PR #34667 完整报告

sgl-project/sglang

Drop the mmlu case from the unified radix cache kit

合并时间：2026-08-13 15:16

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34667

执行摘要

PR #34667 从共享测试基座 UnifiedRadixTreeTestMixin 中删除 MMLU 用例 test_mmlu 与阈值配置 mmlu_threshold，并清理 5 个消费者文件中的显式覆盖与跳过逻辑。根因是 64 题采样下 MMLU 分数只能是 1/64 的倍数，默认阈值 0.8 不可达，测试因算术而非回归持续变红。变更仅涉及测试基础设施，不触碰产品代码，使 unified radix cache 测试回归到 KL 精准守卫 + gsm8k 粗粒度兜底的分层设计。

功能与动机

UnifiedRadixTreeTestMixin 捆绑的 MMLU 用例在仓库内已无人信任：7 个消费者中，两个在 CI 里跳过（SWA 模型标注 mmlu eval not stable enough），四个把阈值降到 0.4 或 0.7，剩下两个停在默认 0.8 并刚出现失败：AssertionError: 0.796875 not greater than or equal to 0.8。

PR body 给出了清晰的算术论证：num_examples=64 时可达分数为 51/64 = 0.7969、52/64 = 0.8125，不存在等于 0.8 的分数，因此 0.8 阈值实际等效于 0.8125，而 64 题评估的 run-to-run 波动有数个百分点。同一文件的三个 KL 用例全部通过，gsm8k 得到 0.965，说明这不是回归而是测试设计缺陷。MMLU 与 gsm8k 同为粗粒度兜底网，在该套件里没有独立失败模式，所以作者选择删除而非重调阈值。

实现拆解

1. 共享基座变更：python/sglang/test/kits/unified_radix_cache_kit.py 删除 UnifiedRadixTreeTestMixin 的 test_mmlu 方法（约 20 行）与 mmlu_threshold 类属性，类 docstring 从 gsm8k, mmlu and multi-turn KL tests 改为 gsm8k and multi-turn KL tests。由于这是全部消费者的共享基座，未显式覆盖该用例的文件会自动停止运行，无需逐一编辑。
2. 消费者清理：test_unified_radix_cache_kl_swa.py 删除 skipIf 包装的 test_mmlu 覆盖和 mmlu_threshold = 0.7，并移除不再使用的 is_in_ci 导入；test_unified_radix_cache_hicache_pp_kl.py、test_unified_radix_cache_kl_cp.py 各删除一行 mmlu_threshold = 0.7；test_unified_radix_cache_kl_dcp.py 删除一行 mmlu_threshold = 0.4。全部为纯删逻辑。
3. 文档同步：test_unified_radix_cache_kl_hybrid_bitexact.py 虽不继承该 mixin，其 docstring 提到 mixin bundles gsm8k and mmlu，本次同步改为只捆绑 gsm8k，避免文档与实现漂移。

4. 范围边界：test_unified_radix_cache_kl_dsv4.py 的 mmlu_num_threads 保留，因为该类基于 AccuracyTwoPassMixin（另一套测试 kit），不在本次范围内。
5. CI 验证：无新增测试；作者通过 /rerun-test 一次重跑 7 个消费者文件，覆盖共享 mixin 变更的全部影响面。

以下是清理后 UnifiedRadixTreeTestMixin 的关键部分：

```
class UnifiedRadixTreeTestMixin:
    """Mixin: gsm8k 与多轮 KL 用例（原 mmlu 用例已移除，见 PR #34667）。

    移除 `test_mmlu` 与 `mmlu_threshold` 的原因：
    - num_examples=64 时分数只能是 1/64 的倍数，没有任何可达分数等于 0.8，
      该阈值实际等效于 0.8125，对 64 题评估的随机波动过于敏感；
    - 三个 KL 用例是精准守卫，gsm8k 是粗粒度兜底网，
      MMLU 与 gsm8k 测的是同一件事，在该套件里没有独立失败模式。
    """

    kl_threshold: float = 0.003
    gsm8k_threshold: float = 0.93
    num_gsm8k_questions: int = 200

    def test_gsm8k(self):
        """Few-shot GSM8K 数学推理准确率，作为粗粒度兜底。"""
        from sglang.test.few_shot_gsm8k import run_eval as run_few_shot_gsm8k

        url = urlparse(self.base_url)
        args = SimpleNamespace(
            num_shots=10,
            data_path=None,
            num_questions=self.num_gsm8k_questions,
            max_new_tokens=16000,
            parallel=128,
            host=f"http://{url.hostname}",
            port=int(url.port),
        )
        metrics = run_few_shot_gsm8k(args)
        print(
            f"[{self.__class__.__name__}] GSM8K accuracy: {metrics['accuracy']:.3f} "
            f"(threshold: {self.gsm8k_threshold})"
        )
        self.assertGreaterEqual(metrics["accuracy"], self.gsm8k_threshold)
```

