PR #35004 完整报告

sgl-project/sglang

[Diffusion] Reuse SRT CLIP encoder blocks

合并时间：2026-08-17 19:51

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35004

执行摘要

- 一句话：复用 SRT CLIP 模块，修复 causal/vision 语义漂移
- 推荐动作：值得精读。重点看三处设计：(1) EncoderTensorParallelMixin 把 TP group 绑定到 encoder 实例生命周期，替代全局 patch，如何规避 SRT 层运行时解析 collect group 的陷阱；(2) use_tensor_parallel_group 对 MMGen TP、SRT TP、SRT attention TP 三处上下文的一致保存 / 恢复；(3) 数值验证方法学——与 HF 逐层比对、负对照（pre-fix 运行时）与 GT 归因审计，是处理行为修正类重构的示范。若团队有 diffusion 或 SRT CLIP 相关模型，建议同步回归 FastHunyuan/Wan 输出与 ROCm/MUSA 平台的 attention 路径。

功能与动机

PR body 明确指出两套 CLIP 实现的行为与维护双重漂移：padded 文本输入在 diffusion 副本中丢失 causal masking；SRT 侧视觉 attention 被意外做成 causal；模型块、TP 权重加载与 mask 处理重复实现两遍。此外 encoder folding 在 folding group 下构建并分片复用的 SRT encoder，forward 前又恢复 DiT TP group，而 SRT 层在执行时才解析 collective group，导致 folded 权重可能跑在错误的组上。此前只能靠各模型内部打补丁，无法覆盖所有 encoder 入口。

实现拆解

1. SRT 侧扩展 CLIP 模块（python/sglang/srt/models/clip.py）：新增 prepare_clip_attention_mask，把 padding mask 与上三角 causal mask 合成一张 additive mask；新增带 causal 标志的 CLIPAttention，从 runtime_context.get_parallel() 解析 attention TP 配置，替代原先固定 causal 的 VisionAttention；CLIPEncoderLayer / CLIPEncoder / CLIPTextTransformer 增加 causal、num_hidden_layers_override、act_layer 参数透传；CLIPMLP 在未传入 act_layer 时按 config.hidden_act 选择激活函数。
2. MMGen 侧删除重复实现（python/sglang/multimodal_gen/runtime/models/encoders/clip.py）：文件从 400+ 行缩减到约 20 行包装，直接导入 SRT 的 CLIPEncoder、CLIPTextEmbeddings、CLIPVisionEmbeddings、prepare_clip_attention_mask；新增 _srt_clip_param_name 把 checkpoint 里的 .out_proj. 映射为 SRT 的 .proj.，CLIPTextModel.load_weights 复用该映射处理 q/k/v 到 qkv_proj 的 stacked 加载。

3. 统一 encoder TP group 生命周期（encoders/base.py + distributed/parallel_state.py + loader/component_loaders/text_encoder_loader.py）：新增 EncoderTensorParallelMixin，加载器在构造完成后调用 bind_encoder_tp_group 绑定构建分片所用的 group，__call__时通过 use_tensor_parallel_group 嵌套切换 MMGen TP、SRT TP、SRT attention TP 并在退出时恢复各自原值（替换旧 patch_tensor_parallel_group 的全局单次 patch 语义）。load_model 不再手动 fold_ctx，而是统一用 use_tensor_parallel_group(get_folding_tp_group(config)) 包裹构造与加载，并要求原生 encoder 必须继承 EncoderTensorParallelMixin。get_folding_tp_group 同步收紧，仅保留 sp / world / None 三种模式，移除 ulysses / ring。
4. 其他原生 encoder 接入 mixin：Gemma3ForConditionalGeneration（删除手写的 _vision_parallel_context 与 _vision_tensor_parallel_group）、Gemma2Model、Mistral3ForConditionalGeneration 改为继承 EncoderTensorParallelMixin；mini-max_h3_qwen3vl.py、ideogram.py 做相应配置微调；warmup_request_builder.py 调整视频 warmup 帧数与模型对齐。
5. 测试与 GT 配套：新增双 GPU 端到端 guard（test_encoder_fold_srt_2_gpu.py）验证 fold 与 replicate 输出 parity（SSIM、PSNR、MAD）以及 folded SRT CLIP 与单 rank 一致；新增单元测试（test_srt_clip_reuse.py）锁定组件复用关系、causal 语义传播与 text/vision attention 分离；扩大 health warmup 就绪测试与 CFG parallel warmup 对齐测试；在 sgl-project/ci-data-diffusion 的 PR #2/#3 中重新 pin FastHunyuan、Wan 480P/720P/LoRA、MiniMax-H3 的 GT 帧，并在此 PR 中固定数据 commit。

