

PR #36626 完整报告

sgl-project/sglang

[Fix] Resolve tool argument types through top-level anyOf/oneOf/allOf

合并时间: 2026-08-28 12:50

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/36626>

执行摘要

- 一句话: 修复顶层 anyOf/oneOf/allOf 工具参数解析, 覆盖 13 个解析器
- 推荐动作: 值得精读。建议关注三点: (1) get_schema_properties 的递归合并与 setdefault 首分支优先策略, 可作为 "兼容组合 schema" 的通用样板; (2) glm 流式收尾从字符串启发式到状态判断的修复思路, 类似的 endswith 误判在流式解析中很常见; (3) 测试裁剪策略——只保留未修复代码上必失败的用例, 避免回归测试 "假绿"。对正在维护工具调用解析器或接入新模型的工程师有直接参考价值。

功能与动机

PR body 明确指出: Tool schemas that compose subschemas with a top-level anyOf/oneOf/allOf (合法的 JSON Schema / OpenAI 规范写法, 例如 discriminated-union 参数) 会破坏大部分 SGLang tool-call 解析器——嵌套 object/array/scalar 参数被当作 JSON 字符串返回, 或 (glm45/glm47 流式) 直接变成非法 JSON。根因是所有受影响的 detector 只导航 parameters["properties"] 顶层; 组合 schema 下没有 properties 键, 类型推断失败后一律按字符串处理。例如 qwen3_coder 的 _get_arguments_config 会返回整个 schema (含 oneOf/type 键) 作为参数表, 所有值原样输出, 标量参数 "count": "7"、"verbose": "True" 也受影响。此外 glm45/glm47 流式用 _last_arguments.endswith("{}") 判断是否补外层闭合大括号, 最后一个参数为 object/array 时尾部 } 属于嵌套值, 外层对象永不闭合, 流式 arguments 成为非法 JSON (仅当模型在 内输出尾部空白时被掩盖)。

实现拆解

1. 新增共享工具 get_schema_properties() (python/sglang/srt/function_call/utils.py, +19 行): 优先返回顶层 properties; 缺失时按 anyOf/oneOf/allOf 顺序递归合并各分支的 properties, setdefault 保证重复 key 取第一个声明它的分支 (与 oneOf 分支优先级语义一致); 非 dict 输入安全返回空字典, 不抛异常。
2. 全量接入 13 个 detector: 把各自 params.get("properties", {}) 的顶层查找替换为 get_schema_properties(params), 涉及 glm4_moe/glm47_moe、qwen3_coder、step3、minicpm5、hunyuan、spark25、minimax_m2/minimax_m3、dots、poolside_v1、mimo、kimik2, 每处仅 1-3 行。两个刻意行为变化: minicpm5 的 allowed-props 过滤此前在组合 schema 下被绕过 (空属性集), 现在激活, 未声明参数会被丢弃 (与 flat schema 行为对齐); qwen3_coder 对声明为空的 properties: {} 保留旧语义, 不产生 "param not defined" 误报。

3. 流式收尾二次修复: glm45/glm47 的 `_finalize_tool_call` 以 `_is_first_param` 状态判断 "是否已开始输出参数" 来闭合外层对象, 替代 `endswith("{}")` 启发式; 空对象分支 (`_sent_empty_object`) 与 `_last_arguments += "{}"` 逻辑保持原样。
4. 测试与验证配套: `test_function_call_parser.py` 新增 `TestGetSchemaProperties` (4 个纯函数用例: flat、顶层组合、anyOf/allOf 嵌套、非 dict/ 缺失输入) 与 `TestTopLevelCompositeToolSchema` (6 个端到端用例: glm45/glm47 oneOf 流式、flat 流式大括号闭合、qwen3_coder 流式 / 非流式); `test_minimax_m3_detector.py` 新增 `TestMinimaxM3TopLevelOneOf` (oneOf 下整数 / 布尔标量强制转换)。完整跑两个文件共 270 个用例无回归; 第三个 commit 主动裁剪了未修复代码上也能通过的弱用例 (glm 非流式有 `json.loads` 兜底、minimax 嵌套标签解析是结构驱动), 改以标量强制转换场景覆盖必坏路径。GPU A/B: 6 类 schema (oneOf 嵌套 object、anyOf、allOf、array/scalar 强制转换、flat 对照、多工具混合) × 流式 / 非流式, Qwen3-Coder 2/12 → 12/12, GLM-4.5-Air-FP8 7/12 → 12/12。
5. 刻意不动的部分: `kimik3_structural_tag` 走 grammar 构建路线, 顶层组合 schema 下已能优雅降级到 `AnyTextFormat`, 不接入本次修复。

