PR #25233 完整报告

sgl-project/sglang

[Fix] DeepSeek-V3.2: build structural tag locally to encode both wrapper and invoke layers

合并时间：2026-05-16 05:32

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25233

执行摘要

- 一句话：本地构建结构标签修复并行工具调用
- 推荐动作：值得精读。PR 展示了如何通过 override detector 方法在本地构建结构化标签来绕过上游 bug，设计清晰、docstring 详尽。团队应关注 xgrammar 上游修复进度，以便未来移除 override。

功能与动机

Fixes two layered regressions in the DeepSeek-V3.2 nightly tool-call test. The xgrammar built-in template forces a double-newline between consecutive invoke blocks, causing parallel tool calls to collapse to one at greedy decoding (mlc-ai/xgrammar#637). The legacy fallback path only constrains the inner invoke block, missing the outer wrapper, leading to list index out of range under tool_choice="required" (introduced by #21593 and #21722).

实现拆解

1. 导入 xgrammar 类型并定义常量：在 deepseekv32_detector.py 顶部添加 try/except 导入 xgrammar 的 StructuralTag 和结构标签构建组件（如 ConstStringFormat、TagsWithSeparatorFormat 等），以及定义 invoke 前后缀、thinking 标签排除 token 等常量。
2. 重写 get_structural_tag 方法：移除原来的 get_structural_tag_name 方法，新增 get_structural_tag 方法。该方法根据 tools 和 tool_choice 参数，构建一个完整的本地 StructuralTag：外层 <｜DSML｜function_calls> 和 </｜DSML｜function_calls> 包装，内层 <｜DSML｜invoke> 块，invoke 之间使用空分隔符（invoke_end 自带换行），从而匹配 chat template 的单换行连接。
3. 新增 _invoke_tag 辅助方法：为每个 tool 生成一个 TagFormat，使用工具名和参数 JSON Schema 作为内嵌标签。
4. 处理 thinking 模式：当 thinking_mode 为 True 时，在结构标签前加入可选的 <think>...</think> 标签前缀。
5. 移除对 xgrammar builtin 的依赖：由于返回了非 None 的 StructuralTag，FunctionCallParser 的 dispatch 逻辑会跳过 xgrammar builtin 和 legacy 回退分支，直接使用本地构建的标签。测试方面：没有新增独立单元测试，但通过 nightly 多 GPU 测试（

test_deepseek_v32_all_variants）验证了 36/36 子测试通过。

关键文件：

- python/sglang/srt/function_call/deepseekv32_detector.py（模块 函数调用；类别 source；类型 core-logic；符号 get_structural_tag, _invoke_tag, get_structural_tag_name）：唯一变更文件，通过 override get_structural_tag 方法实现修复

关键符号：DeepSeekV32Detector.get_structural_tag, DeepSeekV32Detector._invoke_tag

关键源码片段

python/sglang/srt/function_call/deepseekv32_detector.py

唯一变更文件，通过 override get_structural_tag 方法实现修复

```
def get_structural_tag(
    self,
    tools: Union[List[Tool], None] = None,
    tool_choice: Union[ToolChoice, Literal["auto", "required"]] = "auto",
    thinking_mode: bool = False,
) -> Optional["StructuralTag"]:
    """Build an xgrammar StructuralTag locally for DeepSeek-V3.2 to avoid two layered defects in
    the built-in template and legacy fallback."""
    if not tools or StructuralTag is None:
        return None

    invoke_end = self.invoke_end_token + "\n" # "</｜DSML｜invoke>\n"
    function_calls_begin = self.bot_token + "\n" # "<｜DSML｜function_calls>\n"

    # 为每个工具生成 inner invoke 标签，使用空 separator 联合以保持单换行
    invoke_tags = [self._invoke_tag(tool, invoke_end) for tool in tools]

    if tool_choice == "required" or isinstance(tool_choice, ToolChoice):
        # required / named: 外层 wrapper 强制出现
        outer_wrapper = SequenceFormat([
            AnyTextFormat(),
            ConstStringFormat(function_calls_begin),
            TagsWithSeparatorFormat(tags=invoke_tags, separator="", at_least_one=True),
            ConstStringFormat("</｜DSML｜function_calls>"),
        ])
    else: # "auto"
        # auto: 外层 wrapper 需要 <｜DSML｜function_calls> 触发后才出现
        outer_wrapper = SequenceFormat([
            AnyTextFormat(),
            TriggeredTagsFormat(
                ConstStringFormat(function_calls_begin),
                TagsWithSeparatorFormat(tags=invoke_tags, separator="", at_least_one=True),
                ConstStringFormat("</｜DSML｜function_calls>"),
            ),
        ])
```

```
    # 可选思考标签前缀
    if thinking_mode:
        tag = SequenceFormat([
            AnyTextFormat(),
            TagFormat("<think>", _THINK_EXCLUDE_TOKENS, "</think>"),
            outer_wrapper,
        ])
    else:
        tag = outer_wrapper

    return StructuralTag(tag, _XML_STYLE)

def _invoke_tag(self, tool: Tool, invoke_end: str) -> TagFormat:
    """构建单个 invoke 的 TagFormat，参数使用 JSON Schema 约束。"""
    param_schema = JSONSchemaFormat.from_json_schema(tool.function.parameters)
    return TagFormat(
        _INVOKE_BEGIN_PREFIX + tool.function.name + _INVOKE_BEGIN_SUFFIX,
        param_schema,
        invoke_end,
    )
```

评论区精华

PR 作者在评论中解释了 CI 中一个 *decode speed* 测试因 GPU timing flake 失败（179.44 tok/s vs 180 tok/s 阈值，0.3% 偏差），但与修复无关，准确性测试全部通过。同时作者强调本修复使用 stock xgrammar 0.2.0 验证，不依赖上游 PR mlc-ai/xgrammar#638。

- 并行工具调用坍塌与修复验证 (correctness): 修复在 stock xgrammar 0.2.0 上验证通过，所有变体（DP8/DP8+MTP/TP8/TP8+MTP）均通过工具调用测试。

风险与影响

- 风险：
  1. 兼容性风险：本地构建的 StructuralTag 可能与未来 xgrammar 版本中的标签定义产生差异，需要在 xgrammar 修复（#638 合并）后评估是否移除 override。
  2. 维护风险：新增的 _invoke_tag 和 get_structural_tag 增加了 detector 代码复杂度，后续需要同步维护与 chat template 的一致性。
  3. 缺少测试覆盖：本次变更没有新增单元测试，仅依赖 nightly 集成测试，回归检测能力较弱。 - 影响：影响范围限定于 DeepSeek-V3.2 模型的工具调用功能，修复了并行工具调用失效和解析错误两个严重 bug，使用体验恢复正常。无性能影响，对系统其他模块无影响。影响程度中等（仅特定模型，但功能影响大）。 - 风险标记：依赖上游 xgrammar 修复后的兼容性，缺少测试覆盖

关联脉络

- PR #21722 Route DeepSeek-V3.2 to xgrammar builtin structural tag: 引入了 xgrammar 内置标签路径，该路径的双换行 bug 导致并行工具调用坍塌

- PR #21593 Add legacy structural tag fallback for DeepSeek-V3.2: 引入了 legacy 回退路径，该路径缺少外层 wrapper 导致下标越界