PR #34359 完整报告

sgl-project/sglang

[Diffusion] Support native and PEFT MiniMax H3 LoRAs

合并时间：2026-08-12 17:52

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34359

执行摘要

- 一句话：支持 MiniMax H3 原生与 PEFT 双格式 LoRA
- 推荐动作：值得精读。三个设计点有借鉴价值：_compute_lora_delta 用 einsum + flatten 统一 2D/3D 投影；param_names_mapping 用（目标名，偏移，组数）三元组表达 QKV 堆叠契约；alpha 解析采用“显式参数 > adapter_config.json > 默认回退”的三级策略并支持运行时更新。建议关注后续对 lora_param_names_mapping 旧字段的清理，以及 stacked 3D 在 TP 下的切片验证。

功能与动机

PR body 明确两项动机：一是让 MiniMax H3 的 LoRA 支持覆盖社区常见的两种保存格式（native fused 与标准 Diffusers/PEFT split-QKV）；二是显式 alpha 缺失问题——部分单文件 PEFT 适配器不携带 adapter_config.json 或 alpha 元数据，旧实现回退 alpha = rank 并不总是正确：LightX2V 的 rank-128 H3 Turbo 适配器实际 lora_alpha=8，旧回退会把 delta 放大 16 倍。PR 还指出参考实现使用 4 次 denoising 评估，而 SGLang 调度器包含 terminal sigma，因此匹配请求需用 num_inference_steps=5。

实现拆解

1. 参数映射契约（configs/models/dits/minimax_h3.py）：将 lora_param_names_mapping 字段替换为 param_names_mapping，并为 MiniMaxH3DiTArchConfig 内置 Diffusers/PEFT 别名到 H3 原生模块的正则映射；QKV 拆分投影 to_q / to_k / to_v 通过（目标名，组内偏移，组数）三元组表示要堆叠到 fused qkv_proj 的位置；token_refiner 子块与主 block 同构处理。此举让 load_lora_adapter 复用同一套映射逻辑完成 PEFT 键归一化。
2. 计算内核（runtime/layers/lora/linear.py）：新增 _compute_lora_delta，统一处理常规 2D 投影与 stacked 3D 投影（einsum 按组压缩再展开、flatten 拼接），替换 BaseLayerWithLoRA、ColumnParallelLinearWithLoRA、MergedColumnParallelLinearWithLoRA、LinearWithLoRA 四处手写 delta 表达式；同时把 inferred_rank 推断改为 shape[-2]，兼容 3D 堆叠权重。
3. alpha 透传链（server_args.py、lora_pipeline.py、diffusion_generator.py、gpu_worker.py、common_api.py、entrypoints/utils.py、scheduler.py）：新增 --lora-alpha CLI 参数及正整数校验；_normalize_lora_params 增加 lora_alpha 归一化与合法性校验；set_lora / load_lora_adapter 签名扩展，加载时按“显式 alpha >

adapter_config.json > alpha=rank"三级解析；set_lora 检测到 alpha 变化时更新 loaded_adapter_alphas 缓存并触发重新 apply；运行时 /v1/set_lora API 同步支持。

4. 下载与文档（utils/hf_diffusers_utils.py、cookbook）：maybe_download_lora 在指定 --lora-weight-name 时把下载范围收敛为该文件 + *.json 元数据，避免从多适配器 repo 下载全部权重；更新 MiniMax-H3 cookbook、CLI 文档与兼容性矩阵，记录经过验证的适配器与复现命令。
5. 测试配套（test_lora_pipeline.py、test_lora_inference_mode.py、test_minimax_h3_dit_contract.py）：新增 test_lora_alpha_override_updates_cached_adapter_scale、test_pinned_lora_weight_limits_snapshot_download、test_stacked_lora_delta_preserves_projection_order 等 13 个契约测试，覆盖 alpha 缓存更新、pin 单文件下载、stacked 投影顺序等关键行为。

