# PR #26327 完整报告

sgl-project/sglang

[diffusion] fix: fix diffusion LoRA consistency cases

合并时间：2026-05-28 06:29

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26327

## 执行摘要

- 一句话：修复 Diffusion LoRA 精度与一致性验证
- 推荐动作：该 PR 修复了 diffusion LoRA 多个边界情况，并加强了测试覆盖，值得 review 和 merge。特别关注 FP32 合并默认值变更和 lora_alpha 加载的设计决策。

## 功能与动机

根据 PR body，主要动机是修复 LoRA 合并精度（默认 FP32 合并）、保护缓存 LoRA 条目中的合并强度、加载适配器级别的 lora_alpha，以及为 LoRA API 生成添加一致性验证，确保启动 / 应用 LoRA、merge_lora_weights、set_lora、动态切换和多 LoRA 切换路径生成像素与 GT 比较。

## 实现拆解

1. 合并精度提升：在 linear.py 的 _should_merge_in_fp32 中，将环境变量 SGLANG_DIFFUSION_LORA_MERGE_FP32 的默认值从 0 改为 1，即默认启用 FP32 合并；同时在 merge_lora_weights 中，当传入新 strength 时同步更新 lora_weights_list 中每个条目的 strength。
2. lora_alpha 加载与缓存：在 lora_pipeline.py 的 load_lora_adapter 中，新增从 adapter_config.json 读取 lora_alpha，并存入新字典 loaded_adapter_alphas；在 _apply_lora_to_layers 中，当查找 alpha 时优先使用该缓存值，若没有则回退到 rank。
3. set_lora 路径复用：在 set_lora 中，当传入的 lora_paths 某项为 None 时，从 loaded_adapter_paths 中根据 nickname 获取已有路径，避免重复加载。
4. 一致性测试增强：在 test_server_common.py 中新增 _validate_lora_consistency 方法，并在 _test_lora_api_functionality 和 _test_lora_dynamic_switch_e2e 中增加对合并、设置、动态切换后生成内容的一致性检查。
5. CI 数据基线更新：更新 test_utils.py 中的 SGL_TEST_FILES_CI_DATA_REVISION 指向包含新 LoRA GT 的提交。
6. 其他配套：修复 test_lora_pipeline.py 中测试夹具未初始化 loaded_adapter_alphas 的问题，并为 gpu_cases.py 中 Wan2.2 案例添加 --lora-merge-mode dynamic 参数。

关键文件：

- python/sglang/multimodal_gen/runtime/pipelines_core/lora_pipeline.py（模块 LoRA 管线；类别 source；类型 core-logic）：核心 LoRA 管线，新增 lora_alpha 加载、缓存和

set_lora 路径复用逻辑。

- python/sglang/multimodal_gen/runtime/layers/lora/linear.py（模块 LoRA 合并；类别 source；类型 core-logic）：LoRA 合并精度核心逻辑，修改默认 FP32 合并并修复 strength 传播。
- python/sglang/multimodal_gen/test/server/test_server_common.py（模块 服务测试；类别 test；类型 test-coverage；符号 _validate_lora_consistency）：新增 LoRA 一致性验证方法，增强端到端测试覆盖。
- python/sglang/multimodal_gen/test/test_utils.py（模块 测试工具；类别 test；类型 test-coverage）：更新 CI 数据基线版本。
- python/sglang/multimodal_gen/test/unit/test_lora_pipeline.py（模块 单元测试；类别 test；类型 test-coverage）：修复单元测试夹具未初始化新字段。
- python/sglang/multimodal_gen/test/server/gpu_cases.py（模块 GPU 用例；类别 test；类型 test-coverage）：为 Wan2.2 LoRA 用例添加 dynamic merge 模式。

关键符号：_should_merge_in_fp32, merge_lora_weights, load_lora_adapter, set_lora, _apply_lora_to_layers, _validate_lora_consistency, _test_lora_api_functionality, _test_lora_dynamic_switch_e2e

