

PR #35713 完整报告

sgl-project/sglang

[diffusion] feat: support out-of-tree models and pipelines

合并时间: 2026-08-21 00:33

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/35713>

执行摘要

- 一句话: Diffusion 支持外部模型与 pipeline 插件注册
- 推荐动作: 值得精读, 特别是注册表惰性初始化与公共 API 的设计。该 PR 是 SGLang-Diffusion 插件化演进的关键一步, 理解它有助于后续接入第三方模型或开发类似插件机制。如果团队计划扩展模型生态, 建议关注 `register_pipeline` 的语义与 `overwrite` 冲突策略的文档化。

功能与动机

PR body 指出: “An external package can now register its native component models and composed pipeline from `init.py`, then run through the standard `sglang serve` or `sglang generate` CLI without changing SGLang-Diffusion source.” 过去第三方模型只能通过 fork 源码或提交内置支持才能接入, 维护成本高。本 PR 通过开放注册 API 降低接入门槛, 避免每次接入新模型都要改动核心代码。

实现拆解

1. 环境变量与加载器: 在 `python/sglang/multimodal_gen/envs.py` 新增 `SGLANG_EXTERNAL_MODEL_PACKAGE` 环境变量定义; 新增 `runtime/utils/external_model_package.py`, 实现 `load_external_model_package()`, 用 `functools.lru_cache` 确保进程内只导入一次外部包。
2. 注册表惰性初始化与公共 API: `multimodal_gen/registry.py` 引入 `_BUILTIN_PIPELINES_DISCOVERED` 标志替代原先“注册表非空即返回”的逻辑, 解决外部注册后内置发现可能被跳过的问题; 新增 `_ensure_registry_initialized()` 统一执行内置发现与外部包加载; 新增 `get_pipeline_class()`、`get_registered_pipeline_names()`、`register_pipeline()` 三个公共 API, 其中 `register_pipeline()` 以 keyword-only 参数接收 pipeline 类、采样 / 配置类、`hf_model_paths`、`model_detectors` 与 `overwrite`, 并复用 `register_configs()` 完成配置类注册。
3. 模型注册集成: `runtime/models/registry.py` 在 `get_supported_archs()`、`resolve_by_alias()`、`_normalize_archs()` 三处调用 `load_external_model_package()`, 使模型解析前先加载外部包; `_LazyRegisteredModel` 移除冗余的 `component_name` 字段, 改为从 `<module>:<class>` 字符串直接解析模块与类名。
4. 流水线构建接入: `runtime/pipelines_core/__init__.py` 的 `build_pipeline` 中, 显式 `--pipeline` 分支从直接访问私有 `_PIPELINE_REGISTRY` 改为调用

`get_registered_pipeline_names()` 与 `get_pipeline_class()`，保证显式指定的 pipeline 也能命中外部注册。

5. 测试与文档配套：新增 `test/unit/test_external_model_package.py`，在临时目录构造一个外部插件包，通过子进程设置环境变量 `SGLANG_EXTERNAL_MODEL_PACKAGE` 和 `PYTHONPATH`，断言模型解析、`get_model_info`、`get_pipeline_config_classes` 均能命中外部类；文档 `support_new_models.mdx` 与 `environment_variables.mdx` 补充 out-of-tree 示例和环境变量说明。

关键文件：

- `python/sglang/multimodal_gen/registry.py`（模块 注册中心；类别 source；类型 dependency-wiring；符号 `_ensure_registry_initialized`, `get_pipeline_class`, `get_registered_pipeline_names`, `register_pipeline`）：核心变更：引入惰性初始化与公共注册 API，是 out-of-tree 支持的主入口。
- `python/sglang/multimodal_gen/test/unit/test_external_model_package.py`（模块 测试覆盖；类别 test；类型 test-coverage；符号 `test_external_diffusion_package_registration`, `ExternalTransformer`, `ExternalPipelineConfig`, `ExternalSamplingParams`）：集成测试覆盖外部包注册全链路，验证模型解析、pipeline 与配置类均能命中外部注册。
- `python/sglang/multimodal_gen/runtime/utils/external_model_package.py`（模块 插件加载；类别 source；类型 data-contract；符号 `load_external_model_package`）：新增懒加载函数，是外部包导入的唯一入口，进程内仅执行一次。
- `python/sglang/multimodal_gen/runtime/models/registry.py`（模块 模型注册；类别 source；类型 data-contract；符号 `get_supported_archs`, `resolve_by_alias`, `_normalize_archs`, `register_model`）：模型注册表在多个查询入口调用外部包加载，并对模型类字符串解析做了精简。
- `python/sglang/multimodal_gen/runtime/pipelines_core/__init__.py`（模块 流水线构建；类别 source；类型 core-logic；符号 `build_pipeline`）：`build_pipeline` 改用公共访问器，使显式 `--pipeline` 支持外部注册。
- `python/sglang/multimodal_gen/envs.py`（模块 环境配置；类别 source；类型 core-logic；符号 `SGLANG_EXTERNAL_MODEL_PACKAGE`）：新增环境变量定义，连接 SRT 与 Diffusion 的插件机制。
- `docs/docs/sglang-diffusion/support_new_models.mdx`（模块 文档；类别 other；类型 data-contract）：新增 out-of-tree 支持教程，说明如何编写插件包。
- `docs/docs/sglang-diffusion/environment_variables.mdx`（模块 文档；类别 other；类型 core-logic）：记录新环境变量。

