

PR #33255 完整报告

sgl-project/sglang

[misc] Improve benchmark determinism and dataset API coverage

合并时间: 2026-08-02 16:39

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/33255>

执行摘要

- 一句话: 基准数据集采样支持 `--seed` 确定性, 新增图像前缀缓存复用测试
- 推荐动作: 值得精读: 可学习如何用“排序 + RNG 重置”让基于全局随机状态的采样在跨 tokenizer 版本时保持确定性, 以及如何通过端到端贪心请求校验图片前缀缓存正确性。断言修复的 `caution` 说明体现了对隐性坏味道的复盘, 建议关注后续 VLM 套件的失败信号。

功能与动机

PR body 明确列出目标: Make dataset sampling deterministic under `--seed` (sorted vocab; reset RNG after processor init), add gsm8k `--chat-template-kwarg`s, skip worker-state guards gracefully when `one_batch_server` points at a PD router, add an image prefix-cache reuse test, and extend dataset API tests (incl. the media-kwarg processor contract used by the upcoming Kimi K3 processor, #32541)。另外指出 `verify_single_image_response` 中的断言此前恒真 (or "person" 被当 `truthy` 字面量), 重新武装后可能暴露潜在失败。

实现拆解

1. 规范化 vocab 采样池: `python/sglang/benchmark/datasets/common.py` 的 `get_available_tokens` 由列表推导改为 `sorted()`, 消除 tokenizer 版本间 vocab dict 迭代序差异对 `random.choices` 采样结果的影响; 配合 `lru_cache` 保证缓存结果顺序稳定。
2. RNG 重置与 Kimi K3 契约: `python/sglang/benchmark/datasets/image.py` 的 `ImageDataset` 新增 `seed` 字段, `from_args` 从 `args.seed` 取值; `load()` 在 `get_processor()` 之后对 `random` 与 `np.random` 做 `seed(self.seed)` 重设, 抵消 processor 初始化对全局 RNG 的消耗; `create_mm_data_row` 的 `medias` 分支从仅 `KimiK25Processor` 扩展为包含 `KimiK3Processor`。
3. gsm8k chat-template kwarg: `benchmark/gsm8k/bench_sglang.py` 新增 `--chat-template-kwarg`s (`json.loads` 解析, 默认 `{"enable_thinking": true}`), 将 `apply_chat_template` 调用从硬编码 `enable_thinking=True` 改为 `**args.chat_template_kwarg`s, 兼容不同模型的 `thinking` 开关取名差异 (`enable_thinking` vs `thinking`)。
4. PD router 守卫: `python/sglang/benchmark/one_batch_server.py` 在 `server_info` 含 `router_manager` 且 `internal_states` 为空时打印警告并把 `max-running-requests` 与 `token-capacity` 守卫置为 `inf`, 避免对 PD router 场景误断言; 否则维持原有断言逻辑。

5. 测试配套: python/sglang/test/vlm_utils.py 新增 test_image_prefix_cache_reuse (同图重复输出一致、异图输出不同, 拦截图片 KV 错误复用) 并修复 verify_single_image_response 恒真断言; test/registered/bench_fn/test_benchmark_datasets_api.py 新增 KimiK3Processor 模拟类、media contract 测试、seed 与 processor 初始化独立性测试、vocab 顺序无关性测试。

关键文件:

