

PR #35774 完整报告

sgl-project/sglang

[diffusion] Resolve LoRA weight sources deterministically

合并时间: 2026-08-21 21:05

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/35774>

执行摘要

- 一句话: 新增确定性 LoRA 权重源解析
- 推荐动作: 值得精读。该 PR 展示了一个小而美的抽象: 将权重源解析独立成模块, 并通过 fail-closed 策略控制行为确定性, 同时主动收窄 PR 范围 (删除早期尝试的 preflight CLI 等), 是控制变更风险的范例。

功能与动机

PR 目标是让 LoRA 权重来源解析具有确定性。原先 `--lora-path` 的解析依赖 `diffusers` 的 `_best_guess_weight_name` 猜测文件名, 在含多个权重文件的仓库中行为不确定; 同时远程下载未固定 revision, 内容可能漂移。PR body 明确 'pin remote LoRA downloads to the immutable Hub commit returned during file selection', 且设计上 'fails closed for non-safetensors adapters'。

实现拆解

1. 新增权重源解析模块: 新增 `python/sglang/multimodal_gen/runtime/weights/source.py`, 定义 `WeightSource`、`WeightInventory`、`ResolvedWeight` 三个冻结数据类, 以及 `parse_weight_source`、`resolve_weight_inventory`、`resolve_weight` 函数。
`parse_weight_source` 负责将输入字符串归一化为四种形态: 本地路径、owner/repo 仓库、带子目录的仓库、精确的 tree/blob/resolve URL; 并通过 `_merge_revision` 检查 URL 中 revision 与显式 revision 的冲突。`resolve_weight_inventory` 对本地路径列出目录文件, 对远程仓库调用 `HfApi.model_info` 获取不可变 sha 和文件清单, 并按子目录过滤。`resolve_weight` 在清单中选择唯一权重文件; 若存在多个独立权重文件且未指定 `weight_name` 或精确文件 URL, 则抛出歧义错误。
2. 改造 LoRA 下载入口: 修改 `python/sglang/multimodal_gen/runtime/utils/hf_diffusers_utils.py` 的 `maybe_download_lora`, 将 `ModelScope` 与 `Hugging Face` 路径分流。
`Hugging Face` 路径改为先调用 `resolve_weight` 得到 `selected_file`, 校验必须是 `safetensors`, 然后只以 `allow_patterns=["*.json", selected_file]` 和固定的 revision (即不可变 sha) 调用 `maybe_download_model`, 避免下载所有候选权重。本地路径同样经由 `resolve_weight` 归一化后拼接出最终文件路径。
3. 添加单元测试: 新增 `test/unit/test_weight_source.py`, 覆盖本地 / 远程清单解析、revision 冲突拒绝、歧义拒绝、子目录过滤等场景。在 `test/unit/test_lora_pipeline.py` 中新增 `test_lora_tree_url_selects_one_pinned_weight` 和

test_lora_exact_file_url_needs_no_weight_name, 验证 URL 驱动下载时 allow_patterns 和 revision 传递正确。

4. 更新文档: 在 docs/docs/sglang-diffusion/api/cli.mdx 中更新 --lora-path 的说明, 明确支持本地路径、HF 仓库 / 子目录和精确文件 URL。

关键文件:

- python/sglang/multimodal_gen/runtime/weights/source.py (模块 权重源; 类别 source; 类型 core-logic; 符号 WeightSource, WeightInventory, ResolvedWeight, _validate_relative_hub_path): 新增模块, 是 PR 核心: 定义权重源解析的数据结构与函数, 负责本地 / 远程清单、revision 固定与确定性选择。
- python/sglang/multimodal_gen/runtime/utils/hf_diffusers_utils.py (模块 LoRA 加载; 类别 source; 类型 dependency-wiring; 符号 maybe_download_lora): 修改 LoRA 下载入口, 接入 resolve_weight, 按解析结果只下载选中文件与 JSON, 并固定 revision。
- python/sglang/multimodal_gen/test/unit/test_weight_source.py (模块 单元测试; 类别 test; 类型 test-coverage; 符号 test_parse_weight_source_accepts_repo_subfolder_and_exact_url, test_parse_weight_source_rejects_conflicting_url_revision, test_resolve_local_inventory_lists_files_without_loading_tensors, test_resolve_remote_inventory_pins_revision_and_filters_subfolder): 新增单元测试, 覆盖解析、revision 冲突、歧义拒绝、子目录过滤等核心行为。
- python/sglang/multimodal_gen/test/unit/test_lora_pipeline.py (模块 单元测试; 类别 test; 类型 test-coverage; 符号 test_lora_tree_url_selects_one_pinned_weight, test_lora_exact_file_url_needs_no_weight_name): 扩展 LoRA 管线测试, 验证 tree/blob URL 驱动的下下载参数。
- python/sglang/multimodal_gen/runtime/weights/__init__.py (模块 模块导出; 类别 source; 类型 entrypoint): 标识权重源解析包, 便于模块导入。
- docs/docs/sglang-diffusion/api/cli.mdx (模块 CLI 文档; 类别 docs; 类型 documentation): 更新 --lora-path 说明, 反映支持本地路径、HF 仓库 / 子目录和精确 URL。

关键符号: parse_weight_source, resolve_weight_inventory, resolve_weight, maybe_download_lora, _merge_revision, _parse_huggingface_url, _validate_relative_hub_path

