PR #36284 完整报告

sgl-project/sglang

[CI] Add Kimi-K3 MMMU-Pro accuracy coverage

合并时间：2026-08-26 07:33

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/36284

执行摘要

- 一句话：新增 Kimi-K3 B300 MMMU-Pro 精度 CI 覆盖
- 推荐动作：值得精读，尤其关注 run_eval.py 中 preset 与 legacy 参数分支的设计，以及 MMMUProMixin 的阈值校准方法论。对后续为其他推理模型添加 sgl-eval 评测有直接参考价值，也展示了如何通过实测数据驱动 CI 阈值设定。

功能与动机

Kimi-K3 是结合 DSPARK 线性 ReplaySSM 推测解码的推理模型，原 low-latency CI 使用 GSM8K 文本精度阈值把关，无法覆盖多模态长推理场景。PR body 明确指出：'Strengthen Kimi-K3 B300 low-latency coverage with the multimodal MMMU-Pro benchmark. The test should use the model-maintained sgl-eval preset so its generation settings stay aligned with Kimi-K3.' 首次 B300 运行也暴露了旧 GSM8K 阈值（accept length 4.5）对 MMMU-Pro 不再适用（实测 2.62），需要按新任务重新校准。

实现拆解

本 PR 分 5 步完成：

1. 扩展精度测试套件（python/sglang/test/kits/eval_accuracy_kit.py）：新增 MMMUProMixin，提供 test_mmmu_pro 方法；_run_accuracy_eval 将 num_threads 改为 Optional，并允许通过 eval_overrides 传递 model=None 与 load_preset_from_model_id，使模型和采样设置完全交给 sgl-eval preset。
2. 扩展 run_eval.py 的 sgl-eval 桥：新增 --load-preset-from-model-id 参数；有 preset 时省略 --model、--num-threads、--temperature、--top-p、--max-tokens、--thinking 等参数，无 preset 时保留 legacy 默认（如 top_p=1.0、max_tokens=2048、temperature=0.0），避免影响既有 GSM8K 等约 47 个消费者。同时将用户输入的 mmmu-pro 归一化为 sgl-eval 注册名 mmmu_pro。
3. 更新 sgl-eval 版本 pin：scripts/ci/utils/sgl_eval_ref.sh 更新到包含 Kimi-K3 preset 的修订版本。
4. 拆分并改造 B300 测试：从 test/registered/models_e2e/test_kimi_k3_b300.py 中移出 TestKimiK3B300LowLatency，新建独立文件 test_kimi_k3_b300_low_latency.py，使用 MMMUProMixin（score_threshold=0.75、200 例、accept_length=2.4、preset 模型 moonshotai/Kimi-K3），并保留单请求 accept_length=4.0 和 300 token/s 速度断言；原

文件保留 Balanced 和 MegaMoE 的 GSM8K 测试。注册 est_time=1800。

5. 补充单元测试：在 test/registered/unit/test_eval_accuracy_kit_sgl_eval.py 和 test/registered/unit/bench/test_simple_eval_gsm8k.py中新增命令构造、preset优先级、legacy 默认值、任务名归一化和 mixin 阈值行为测试。

测试配套：新增 / 修改 6 个测试文件，无 serving 代码变更，未添加 workflow 或强制 bin 覆盖。

关键文件：

- python/sglang/test/kits/eval_accuracy_kit.py（模块 精度测试套件；类别 test；类型 test-infra；符号 MMMUProMixin, test_mmmu_pro, _run_accuracy_eval）：新增 MMMUProMixin 和 test_mmmu_pro 方法，是 MMMU-Pro 评测的入口；同时调整 _run_accuracy_eval 的参数传递，使其支持 preset 模式。
- python/sglang/test/run_eval.py（模块 评测脚本；类别 test；类型 test-infra；符号 _run_sgl_eval, run_eval, run_eval_once）：扩展 sgl-eval 桥，新增 --load-preset-from-model-id 支持和参数透传逻辑，是 MMMUProMixin 能工作的关键。
- test/registered/models_e2e/test_kimi_k3_b300_low_latency.py（模块 B300 低延迟测试；类别 test；类型 test-coverage；符号 TestKimiK3B300LowLatency, setUpClass, tearDownClass, _stop_server）：新增的独立低延迟测试文件，使用 MMMUProMixin 替换 GSM8K，并设置校准后的阈值，是本次 CI 覆盖的核心载体。
- test/registered/models_e2e/test_kimi_k3_b300.py（模块 B300 测试；类别 test；类型 test-refactor；符号 TestKimiK3B300Balanced）：移除低延迟测试类及相关导入，保留 Balanced 和 MegaMoE 的 GSM8K 覆盖，避免重复注册。
- test/registered/unit/test_eval_accuracy_kit_sgl_eval.py（模块 精度套件单测；类别 test；类型 test-coverage；符号 _run_mmmu_pro, test_mmmu_pro_uses_kimi_preset_and_300_examples, test_mmmu_pro_score_threshold_gates_result）：为 MMMUProMixin 增加 Hermetic 单元测试，验证 preset 参数拼接和分数阈值 gating。
- test/registered/unit/bench/test_simple_eval_gsm8k.py（模块 评测脚本单测；类别 test；类型 test-coverage；符号 test_model_preset_owns_model_and_sampling_defaults, test_non_preset_cli_keeps_legacy_top_p_default, test_run_eval_dispatches_hyphenated_mmmu_pro_name）：为 run_eval.py 的 preset 参数透传、legacy 默认值和任务名归一化增加测试，防止回归。
- scripts/ci/utils/sgl_eval_ref.sh（模块 CI 脚本；类别 infra；类型 configuration）：更新 sgl-eval 版本 pin 到包含 Kimi-K3 preset 的修订，是评测可用的前提。