## 关键源码片段

### python/sglang/multimodal_gen/runtime/pipelines_core/lora_pipeline.py

核心 LoRA 管线，新增 lora_alpha 加载、缓存和 set_lora 路径复用逻辑。

```
# python/sglang/multimodal_gen/runtime/pipelines_core/lora_pipeline.py
# 关键变更：加载适配器级别的 lora_alpha 并缓存

def load_lora_adapter(self, lora_path: str, lora_nickname: str) -> None:
    # ... 下载和归一化 state_dict ...
    raw_state_dict = load_file(lora_local_path)
    lora_state_dict = normalize_lora_state_dict(raw_state_dict, logger=logger)

    # 尝试从 adapter_config.json 读取 lora_alpha（适配器级别）
    adapter_lora_alpha = None
    adapter_config_path = os.path.join(
        os.path.dirname(lora_local_path), "adapter_config.json"
    )
    if os.path.isfile(adapter_config_path):
        with open(adapter_config_path, encoding="utf-8") as f:
            adapter_config = json.load(f)
        if adapter_config.get("lora_alpha") is not None:
            adapter_lora_alpha = int(adapter_config["lora_alpha"])
    # ... 清空或初始化 self.lora_adapters[lora_nickname] ...

    # 填充 lora_adapters 字典
    for target_name, weight in lora_state_dict.items():
        # ... 过滤和映射 ...
```

```
            self.lora_adapters[lora_nickname][target_name] = weight.to(self.device)

        # 缓存路径和 alpha
        self.loaded_adapter_paths[lora_nickname] = lora_path
        self.loaded_adapter_alphas[lora_nickname] = adapter_lora_alpha # 新增：缓存适配器级 alpha

        logger.info("Rank %d: loaded LoRA adapter %s", rank, lora_path)
```

## python/sglang/multimodal_gen/runtime/layers/lora/linear.py

LoRA 合并精度核心逻辑，修改默认 FP32 合并并修复 strength 传播。

```
# python/sglang/multimodal_gen/runtime/layers/lora/linear.py
# 关键变更：默认 FP32 合并 + 更新 strength 到列表

def _should_merge_in_fp32(self, lora_list: list[LoRAWeightEntry]) -> bool:
    # 环境变量默认值从 "0" 改为 "1"：默认启用 FP32 合并
    if os.getenv("SGLANG_DIFFUSION_LORA_MERGE_FP32", "1") != "1":
        return False
    # 对 distilled-lora 路径仍然不使用 FP32（防止精度问题）
    for _, _, lora_path, _, _, _ in lora_list:
        if lora_path and "distilled-lora" in lora_path.lower():
            return False
    return True

@torch.no_grad()
def merge_lora_weights(self, strength: float | None = None) -> None:
    if strength is not None:
        self.strength = strength
        # 新增：将新 strength 应用到 lora_weights_list 的每个条目，保证强度一致
        if self.lora_weights_list:
            self.lora_weights_list = [
                (lora_A, lora_B, lora_path, strength, lora_rank, lora_alpha)
                for (lora_A, lora_B, lora_path, _, lora_rank, lora_alpha)
                in self.lora_weights_list
            ]
    # ... 后续合并逻辑不变 ...
```

## python/sglang/multimodal_gen/test/server/test_server_common.py

新增 LoRA 一致性验证方法，增强端到端测试覆盖。

```
# python/sglang/multimodal_gen/test/server/test_server_common.py
# 新增：LoRA 一致性验证辅助方法

def _validate_lora_consistency(
    self, case: DiffusionTestCase, content: bytes, operation: str
) -> None:
    """
    验证 LoRA 操作后的生成内容与 ground truth 一致性。
    Args:
```

```
        case: 测试用例配置，包含一致性检查开关
        content: 生成的图像或视频字节
        operation: 操作描述，用于日志
    """
    if not case.run_consistency_check:
        logger.info(
            "[LoRA Consistency] Skipping %s consistency for %s: disabled for case",
            operation,
            case.id,
        )
        return

    logger.info(
        "[LoRA Consistency] Validating %s output for %s", operation, case.id
    )
    self._validate_consistency(case, content)
```

# 评论区精华

本次 PR 无公开 review 讨论。

- 暂无高价值评论线程

# 风险与影响

- 风险：
  - FP32 合并默认启用：增加显存占用和计算开销，但对 diffusion 推理整体影响较小（仅合并阶段），精度提升是值得的。
  - 一致性测试依赖远程 GT 仓库：新的 _validate_lora_consistency 会通过网络下载 GT 文件，如果网络不稳定或 GT 版本未同步，可能导致测试失败。
  - lora_alpha 文件缺失：adapter_config.json 可能不存在或格式异常，当前代码仅做了存在性检查，缺少异常处理，但 fallback 到 rank 机制可缓解。
  - 路径复用逻辑：set_lora 中依赖 loaded_adapter_paths 缓存，若缓存被错误清除可能引发后续错误。
- 影响：
  - 用户：使用 diffusion LoRA API 的用户将获得更精确的权重合并和更稳定的多 adapter 切换体验。
  - 系统：默认 FP32 合并增加了少量计算负荷，但提升了生成质量；新增的一致性校验在 CI 中运行，不影响线上服务。
  - 团队：新增的端到端测试覆盖了 LoRA 核心 API 路径，降低回归风险，便于后续重构。
  - 风险标记：FP32 合并默认启用可能影响性能，一致性测试依赖外部 GT 仓库，lora_alpha 文件可能缺失

# 关联脉络

- 暂无明显关联 PR