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；符号 set_lora, load_lora_adapter, _normalize_lora_params）：LoRA 管线核心：alpha 参数贯穿参数规范化、加载与缓存更新，并修正 stacked 3D 权重下的 rank 推断。
- python/sglang/multimodal_gen/runtime/layers/lora/linear.py（模块 线性层；类别 source；类型 core-logic；符号 _compute_lora_delta）：新增 _compute_lora_delta 统一 2D/3D LoRA 投影计算，是 stacked QKV 适配器的数值核心。
- python/sglang/multimodal_gen/configs/models/dits/minimax_h3.py（模块 H3 配置；类别 source；类型 data-contract；符号 MiniMaxH3DiTArchConfig）：以 param_names_mapping 定义 PEFT 别名到 H3 原生模块的映射，QKV 用（目标，偏移，组数）三元组表达堆叠契约。
- 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_alpha_override_updates_cached_adapter_scale, test_pinned_lora_weight_limits_snapshot_download）：新增测试覆盖 alpha 覆盖更新缓存与 pin 单文件下载行为。
- python/sglang/multimodal_gen/runtime/server_args/server_args.py（模块 参数配置；类别 source；类型 core-logic；符号 ServerArgs, add_cli_args）：新增 --lora-alpha CLI 参数与正整数校验，是显式 alpha 覆盖的配置入口。
- python/sglang/multimodal_gen/runtime/utils/hf_diffusers_utils.py（模块 下载工具；类别 source；类型 core-logic；符号 maybe_download_lora）：maybe_download_lora 支持仅下载 pin 的权重文件与 JSON 元数据，避免多适配器 repo 全量下载。
- python/sglang/multimodal_gen/runtime/entrypoints/diffusion_generator.py（模块 生成入口；类别 source；类型 dependency-wiring；符号 DiffusionGenerator.set_lora）：set_lora 入口透传 lora_alpha 到 worker 层。
- python/sglang/multimodal_gen/runtime/managers/gpu_worker.py（模块 Worker；类别 source；类型 core-logic；符号 GPUWorker.set_lora）：GPUWorker.set_lora 将 lora_alpha 转发给 LoRAPipeline.set_lora。
- python/sglang/multimodal_gen/test/unit/test_lora_inference_mode.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_stacked_lora_delta_preserves_projection_or

der）：验证 stacked LoRA delta 保持投影顺序，防止 QKV 堆叠错位。

- python/sglang/multimodal_gen/runtime/entrypoints/openai/common_api.py（模块 API 入口；类别 source；类型 entrypoint；符号 set_lora）：/v1/set_lora 请求体新增 lora_alpha 字段，支持运行时更新 alpha。
- python/sglang/multimodal_gen/test/unit/test_minimax_h3_dit_contract.py（模块 单元测试；类别 test；类型 test-coverage）：H3 契约测试扩展，覆盖 PEFT 键到原生模块的映射。
- docs/cookbook/diffusion/MiniMax/MiniMax-H3.mdx（模块 文档；类别 other；类型 documentation）：记录已验证的 H3 适配器、--lora-alpha 用法与可复现命令。
- python/sglang/multimodal_gen/runtime/managers/scheduler.py（模块 调度器；类别 source；类型 dependency-wiring）：set_lora 调度调用透传 lora_alpha 参数（1 行）。
- python/sglang/multimodal_gen/runtime/entrypoints/utils.py（模块 工具；类别 source；类型 dependency-wiring）：工具函数适配 lora_alpha 消息格式化（1 行）。
- docs/docs/sglang-diffusion/api/cli.mdx（模块 文档；类别 other；类型 documentation）：CLI 文档补充 --lora-alpha 参数说明。
- docs/docs/sglang-diffusion/compatibility_matrix.mdx（模块 文档；类别 other；类型 documentation）：更新 LoRA 兼容性矩阵说明。

关键符号：_compute_lora_delta, LoRAPipeline.set_lora, LoRAPipeline.load_lora_adapter, LoRAPipeline._normalize_lora_params, ServerArgs.add_cli_args, maybe_download_lora, DiffusionGenerator.set_lora, GPUWorker.set_lora, common_api.set_lora

关键源码片段

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

LoRA 管线核心：alpha 参数贯穿参数规范化、加载与缓存更新，并修正 stacked 3D 权重下的 rank 推断。

```
def load_lora_adapter(
    self,
    lora_path: str,
    lora_nickname: str,
    rank: int,
    weight_name: str | None = None,
    lora_alpha: int | None = None,
):
    # 仅 rank 0 先下载，其他 rank 等待 barrier 后命中缓存，
    # 避免多 rank 并发读取不完整文件
    if rank == 0:
        lora_local_path = maybe_download_lora(lora_path, weight_name=weight_name)
    else:
        lora_local_path = None
    if dist.is_initialized():
        dist.barrier()
    if rank != 0:
        lora_local_path = maybe_download_lora(lora_path, weight_name=weight_name)
```

```
    raw_state_dict = load_file(lora_local_path)
    lora_state_dict = normalize_lora_state_dict(raw_state_dict, logger=logger)
    # alpha 解析优先级：显式 --lora-alpha > adapter_config.json > 回退 alpha = rank
    # 旧实现一旦缺少元数据就回退 alpha = rank，对 LightX2V 等适配器会放大 16 倍 delta
    adapter_lora_alpha = lora_alpha
    adapter_config_path = os.path.join(
        os.path.dirname(lora_local_path), 'adapter_config.json'
    )
    if adapter_lora_alpha is None and 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'])
```

