# PR #33459 完整报告

sgl-project/sglang

[Spec] Support logprobs with DFlash

合并时间：2026-08-06 03:37

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33459

## 执行摘要

- 一句话：DFlash 推测解码支持 return_logprob 请求
- 推荐动作：值得快速阅读。改动小但覆盖了 admission 控制、worker 验证路径和测试复用，是 DFlash 功能扩展的一个清晰范例。重点关注 compute_spec_v2_logprobs 的调用时机和 output_indices 的构造方式，以及 scheduler 中对 DFlash/DSPark 的差异化处理。

## 功能与动机

PR body 明确说明：目标是 'Enable return_logprobs requests when running DFlash speculation via Spec v2'。此前 DFlash 在 admission 阶段和 worker 入口都会拒绝 return_logprob 请求，导致使用 DFlash 加速且需要 logprobs（如评估、RLHF 场景）的用户无法使用该功能。

## 实现拆解

1. 解除 worker 入口限制：在 python/sglang/srt/speculative/dflash_worker_v2.py 的 forward_batch_generation 中删除开头的 raise ValueError('DFLASH speculative decoding does not support return_logprob yet.')，并新增 import compute_spec_v2_logprobs。
2. 补算 logprobs：在验证阶段完成、SIMULATE_ACC_LEN 分支之后，当 batch.return_logprob 为真时，构造 output_indices（形状为 bs * block_size 的平铺索引），调用 compute_spec_v2_logprobs(batch, logits_output, out_tokens.reshape(-1), output_indices, block_size - 1)，复用 Spec v2 的 logprob 处理器。
3. 调整 admission 校验：在 python/sglang/srt/speculative/dflash_utils.py 的 validate_dflash_request 中删除 return_logprob 检查，仅保留 return_hidden_states 检查；在 python/sglang/srt/managers/scheduler.py 的 handle_generate_request 中，对 DFlash 家族改为区分处理：is_dspark() 且 return_logprob 时仍拒绝，DFlash 则放行。
4. 更新测试：在 test/registered/spec/dflash/test_dflash.py 中移除 @unittest.skip 的占位测试 test_grammar_logprob_count_matches_completion_tokens，改为引入 SpecLogprobKit mixin，复用现有 logprob 校验能力。

关键文件：

- python/sglang/srt/speculative/dflash_worker_v2.py（模块 推测解码；类别 source；类型 core-logic；符号 forward_batch_generation）：核心改动文件：删除 return_logprob 禁

令，在验证路径中接入 compute_spec_v2_logprobs，是实现 logprob 支持的关键。

- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 handle_generate_request）：调整了 DFlash 家族的 admission 控制：DFlash 放行 return_logprob，DSPark 仍拒绝，避免误放行不支持的功能。
- python/sglang/srt/speculative/dflash_utils.py（模块 推测解码；类别 source；类型 core-logic；符号 validate_dflash_request）：校验函数中移除 return_logprob 限制，是 admission 放行的直接来源。
- test/registered/spec/dflash/test_dflash.py（模块 测试；类别 test；类型 test-coverage；符号 test_grammar_logprob_count_matches_completion_tokens, TestDFlashServerBase）：测试配套：引入 SpecLogprobKit 替代手写跳过测试，验证 logprob 与 completion token 一致性。

关键符号：forward_batch_generation, handle_generate_request, validate_dflash_request

## 关键源码片段

### python/sglang/srt/speculative/dflash_worker_v2.py

核心改动文件：删除 return_logprob 禁令，在验证路径中接入 compute_spec_v2_logprobs，是实现 logprob 支持的关键。

```
# dflash_worker_v2.py: forward_batch_generation 中取消 return_logprob 禁令，
# 在验证结果就绪后（SIMULATE_ACC_LEN 分支之后）补算 Spec v2 logprobs。
if batch.return_logprob:
    # 每个序列固定 block_size 个输出 token，构造平铺索引便于按块取数
    output_indices = torch.arange(
        bs * block_size, dtype=torch.int64, device=device
    ).view(bs, block_size)
    # 直接复用 Spec v2 的 logprob 处理器，传入验证输出 tokens 与块内位置
    compute_spec_v2_logprobs(
        batch,
        logits_output,
        out_tokens.reshape(-1),
        output_indices,
        block_size - 1,
    )
```

