PR #24859 完整报告

sgl-project/sglang

[Spec V1] Split draft-extend phase from `EagleDraftInput` into new `EagleDraftExtendInput`

合并时间：2026-05-10 16:07

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24859

执行摘要

- 一句话：拆分推测解码 V1 Draft/Extend 数据结构
- 推荐动作：该 PR 值得精读，尤其是 eagle_info.py 与 frozen_kv_mtp_info.py 中的数据结构设计。对于从事推测解码开发的工程师，可以学习如何通过类型拆分消除阶段混淆。PR body 中的"Looks confusing but is correct"部分对设计权衡有清晰解释，可作为代码注释的典范。建议在合并前或合并后补充 V2 对齐的 issue 跟踪。

功能与动机

PR body 指出：重构前 EagleDraftInput.hidden_states 在同一实例上由 draft 阶段的 [bs, hidden] 切换为 draft-extend 阶段的 [total_accepted, hidden]；EagleVerifyOutput 的 next_draft_input 名不符实（实际包含 extend 数据），且携带 4 个仅用于 verify→extend 衔接的临时字段。这种阶段混淆增加了 attention backend 的特殊判断和 worker 中的维护负担。通过明确分离两种阶段的数据结构，使数据流更清晰且类型安全。

实现拆解

1. 在 eagle_info.py 中新增 EagleDraftExtendInput dataclass，集中 extend 阶段全部字段（per-accept-token hidden_states、accept counts、input_ids、seq_lens、req_pool_indices 等），并将 prepare_extend_after_decode、generate_attn_arg_prefill、filter_batch、merge_batch 等操作移入该类。同时精简 EagleDraftInput，只保留 draft 阶段必要字段（topk_p、topk_index、hidden_states[bs, h] 等），V2 专用字段以 Optional 保留并注释。
2. 修改 EagleVerifyOutput，将 next_draft_input 替换为 draft_extend_input，将 4 个过渡字段（unfinished_accept_tokens、seq_lens_for_draft_extend、seq_lens_for_draft_extend_cpu、req_pool_indices_for_draft_extend）直接归入 EagleDraftExtendInput。verify 方法构造并返回 EagleDraftExtendInput 实例。
3. 调整 worker 控制流：eagle_worker.py、multi_layer_eagle_worker.py、frozen_kv_mtp_worker.py 中的 forward_batch_generation 在 draft 后安装 verify_input 到 batch.spec_info，调用 self.verify(batch)（不再传 spec_info），然后从 verify_output.draft_extend_input 取出 extend 数据安装到 batch.spec_info，调用 forward_draft_extend_after_decode，该方法返回下一轮 EagleDraftInput，由调用者安装。当所有请求完成时安装一个空的 EagleDraftInput(capture_hidden_mode=LAST)，确保下一轮 merge_batch 能正确处理（EagleVerifyInput 无 merge_batch）。

4. 在 frozen_kv_mtp_info.py 中新增 FrozenKVMTPDraftExtendInput 作为 EagleDraftExtendInput 的标记子类，并重命名转换函数 _to_frozen_kv_mtp_draft_extend_input；frozen_kv_mtp_worker.py 的 forward_draft_extend_after_decode 改为从 batch.spec_info 读取 extend 输入，空闲时安装空输入。
5. 在 forward_batch_info.py 中，_pad_inputs_to_size 改成 getattr 守卫以兼容两个 draft 类的字段差异；spec_info.py 增加 SpecInputType.EAGLE_DRAFT_EXTEND 和 FROZEN_KV_MTP_DRAFT_EXTEND，并确保它们被 is_draft_input() 覆盖。相关 cuda graph runner 更新导入。

关键文件：

- python/sglang/srt/speculative/eagle_info.py（模块 推测解码；类别 source；类型 core-logic；符号 filter_batch, merge_batch, EagleDraftExtendInput, post_init）：核心数据结构文件：新增 EagleDraftExtendInput 并精简 EagleDraftInput，实现了阶段分离的基石；修改 EagleVerifyOutput 字段结构，调整 verify 方法返回类型；将 filter_batch/merge_batch 移入新类。
- python/sglang/srt/speculative/frozen_kv_mtp_info.py（模块 推测解码；类别 source；类型 core-logic；符号 FrozenKVMTPDraftExtendInput, post_init, _to_frozen_kv_mtp_draft_input, _to_frozen_kv_mtp_draft_extend_input）：对应 Frozen-KV MTP 的数据结构：新增 FrozenKVMTPDraftExtendInput 子类，重命名转换函数，同步修改 FrozenKVMTPVerifyInput.verify 以返回扩展后的输入。
- python/sglang/srt/speculative/eagle_worker.py（模块 推测解码；类别 source；类型 core-logic；符号 verify）：EAGLE Worker 控制流：修改 forward_batch_generation 以安装 verify_input、提取 draft_extend_input、调用新的 forward_draft_extend_after_decode（返回 EagleDraftInput），并在空分支安装空输入。
- python/sglang/srt/speculative/multi_layer_eagle_worker.py（模块 推测解码；类别 source；类型 core-logic；符号 verify）：Multi-layer EAGLE Worker：与 EAGLE Worker 做相同控制流调整，包括 spec_info 安装、verify 签名变更、extend 后输入装回。
- python/sglang/srt/speculative/frozen_kv_mtp_worker.py（模块 推测解码；类别 source；类型 core-logic；符号 forward_draft_extend_after_decode, verify）：Frozen-KV MTP Worker：调整 forward_draft_extend_after_decode 签名，改为从 batch.spec_info 读取 FrozenKVMTPDraftExtendInput；修改 _select_last_verified_seed 类型注解；导入新类型。
- python/sglang/srt/model_executor/forward_batch_info.py（模块 前向批处理；类别 source；类型 data-contract；符号 _pad_inputs_to_size）：padding 兼容性调整：_pad_inputs_to_size 从 if spec_info.topk_p is not None 改为 if getattr(spec_info, "topk_p", None) is not None，以支持两个 draft 类可能缺失相关字段。

