PR #24081 完整报告

sgl-project/sglang

[Spec] Rename `accepted_drafts` -> `correct_drafts` for unambiguous naming

合并时间：2026-05-12 13:12

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24081

执行摘要

- 一句话：重命名投机解码外部 API 中 accepted_* 为 correct_drafts
- 推荐动作：该 PR 为纯命名清理，技术含量不高，但体现了项目对语义一致性的坚持。建议快速合并，并关注下游用户在下一个版本移除别名前的适配情况。对于希望了解项目命名规范的开发者，可以审阅 .claude/rules/speculative-naming.md 及本次变更作为案例。

功能与动机

Per speculative-naming rule, accept_* means "with bonus" and correct_* means "drafts only". Old external keys/params (spec_accepted_drafts, spec_proposed_drafts, spec_accept_histogram, accepted_tokens=) all carried drafts-only counts but used accept_* root — semantically misleading. Internal rename already landed; this finishes external API surface.

实现拆解

1. 更新核心 API 函数签名与 trace 输出：在 python/sglang/srt/observability/req_time_stats.py 中，将 set_spec_verify_end_time 的第一个参数从 accepted_tokens: int = 0 改为 num_correct_drafts: int = 0，并增加可选的 accepted_tokens: Optional[int] = None 作为向后兼容别名。trace_slice 的元数据键也由 "accepted_tokens" 改为 "num_correct_drafts"，同时写入旧键作为别名。
2. 更新所有调用者使用新参数名：修改 eagle_worker.py、frozen_kv_mtp_worker.py、ngram_worker.py 中的对应调用，将 accepted_tokens=accepted 替换为 num_correct_drafts=num_correct_drafts，并将局部变量从 accepted 重命名为 num_correct_drafts，逻辑不变。
3. 更新 eagle_info.py 中的局部变量命名：将 num_accepted 改为 num_accept_tokens，以精确反映其含义（包含初始 token 的总接受数），避免与 correct_drafts 混淆。
4. 更新 spec_utils.py 中的局部变量命名：将 traverse_tree 函数中的 accepted 布尔变量改为 is_accepted，与外部命名规范一致。
5. 更新 meta_info JSON 键：在 tokenizer_manager.py 中，将输出到用户侧 API 的 meta_info 字典键从 spec_accepted_drafts 等改为 spec_num_correct_drafts、spec_num_proposed_drafts、spec_correct_drafts_histogram，同时保留旧键作为向后兼容别名。

关键文件：

- python/sglang/srt/observability/req_time_stats.py（模块 可观测性；类别 source；类型 core-logic；符号 set_spec_verify_end_time）：核心变更点：修改了 set_spec_verify_end_time 函数签名和 trace 输出键名，影响所有 speculative decoding 可观测性数据输出。
- python/sglang/srt/speculative/eagle_worker.py（模块 投机解码；类别 source；类型 core-logic）：主要调用者之一：将 accepted_tokens=accepted 调用更新为 num_correct_drafts=num_correct_drafts，并重命名局部变量。
- python/sglang/srt/speculative/eagle_info.py（模块 投机解码；类别 source；类型 core-logic）：内部局部变量重命名：将 num_accepted 改为 num_accept_tokens，明确区分 correct_drafts 与总接受数。
- python/sglang/srt/speculative/spec_utils.py（模块 投机解码；类别 source；类型 core-logic）：局部变量 accepted 重命名为 is_accepted，与命名规范对齐。
- python/sglang/srt/speculative/frozen_kv_mtp_worker.py（模块 投机解码；类别 source；类型 core-logic）：与 eagle_worker.py 相同的调用端更新。
- python/sglang/srt/speculative/ngram_worker.py（模块 投机解码；类别 source；类型 core-logic）：与 eagle_worker.py 相同的调用端更新，额外包含对 verify_input.num_correct_drafts 的取值逻辑（未变）。
- python/sglang/srt/managers/tokenizer_manager.py（模块 请求管理；类别 source；类型 core-logic）：用户可见的 meta_info JSON 键重命名，涉及 API 兼容性。

