PR #2013 完整报告

radixark/miles

[tito] Add the Inkling TITO family (Inkling / Inkling-Small)

合并时间：2026-08-01 01:12

原文链接：http://prhub.com.cn/radixark/miles/pull/2013

执行摘要

- 一句话：新增 Inkling 家族 TITO 支持，可跑多轮会话
- 推荐动作：值得精读，重点看三个设计决策： 1) 优先复用 HF 原生 chat_template.jinja，仅把固定模板用作测试渲染基准，避免双份渲染逻辑长期漂移； 2) 用四个角色 sentinel 固化 comparator 边界，把 assistant 轮之后的非 assistant 内容差异提升为硬失败，这是 TITO 追加式会话正确性的关键； 3) 大权重进不了 CI 时先用 slice 手动验证并如实记录缺陷边界，而不是注册无人验证过的配置。建议结合 sgl-project/sglang#32958 的解析器修复一起阅读，理解 serving 侧约束。

功能与动机

PR body 明确说明目标：'Registers TITOTokenizerType.INKLING so the session server can drive multi-turn / agentic rollouts on Inkling and Inkling-Small'。Inkling 家族靠 HF 仓库自带的 chat_template.jinja 渲染，且与 sglang 的 token 级 renderer 渲染等价，因此接入时不需要固定模板 override 或边界胶水：子类只需固定 assistant 起始标记、声明消息边界特殊 token、绑定 sglang 的 inkling 推理 / 工具解析器。工具调用解析侧的能力来自关联的 sgl-project/sglang#32958（Fix Inkling tool-call parsing recovery），本 PR 依赖该修复后的 serving 行为。

实现拆解

1. 注册 TITO 子类与工厂：在 miles/utils/chat_template_utils/tito_tokenizer.py 中新增 InklingTITOTokenizer，类属性把 reasoning_parser、tool_call_parser 均绑定为 inkling；__init__ 将 <|message_user|>、<|message_model|>、<|message_system|>、<|message_tool|> 四个角色 sentinel 传入 special_token_ids，使 comparator 在 assistant 轮之后把非 assistant 内容差异判为硬失败（NON_ASSISTANT_TEXT）。随后在 TITOTokenizerType 枚举中新增 INKLING = inkling，并在 get_tokenizer_class 工厂中完成映射，使 CLI 参数 --tito-model inkling 可解析。
2. 新增固定渲染模板：新增 miles/utils/chat_template_utils/templates/inkling_fixed.jinja（133 行），按 HF 原生模板语义复刻 sglang token 级 renderer 的空 scalar-content 行为：空 content 不渲染空 text block、纯工具调用轮不输出 bare assistant terminator、tool call 参数必须是已解析对象而非 JSON 字符串；模板同时登记到 tests/fast/utils/chat_template_utils/test_fixed_templates.py 的固定模板注册表，供家族覆盖测试校验。

3. 扩展会话验证 harness：miles/utils/test_utils/session_verify_runner.py 的 namespace_to_train_args 新增 --sglang-context-length 与 --sglang-cuda-graph-backend-prefill 条件透传，值为 None 时不输出 flag，保持既有家族命令行不变；tests/e2e/sglang/test_session_server_multi_role/_common.py 的 ModelConfig 新增 context_length、rollout_max_response_len、cuda_graph_backend_prefill 字段，并让 global_batch_size 按 rollout_batch_size * n_samples_per_prompt 对齐，避免大模型 lane 减少采样数后训练侧 batch 错配。
4. 测试配套：fast 层新增 comparator 边界测试（TestInklingComparatorBoundaries）、固定模板渲染测试（TestInklingFixedTemplate，如 test_tool_call_only_turn_skips_empty_text_block）、parser 绑定参数化行，以及 test_session_verify_runner.py 的新参数透传与 run_one batch 对齐用例；e2e 层新增 tests/e2e/sglang/test_session_server_multi_role/test_inkling.py，注册 thinkingmachines/Inkling-Small-NVFP4 在 4×H200（TP4、context 32768、cycles 2、append_tool 失败恢复）下的多角色 session 验证。