关键符号：EagleDraftExtendInput.init, EagleDraftExtendInput.prepare_extend_after_decode, EagleDraftExtendInput.generate_attn_arg_prefill, EagleDraftExtendInput.filter_batch, EagleDraftExtendInput.merge_batch, EagleDraftInput.filter_batch, EagleDraftInput.merge_batch, EagleVerifyInput.verify, FrozenKVMTPDraftExtendInput.post_init, FrozenKVMTPVerifyInput.verify,

_to_frozen_kv_mtp_draft_extend_input, EagleWorker.forward_draft_extend_after_decode, MultiLayerEagleWorker.forward_draft_extend_after_decode, FrozenKVMTPWorker.forward_draft_extend_after_decode, ForwardBatch._pad_inputs_to_size

关键源码片段

python/sglang/srt/speculative/frozen_kv_mtp_info.py

对应 Frozen-KV MTP 的数据结构：新增 FrozenKVMTPDraftExtendInput 子类，重命名转换函数，同步修改 FrozenKVMTPVerifyInput.verify 以返回扩展后的输入。

```
# frozen_kv_mtp_info.py (head) — 标记子类与转换函数

@dataclass
class FrozenKVMTPDraftExtendInput(EagleDraftExtendInput):
    """Draft-extend input for Frozen-KV MTP. Tag-only subclass."""
    def __post_init__(self):
        SpecInput.__init__(self, SpecInputType.FROZEN_KV_MTP_DRAFT_EXTEND)

@dataclass
class FrozenKVMTPVerifyInput(EagleVerifyInput):
    def verify(self, *args, **kwargs) -> EagleVerifyOutput:
        output = super().verify(*args, **kwargs)
        # Convert the extend input from EAGLE type to Frozen-KV MTP type
        output.draft_extend_input = _to_frozen_kv_mtp_draft_extend_input(
            output.draft_extend_input
        )
        return output

def _to_frozen_kv_mtp_draft_extend_input(
    draft_extend_input: EagleDraftExtendInput,
) -> FrozenKVMTPDraftExtendInput:
    """Field-wise copy guard: skip if already the right type."""
    if isinstance(draft_extend_input, FrozenKVMTPDraftExtendInput):
        return draft_extend_input
    return FrozenKVMTPDraftExtendInput(
        **{
            field.name: getattr(draft_extend_input, field.name)
            for field in fields(EagleDraftExtendInput)
        }
    )
```

评论区精华

PR body 中作者主动解释了多处“看似混淆但正确”的细节，可视为设计讨论：

- filter_batch/merge_batch 虽然 diff 显示被重写，但字节级对比与原来完全一致，仅是位置移动。

- EagleDraftInput 仍保留 num_accepted_drafts/num_accepted_tokens 等字段，是因为 V2 Overlap Worker 仍复用同一实例跨阶段，这些字段会留在 V2 对齐时清理。
- bonus_tokens 同时存在于两个 dataclass，但职责不同：kernel 写入 extend-input，worker 拷贝到下一轮 draft-input 供 draft forward 使用。
- 在所有请求完成分支安装空的 EagleDraftInput 而非留用 EagleVerifyInput，是因为 merge_batch 只定义在 EagleDraftInput 上，空实例的 hidden_states is None 使下次迭代短路。
- 非 CUDA Graph 路径下的 softmax + fast_topk 内联替换从 capture_for_decode 中提取，语义等价且避免修改即将丢弃的 EagleDraftExtendInput。 这些解释降低了代码审查成本，也体现了作者对隐式契约的理解。
- 暂无高价值评论线程

风险与影响

- 风险：
  - 核心路径变更：V1 推测解码三路 worker 均修改了 verify 和 forward_draft_extend_after_decode 接口，非 CUDA Graph 路径下用内联 softmax+fast_topk 代替 capture_for_decode，虽声明等价但仍需回归验证。
  - V2 兼容性：V2 Overlap Worker 仍使用旧接口（EagleDraftInput 保留 V2 字段），本次 PR 未对齐 V2，后续清理时需注意双向兼容。
  - 缺少测试覆盖：本次 PR 未附带新的单元测试或集成测试，依赖现有 CI（CI 标签 run-ci 已触发），但风险仍存。
  - 数据结构契约：_pad_inputs_to_size 使用 getattr 守卫，若未来在两个 dataclass 上增加同名字段但语义不同，可能导致静默错误。
- 影响：
  - 用户影响：无直接用户可见变化，推测解码行为应与之前一致（PR 声明无行为改变）。
  - 系统影响：清理了大量过渡字段，简化了 verify→extend 路由，降低了 speculative 代码维护复杂度。V2 对齐作为 follow-up，需协调统一方向。
  - 团队影响：开发者阅读 spec 代码更易理解阶段边界；该 PR 可作为重构教科书式的示例，体现数据结构分离消除隐式状态的思路。
  - 风险标记：核心路径变更，缺少测试覆盖，V2 兼容性未对齐

关联脉络

- PR #24865 speculative: drop dead params/returns/no-ops: 同一作者在同一模块（推测解码）的先行清理 PR，删除了死代码、无操作参数，为本 PR 的分阶段分离预清理了冗余逻辑。多个文件（eagle_worker.py, multi_layer_eagle_worker.py 等）在本 PR 中继续修改。