关键符号：set_spec_verify_end_time

关键源码片段

python/sglang/srt/observability/req_time_stats.py

核心变更点：修改了 set_spec_verify_end_time 函数签名和 trace 输出键名，影响所有 speculative decoding 可观测性数据输出。

```
def set_spec_verify_end_time(
    self,
    ts=None,
    num_correct_drafts: int = 0,
    # FIXME: backward-compat alias, remove in next release.
    accepted_tokens: Optional[int] = None,
):
    # 如果调用者仍使用旧参数名 accepted_tokens，则自动映射到 num_correct_drafts
    if accepted_tokens is not None:
        num_correct_drafts = accepted_tokens
    ts = ts or time.perf_counter()

    if self.trace_ctx.tracing_enable:
        stage = RequestStage.SPEC_VERIFY
        self.trace_slice(
            stage,
```

```
            self.spec_verify_start_time,
            ts,
            {
                "num_correct_drafts": num_correct_drafts,
                # FIXME: backward-compat alias, remove in next release.
                "accepted_tokens": num_correct_drafts,
            },
        )
```

python/sglang/srt/speculative/eagle_worker.py

主要调用者之一：将 accepted_tokens=accepted 调用更新为 num_correct_drafts=num_correct_drafts，并重命名局部变量。

```
if get_global_tracing_enabled():
    for idx, req in enumerate(batch.reqs):
        # 从 verify_output 获取每请求 correct_drafts 数量
        num_correct_drafts = verify_output.num_correct_drafts_per_req_cpu[
            idx
        ]
        req.time_stats.set_spec_verify_end_time(
            num_correct_drafts=num_correct_drafts # 使用新参数名
        )
```

python/sglang/srt/speculative/eagle_info.py

内部局部变量重命名：将 num_accepted 改为 num_accept_tokens，明确区分 correct_drafts 与总接受数。

```
for i, (req, accept_index_row) in enumerate(zip(batch.reqs, accept_index_cpu)):
    num_accept_tokens = 0 # 原为 num_accepted，更名以反映包含 bonus 的计数
    for j, idx in enumerate(accept_index_row):
        if idx == -1:
            break
        num_accept_tokens += 1
        # ... 处理 token
    # 更新 KV cache 追踪
    req.kv_committed_len += num_accept_tokens
    req.kv_allocated_len = req.kv_committed_len
```

评论区精华

无 review 讨论，仅存在 CI 触发评论（/rerun-test 执行测试并报告结果）。PR 作者自行合并，表明变更无争议。

- 暂无高价值评论线程

风险与影响

- 风险：向后兼容风险：旧 API 键和参数名虽保留为别名，但已标记 FIXME 将在下一版本移除。依赖旧名称的外部用户若未及时更新可能中断。 语义一致性风险：

num_correct_drafts 仅统计 draft tokens 计数（不含 bonus 的初始 token），而 num_accept_tokens 则包含 bonus，若混淆可能导致下游理解错误。本次变更通过明确命名区分，风险较低。 回归风险：由于仅更改变量名和键名，逻辑未变，且所有修改均经 CI 测试运行通过（见 comment 中的测试结果），回归可能性低。

- 影响：影响范围：所有接入投机解码（Speculative Decoding）且依赖 meta_info 中 accepted_* 字段或通过 trace 分析性能的用户。 影响程度：低。旧名称仍在当前版本中工作，但有 FIXME 标记，用户应尽快迁移至新名称。内部逻辑完全等价，无需额外配置变更。 团队影响：统一了内部与外部命名，降低未来开发者的认知负荷。
- 风险标记：向后兼容别名将在下版本移除，避免混淆 correct_drafts 与 accept_tokens 含义

关联脉络

- PR #25014 [Spec] Internal rename per N2 v2 naming rule: 内部标识符重命名先驱，本 PR 完成外部 API 对应部分。
- PR #25029 [Spec] Mamba scatter cleanup; fix multi-layer positional bug; dflash naming: 涉及投机解码内部变量重命名，与本 PR 同属命名规范系列。
- PR #25030 [Spec] Multi-layer mamba scatter cleanup; fix positional call bug: 同样为投机解码内部重命名与修复，与本 PR 关联。