关键文件：

- miles/utils/chat_template_utils/tito_tokenizer.py（模块 分词器；类别 source；类型 core-logic；符号 InklingTITOTokenizer, init）：核心注册点：新增 InklingTITOTokenizer 子类、TITOTokenizerType.INKLING 枚举与工厂映射，绑定 sglang 的 inkling 推理 / 工具解析器，并设置四个消息角色 sentinel 作为 comparator 边界。
- miles/utils/chat_template_utils/templates/inkling_fixed.jinja（模块 模板；类别 other；类型 core-logic）：新增 133 行固定模板，复刻 HF 原生 chat_template 的空 scalar-content 行为（空文本块、bare terminator、tool call 对象化参数），是 fast 测试的渲染基准。
- miles/utils/test_utils/session_verify_runner.py（模块 会话验证；类别 source；类型 core-logic；符号 namespace_to_train_args）：namespace_to_train_args 新增 --sglang-context-length 与 --sglang-cuda-graph-backend-prefill 透传，是 e2e 配置进入训练参数链路的必经环节。
- tests/fast/utils/chat_template_utils/test_tito_tokenizer.py（模块 分词器测试；类别 test；类型 test-coverage；符号 TestInklingComparatorBoundaries, _build_comparator, test_uses_exact_message_role_boundaries, test_appended_non_assistant_content_is_not_assistant_text）：新增 Inkling comparator 边界、固定模板渲染与 parser 绑定测试，是 fast 层主要验证载体。
- tests/e2e/sglang/test_session_server_multi_role/test_inkling.py（模块 会话测试；类别 test；类型 test-coverage；符号 test_inkling）：新增的 e2e 入口，注册 Inkling-Small-NVFP4 在 4×H200 上的多角色 session 验证，是本 PR 对外部可见的端到端门禁。
- tests/e2e/sglang/test_session_server_multi_role/_common.py（模块 会话测试；类别 test；类型 test-coverage）：ModelConfig 扩展 context_length、rollout_max_response_len、cuda_graph_backend_prefill 字段，并修正 global_batch_size 与样本数对齐，是所有多角色 e2e 的公共基座。
- tests/fast/utils/test_utils/test_session_verify_runner.py（模块 会话验证；类别 test；类型 test-coverage；符号 test_namespace_to_train_args_omits_context_length_by_defau

lt, test_namespace_to_train_args_emits_model_context_length,test_namespace_to_train_args_omits_prefill_cuda_graph_backend_by_default,test_namespace_to_train_args_emits_prefill_cuda_graph_backend）：覆盖新参数透传的默认省略与显式发射行为，以及 run_one 的 batch 对齐逻辑。

- tests/fast/utils/chat_template_utils/test_fixed_templates.py（模块 模板测试；类别 test；类型 test-coverage）：固定模板注册表新增 INKLING 行，确保家族覆盖测试能校验 inkling_fixed.jinja 的存在与参数。

关键符号：InklingTITOTokenizer, TITOTokenizerType.get_tokenizer_class, namespace_to_train_args, run_one

关键源码片段

miles/utils/chat_template_utils/tito_tokenizer.py

核心注册点：新增 InklingTITOTokenizer 子类、TITOTokenizerType.INKLING 枚举与工厂映射，绑定 sglang 的 inkling 推理 / 工具解析器，并设置四个消息角色 sentinel 作为 comparator 边界。

InklingTITOTokenizer 与工厂注册

```
class InklingTITOTokenizer(TITOTokenizer):
    """Inkling 家族（Inkling / Inkling-Small）的 TITO 子类。

    线上由 sglang 的 token 级 renderer 服务，本子类只做三件事：
    绑定 inkling 解析器、固定 assistant 起始标记、声明 comparator 边界。
    """

    reasoning_parser = 'inkling'
    tool_call_parser = 'inkling'

    FIXED_TEMPLATE = FixedTemplate(template='inkling_fixed.jinja')

    # Inkling 的 assistant 消息以 <|message_model|> 开头
    _DEFAULT_ASSISTANT_START = '<|message_model|>'

    def __init__(self, tokenizer, chat_template_kwargs=None, assistant_start_str=None):
        super().__init__(
            tokenizer,
            chat_template_kwargs=chat_template_kwargs,
            assistant_start_str=assistant_start_str or self._DEFAULT_ASSISTANT_START,
            # 四个角色 sentinel 都是 comparator 边界：assistant 轮之后
            # 非 assistant 内容差异按硬失败（NON_ASSISTANT_TEXT）处理
            special_token_ids={
                tokenizer.convert_tokens_to_ids('<|message_user|>'),
                tokenizer.convert_tokens_to_ids('<|message_model|>'),
                tokenizer.convert_tokens_to_ids('<|message_system|>'),
                tokenizer.convert_tokens_to_ids('<|message_tool|>'),
            },
```

```
        )


class TITOTokenizerType(StrEnum):
    # ... 既有家族省略 ...
    INKLING = 'inkling'

    @classmethod
    def get_tokenizer_class(cls, t):
        match t:
            # ... 既有分支省略 ...
            case cls.INKLING:
                return InklingTITOTokenizer
            case _:
                raise ValueError(f'Unknown TITOTokenizerType: {t!r}')
```

