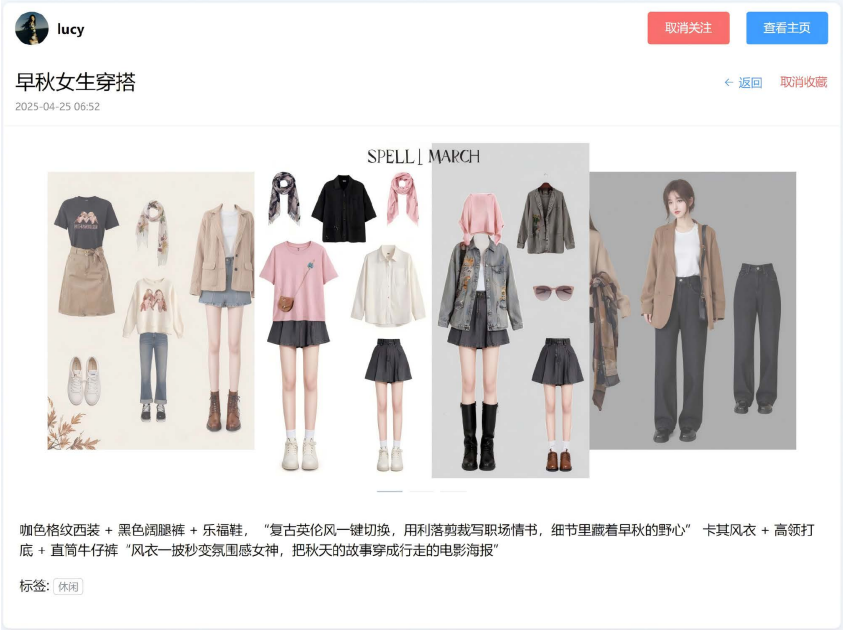

Figure 11. Post details page diagram
图 11. 帖子详情页面图

5.2.5. 智能穿搭生成页面

用户点击头部导航栏“AI 穿搭生成”选项，系统跳转 AI 穿搭生成页面，用户可以通过选择基础信息、服装细节、场景环境和输入，如图 12 所示。

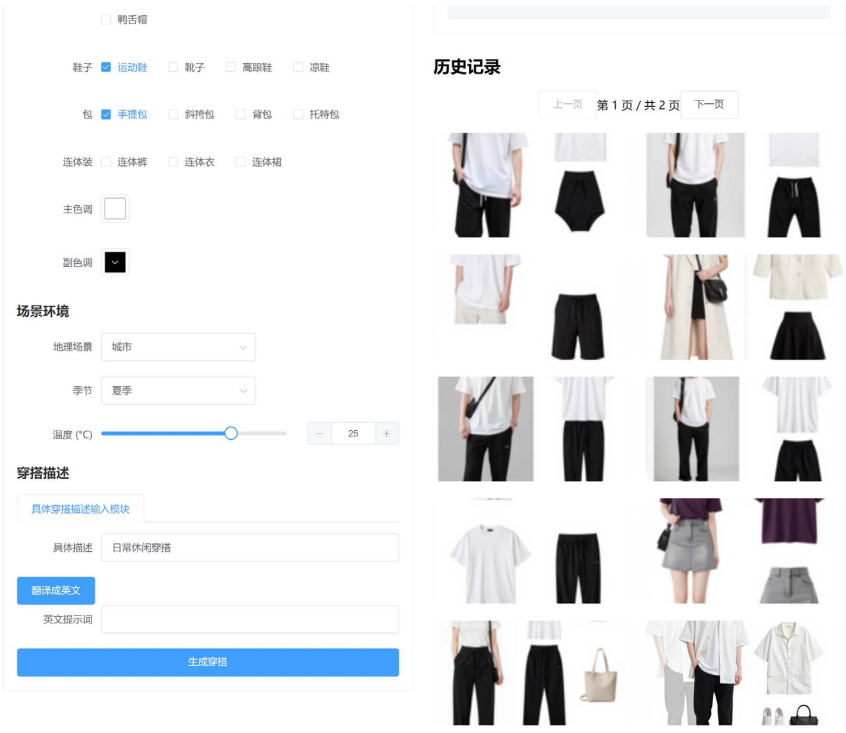

Figure 12. Intelligent outfit generation page diagram
图 12. 智能穿搭生成页面图

6. 系统测试与性能评估

测试不仅是保障系统质量的关键手段，更是提升用户体验、降低维护成本和确保系统稳定运行的基础。

系统测试目标与实验设计

为确保本系统在实际运行环境中的稳定与高效，本章节旨在通过设计严谨的功能性测试和性能测试，对系统的稳健性与服务承载能力进行客观评估。本次评估将聚焦于回答以下核心问题：系统功能是否正确？系统在模拟高并发访问的压力下，其关键性能指标是否仍能保持在可接受的范围内？

具体而言，本次性能测试将重点关注：请求响应时间：衡量系统处理单个请求所需的平均时间及 P95 分位值，直接关乎用户体验的流畅度。系统吞吐量：评估系统在单位时间内成功处理请求的能力，反映其整体处理效率。

