PR #34471 完整报告

sgl-project/sglang

[diffusion] Support LTX-2.5

合并时间：2026-08-15 23:36

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34471

执行摘要

- 一句话：为扩散管线新增 LTX-2.5 完整支持
- 推荐动作：值得精读。建议重点关注三点：扩散解码器的性能优化组合（NATTEN 惰性探测、FlexAttention 编译、SwiGLU tile 大小、tiling 策略）、pipeline 的组件化装配（component_uses + use_declared_component 机制），以及蒸馏 / 完整权重双路径的 sigma 调度处理。同时跟踪后续 PR #36026。

功能与动机

PR body 明确列出目标：支持 LTX-2.5 的蒸馏与 dev（full）checkpoint、两阶段上采样、T2I/T2AV/I2AV、TP/SP/CFG 并行、在线 FP8、BCG、DiT 分层卸载、Cache-DiT、时长头、扩散视频解码器、tiling 以及 cookbook。作者在 commit 中说明 LTX-2.5 实为 LTX-2.3 架构，因此复用既有 pipeline 与阶段，在降低维护成本的同时把新能力补齐。

实现拆解

1. 配置与契约层：新增 LTX-2.5 的 arch config（configs/models/dits/ltx_2_5.py、configs/models/vaes/ltx_2_5_video.py、configs/models/decoders/ltx_2_5_diffusion_decoder.py、configs/models/encoders/gemma_4_unified.py），并为 vocoder 扩展嵌套 BWE 配置（__post_init__ 归一化）。这些配置决定权重映射和加载契约，是后续模型 forward 的前提。
2. 模型实现层：新增 ltx_2_5_diffusion_decoder.py（约 1000 行），实现阶段 1-4 确定性上采样 + 阶段 5 patchified 像素去噪；注意力优先走 NATTEN na3d，缺失时回退到编译版 flex_attention。新增 ltx_2_duration_head.py，用 LTX2DurationAttentionPooler 跨注意力池化文本 token，回归视频时长（log-seconds 训练，exp 输出秒）。
3. 管线集成层：ltx_2_pipeline.py 增加 dev/full/sft variant 路由（--model-variant dev 加载 transformer_full）、蒸馏权重固定 sigma 调度、自动插入 LTX2DurationStage（在 latent 准备前改写 num_frames）、支持 --load-diffusion-decoder；decoding_av.py 增加 _decode_with_diffusion_decoder，含 tiling 和 seed 可复现。
4. 并行与性能修复：IPC all-to-all 拒绝 cross-attention 形状（_ipc_input_a2a_qkv）；fp8 路径在 producer 端保证 latent 连续，修复量化内核断言；帧数对齐改为按 SP degree（后因改变 LTX-2/2.3 行为被 revert）；整体通过 NATTEN 与 tiling 将解码时间从超 10 分钟降至可用水平。

5. 文档与测试配套：新增交互式部署命令生成器 ltx25-deployment.jsx 和完整 cookbook（含 NATTEN 安装提示、FP8、CFG 并行）；新增 test_ltx2_5_config.py 等配置接线单测，并移除对 Hub 的依赖。