关键文件:

- `python/sglang/srt/function_call/utils.py` (模块 工具函数; 类别 source; 类型 core-logic; 符号 `get_schema_properties`): 新增共享函数 `get_schema_properties()`, 是本 PR 的核心修复入口: 递归展开 anyOf/oneOf/allOf 组合分支, 供全部 13 个 detector 复用。
- `test/registered/unit/function_call/test_function_call_parser.py` (模块 解析器测试; 类别 test; 类型 test-coverage; 符号 `TestGetSchemaProperties`, `test_flat_properties`, `test_top_level_combinators`, `test_anyof_allof_and_nesting`): 新增 `TestGetSchemaProperties` 与 `TestTopLevelCompositeToolSchema` 共 10 个用例, 覆盖组合 schema 类型解析与流式大括号闭合, 是本次修复的主要回归防线。
- `test/registered/unit/function_call/test_minimax_m3_detector.py` (模块 检测器测试; 类别 test; 类型 test-coverage; 符号 `TestMinimaxM3TopLevelOneOf`, `setUp`, `test_detect_and_parse`): 新增 `TestMinimaxM3TopLevelOneOf`, 专门验证 oneOf 组合 schema 下整数 / 布尔标量强制转换路径, 覆盖 `minimax_m3` 的非流式解析修复。
- `python/sglang/srt/function_call/minicpm5_detector.py` (模块 MiniCPM 检测器; 类别 source; 类型 dependency-wiring; 符号 `get_argument_type`, `detect_and_parse`): `get_argument_type` 与 `allowed-props` 过滤改用 `get_schema_properties`, 组合 schema 下未声明参数开始被过滤, 是 PR 主动声明的行为变化点。
- `python/sglang/srt/function_call/glm47_moe_detector.py` (模块 GLM 检测器; 类别 source; 类型 core-logic; 符号 `get_argument_type`, `_finalize_tool_call`): 除接入共享工具外, 还修复了流式收尾 `endswith("{}")` 启发式缺陷, 是本 PR 中第二处核心逻辑变更。
- `python/sglang/srt/function_call/dots_detector.py` (模块 Dots 检测器; 类别 source; 类型 dependency-wiring; 符号 `_tool_schema`): `_tool_schema` 返回的属性表改用 `get_schema_properties` 解析, 保证 Dots 模型在组合 schema 下也能拿到完整属性集合。
- `python/sglang/srt/function_call/spark25_detector.py` (模块 Spark 检测器; 类别 source; 类型 dependency-wiring): Spark25 检测器的参数类型查找路径接入 `get_schema_properties`, 属于 13 个接入点之一。

- python/sclang/srt/function_call/glm4_moe_detector.py (模块 GLM 检测器; 类别 source; 类型 core-logic) : glm45 与 glm47 同构, 参数类型查找路径统一接入共享工具。
- python/sclang/srt/function_call/minimax_m2.py (模块 Minimax 检测器; 类别 source; 类型 dependency-wiring) : MiniMax M2 检测器的参数 schema 导航接入共享工具, 与历史 PR #35290 的模型加载修复形成配套。
- python/sclang/srt/function_call/minimax_m3.py (模块 Minimax 检测器; 类别 source; 类型 dependency-wiring; 符号 _get_child_schema) : _get_child_schema 此前只导航顶层 properties, 组合 schema 下解析为 None 并损坏非流式嵌套 object 参数, 本 PR 修复该路径。
- python/sclang/srt/function_call/qwen3_coder_detector.py (模块 Qwen3 检测器; 类别 source; 类型 core-logic; 符号 _get_arguments_config) : qwen3_coder 是受影响最严重的解析器 (GPU 基线仅 2/12) : _get_arguments_config 此前把整个 schema 当作参数表, 所有值按字符串返回。
- python/sclang/srt/function_call/mimo_detector.py (模块 MIMO 检测器; 类别 source; 类型 dependency-wiring) : MIMO 检测器的参数类型查找路径接入共享工具, 与其他 detector 保持一致。