关键符号：`_ensure_registry_initialized`, `get_pipeline_class`, `get_registered_pipeline_names`, `register_pipeline`, `load_external_model_package`, `build_pipeline`

关键源码片段

`python/sglang/multimodal_gen/registry.py`

核心变更：引入惰性初始化与公共注册 API，是 out-of-tree 支持的主入口。

```

# python/sglang/multimodal_gen/registry.py
# 惰性初始化标志：确保内置 pipeline 发现只执行一次
_BUILTIN_PIPELINES_DISCOVERED = False

def _ensure_registry_initialized() -> None:
    # 统一入口：先发现内置 pipeline，再导入外部模型包
    _discover_and_register_pipelines()
    load_external_model_package()

def get_pipeline_class(pipeline_class_name: str) -> Type[ComposedPipelineBase] | None:
    # 按名称获取 pipeline 类，查找时自动完成注册初始化
    _ensure_registry_initialized()
    return _PIPELINE_REGISTRY.get(pipeline_class_name)

def get_registered_pipeline_names() -> List[str]:
    # 返回全部已注册 pipeline 名称，含外部包注册项
    _ensure_registry_initialized()
    return list(_PIPELINE_REGISTRY)

```

python/sglang/multimodal_gen/runtime/utils/external_model_package.py

新增懒加载函数，是外部包导入的唯一入口，进程内仅执行一次。

```

# python/sglang/multimodal_gen/runtime/utils/external_model_package.py
import importlib
from functools import lru_cache

from sglang.multimodal_gen import envs
from sglang.multimodal_gen.runtime.utils.logging_utils import init_logger

logger = init_logger(__name__)

@lru_cache(maxsize=1)
def load_external_model_package() -> None:
    # 按配置导入 out-of-tree 模型包，进程内只执行一次
    package_name = envs.SGLANG_EXTERNAL_MODEL_PACKAGE
    if not package_name:
        return

    logger.info("Loading external model package: %s", package_name)
    importlib.import_module(package_name)

```

评论区精华

本 PR 没有收到 review 评论；唯一 issue 评论是作者触发的 [/tag-and-rerun-ci](#) CI 重跑。作者在 PR body 中自述了验证结果：在 RTX 5090 devbox 上以独立外部包注册

ExternalMiniMaxH3Pipeline 并成功跑通 4 秒 T2VA 请求。设计层面的取舍（如懒加载时机、`overwrite` 冲突处理）未经过额外公开讨论，由作者直接决定。

- 暂无高价值评论线程

风险与影响

- 风险：

1. 外部包导入失败即启动失败：`load_external_model_package()` 直接调用 `importlib.import_module()` 且未捕获异常，若外部包存在依赖缺失或语法错误，服务进程会在注册初始化时直接退出。
2. 同名 pipeline 注册冲突：`register_pipeline()` 提供 `overwrite` 参数，但默认值 `False` 下若与内置 pipeline 同名，具体处理是抛错还是忽略取决于实现，存在行为模糊性，且外部包与内置包的加载顺序会影响最终结果。
3. 懒加载入口覆盖不全：`_ensure_registry_initialized()` 是统一入口，但若未来有代码绕过该入口直接访问 `_PIPELINE_REGISTRY`（例如旧代码或第三方模块），外部注册将不可见。
4. 子进程环境依赖：模型架构检查通过 `_run_in_subprocess` 在子进程中执行，依赖环境变量传递；若外部包除环境变量外还依赖配置文件或工作目录，子进程可能无法正确加载模型。
 - 影响：对第三方模型开发者是重大利好，接入成本从“改源码”降为“写一个 Python 插件包并设置环境变量”。对现有用户无行为变化（未设置 `SGLANG_EXTERNAL_MODEL_PACKAGE` 时路径与原先一致）。对系统而言，服务启动时会额外导入外部包，增加启动时间和潜在依赖冲突风险。对团队而言，新增了公共 API 和文档的长期维护负担，且 `SGLANG_EXTERNAL_MODEL_PACKAGE` 与 SRT 模型插件机制共享环境变量名，需注意命名空间的相互影响。改动集中在 `multimodal_gen` 模块，不影响 SRT 核心路径。
 - 风险标记：外部包导入失败即启动失败，同名 pipeline 注册冲突，懒加载入口覆盖不全，子进程环境依赖敏感

关联脉络

- PR #35668 [diffusion] feat: add weight source reader: 重构 checkpoint 加载逻辑，引入 weight reader 协议，为外部权重源接入铺路，与本 PR 的 out-of-tree 支持互补。
- PR #35688 [diffusion] feat: let every layerwise component be configurable: 增强 diffusion 运行时组件的可配置性，体现生态扩展趋势，与本 PR 共同构成插件化能力。