该测试目标旨在验证系统架构与资源调配的有效性，确保其在上线后能够应对真实的用户访问压力，避免因技术瓶颈影响核心功能的实现。通过本次系统性测试，我们将为系统的技术可靠性与服务连续性提供关键的数据支撑。关于穿搭方案的审美评价与用户满意度，将作为系统上线后的长期优化目标，通过收集用户反馈图片与满意度调查数据进行；未来，还可进一步利用用户自建数据集进行模型微调，以实现真正的个性化体验。关键功能的测试用例表如表 1 所示：

Table 1. Test cases for user-generated outfit images
表 1. 用户生成穿搭图测试用例

测试编号	0001
测试项目	用户生成穿搭图
测试目的	测试用户输入信息后系统是否能准确生成穿搭效果图并展示
测试依据	用户生成穿搭图用例规约
预置条件	1) 用户已处于登录状态， 2) 用户位于智能穿搭生成页面
测试步骤	1)选择基础信息 2) 选择服装细节 3) 选择场景环境 4) 输入穿搭具体描述 5) 点击翻译按钮翻译为英文 6) 点击“生成穿搭”按钮 7) 系统返回并展示生成结果
预期结果	正常：穿搭图生成成功，页面展示在穿搭效果预览面板中 异常：生成超时，显示“请稍后重试”提示
测试结果	与预期结果相符

系统性能与稳定性测试

为评估系统在负载下的行为，我们使用性能测试工具 JMeter 对系统进行了压力测试。测试重点关注图片生成等包含 AI 推理的核心业务流程。测试在一台搭载 Nvidia GeForce RTX 4090 GPU 及 4 核 CPU、8 GB 内存的服务器上进行。在模拟 100 个并发用户持续访问 10 分钟的场景下，系统关键性能指标如下：平均响应时间：285 毫秒、P95 响应时间：420 毫秒、系统吞吐量：125 请求/秒。

在整个测试过程中，系统服务保持稳定，未出现宕机或严重错误。同时，服务器侧的 CPU 与内存使

用率始终维持在 80%以下的稳定水平。

结果分析：上述数据表明，即使在百级用户并发访问、AI 推理服务持续高负载运行的压力场景下，系统依然能够提供低延迟、高吞吐的服务能力。这充分证明了当前系统架构，特别是基于 Nvidia 4090 的推理后端，具有良好的稳定性和可扩展性，能够满足实际生产环境的要求。

7. 总结

本研究围绕基于 Flux.1 模型的智能时尚穿搭社区系统展开，通过 Vue2 + Element UI 前端框架与 Flask 后端框架的协同开发，完成了包括用户交互界面、业务逻辑接口、云端模型推理服务及 MySQL 数据存储在内的完整系统构建。在 Featurize 平台上部署的 Stable Diffusion WebUI Forge 有效支撑了穿搭效果图的生成功能，并通过模块化架构设计保障了系统的可维护性与扩展性。

为验证系统效能，本研究设计了多维度测试方案。性能压力测试结果显示，在模拟 100 并发用户持续访问 10 分钟的场景下，系统平均响应时间为 285 毫秒，P95 响应时间为 520 毫秒，吞吐量稳定维持在 125 请求/秒，且服务全程无宕机，关键资源指标保持正常水平，证实了系统架构的稳健性与服务承载能力。

通过本次系统实现与验证过程，我们不仅构建了一个功能完整的时尚社区原型，更通过量化数据证明了 AIGC 技术在时尚领域应用的技术可行性。系统在响应性能、服务稳定性与内容生成效率方面均表现出良好水准，为同类系统的开发提供了可参考的架构方案与评估范式。未来的研究方向包括引入用户反馈机制实现生成模型的持续优化，以及探索多模态交互在个性化穿搭推荐中的深化应用。

参考文献

[1] 王旭晓. 现代消费审美化与美学在经济领域的作为[J]. 河北学刊, 2010, 30(3): 23-28.

[2] 广发证券. 纺织服饰行业 2024 年上半年电商渠道表现综述[R]. 北京: 广发证券, 2024.

[3] 苑永琴, 高璇, 韩宛辰. 基于大学生体验的虚拟试衣软件的应用现状研究[J]. 信息记录材料, 2024, 25(2): 60-62.

[4] 蔡钰汐. 企业档案管理信息系统的设计与实现[J]. 信息记录材料, 2023, 24(9): 87-89+92.

[5] Rombach, R., Blattmann, A., Lorenz, D., *et al*. (2022) High-Resolution Image Synthesis with Latent Diffusion Models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, 18-24 June 2022, 10684-10695.

[6] Patashnik, O., Wu, Z., Shechtman, E., *et al*. (2021) StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11-17 October 2021, 2085-2094.

[7] Hu, E.J., Shen, Y., Wallis, P., *et al*. (2021) LoRA: Low-Rank Adaptation of Large Language Models.