关键符号: get_schema_properties, get_argument_type, _finalize_tool_call, _get_child_schema, _tool_schema, _get_arguments_config

关键源码片段

python/sclang/srt/function_call/utils.py

新增共享函数 get_schema_properties(), 是本 PR 的核心修复入口: 递归展开 anyOf/oneOf/allOf 组合分支, 供全部 13 个 detector 复用。

utils.py 新增的共享入口: 所有 detector 的参数属性查找统一走这里

```
def get_schema_properties(schema: Any) -> Dict[str, Any]:
    """返回 tool parameters schema 顶层的 properties 字典。
```

当顶层没有直接声明 properties 时, 递归下探 anyOf/oneOf/allOf 组合分支 (JSON Schema 允许这种写法, 例如 discriminated-union 参数)。

```
"""
```

```
if not isinstance(schema, dict):
    return {}
```

顶层直接声明 properties 时优先返回, 保持 flat schema 的旧行为不变

```
properties = schema.get("properties")
```

```
if isinstance(properties, dict):
```

```
    return properties
```

否则按 anyOf/oneOf/allOf 顺序合并各分支的 properties;

setdefault 保证重复 key 取第一个声明它的分支, 与 oneOf 分支优先级一致

```
merged: Dict[str, Any] = {}
```

```
for keyword in ("anyOf", "oneOf", "allOf"):
```

```
    branches = schema.get(keyword)
```

```
    if isinstance(branches, list):
```

```
        for branch in branches:
```

```

        # 递归处理嵌套组合，支持 anyOf 里再套 oneOf 等深层结构
        for key, value in get_schema_properties(branch).items():
            merged.setdefault(key, value)
    return merged

```

python/sclang/srt/function_call/glm47_moe_detector.py

除接入共享工具外，还修复了流式收尾 `endswith("{}")` 启发式缺陷，是本 PR 中第二处核心逻辑变更。

```

# glm47_moe_detector.py: 两处核心改动
def get_argument_type(
    func_name: str, arg_key: str, name_to_tool: Dict[str, Tool]
) -> Optional[str]:
    # ...
    # 兼容 tool.function 上可能不存在 parameters 属性的情况
    params = getattr(tool.function, "parameters", None)
    # 统一走 get_schema_properties: 顶层为 anyOf/oneOf/allOf 组合时也能取到参数定义
    arg_spec = get_schema_properties(params).get(arg_key)
    if isinstance(arg_spec, dict):
        # 复用复杂 JSON Schema 类型推断（支持 type 数组 / enum / 组合类型）
        return infer_type_from_json_schema(arg_spec)
    # 无类型信息时返回 None，调用方沿用字符串兜底
    return None

# _finalize_tool_call 的流式收尾分支：
# 旧逻辑用 _last_arguments.endswith("{}") 判断是否补外层 "{}";
# 当最后一个参数是 object/array 时，尾部 "{}" 属于嵌套值，
# 外层对象永不闭合，流式输出的 arguments 成为非法 JSON。
# 新逻辑改为跟踪参数流式输出状态：
if <空参数分支，保持原逻辑>:
    self._last_arguments += "{}"
    self.streamed_args_for_tool[self.current_tool_id] += "{}"
    self._sent_empty_object = True
elif not self._is_first_param and not self._sent_empty_object:
    # 已开始输出至少一个参数时，在这里闭合外层对象
    calls.append(ToolCallItem(tool_index=self.current_tool_id))