关键文件：

- python/sglang/multimodal_gen/runtime/models/decoders/ltx_2_5_diffusion_decoder.py（模块 扩散解码器；类别 source；类型 data-contract；符号 _na3d, LTX2VideoDecoderTimestepEmbedder, init, forward）：新增约 1000 行的扩散视频解码器，是 LTX-2.5 的核心新组件。实现 3D 邻域注意力（NATTEN na3d 优先，FlexAttention 编译回退）、patchify/unpatchify 与 tiling 解码，直接决定视频解码质量和性能。
- python/sglang/multimodal_gen/runtime/models/adapter/ltx_2_duration_head.py（模块 时长头适配；类别 source；类型 data-contract；符号 LTX2DurationAttentionPooler, init, forward, LTX2DurationHead）：新增时长头，让 LTX-2.5 无需 --num-frames 即可根据 caption 自动决定片长。通过跨注意力池化把 video/audio 连接器输出回归为时长，是采样参数与扩散帧数之间的关键契约。
- python/sglang/multimodal_gen/runtime/pipelines/ltx_2_pipeline.py（模块 管线装配；类别 source；类型 dependency-wiring；符号 init, _is_dev_variant, _maybe_route_dev_transformer, _declares_component）：管线装配核心：负责 dev variant 路由、时长阶段插入、扩散解码器可选加载、蒸馏 sigma 固定。决定 LTX-2.5 各功能开关如何串联。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/ltx_2/duration.py（模块 时长阶段；类别 source；类型 data-contract；符号 LTX2DurationStage, init, component_uses, forward）：新增自动时长阶段，运行在 latent 准备之前，通过改写 batch.num_frames 实现 auto-duration。展示了 SGLang pipeline stage 的组件化模式（component_uses + use_declared_component）。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/ltx_2/decoding_av.py（模块 解码阶段；类别 source；类型 data-contract；符号 init, _decode_with_diffusion_decoder, _prepare_video_latents）：解码阶段集成扩散解码器替换传统 VAE 解码，新增 _decode_with_diffusion_decoder 与 _prepare_video_latents，处理 tiling、seed 和外部反归一化。
- docs/src/snippets/diffusion/ltx25-deployment.jsx（模块 交互文档；类别 source；类型 core-logic；符号 LTX25Deployment, checkDarkMode, handleRadioChange, getParallelFlags）：新增交互式部署命令生成器，把 LTX-2.5 的硬件、精度、权重、阶段、解码器和时长选项组合成可复制命令，是文档的核心交互组件。
- python/sglang/multimodal_gen/configs/models/dits/ltx_2_5.py（模块 配置层；类别 source；类型 data-contract；符号 LTX25ArchConfig, LTX25Config）：LTX-2.5 DiT 架构配置，定义参数字段映射、继承关系和默认值，是权重加载与模型初始化的契约基础。
- python/sglang/multimodal_gen/test/unit/test_ltx2_5_config.py（模块 单测覆盖；类别 test；类型 test-coverage；符号 TestLTX25DiTConfig, test_inherits_ltx23_audio_video_base, test_feed_forward_bias_is_video_only, test_param_names_mapping_extends_ltx2）：新增 613 行配置接线测试，验证 DiT/VAE

配置继承、字段重命名与默认值不变，是避免回归的关键保障。

关键符号：_na3d, _flex_attention_fn, _patchify, _unpatchify, LTX2DurationHead.forward, LTX2DurationHead.predict_num_frames, LTX2DurationStage.forward, LTX2AVDecodingStage._decode_with_diffusion_decoder, _BaseLTX2Pipeline._maybe_route_dev_transformer

关键源码片段

python/sglang/multimodal_gen/runtime/models/decoders/ltx_2_5_diffusion_decoder.py