关键文件：

- python/sglang/multimodal_gen/runtime/models/encoders/clip.py（模块 编码器；类别 source；类型 data-contract；符号 CLIPVisionEmbeddings, init, forward, CLIPTextEmbeddings）：核心消除重复：删除 MMGen 侧 400+ 行 CLIP 实现，改为复用 SRT 组件，仅保留 diffusion 输出 / 池化 / 权重加载包装与参数名映射。
- python/sglang/srt/models/clip.py（模块 编码器；类别 source；类型 data-contract；符号 prepare_clip_attention_mask, CLIPAttention, init, forward）：SRT 侧扩展：新增可区分 text-causal 与 vision-bidirectional 的 CLIPAttention，新增 prepare_clip_attention_mask，并把 causal 语义透传到 CLIPEncoder 与 CLIPTextTransformer。
- python/sglang/multimodal_gen/runtime/models/encoders/base.py（模块 编码器基类；类别 source；类型 data-contract；符号 TextEncoder, EncoderTensorParallelMixin, bind_encoder_tp_group, call）：新增 EncoderTensorParallelMixin，把 TP group 绑定到 encoder 实例生命周期，是修复 fold 场景组错误的核心机制；TextEncoder/ImageEncoder 均接入。
- python/sglang/multimodal_gen/runtime/distributed/parallel_state.py（模块 并行状态；类别 source；类型 core-logic；符号 patch_tensor_parallel_group, use_tensor_parallel_group）：patch_tensor_parallel_group 重构为 use_tensor_parallel_group，从全局单次 patch 改为对 MMGen TP、SRT TP、SRT attention TP 三处一致保存 / 恢复的嵌套 context。

- python/sglang/multimodal_gen/runtime/loader/component_loaders/text_encoder_loader.py（模块 加载器；类别 source；类型 dependency-wiring；符号 load_model, TextEncoderLoader）：加载器接线：load_model 统一用 use_tensor_parallel_group(encoder_tp_group) 包裹构造与权重加载，并要求原生 encoder 继承 EncoderTensorParallelMixin 后绑定 group。
- python/sglang/multimodal_gen/runtime/models/encoders/gemma_3.py（模块 编码器；类别 source；类型 data-contract；符号 Gemma3ForConditionalGeneration, _vision_parallel_context）：Gemma3ForConditionalGeneration 从手写 _vision_parallel_context 迁移到 EncoderTensorParallelMixin，是 mixin 替换手工补丁的代表案例。
- python/sglang/multimodal_gen/test/single_test_file/test_encoder_fold_srt_2_gpu.py（模块 端到端测试；类别 test；类型 test-coverage；符号 _tiny_clip_config, _deterministic_state_dict, _clip_checkpoint_weights, _worker）：新增双 GPU 端到端 guard：fold vs replicate 的 tiny SD3 输出 parity，以及 folded SRT CLIP 与单 rank 数值一致。
- python/sglang/multimodal_gen/test/unit/test_srt_clip_reuse.py（模块 单元测试；类别 test；类型 test-coverage；符号 _clip_config, _FakeQKV, forward, _FakeProjection）：新增单元测试锁定 CLIP 复用关系与语义：text-causal、vision-bidirectional 分离，mask 合成与 fast path。