```

评论区精华

流程上 Fridge003 直接 APPROVED，无任何行内 review 评论；PR 评论区只有 4 次 CI 指令（/tag-and-rerun-ci 与 3 次 /rerun-failed-ci），无技术交锋。技术决策集中记录在 PR body 与 commit message 中：

duplicate keys resolve to the first branch that declares them, matching oneOf preference order

minicpm5: for top-level-combinator tools the allowed-props filter was previously bypassed (empty property set); it now activates, so undeclared arguments are

dropped

qwen3_coder: a declared-but-empty properties: {} keeps its old semantics

- 第二个 commit 提到 "review follow-ups", 说明合入前有 reviewer 反馈 (未留下评论痕迹), 包括 qwen3_coder 重构与 minimax_m3 补漏。
- 第三个 commit 体现测试纪律: > Drop cases that also pass on unfixed code ... rewrite the minimax_m3 case around scalar coercion, which is the path that actually breaks without schema info
- glm45/glm47 流式收尾 endswith("{}") 启发式缺陷 (correctness): 改为以 _is_first_param 状态判断是否已开始输出参数, 有参数即闭合外层对象; 空对象分支 _sent_empty_object 保持不变。
- minicpm5 allowed-props 过滤行为变化 (design): 接受该行为变化并在 commit 中保留说明; 对依赖透传行为的客户端属于 breaking。
- 组合分支重复 key 的首分支优先约定 (design): 对 anyOf/allOf 而言这是约定而非语义保证, 实际调用匹配后声明 key 的分支时仍可能类型误判, 属已知局限。
- 测试裁剪: 只保留未修复代码上必失败的用例 (testing): 保留真正有回归价值的用例, 避免 "假绿" 测试; 每个新增用例都先在未修复 main 上验证为失败。

风险与影响

- 风险:
 - 回归面: 改动横跨 13 个 detector 的参数属性查找路径, 每处只有 1-3 行但都处于工具调用解析关键路径; step3/hunyuan/poolside_v1/kimik2 等接入点没有专门新增用例, 主要依赖 270 个全量单测与 GPU A/B 兜底。
 - 行为变更: minicpm5 在组合 schema 下未声明参数从 "透传" 变为 "丢弃", 依赖透传行为的客户端可能静默丢参; 这是 PR 主动声明并接受的变化, 但对使用者是 breaking。
 - 流式收尾边界: _is_first_param 与 _sent_empty_object 的组合边界由新增用例覆盖 (flat 流式大括号闭合), 但 "空参数 + 首参数为嵌套 object" 等更细边界仍建议在真实模型上观察。
 - 合并策略局限: anyOf/allOf 下 "首分支优先" 是约定而非语义保证, 实际调用匹配后声明 key 的分支时仍可能类型误判, qwen3_coder 受影响面最大。
 - 递归深度: get_schema_properties 沿组合分支递归, 极端深层嵌套 schema 存在栈深度风险; 不跟踪 \$ref 所以无循环引用问题。
 - 兼容性: 对 flat schema 完全兼容 (顶层 properties 直接短路返回), 非 dict 输入返回空字典, 行为安全。
 - 影响: 用户侧: 使用顶层 anyOf/oneOf/allOf tool schema (discriminated-union 参数) 的 API 调用方从 "参数变字符串 / 非法 JSON" 恢复为 "类型正确、JSON 合法"; Qwen3-Coder 与 GLM-4.5/4.7 系列是主要受益模型, scalar 参数 (count="7"、verbose="True") 也随 qwen3_coder 修复恢复正常。系统侧: function_call 解析体系开始收敛共享逻辑, 后续新增 detector 可复用 get_schema_properties, 降低同类 bug 概率。团队侧: 该 PR 建立了 "解析器行为变化必须显式声明 + 用例必须先在未修复代

码上失败 " 的测试纪律范例。

- 风险标记：跨 13 个解析器的统一改动，流式收尾逻辑替换启发式判断，minicpm5 未声明参数行为变更，组合分支采用首分支优先约定

关联脉络

- PR #35290 [XPU] Lazily import tvn_ffi-dependent all_reduce kernel in minimax_m2: 同属 MiniMax M2 服务链路修复：一个修模型加载崩溃，一个为 function_call/minimax_m2.py 接入顶层组合 schema 解析，前后端配套完善 MiniMax M2 支持。
- PR #36603 fix(kimi-k3): preserve dense ModelSlim MLA weights: 同属 Kimi 系列模型链路修复：本 PR 为 kimik2 detector 接入联合 schema 解析并明确不触碰 kimik3_structural_tag，与 Kimi 系列量化 / 解析修复互为补充。