新增约 1000 行的扩散视频解码器，是 LTX-2.5 的核心新组件。实现 3D 邻域注意力（NATTEN na3d 优先，FlexAttention 编译回退）、patchify/unpatchify 与 tiling 解码，直接决定视频解码质量和性能。

```
# NATTEN 的 fused `na3d` 是实现 3D 邻域注意力的首选后端，
# 它不需要任何 mask，性能约为编译版 `flex_attention` 回退路径的 4.8 倍。
# 由于 `sglang[diffusion]` 不强制安装 NATTEN，这里做惰性探测并缓存结果。
_na3d_fn: object = None
_NA3D_UNAVAILABLE = object()


def _na3d():
    """返回 NATTEN 的 `na3d`，缺失时返回 `None`。"""
    global _na3d_fn
    if _na3d_fn is None:
        try:
            from natten.functional import na3d

            _na3d_fn = na3d
        except ImportError:
            _na3d_fn = _NA3D_UNAVAILABLE
    return None if _na3d_fn is _NA3D_UNAVAILABLE else _na3d_fn


# 编译并缓存的 `flex_attention` 回退：未编译时会物化完整 `S x S` 分数矩阵，
# 在 121 帧 960x544 这类网格上是数十 GiB 级别；编译后才真正利用邻域窗口
# 的稀疏性，且不同 stage 形状各触发一次重编译。
_compiled_flex_attention = None


def _flex_attention_fn():
    global _compiled_flex_attention
    if _compiled_flex_attention is None:
        from torch.nn.attention.flex_attention import flex_attention

        _compiled_flex_attention = torch.compile(flex_attention, dynamic=False)
    return _compiled_flex_attention
```

```
def _patchify(x: torch.Tensor, patch_size: int) -> torch.Tensor:
    """对 H/W 做 space-to-depth：`(B, C, F, H, W)` -> `(B, C*p**2, F, H//p, W//p)`。

    Channel 的打包顺序是 `(channel, width_offset, height_offset)`，必须
    与上游 checkpoints 一致，否则后续注意力看到的 token 含义会错位。
    """
    batch_size, num_channels, num_frames, height, width = x.shape
    x = x.reshape(
        batch_size,
        num_channels,
        num_frames,
        height // patch_size,
        patch_size,
        width // patch_size,
        patch_size,
    )
    x = x.permute(0, 1, 6, 4, 2, 3, 5)
    return x.reshape(
        batch_size,
        num_channels * patch_size * patch_size,
        num_frames,
        height // patch_size,
        width // patch_size,
    )
```

评论区精华

mickqian 在 review 中对扩散解码器提出三点：能否原生实现 PixArtAlphaCombinedTimestepSizeEmbeddings、是否适配 ParallelTiledVAE / wan_common_utils、是否支持 parallel 与 tiled 解码。AgainstEntropy 回应：cache-dit 可用（复用 LTX-2/2.3 的 DiT），tiled 解码已在 commit 2cd01c0 实现，native adaptation 正在进行；同时澄清它并非 VAE 而是扩散模型，parallel decode 将作为后续 PR 跟进（#36026）。

- 扩散解码器原生化与并行解码支持 (design): 本 PR 先保留可用实现：tiled 已落地，timestep embedder 已改为复用 LTX-2 的原生实现；剩余原生化和并行解码交由后续 PR #36026。

风险与影响

- 风险：
  1. 性能与依赖：NATTEN 不是 sglang[diffusion] 的强制依赖，未安装时静默回退到慢约 4.8 倍的 FlexAttention，普通用户可能无感知地承受低性能；文档已提示安装，但代码无显式警告。
  2. fp8 路径脆弱：曾因 latent 非连续触发量化内核断言，修复放在 producer 端，仍需要回归覆盖不同 batch 形状。

3. 分布式组合复杂：TP/SP/CFG 多轴并行组合容易产生 shape 分歧，帧数对齐修改就曾因改变 LTX-2/2.3 行为被 revert，说明该区域对行为变更敏感。
4. 变更面大：+3693 行、50 个文件，含大量文档和新增模型；PR body 显示 extra CI 曾有失败，合并后仍需关注边界场景。 - 影响：对用户：LTX-2.5 用户可直接用 sglang serve 部署，获得多级并行、FP8、Cache-DiT、两阶段生成等能力；对系统：multimodal_gen 新增两个重量级组件（扩散解码器可选加载，内存占用高），但未触及核心 scheduler/memory 路径；对团队：复用 LTX-2/2.3 架构降低长期维护成本，文档与配置面扩大，后续 #36026 继续推进原生化。 - 风险标记：大面积新增代码，NATTEN 可选依赖，fp8 路径历史缺陷，并行组合复杂，CI extra 失败

关联脉络

- PR #36026 (review 中提及的后续 PR): AgainstEntropy 在 review 中明确指出该 PR 为扩散视频解码器的原生化和 parallel decode 的后续工作。
- PR #36169 [diffusion] docs: desktop-safe 24 GB recipe and the DGX Spark tier: 同为 diffusion 模块的文档演进，两者共同构建 MiniMax-H3 与 LTX-2.5 的部署 cookbook 体系。
- PR #35963 Add Spark3 Model: 同类新增模型架构支持 PR，展示了新模型接入时的配置、测试与文档配套模式。