- test/registered/bench_fn/test_benchmark_datasets_api.py (模块 数据集 API; 类别 test; 类型 test-coverage; 符号 KimiK3Processor, test_image_sampler_uses_kimi_k3_media_contract, test_image_dataset_seed_is_independent_of_processor_initialization, test_gen_mm_prompt_is_independent_of_vocab_order) : 新增大规模数据集 API 测试: Kimi K3 media contract、seed 与 processor 初始化独立性、vocab 顺序无关性
- python/sglang/test/vlm_utils.py (模块 VLM 测试; 类别 test; 类型 test-coverage; 符号 test_image_prefix_cache_reuse, describe, verify_single_image_response) : 新增图片前缀缓存复用端到端测试, 并修复 verify_single_image_response 恒真断言
- python/sglang/benchmark/one_batch_server.py (模块 基准服务器; 类别 source; 类型 core-logic) : 在 PD router 场景下优雅禁用 worker 内部状态守卫, 避免 benchmark 崩溃
- python/sglang/benchmark/datasets/image.py (模块 基准数据集; 类别 source; 类型 dependency-wiring; 符号 ImageDataset.load, ImageDataset.from_args, create_mm_data_row) : 核心确定性修复: ImageDataset 增加 seed 并在 load() 重置 RNG; 同时支持 KimiK3Processor 的 medias 契约
- benchmark/gsm8k/bench_sglang.py (模块 GSM8K 基准; 类别 source; 类型 core-logic; 符号 main) : 新增 --chat-template-kwarg 参数, 支持按模型透传 thinking 开关
- python/sglang/benchmark/datasets/common.py (模块 基准数据集; 类别 source; 类型 core-logic; 符号 get_available_tokens) : 对 vocab 采样池排序, 消除跨 tokenizer 版本的迭代序差异带来的非确定性

关键符号: ImageDataset.load, ImageDataset.from_args, get_available_tokens, create_mm_data_row, verify_single_image_response, test_image_prefix_cache_reuse, KimiK3Processor.call, test_image_sampler_uses_kimi_k3_media_contract, test_image_dataset_seed_is_independent_of_processor_initialization, test_gen_mm_prompt_is_independent_of_vocab_order

关键源码片段

test/registered/bench_fn/test_benchmark_datasets_api.py

新增大规模数据集 API 测试: Kimi K3 media contract、seed 与 processor 初始化独立性、vocab 顺序无关性

```
def test_image_dataset_seed_is_independent_of_processor_initialization(self):
    # 构造带 seed 的 ImageDataset, 并用“带 RNG 副作用”的 processor 初始化
    # 模拟真实环境: get_processor 初始化本身会偷偷消耗全局 random / np.random 状态,
    # 若不重置, 同一 seed 会因 processor 初始化次数不同而产生不同采样结果。
    dataset = ImageDataset.from_args(
        make_args(
```

```

        num_prompts=3,
        image_resolution="random:8x16-16x32",
        seed=20260717,
    )
)
processor_init_count = 0

def get_processor_with_rng_side_effects(_model_id):
    nonlocal processor_init_count
    processor_init_count += 1
    random.random()
    np.random.random(processor_init_count)
    return self.processor

with patch(
    "sglang.benchmark.datasets.image.get_processor",
    side_effect=get_processor_with_rng_side_effects,
):
    first = dataset.load(model_id="test-model")
    random.seed(999)
    np.random.seed(999)
    second = dataset.load(model_id="test-model")

# 两次 load 之间刻意重置 RNG，验证数据行完全一致，
# 证明确定性来自 load() 内部的 seed 重置而不是外部巧合。
self.assertEqual(
    [
        (row.prompt, row.prompt_len, row.output_len, row.image_data)
        for row in first
    ],
    [
        (row.prompt, row.prompt_len, row.output_len, row.image_data)
        for row in second
    ],
)

```

python/sglang/test/vlm_utils.py

新增图片前缀缓存复用端到端测试，并修复 verify_single_image_response 恒真断言

```

def test_image_prefix_cache_reuse(self):
    """图片前缀（radix）缓存跨请求正确性测试。

```

重复相同图片必须复用多模态前缀且输出不变；不同图片绝不能用前一张图片的 KV。这能拦截 image-token pad_value 或 feature-hash 回归导致的“静默复用错误图片 KV”级正确性 bug，而这种 bug 在单请求测试中不可见。纯贪心请求级校验：无需额外 server 参数，radix cache 默认开启。

```

client = openai.Client(api_key=self.api_key, base_url=self.base_url)

```

```

def describe(url: str) -> str:

```

```

response = client.chat.completions.create(
    model="default",
    messages=[
        {
            "role": "user",
            "content": [
                {"type": "image_url", "image_url": {"url": url}},
                {"type": "text", "text": "Describe this image in one sentence."},
            ],
        },
    ],
    temperature=0,
    max_tokens=32,
    **(self.get_vision_request_kwargs()),
)
assert response.usage.prompt_tokens > 0
content = response.choices[0].message.content
assert isinstance(content, str) and content
return content