SWA 消费者在清理后的形态：

```
class TestUnifiedSWARadixCache(UnifiedRadixTreeTestMixin, CustomTestCase):
    """SWA 混合模型 + UnifiedRadixCache。

    变更前这里还有 `mmlu_threshold = 0.7` 以及
    `@unittest.skipIf(is_in_ci(), "SWA model mmlu eval not stable enough")`
    包装的 `test_mmlu` 覆盖；mixin 删除该用例后，这些显式覆盖
```

```
    一并清理，避免留下“半转换”的消费者。
    """

    kl_threshold = 0.03
    gsm8k_threshold = 0.7

    @classmethod
    def setUpClass(cls):
        cls.model = SWA_MODEL
        cls.base_url = DEFAULT_URL_FOR_TEST
        cls.process = popen_launch_server(
            cls.model,
            cls.base_url,
            timeout=DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
            other_args=[
                "--tp-size",
                "2",
                "--mem-fraction-static",
                "0.7",
                "--disable-piecewise-cuda-graph",
            ],
            env={"SGLANG_ENABLE_UNIFIED_RADIX_TREE": "1"},
        )
        cls.input_ids = get_input_ids(cls.model, num_samples=18)

    @classmethod
    def tearDownClass(cls):
        kill_process_tree(cls.process.pid)
```

评论区精华

由于 review 评论为空，讨论集中在 PR body 与 issue 评论：

> 阈值不可达是算术而非回归：51/64 = 0.7969、52/64 = 0.8125，没有可达分数等于 0.8，实际门槛是 0.8125。
>
> 三个 KL 用例是精准仪器，gsm8k 是粗粒度兜底网，MMLU 是第二张测量同一件事的粗网——在该套件里没有独立失败模式。

针对 /rerun-test，作者强调：

> The change is in the shared mixin, so this covers every consumer of it, not only the files edited here.

github-actions 的回复显示 4-gpu-h100（4 个测试运行）仍标记失败、4-gpu-b200（1 个测试运行）通过，失败用例细节未再澄清，PR 随后合并。

风险与影响

- 测试覆盖收紧：7 个 unified radix cache 配置不再有 MMLU 兜底，若未来 MMLU 恰好暴露某类缓存回归将失去保护；缓解点是三个 KL 用例仍为精准守卫。

- 共享基座兼容性：sglang.test.kits 是测试基础设施，本次已清空仓库内所有显式引用；仓库外或未来消费者若仍引用 mmlu_threshold / test_mmlu 会触发 AttributeError。
- CI 状态存疑：重跑中 4-gpu-h100 4 个测试运行仍标红且未说明归属，合并前无全绿记录，可能存在与本次变更无关的既有 flake，后续需留意 KL/PP/CP/Mamba 测试稳定性。
- 对用户无影响：变更不涉及产品源码、配置、部署路径，无性能与安全影响。

关联脉络

本 PR 是 unified radix cache 测试体系持续收敛的一环。此前 #34607 新增 bit-exact KL 守卫测试、#34656 记录双架构位精确数据，本次删除 mixin 中冗余的 MMLU 用例并同步相关 docstring，与这两次变更在同一文件、同一测试线路上演进。同时 #34477 正在把 MMLU/MMMU-Pro 评测迁移到 sgl-eval 基础设施，本 PR 移除基于 run_eval 的 MMLU 捆绑用例，两条线共同收敛 MMLU 在 CI 测试中的定位。整体趋势是让 unified radix cache 的回归测试更聚焦、更稳定，减少粗粒度评测带来的随机波动噪音。