### python/sglang/srt/managers/scheduler.py

调整了 DFlash 家族的 admission 控制：DFlash 放行 return_logprob，DSPark 仍拒绝，避免误放行不支持的功能。

```
# scheduler.py: handle_generate_request 中 DFlash 家族的 admission 控制，
# 仅保留 DSPark 的 return_logprob 限制，DFlash 放行交给 validate_dflash_request。
if self.spec_algorithm.is_dflash_family():
    error_msg = (
        "DSpark speculative decoding does not support return_logprob yet."
        if self.spec_algorithm.is_dspark() and req.return_logprob
        else validate_dflash_request(req, self.enable_overlap)
```

```
        )
        if error_msg is not None:
            req.set_finish_with_abort(error_msg)
            self.init_req_max_new_tokens(req)
            self._add_request_to_queue(req)
            return
```

### python/sglang/srt/speculative/dflash_utils.py

校验函数中移除 return_logprob 限制，是 admission 放行的直接来源。

```
# dflash_utils.py: validate_dflash_request 移除 return_logprob 检查，
# 因为 worker 已具备 compute_spec_v2_logprobs 能力，仅剩下 hidden_states 限制。
def validate_dflash_request(req: Req, enable_overlap: bool) -> Optional[str]:
    if enable_overlap and req.return_hidden_states:
        return "DFLASH speculative decoding does not support return_hidden_states yet."

    return None
```

## 评论区精华

review 中 kpham-sgl 建议不要手写 test_logprobs，而是直接复用现有的 SpecLogprobKit（并给出 spec_server_kits.py 中 L389 的参考实现），作者 jvmncs 回复 'resolved in latest, thank you!'。最终提交采用 SpecLogprobKit mixin 方案，删除了手写测试。

- 测试方式选择：手写 test_logprobs 还是复用 SpecLogprobKit (testing): 最终采用 SpecLogprobKit mixin，删除手写 test_logprobs 和 skip 的旧测试。

## 风险与影响

- 风险：
  1. logprob 计算正确性：compute_spec_v2_logprobs 复用自 Spec v2，但 DFlash 验证路径有自身特点（如 out_tokens 包含 bonus token、commit_lens 变化、mamba 状态更新），若索引或 token 顺序与 Spec v2 假设不一致，可能产生错误的 logprob 数值。
  2. 索引构造假设：output_indices 按 bs * block_size 平铺，隐式假设所有序列在验证时都占满 block_size 个位置；若存在提前终止或 padding 不一致，索引可能错位。
  3. 行为差异：scheduler 中 DFlash 放行而 DSPark 仍拒绝 return_logprob，用户可能对差异感到困惑。
  4. CI 稳定性：test_dflash.py 在 PR 中重跑了三次才通过（1-gpu-5090），说明该测试或环境存在一定抖动。 - 影响：对用户：使用 DFlash 推测解码且需要 logprobs 的用户（如评估、RLHF、日志分析）现在可以正常请求，不再需要关闭推测解码或改用其他算法。对系统：正常路径不触发 compute_spec_v2_logprobs，无性能影响；仅在 return_logprob 请求时增加一次 logprob 计算。对团队：填补了 DFlash 在 Spec v2 下 logprob 支持的功能空白，并为后续其他 spec worker 复用 compute_spec_v2_logprobs 提供了模式。 - 风险标记：核心 spec 路径变更，logprob 索引构造依赖 block 布局，CI 测试稳定性问题，DFlash/DSPark 行为差异

## 关联脉络

- PR #33348 [DCP] Match the replicated draft KV pool's page granularity to its allocator: 同属 DFlash / draft KV 基础设施演进，本 PR 的 compute_spec_v2_logprobs 依赖 draft KV 验证路径的稳定行为。
- PR #33580 [Unified Radix Cache] Complete the tree-core interface boundary: 统一 radix cache 接口重构为 DFlash 等 spec worker 提供稳定缓存接口，本 PR 在此类基础设施之上解除 logprob 限制。