评论区精华

review 交互非常轻：guapisolo 先以 DISMISSED 状态评论 LGTM，随后在 CI passed 后正式 APPROVED，无实质代码讨论。核心经验记录在 PR body 中：作者手动在 6 层 surgery checkpoint（8×H200、双 sglang 引擎 TP4/EP4、megatron actor TP4+SP）上跑 session-server e2e，128 个样本 0 个 token 序列失配；同时记录了两个非 TITO 缺陷——1) slice 无法产生格式良好的 tool call，无法满足 harness 的 append_tool 覆盖门；

2) temperature 0.7 sampling 下近随机输出偶尔复现 Inkling 自己的消息边界特殊 token，造成 segment-count mismatch（37 vs 39 段），greedy 解码可消除。这些记录为后续全权重验证提供了直接参考。

- slice 验证的局限与两个非 TITO 缺陷 (testing): 均判定为非 TITO 缺陷并记录；后续第二个 commit 用 NVFP4 完整权重注册正式 e2e，覆盖门由完整权重的 append_tool 模式承担。
- 审核流程：DISMISSED 后 APPROVED (other): 审核通过并合入 main。

风险与影响

- 风险： 1) 模板漂移：inkling_fixed.jinja 是对 sglang token 级 renderer 的复刻，且仅用于测试对照；线上渲染仍以 HF 原生 chat_template.jinja 为准。若 sglang 上游 renderer 或 inkling 解析器更新（如 sgl-project/sglang#32958 的后续演进），模板、comparator 边界与线上渲染可能失配，导致 session server 前缀校验 500。 2) 兼容性：namespace_to_train_args 新增读取 ns.sglang_context_length 与 ns.sglang_cuda_graph_backend_prefill，若仓库内其他调用方构造 Namespace 时未设置这两个属性，将直接触发 AttributeError；本轮已更新 _common.run_one 与 fast 测试的默认值，但仍需核查其余调用路径。 3) e2e 资源与稳定性：test_inkling 注册 4×H200（est_time 1200 秒）、上下文截断至 32768（官方 1M 上下文 recipe 需 TP8），与生产形态有差异；CI 拉取 276 B 级 NVFP4 checkpoint 有网络与存储压力，且若跑 sampling 可能复现 body 中记录的 segment mismatch，存在 flaky 风险。 4) 依赖外部权重：thinkingmachines/Inkling-Small-NVFP4 属于远端模型，权重变更会影响 e2e 结果。

- 影响：对训练 / 推理使用方：获得 --tito-model inkling 入口，session server 可对 Inkling / Inkling-Small 执行 append_user、append_system、append_assistant、force_final、rollback、append_tool 全动作的多轮验证与 rollout 数据组装。对系统：TITO 家族注册表、comparator 边界、固定模板注册表同步扩展，与既有 qwen3 / glm47 / deepseekv4 等家族的接入模式完全一致，无架构性改动。对团队：新增一个必须跟随 sglang inkling parser 演进而维护的模型家族；CI 时长与资源占用上升（一条 4×H200 lane），且未来 sglang bump 时需要回归该家族的 e2e。
- 风险标记：新模型家族接入，手写模板与 sglang 渲染一致性，e2e 依赖大权重资源，namespace 新属性兼容性

关联脉络

- PR #2009 [docs] Add Inkling-Small model page: 同一模型家族（Inkling-Small）的文档接入，说明该家族是持续布局的功能线。
- PR #1820 feat(session): pass request chat_template_kwargs to apply_chat_template: 同属 tito_tokenizer / session 渲染链路的演进，为本 PR 的 chat_template_kwargs 处理提供基础。
- PR #1795 Bump sglang to v0.5.16: sglang 版本升级会带入 inkling parser 相关修复（如 sgl-project/sglang#32958），影响本家族的解析行为与 e2e 稳定性。