关键符号：MMMUProMixin.test_mmmu_pro, _run_accuracy_eval, _run_sgl_eval, run_eval, TestKimiK3B300LowLatency.setUpClass, TestKimiK3B300LowLatency.tearDownClass

关键源码片段

python/sglang/test/kits/eval_accuracy_kit.py

新增 MMMUProMixin 和 test_mmmu_pro 方法，是 MMMU-Pro 评测的入口；同时调整 _run_accuracy_eval 的参数传递，使其支持 preset 模式。

```
class MMMUProMixin:
    """基于 sgl-eval 的标准 10 选项 MMMU-Pro 评测 mixin。

    模型 preset 提供 endpoint 模型和全部生成设置。将模型和采样参数
    交给 sgl-eval 管理，对推理模型很重要——其推荐 token 预算和采样
    设置与 run_eval 默认值不同。
    """

    mmmu_pro_score_threshold: float = _THRESHOLD_NOT_SET
    mmmu_pro_accept_length_thres: Optional[float] = None
    mmmu_pro_num_examples: Optional[int] = 300
    mmmu_pro_num_threads: Optional[int] = None
    mmmu_pro_load_preset_from_model_id: Optional[str] = None

    def test_mmmu_pro(self):
        # 必须显式指定 preset，否则无法获得与模型对齐的生成配置
        assert self.mmmu_pro_load_preset_from_model_id, (
            f"{type(self).__name__} 必须设置 mmmu_pro_load_preset_from_model_id"
        )
        _run_accuracy_eval(
            self,
            eval_name="mmmu_pro",
            score_threshold=self.mmmu_pro_score_threshold,
            num_examples=self.mmmu_pro_num_examples,
            num_threads=self.mmmu_pro_num_threads,
            accept_length_thres=self.mmmu_pro_accept_length_thres,
            model=None, # 由 preset 决定模型
            load_preset_from_model_id=self.mmmu_pro_load_preset_from_model_id,
        )
```

python/sglang/test/run_eval.py

扩展 sgl-eval 桥，新增 --load-preset-from-model-id 支持和参数透传逻辑，是 MMMUProMixin 能工作的关键。

```
def _run_sgl_eval(eval_name, args) -> dict:
    # ... 前面处理 out_dir 等
    model_preset_id = getattr(args, "load_preset_from_model_id", None)
    cmd = [
        "sgl-eval", "run", eval_name,
        "--base-url", base_url,
        "--out-dir", str(out_parent),
    ]

    if model_preset_id:
        cmd += ["--load-preset-from-model-id", model_preset_id]

    # 有 preset 时不传 model 与采样参数，交给 preset 统一决定；
    # 无 preset 时保留 legacy 默认值，避免影响既有 GSM8K 调用方。
    if getattr(args, "model", None):
```

```
        cmd += ["--model", args.model]
    if getattr(args, "num_examples", None) is not None:
        cmd += ["--num-examples", str(args.num_examples)]
    if getattr(args, "num_threads", None) is not None:
        cmd += ["--num-threads", str(args.num_threads)]
    if getattr(args, "temperature", None) is not None:
        cmd += ["--temperature", str(args.temperature)]
    elif not model_preset_id:
        cmd += ["--temperature", "0.0"]
    if getattr(args, "top_p", None) is not None:
        cmd += ["--top-p", str(args.top_p)]
    elif not model_preset_id and getattr(args, "_sgl_eval_from_cli", False):
        cmd += ["--top-p", "1.0"]
    # ... 其余参数（seed、max-tokens、thinking 等）
```

