PR #25591 完整报告

sgl-project/sglang

[diffusion] fix: use dynamic LoRA for LTX2 original stage-two

合并时间：2026-05-18 23:05

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25591

执行摘要

- 一句话：LTX2 original 模式 stage-2 改用动态 LoRA
- 推荐动作：值得精读。该 PR 展示了在共享权重场景下，通过动态 LoRA 避免请求路径上权重变动的设计思路。建议关注 review 中关于阶段无关性的建议，考虑在后续迭代中统一处理 original 模式的所有阶段。

功能与动机

original 模式复用单个 DiT 执行 stage 1 和 stage 2，原先将 stage-2 蒸馏 LoRA 融合到共享 DiT 上，导致请求切换阶段时产生沉重的 merge/unmerge 开销。改用动态 LoRA 后，虽然 stage-2 前向略有额外开销，但避免了改变共享 DiT 权重。PR body 验证显示 *e2e* 时延 11854.79 ms，平均降噪 310.04 ms，GPU 峰值显存 60.35 GB。

实现拆解

1. 新增 _should_merge_lora_for_phase 方法（于 ltx_2_pipeline.py 第 809-814 行）：接收 phase 参数，若 phase 为 "stage2" 且当前驻留模式为 "original"，则返回 False（动态 LoRA）；否则回退到原有的 _should_merge_stage2_distilled_lora 逻辑，即仅 LTX-2.3 原生变体才融合。
2. 修改 switch_lora_phase 中的调用点（第 976-986 行）：原来 stage2 和 HQ 的 stage1 分支直接调用 _should_merge_stage2_distilled_lora，现在改为调用 _should_merge_lora_for_phase(phase)，将决策逻辑统一到新方法中。
3. 更新性能基线（perf_baselines.json）：调整 fastwan2_2_ti2v_5b、ltx_2_3_two_stage_ti2v_2gpus 等测试用例的 *expected_e2e_ms*、*expected_avg_denoise_ms*、LTX2LoRASwitchStage 等数值，以匹配动态 LoRA 带来的时延变化。

关键文件：

- python/sglang/multimodal_gen/runtime/pipelines/ltx_2_pipeline.py（模块 扩散模型；类别 source；类型 core-logic；符号 _should_merge_lora_for_phase）：核心逻辑变更：新增 _should_merge_lora_for_phase 方法并替换 switch_lora_phase 中的调用点，实现 dynamic LoRA 决策。
- python/sglang/multimodal_gen/test/server/perf_baselines.json（模块 测试基线；类别 test；类型 test-coverage）：性能基线更新：调整了 fastwan2_2_ti2v_5b 和

ltx_2_3_two_stage_ti2v_2gpus 等测试用例的预期耗时，反映 dynamic LoRA 的加速效果。

关键符号：_should_merge_lora_for_phase

关键源码片段

python/sglang/multimodal_gen/runtime/pipelines/ltx_2_pipeline.py

核心逻辑变更：新增 _should_merge_lora_for_phase 方法并替换 switch_lora_phase 中的调用点，实现 dynamic LoRA 决策。

```
# 新增方法：根据 phase 和驻留模式决策是否融合 LoRA 权重
# 仅在 original 模式的 stage2 返回 False（动态 LoRA），避免请求路径上权重融合
def _should_merge_lora_for_phase(self, phase: str) -> bool:
    if phase == "stage2" and self._ltx2_residency.mode == "original":
        # original 模式共享一个 DiT，动态 LoRA 避免请求时的 merge/unmerge
        return False
    return self._should_merge_stage2_distilled_lora(self.server_args)


# switch_lora_phase 中原先直接调用 _should_merge_stage2_distilled_lora 的位置
# 改为调用新方法，使决策统一
if phase == "stage2":
    set_lora_kwargs["merge_weights"] = self._should_merge_lora_for_phase(phase)
elif phase == "stage1" and self.pipeline_name == "LTX2TwoStageHQPipeline":
    set_lora_kwargs["merge_weights"] = self._should_merge_lora_for_phase(phase)
```

python/sglang/multimodal_gen/test/server/perf_baselines.json

性能基线更新：调整了 fastwan2_2_ti2v_5b 和 ltx_2_3_two_stage_ti2v_2gpus 等测试用例的预期耗时，反映 dynamic LoRA 的加速效果。

```
// 以 ltx_2_3_two_stage_ti2v_2gpus 为例，LTX2LoRASwitchStage 从 114.37 ms 降至 70.0 ms
{
  "ltx_2_3_two_stage_ti2v_2gpus": {
    "stages_ms": {
      "LTX2LoRASwitchStage": 70.0, // 原为 114.37
      // ... 其他阶段不变
    },
    "expected_e2e_ms": 12000.0, // 原为 17660.77
    "expected_avg_denoise_ms": 238.8
  }
}
```

评论区精华

gemini-code-assist[bot] 指出 _should_merge_lora_for_phase 当前仅对 stage2 返回 False，但在 original 模式下，所有阶段都应避免权重融合，以确保共享 DiT 在请求生命周期内不被修改。该建议未在后续提交中采纳，意味着当前实现仍允许 stage-1 的 LoRA 融合（对于 LTX-2.3 HQ 等原生变体），可能保留部分 merge/unmerge 成本。

- dynamic LoRA 应应用于 original 模式的所有阶段 (design): 未采纳。当前实现仅对 stage2 做动态 LoRA，stage-1 仍沿用原有融合逻辑（对于 LTX-2.3 HQ 等原生变体）。

风险与影响

- 风险：
  1. 回归风险：修改了 switch_lora_phase 中 merge_weights 的决策逻辑，可能影响非 original 模式（snapshot/resident）的行为。但 PR 明确声明保持这些模式不变，且变更仅新增方法并替换调用点，逻辑等价性较高。
  2. 性能基线漂移：perf_baselines.json 中的数值调整若未经过充分测试，可能导致 CI 基线检查失败。
  3. 未完全消除风险：review 指出的 stage-1 融合问题未处理，若 original 模式下 stage-1 也使用融合 LoRA，切换 stage-2 时仍需 unmerge。 - 影响：影响范围：仅影响 LTX2TwoStagePipeline 的 original 模式用户。性能：动态 LoRA 避免了昂贵的权重融合操作，预计减少阶段切换延迟（基线中 LTX2LoRASwitchStage 从 114.37 ms 降至 70.0 ms）。兼容性：original 模式组件放置（含默认非 DiT 层卸载）保持不变，snapshot/resident 行为也保持不变。 - 风险标记：review 建议未完全采纳，性能基线需验证

关联脉络

- 暂无明显关联 PR