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

新增 _compute_lora_delta 统一 2D/3D LoRA 投影计算，是 stacked QKV 适配器的数值核心。

```
def _compute_lora_delta(
    x: torch.Tensor, lora_A: torch.Tensor, lora_B: torch.Tensor
) -> torch.Tensor:
    '''Apply a regular or stacked LoRA projection to the last dimension.'''
    if lora_A.dim() == 2 and lora_B.dim() == 2:
        # 常规 2D 路径：与原有 x @ lora_A.T @ lora_B.T 行为完全一致，
        # 保证存量 LoRA 适配器不因本次重构改变数值结果
        return x @ lora_A.T @ lora_B.T
    if lora_A.dim() == 3 and lora_B.dim() == 3:
        # H3 将拆分的 Q/K/V LoRA 堆叠成 3D 投影（组数 = 3）
        if lora_A.shape[0] != lora_B.shape[0]:
            raise ValueError(
                'Stacked LoRA A/B projections must have the same group count, got '
                f'{lora_A.shape[0]} and {lora_B.shape[0]}'
            )
        # 先按组压缩到低秩中间态（...i 是输入维度，nri 是第 n 组的 A 矩阵）
        hidden = torch.einsum('...i,nri->...nr', x, lora_A)
        # 再按组投影回输出维度，最后 flatten 拼接回单个 delta
        delta = torch.einsum('...nr,nor->...no', hidden, lora_B)
        return delta.flatten(start_dim=-2)
    raise ValueError(
        'LoRA A/B projections must both be 2D or both be 3D, got '
        f'{tuple(lora_A.shape)} and {tuple(lora_B.shape)}'
    )
```

python/sglang/multimodal_gen/configs/models/dits/minimax_h3.py

以 param_names_mapping 定义 PEFT 别名到 H3 原生模块的映射，QKV 用（目标，偏移，组数）三元组表达堆叠契约。

```
@dataclass
class MiniMaxH3DiTArchConfig(DiTArchConfig):
    # H3 原生融合了 Q/K/V 投影，因此 PEFT/Diffusers 的拆分 to_q / to_k / to_v
```

```
    # 需要通过 param_names_mapping 重映射并按序堆叠成 fused LoRA 层
    param_names_mapping: dict = field(
        default_factory=lambda: {
            # 通用规则：剥离 PEFT 的 base_model.model / transformer 包装前缀
            r'^base_model\.model\.(.*\.lora_[AB])$': r'\1',
            r'^transformer\.(.*\.lora_[AB])$': r'\1',
            # 输入投影别名：proj_in -> video_patch_proj，audio_proj_in -> audio_patch_proj
            r'^proj_in\.(lora_[AB])$': r'video_patch_proj.\1',
            r'^audio_proj_in\.(lora_[AB])$': r'audio_patch_proj.\1',
            r'^context_embedder\.(lora_[AB])$': r'condition_proj.\1',
            # 时间条件投影：linear_1 / linear_2 对应 proj_in / proj_out
            r'^time_embedder\.linear_1\.(lora_[AB])$': r'time_embedder.proj_in.\1',
            r'^time_embedder\.linear_2\.(lora_[AB])$': r'time_embedder.proj_out.\1',
            # 注意力 QKV：映射值内的三元组（目标名，组内偏移，组数）表示堆叠位置
            r'^transformer_blocks\.(\d+)\.attn\.to_q\.(lora_[AB])$': (
                r'blocks.\1.attn.qkv_proj.\2', 0, 3,
            ),
            r'^transformer_blocks\.(\d+)\.attn\.to_k\.(lora_[AB])$': (
                r'blocks.\1.attn.qkv_proj.\2', 1, 3,
            ),
            r'^transformer_blocks\.(\d+)\.attn\.to_v\.(lora_[AB])$': (
                r'blocks.\1.attn.qkv_proj.\2', 2, 3,
            ),
            # 输出投影与 MLP、AdaLN 别名
            r'^transformer_blocks\.(\d+)\.attn\.to_out\.0\.(lora_[AB])$': r'blocks.\1.attn.out_proj.
            \2',
            r'^transformer_blocks\.(\d+)\.ff\.net\.0\.proj\.(lora_[AB])$': r'blocks.\1.mlp.fc1.\2',
            r'^transformer_blocks\.(\d+)\.ff\.net\.2\.(lora_[AB])$': r'blocks.\1.mlp.fc2.\2',
            r'^transformer_blocks\.(\d+)\.adaln_proj\.linear\.(lora_[AB])$': r'blocks.\1.adaln_proj.
            linear.\2',
            r'^norm_out\.linear\.(lora_[AB])$': r'final_layer.adaln_proj.linear.\1',
            r'^proj_out\.(lora_[AB])$': r'final_layer.video_out.\1',
            r'^audio_proj_out\.(lora_[AB])$': r'final_layer.audio_out.\1',
            # token_refiner 子块与主 block 同构，同样需要 QKV 三元组堆叠
            r'^token_refiner\.refiner_blocks\.(\d+)\.attn\.to_q\.(lora_[AB])$': (
                r'token_refiner.blocks.\1.attn.qkv_proj.\2', 0, 3,
            ),
            r'^token_refiner\.refiner_blocks\.(\d+)\.attn\.to_k\.(lora_[AB])$': (
                r'token_refiner.blocks.\1.attn.qkv_proj.\2', 1, 3,
            ),
            r'^token_refiner\.refiner_blocks\.(\d+)\.attn\.to_v\.(lora_[AB])$': (
                r'token_refiner.blocks.\1.attn.qkv_proj.\2', 2, 3,
            ),
            # 其余 token_refiner MLP / out_proj 条目与主 block 规则同构，此处省略
        }
    )
```