test/registered/models_e2e/test_kimi_k3_b300_low_latency.py

新增的独立低延迟测试文件，使用 MMMUProMixin 替换 GSM8K，并设置校准后的阈值，是本次 CI 覆盖的核心载体。

```
class TestKimiK3B300LowLatency(MMMUProMixin, SpecDecodingMixin, CustomTestCase):
    """TP8 Low Latency recipe with DSPARK linear ReplaySSM speculation."""

    mmmu_pro_score_threshold = 0.75
    mmmu_pro_num_examples = 200
    mmmu_pro_load_preset_from_model_id = MODEL_PATH
    # MMMU-Pro 的多模态长推理平均接受长度低于 GSM8K（首轮实测 2.62），
    # 所以这里用任务专属回归门限；单请求速度门限保持更严格。
    mmmu_pro_accept_length_thres = 2.4
    # 单请求贪婪解码的结束位置会影响这两个指标，且速度是端到端的，
    # 启动和 TTFT 摊到输出里，所以只作为粗粒度保护。
    accept_length_thres = 4.0
    bs_1_speed_thres = 300

    @classmethod
    def setUpClass(cls):
        cls.model = MODEL_PATH
        cls.base_url = DEFAULT_URL_FOR_TEST
        cls.process = popen_launch_server(
            cls.model,
            cls.base_url,
            timeout=SERVER_LAUNCH_TIMEOUT,
            other_args=[
                "--trust-remote-code",
                "--tp-size", "8",
                "--mem-fraction-static", "0.85",
                "--model-loader-extra-config", MODEL_LOADER_EXTRA_CONFIG,
                "--reasoning-parser", "kimi_k3",
                "--tool-call-parser", "kimi_k3",
                "--mamba-full-memory-ratio", "0.86",
```

```
                "--speculative-algorithm", "DSPARK",
                "--speculative-draft-model-path", DSPARK_DRAFT_MODEL,
                "--speculative-dspark-block-size", "7",
                "--enable-linear-replayssm-spec",
            ],
        )

    @classmethod
    def tearDownClass(cls):
        _stop_server(getattr(cls, "process", None))
```

评论区精华

本 PR 没有实质性的 review 评论。评论区仅有作者触发的 rerun-test 交互：第一次低延迟测试在 8-gpu-b300 上失败（run #32818230199），第二次通过（run #32896985069），拆分后新文件的 rerun 再次失败（run #32910692557，PR 中未说明原因）。作者在 PR body 中记录了关键校准过程：首轮 300 例 MMMU-Pro 评测约 29 分钟，因旧 GSM8K 阈值（accept length 4.5）误报失败，CustomTestCase 重试导致 workflow 60 分钟超时；第二轮 200 例通过，得分 0.83，平均接受长度 2.6096，据此将接受长度阈值从 2.6204 校准为 2.4。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 阈值过拟合风险：mmmu_pro_accept_length_thres = 2.4 仅基于一次 B300 运行校准，若 sgl-eval 或模型更新导致平均接受长度波动，可能产生误报。
  2. run_eval.py 回归风险：参数透传逻辑改动影响所有走该入口的评测（含既有 GSM8K 消费者），虽然单元测试覆盖 legacy 默认值，但仍有回归可能。
  3. CI 时长增加：新测试 est_time=1800（30 分钟）且需 8 卡 B300，首次运行就曾因重试触发 60 分钟超时，可能增加 CI 队列压力。
  4. sgl-eval 版本更新：更新 pin 可能引入新依赖或行为变化，影响所有使用 sgl-eval 的评测。 - 影响：用户 / 系统：对生产 serving 无影响，纯测试与 CI 基础设施变更。团队：Kimi-K3 B300 low-latency 配方获得更贴近多模态推理场景的精度把关，替代原来 GSM8K 文本基准，能更好捕捉 DSPARK 推测解码在长多模态推理中的质量问题。测试资产：MMMUProMixin 成为可复用组件，其他模型可低成本接入 sgl-eval preset 评测；run_eval.py 的 preset 机制为推理模型评测提供更准确的生成配置。CI：新增独立测试文件和注册项，需要占用 B300 资源，可能延长 base-c 阶段时间。
- 风险标记：阈值基于单次运行校准，run_eval 参数透传回归风险，CI 时长增加（est_time=1800），sgl-eval 版本更新兼容性

关联脉络

- 暂无明显关联 PR