关键符号：prepare_clip_attention_mask, CLIPAttention.init, CLIPAttention.forward, EncoderTensorParallelMixin.call, EncoderTensorParallelMixin.bind_encoder_tp_group, use_tensor_parallel_group, CLIPTextTransformer.forward, CLIPTextModel.load_weights, Gemma3ForConditionalGeneration, _srt_clip_param_name

关键源码片段

python/sglang/multimodal_gen/runtime/models/encoders/clip.py

核心消除重复：删除 MMGen 侧 400+ 行 CLIP 实现，改为复用 SRT 组件，仅保留 diffusion 输出 / 池化 / 权重加载包装与参数名映射。

```
# SRT 的注意力输出投影参数名是 `proj`，而 HF / diffusion checkpoint 里是
# `out_proj`，加载时统一映射，避免为改名去改动 SRT 侧模块本身。
def _srt_clip_param_name(name: str) -> str:
    return name.replace(".out_proj.", ".proj.")


class CLIPTextTransformer(nn.Module):
    def __init__(
        self,
        config: CLIPTextConfig,
        quant_config: QuantizationConfig | None = None,
        num_hidden_layers_override: int | None = None,
        prefix: str = "",
    ):
        super().__init__()
```

```
        self.config = config
        embed_dim = config.hidden_size

        # embedding 与 encoder 直接复用 SRT 原生模块；CLIPEncoder 传
        # causal=True 显式声明文本语义，视觉侧不传则保持双向 attention
        self.embeddings = CLIPTextEmbeddings(config)
        self.encoder = CLIPEncoder(
            config,
            quant_config=quant_config,
            num_hidden_layers_override=num_hidden_layers_override,
            prefix=prefix,
            causal=True,
        )
        self.final_layer_norm = nn.LayerNorm(embed_dim, eps=config.layer_norm_eps)
        # 用于 pooled_output 计算：定位 eos 位置
        self.eos_token_id = config.eos_token_id

    def forward(
        self,
        input_ids: torch.Tensor | None,
        position_ids: torch.Tensor | None = None,
        attention_mask: torch.Tensor | None = None,
        inputs_embeds: torch.Tensor | None = None,
        output_hidden_states: bool | None = None,
    ) -> BaseEncoderOutput:
        if input_ids is None:
            raise ValueError("You have to specify input_ids")
        input_shape = input_ids.size()
        input_ids = input_ids.view(-1, input_shape[-1])
        hidden_states = self.embeddings(input_ids=input_ids, position_ids=position_ids)

        # mask 构造也统一到 SRT 侧：causal mask 与 padding mask 一次合成，
        # 修复旧实现「给了 mask 就丢掉因果性」的漂移
        attention_mask = prepare_clip_attention_mask(
            input_shape,
            hidden_states.dtype,
            hidden_states.device,
            attention_mask,
        )
        encoder_outputs = self.encoder(
            inputs_embeds=hidden_states,
            return_all_hidden_states=output_hidden_states,
            attention_mask=attention_mask,
        )
        # 后续 final_layer_norm 与 pooled_output 逻辑保持 diffusion 侧原有行为
        last_hidden_state = self.final_layer_norm(encoder_outputs)
        return BaseEncoderOutput(last_hidden_state=last_hidden_state)
```

