PR #26950 完整报告

sgl-project/sglang

[diffusion] Align Cosmos3 text packing with official pipeline

合并时间：2026-06-02 02:07

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26950

执行摘要

- 一句话：Cosmos3 文本处理对齐官方 packed-text
- 推荐动作：建议精读本 PR，特别是理解 packed-text 对齐如何影响扩散模型的 UND 通路与生成质量。这是 Cosmos3 功能正确性的关键修复，值得关注。

功能与动机

官方 diffusers 使用 packed-text（即真实文本 token 序列长度）运行 UND 通路，而之前 SGLang 路径在 tokenization 后保留填充，直到 UND K/V cache 之后才切片，导致数值上与官方语义不同。为修复这一差异，需要对齐 packed-text 实现。

实现拆解

1. 暴露真实文本长度：在 cosmos3.py 的 _tokenize_prompt 方法中，返回值从 (input_ids, attention_mask) 扩展为 (input_ids, attention_mask, seq_len)，其中 seq_len 是 padded 之前的真实 token 序列长度。
2. CFG 批处理长度对齐：在 forward 中，对 conditional 和 unconditional 文本，取二者真实长度的最大值作为 shared_seq_len，并在 tokenization 后立即截断两个分支的 text_ids 和 text_mask 至该长度，确保 CFG 双分支 tensor shape 一致。
3. 传递真实长度到模型：将 cond_text_seq_len 和 uncond_text_seq_len 存入 batch.extra，随后经 _run_transformer 传入 cosmos3video.py 的 forward 作为新的 max_text_seq_len 参数。
4. 模型层立即截断：在 cosmos3video.py 的 forward 入口，若 max_text_seq_len 小于 text_ids 的第二维，则立即对 text_ids 和 text_mask 执行截断（取代原来在生成层循环内对 K/V 的切片 k_und[:, :max_real_len]），避免填充通过 UND 通路。
5. 更新一致性 GT：将 test_utils.py 中的 SGL_TEST_FILES_CI_DATA_REVISION 指向 ci-data 仓库新 commit，该 commit 包含了重新生成的 Cosmos3 Nano T2I 一致性 GT。

关键文件：

- python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/cosmos3.py（模块 扩散管道；类别 source；类型 data-contract；符号 _tokenize_prompt, forward, _run_transformer）：主要变更文件：修改了 tokenization 方法签名、forward 中增加了长度对齐逻辑、传递真实文本长度到下游。

- python/sglang/multimodal_gen/runtime/models/dits/cosmos3video.py（模块 扩散模型；类别 source；类型 data-contract；符号 forward, Cosmos3VideoTransformer）：模型 forward 方法新增 max_text_seq_len 参数，并在入口处立即截断 text_ids 和 text_mask，移除原来的 K/V 切片逻辑。
- python/sglang/multimodal_gen/test/test_utils.py（模块 通用工具；类别 test；类型 test-coverage；符号 SGL_TEST_FILES_CI_DATA_REVISION）：更新 ci-data revision 以指向包含重新生成 Cosmos3 Nano T2I 一致性 GT 的 commit，确保测试通过。

关键符号：_tokenize_prompt, Cosmos3TokenizationStage.forward, _run_transformer, Cosmos3VideoTransformer.forward

关键源码片段

python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/cosmos3.py

主要变更文件：修改了 tokenization 方法签名、forward 中增加了长度对齐逻辑、传递真实文本长度到下游。

```
def _tokenize_prompt(
    self,
    text: str,
    max_sequence_length: int,
    device: torch.device,
    use_system_prompt: bool = False,
    system_prompt: str | None = None,
) -> tuple[torch.Tensor, torch.Tensor, int]:
    """Tokenize a prompt using Qwen2 chat template.

    Returns (input_ids, attention_mask, seq_len) as [1, S] tensors.
    """
    # ... (tokenization logic unchanged) ...
    seq_len = len(token_ids) # real length before padding
    # Pad to max_sequence_length (as before)
    # ...
    return input_ids, attention_mask, seq_len # now expose real length

def forward(self, batch: Req, server_args: ServerArgs) -> Req:
    # ... tokenize both cond and uncond ...
    cond_ids, cond_mask, cond_seq_len = self._tokenize_prompt(...)
    uncond_ids, uncond_mask, uncond_seq_len = self._tokenize_prompt(...)
    # Align lengths for CFG batching
    shared_seq_len = max(cond_seq_len, uncond_seq_len)
    cond_ids = cond_ids[:, :shared_seq_len]
    cond_mask = cond_mask[:, :shared_seq_len]
    uncond_ids = uncond_ids[:, :shared_seq_len]
    uncond_mask = uncond_mask[:, :shared_seq_len]
    batch.extra["cond_text_seq_len"] = cond_seq_len
    batch.extra["uncond_text_seq_len"] = uncond_seq_len
```

```
    # ...
```

python/sglang/multimodal_gen/runtime/models/dits/cosmos3video.py

模型 forward 方法新增 max_text_seq_len 参数，并在入口处立即截断 text_ids 和 text_mask，移除原来的 K/V 切片逻辑。

```
def forward(
    self,
    hidden_states: torch.Tensor,
    encoder_hidden_states: torch.Tensor | list[torch.Tensor],
    timestep: torch.LongTensor,
    encoder_hidden_states_image: ... = None,
    guidance=None,
    text_ids: torch.Tensor | None = None,
    text_mask: torch.Tensor | None = None,
    fps: float | None = None,
    cache_key: str = "default",
    noisy_frame_mask: torch.Tensor | None = None,
    max_text_seq_len: int | None = None, # new parameter
    **kwargs,
) -> torch.Tensor:
    # ...
    if max_text_seq_len is None:
        max_text_seq_len = int(text_mask.sum(dim=1).max().item())
    if max_text_seq_len < text_ids.shape[1]:
        # Trim text tensors immediately, not after UND cache
        text_ids = text_ids[:, :max_text_seq_len]
        text_mask = text_mask[:, :max_text_seq_len]
    # ... UND pathway uses trimmed text_ids directly ...
    # Removed: k_und = k_und[:, :max_real_len] inside GEN layer loop
```

评论区精华

无有意义 review 讨论（仅有自动化机器人留言和标签命令）。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 回归风险：文本截断逻辑变更可能影响非 CFG 场景（CFG scale=1.0），但代码中已处理。
  2. 兼容性：max_text_seq_len 新参数设为可选（None 时 fallback 到旧行为），向后兼容。
  3. 性能：截断在 tokenization 后立即执行，相比之前 K/V cache 内切片，减少了 UND 通路的无效计算，有正面性能影响。
  4. 一致性 GT 更新：由于输出语义变化，所有依赖旧 GT 的测试必须更新 ci-data 引用，否则 CI 会失败。

- 影响：
  - 影响范围：仅限于 Cosmos3 文本到图像 / 视频生成路径。
  - 用户影响：对于 SGLang 用户，Cosmos3 生成结果将与官方 diffusers 一致，消除因填充导致的数值差异。
  - 团队影响：需要维护者确保 ci-data 仓库的 GT 文件及时更新以匹配当前代码。
  - 风险标记：数据契约变更，一致性 GT 依赖更新，核心路径变更

关联脉络

- PR #26926 [diffusion] feat: improve cosmos3 serve API support: 本 PR 堆叠在 #26926 之上，扩展了 Cosmos3 的服务 API 支持。
- PR #9 [diffusion] Regenerate Cosmos3 packed-text GT: 关联 ci-data issue，重新生成了 Cosmos3 一致性 GT 以匹配本 PR 的语义变化。