评论区精华

niehen6174 在评论区给出 LGTM，并明确 “This supersedes #34258; I'll close that one in favor of this more complete change.”，说明本 PR 是更完整的实现，取代了早期 PR #34258。PR body 中对 alpha 问题的说明构成核心讨论：回退 alpha = rank 对 LightX2V（实际 lora_alpha=8、rank=128）会放大 16 倍 delta，因此引入显式覆盖参数；验证通过 alpha=8, scale=1.0 与 alpha=128, scale=0.0625 输出哈希一致来证明等价性。

- 显式 alpha 覆盖与 16 倍强度偏差修复 (correctness): 引入显式 --lora-alpha 与运行时 /v1/set_lora 参数，解析优先级为：显式 alpha > adapter_config.json > 默认回退 alpha = rank。
- 取代早期实现 #34258 (design): #34258 被关闭，本 PR 作为唯一完整实现合入。

风险与影响

- 风险： 1) 核心计算路径回归：_compute_lora_delta 替换了全部 LoRA 前向的 delta 表达式，2D 分支必须与旧行为数值等价；现有测试仅覆盖 2D 切片与单一 3D 顺序用例，TP 并行下 stacked 3D 的 slice_lora_b_weights 切分正确性缺乏端到端验证。 2) 参数映射字段迁移：MiniMaxH3DiTArchConfig 用 param_names_mapping 取代 lora_param_names_mapping，lora_pipeline 读取逻辑随之变更；若其他模型配置仍依赖旧字段，需要彻底清理，否则易出现静默失效（提交 b5b9f31 专门做了别名整合，但其他模型使用情况未在本 PR 中确认）。 3) 下载收敛副作用：maybe_download_lora 在 pin weight_name 时排除其他 .safetensors 分片，若该文件是多分片权重的一部分会加载失败；这是 pin 单个文件的语义边界，需文档明确。 4) alpha 回退仍存在：对既无元数据也未显式传 alpha 的适配器，仍回退 alpha=rank，可能延续强度偏差，只能靠文档提示用户显式指定。 5) 全链路覆盖不足：--lora-alpha 需要穿透 HTTP API -> diffusion_generator -> gpu_worker -> lora_pipeline 四层，新增单元测试只覆盖 pipeline 层，API 层缺少集成测试。
  - 影响：用户侧：MiniMax H3 生态的三类社区适配器（Larry native 格式、fal PEFT 格式、LightX2V PEFT 格式）可直接加载，并新增 --lora-alpha 启动参数与 /v1/set_lora 运行时字段，显著降低使用门槛。系统侧：LoRA 计算层获得统一的 2D/3D 抽象，为其他 fused 投影模型（如 QKV 融合）复用 stacked LoRA 铺路；下载流程对多适配器 repo 更精准。团队侧：取代 #34258，避免双实现维护；cookbook 提供可复现的验证基线，压缩未来排查成本。影响范围主要集中在 diffusion 多模态生成子系统，不影响 LLM 文本推理主链路。
  - 风险标记：核心 LoRA 计算路径变更，参数映射字段迁移兼容性，stacked 3D 投影 TP 分片待验证，API 全链路测试覆盖不足，alpha 缺失回退仍可能偏差

关联脉络

- PR #34258：niehen6174 评论确认本 PR 取代 #34258，提供了更完整的 H3 LoRA 实现，原 PR 被关闭。
- PR #34464 Refocus LoRA tests on regression coverage: 同一 LoRA 测试体系近期向回归契约聚焦，本 PR 新增的契约测试与该方向一致。