python/sglang/multimodal_gen/runtime/models/encoders/base.py

新增 EncoderTensorParallelMixin，把 TP group 绑定到 encoder 实例生命周期，是修复 fold 场景组错误的核心机制；TextEncoder/ImageEncoder 均接入。

```
class EncoderTensorParallelMixin:
    """让编码器始终运行在构建分片时所用的 TP group 上。

    旧实现只在 fold 加载期间临时 patch 全局 TP group，forward 前又恢复
    DiT 的 TP group，导致运行时才解析 collective group 的 SRT 层拿到
    错误分组。这里把 group 绑定到 encoder 实例生命周期上，构造、加载、
    每次公开 forward 都使用同一组。
    """

    _encoder_tp_group: GroupCoordinator | None = None

    def bind_encoder_tp_group(self, tp_group: GroupCoordinator) -> None:
        # 由加载器在构造、分片之后调用，记录构建 shards 所用的 group
        self._encoder_tp_group = tp_group

    def __call__(self, *args, **kwargs):
        tp_group = self._encoder_tp_group
        if tp_group is None:
            # 未显式绑定（例如单卡、replicate 场景）时保持原行为
            return super().__call__(*args, **kwargs)
        # 嵌套上下文同时切换 MMGen TP、SRT TP、SRT attention TP，
        # 退出时恢复各自原值，避免污染外层 DiT forward 的组上下文
        with use_tensor_parallel_group(tp_group):
            return super().__call__(*args, **kwargs)


class TextEncoder(
    EncoderTensorParallelMixin, nn.Module, ABC, LayerwiseOffloadableModuleMixin
):
    # 原有 supports_dp_encode、layer_names、_fsdp_shard_conditions 等保持不变
    pass


class ImageEncoder(
    EncoderTensorParallelMixin, nn.Module, ABC, LayerwiseOffloadableModuleMixin
):
    pass
```

评论区精华

该 PR 的讨论集中在作者 mickqian 对行为修正的数值归因与 GT 更新依据上，核心交锋如下：

- FastHunyuan GT 不是阈值放松：修正 causal CLIP 文本语义后，旧实现（只要给了 attention mask 就关闭 causal，包括 FastHunyuan 正常的 all-one mask）与 HF 的池化 cosine 仅 0.817795，sglang 新实现与 HF cosine 0.9999996；作者强调新 GT 遵循修正后的 HF CLIP 语义，并用固定其余变量、只切换 CLIP pooling 的对照证明 mask 变化有真

实的确定性端到端影响。

- Wan I2V 失败归因：2-GPU consistency 失败被证明是 CLIP vision 修复的直接结果——旧 diffusion 实现给视觉 token 加 causal attention，而 HF 是双向的；HF vs 新实现 hidden_states[-2] 的 max_abs=0、cosine 1.0000001，HF vs 旧行为 cosine 仅 0.25202，因此更新 GT 而非改代码。
- MiniMax-H3 GT 损坏排查：全局 ci-data bump 拉入了无关的坏 GT，PR head、base 与 pre-VAE-residency 实现实际生成 byte-identical 帧，最终只恢复三个经校验的 H3 帧。
- LTX-2.3 延迟差异：CI 上的慢样本在隔离 2xH100 上无法复现，同机 A/B 对比 head 与 merge base 中位数约 138 ms vs 134 ms，判定为 run-to-run variance，未改代码或基线。
- 设计决策：旧 patch_tensor_parallel_group 是全局单次 patch 且带断言，新版 use_tensor_parallel_group 改为保存 / 恢复三处（MMGen TP、SRT TP、SRT attention TP）的嵌套 context，配合 EncoderTensorParallelMixin 将 group 绑定到 encoder 实例生命周期，替代各模型手写 group 恢复补丁。
  - 编码器 fold 场景下 SRT 层使用错误 collective group (design): 引入 EncoderTensorParallelMixin 与 use_tensor_parallel_group，构造、权重加载、每次 forward 统一绑定同一 TP group，并保持 MMGen/SRT 三处状态一致恢复。
  - FastHunyuan GT 更新依据：causal 语义修正而非阈值放松 (correctness): 新 GT 遵循修正后的 HF CLIP 语义，CI 数据在 sgl-project/ci-data-diffusion#2 中更新，阈值未放松。
  - Wan I2V 2-GPU consistency 失败归因 (correctness): 更新 480P/720P/LoRA GT 帧并 pin 数据 commit 915a2e312846e，不修改代码逻辑。
  - MiniMax-H3 GT 损坏排查 (testing): 在 ci-data-diffusion#3 中恢复三个经校验的 H3 帧并 pin 到 8917ada9d270a；Wan LoRA 失败不使用 CLIP，视为 rerun 候选。
  - LTX-2.3 延迟差异是否编码器回归 (performance): 判定为 run-to-run variance，不修改 LTX 代码与基线。