关键源码片段

python/sglang/multimodal_gen/runtime/weights/source.py

新增模块, 是 PR 核心: 定义权重源解析的数据结构与函数, 负责本地 / 远程清单、revision 固定与确定性选择。

```
# 权重源解析核心: 将用户输入归一化为 WeightSource。
# 支持本地路径、owner/repo、子目录、以及 tree/blob/resolve URL。
def parse_weight_source(source: str, *, revision: str | None = None) -> WeightSource:
    expanded = os.path.expanduser(source)
    parsed = urlparse(source)
```

```

# 带 http/https 前缀的直接走 Hugging Face URL 解析。
if parsed.scheme in ("http", "https"):
    return _parse_huggingface_url(source, revision)

# 判断是否为本地路径：存在、绝对路径或以 ./ ../ ~ 开头。
looks_local = (
    os.path.exists(expanded)
    or os.path.isabs(expanded)
    or source.startswith(("./", "../", "~"))
)
if looks_local:
    return WeightSource(
        original=source,
        kind="local",
        local_path=os.path.abspath(expanded),
    )

# 其余按 owner/repo 处理，前两段为仓库 ID。
parts = source.split("/")
if len(parts) < 2 or not all(parts[:2]):
    raise ValueError(
        f"Weight source {source!r} is neither a local path nor an "
        "owner/repo Hugging Face reference"
    )
repo_id = "/".join(parts[:2])
validate_repo_id(repo_id)
tail = "/".join(parts[2:]) or None

# 以权重后缀结尾的视为精确文件名，否则视为子目录。
# （该分支依据测试行为推断：owner/repo/adapters.safetensors -> filename,
# owner/repo/text_encoder -> subfolder）
filename = (
    _validate_relative_hub_path(tail, "filename")
    if tail is not None and tail.lower().endswith(_WEIGHT_SUFFIXES)
    else None
)
subfolder = (
    _validate_relative_hub_path(tail, "subfolder")
    if tail is not None and filename is None
    else None
)
return WeightSource(
    original=source,
    kind="huggingface",
    repo_id=repo_id,
    revision=revision,
    subfolder=subfolder,
    filename=filename,
)

```

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

修改 LoRA 下载入口，接入 resolve_weight，按解析结果只下载选中文件与 JSON，并固定 revision。

LoRA 下载入口：按来源类型定向处理。

```
def maybe_download_lora(model_name_or_path: str, local_dir: str | None = None,
                        download: bool = True, weight_name: str | None = None) -> str:
    """
    ...
    # ModelScope 走原有逻辑（猜测文件名 + 全量下载）。
    if envs.SGLANG_USE_MODELSCOPE.get():
        allow_patterns = (
            ["*.json", weight_name, f"**/{weight_name}"]
            if weight_name is not None
            else ["*.json", "*.safetensors", "*.bin"]
        )
        local_path = maybe_download_model(
            model_name_or_path, local_dir, download, is_lora=True,
            allow_patterns=allow_patterns,
        )
        # 与旧行为一致的处理（省略）

    # Hugging Face 路径：先解析权重源，再按解析结果下载。
    resolved_weight = resolve_weight(model_name_or_path, weight_name=weight_name)
    selected_file = resolved_weight.selected_file
    if not selected_file.endswith(".safetensors"):
        raise ValueError(
            "Native diffusion LoRA loading requires a safetensors file, got "
            f"{selected_file!r}"
        )

    source = resolved_weight.inventory.source
    if source.kind == "local":
        assert source.local_path is not None
        if os.path.isfile(source.local_path):
            return source.local_path
        return os.path.join(source.local_path, selected_file)

    # 远程：固定到不可变 revision，只下载选中文件加 JSON sidecars。
    assert source.repo_id is not None
    local_path = maybe_download_model(
        source.repo_id,
        local_dir,
        download,
        is_lora=True,
        allow_patterns=["*.json", selected_file],
        revision=resolved_weight.inventory.resolved_revision or source.revision,
    )
    return os.path.join(local_path, selected_file)
```

评论区精华

PR 无 review 评论，作者在 body 中明确了设计边界：不引入 inspection CLI、tensor metadata parser 等额外能力，仅保留源解析和确定性选择；Hugging Face 路径对非 safetensors 适配器 fail closed，原因是当前 diffusion LoRA loader 只能消费该格式。作者也在 issue comment 中说明 NVIDIA CI 的失败 (hunyu3d_shape_gen) 与本 PR 无关。

- 暂无高价值评论线程

风险与影响

- 风险：
 1. 行为变更：现有依赖 `--lora-path` 猜测行为的用户，若指向含多个 safetensors 的仓库且未给 `--lora-weight-name`，将收到歧义错误，属于破坏性变更。
 2. 网络依赖：远程解析依赖 `HfApi.model_info`，网络抖动或 Hub 不可用会导致加载失败。
 3. 兼容性：ModelScope 路径保留原逻辑，但 Hugging Face 路径的 `allow_patterns` 从原来的 `[".safetensors", ".bin"]` 收窄为仅选中文件，若所选文件实际是 `.bin` 且被拒绝，可能影响未预料的适配器。
 4. 新模块需要持续维护，且测试通过 mock 模拟 Hub 响应，未覆盖真实网络场景。
 - 影响：影响范围限于 diffusion 多模态生成的 LoRA 加载路径。用户可通过精确 URL 或 `weight_name` 获得确定性选择，远程下载量显著减少（只取一个文件 +JSON）。对现有 ModelScope 用户无影响。团队需要维护新的 `weights/source.py` 模块，并注意与 `diffusers` 内部 API（如 `_best_guess_weight_name`）的耦合。
 - 风险标记：破坏现有 LoRA 路径语义，远程下载依赖 Hugging Face Hub，新模块维护成本

关联脉络

- PR #35701 [diffusion] feat: let offloaded weights stay on the checkpoint mapping: 同属 diffusion runtime 权重加载链路，涉及 checkpoint 权重映射与加载方式，与本 PR 的权重源解析目标相通，但无直接文件交集。