风险与影响

- 风险：
  - CLIP 行为变化影响面广：padded 文本恢复 causal mask、vision 改回双向 attention 是确定性行为修正，会改变所有使用 CLIP 的 diffusion 模型（FastHunyuan、Wan 系列、MiniMax-H3、LTX-2 等）的生成结果。GT 虽已在 ci-data-diffusion 中重新 pin，但已部署用户升级后会观察到输出变化，需要作为 breaking 行为告知。
  - SRT 侧回归面：python/sglang/srt/models/clip.py 从 VisionAttention 切换到新的 CLIPAttention，SRT 多模态链路中任何直接或间接使用该模型类的路径（包括非 diffusion 的 CLIP 图文模型）都会受影响；虽有单元测试覆盖 text/vision 语义分离，但真实模型层回归依赖现有 SRT 测试网。
  - 配置项收紧：get_folding_tp_group 移除了 ulysses / ring 两种折叠模式，存量配置若使用这两种模式会直接抛出 ValueError，属于行为收紧而非渐进兼容。
  - 多平台路径差异：旧 MMGen CLIPAttention 对 ROCm / MUSA / XPU 有 is_causal 与 attn_mask 不能同时使用的特殊规避，新 SRT CLIPAttention 走标准 SDPA，这些平台

需要重新验证数值与性能。

- TP group 全局切换语义变化：use_tensor_parallel_group 现在无条件覆盖 SRT 的 _TP / _ATTN_TP 并恢复，若与 speculative decoding 的 draft worker 等依赖 TP patch 的既有路径叠加，嵌套行为需要额外留意；新版保存 / 恢复语义比旧版更稳健，但缺少针对该组合的专项测试。

- 影响：
  - 代码规模：MMGen 侧 clip.py 从约 400 行减至 20 余行包装，重复维护面消失；SRT 侧 clip.py 增加约 136 行但被多处共享，净维护成本下降。
  - 用户影响：所有依赖 CLIP 的 diffusion 管线输出会发生确定性变化（语义修正），vision encoder 在 H100 batch=4 下中位延迟从 1.814 ms 降至 1.584 ms（-12.7%）。
  - 团队影响：encoder 接入有了统一范式（继承 EncoderTensorParallelMixin），新增编码器不再需要手写 group 恢复补丁；fold 的端到端双 GPU guard 与组件级单测提供了回归防线。
  - 测试影响：新增 / 改动 9+ 个测试文件，包含双 GPU fold parity、SRT 组件复用语义、warmup 就绪与帧对齐，并跨仓库更新 FastHunyuan、Wan、MiniMax-H3 的 GT 基线。
  - 风险标记：核心路径变更，行为兼容性，跨模块重构，配置项收紧，多平台回归

关联脉络

- PR #34940 [Diffusion] Fix H3 swap PEFT SwiGLU lora_B halves when loading FFN Lora: 同属 multimodal_gen diffusion 管线（lora_pipeline 与 encoder 加载），且本 PR 评论中详细讨论了 MiniMax-H3 的 GT 变化与 ci-data 修复。
- PR #35111 [AMD] diffusion: normalize ModelOpt-FP8 weights to e4m3fnuz on gfx942: 同属 diffusion 编码器 / 权重加载链路，且都涉及 diffusion 编码器的量化与跨平台数值路径，需要联动回归。
- PR #34999 [Engine] Freeze GC after server warmup: 本 PR 改动了 warmup_request_builder 并新增 health warmup 就绪测试，与 warmup 相关的基础设施改动存在交叉。