```

```

# miss -> compute, 然后 hit -> 复用同一张图片的 prefix
first = describe(IMAGE_MAN_IRONING_URL)
repeat = describe(IMAGE_MAN_IRONING_URL)
# 不同图片必须独立计算, 不能复用 `first` 的 KV
other = describe(IMAGE_SGL_LOGO_URL)
# 另一张图片占据缓存后再回来, 同一张图片的 KV 也不能被污染
first_again = describe(IMAGE_MAN_IRONING_URL)

```

```

self.assertEqual(first, repeat, "重复相同图片输出变化, prefix 复用破坏贪心确定性。")
self.assertEqual(first, first_again, "相同图片隔了其他请求后输出变化, KV 可能跨请求串扰。")
self.assertNotEqual(first, other, "不同图片输出却相同, 可能错误复用了 prefix 缓存中的 KV。")

```

python/sglang/benchmark/datasets/image.py

核心确定性修复: ImageDataset 增加 seed 并在 load() 重置 RNG; 同时支持 KimiK3Processor 的 medias 契约

```

def load(self, tokenizer=None, model_id=None) -> List[DatasetRow]:
    processor = get_processor(model_id)
    # processor 初始化可能消耗全局 RNG 状态, 例如加载词表或构建 chat template;
    # 这里在采样前重置随机种子, 确保 --seed 能固定生成的 prompts、图片尺寸与内容。
    random.seed(self.seed)
    np.random.seed(self.seed)
    return sample_image_requests(
        num_requests=self.num_requests,
        image_count=self.image_count,
        input_len=self.input_len,
        output_len=self.output_len,
        range_ratio=self.range_ratio,
        processor=processor,
    )

```

```
image_content=self.image_content,
image_format=self.image_format,
image_resolution=self.image_resolution,
backend=self.backend,
random_image_count=self.random_image_count,
)
```

评论区精华

本 PR 没有 review 评论。PR body 中作者主动提示：verify_single_image_response 的关键字断言此前恒真（“person”被当作 truthy 字面量），重新武装后可能让现有 VLM 套件浮现潜在失败。合入前作者通过 `/rerun-test test_benchmark_datasets_api.py` `test_vision_openai_server_a.py` 确认了 ubuntu-latest 与 1-gpu-h100 两个测试通过。

- verify_single_image_response 断言重新启用 (correctness): PR 已通过 any(keyword in text for keyword in ("man", "person", "driver")) 修复，并以 /rerun-test 验证 test_vision_openai_server_a.py 在 H100 上通过。

风险与影响

- 风险：

1. vlm_utils.py 断言修复：verify_single_image_response 现在真正要求输出包含关键字，MiniCPM 等模型可能因输出措辞不含关键字而在现有套件失败；注释已为 MiniCPMO 的 iron 场景留了后备词，但覆盖面仍有限。
2. 排序 vocab 行为变化：get_available_tokens 排序会改变任意随机 prompt 的具体 token 序列，若调用方隐式依赖旧顺序，复现基线可能变化（这正是修复目标，但需留意对现有 benchmark 结果对比的影响）。
3. 全局 RNG 重置副作用：ImageDataset.load 重置 random / np.random 全局状态，会使基准进程后续逻辑的随机序列与参数 seed 耦合；多线程或复用同一进程的调用方需注意全局状态副作用。
4. PD router 守卫绕过：依赖 server_info 中的 router_manager 字段存在；若 router 响应结构变化或字段缺失，可能误入断言分支或过度放宽并发边界。
5. gsm8k kwargs 兼容性：--chat-template-kwags 用 json.loads 解析，非法 JSON 会在命令行阶段抛错；模型不认识的 kwarg（如对 Qwen 传 thinking）会被 apply_chat_template 拒绝。- 影响：对使用 --seed 的 benchmark 用户，图片与文本数据集生成从此可复现；gsm8k 可针对不同模型透传 thinking 开关；PD router 部署下 one_batch_server 不再因 internal states 缺失而崩溃。对测试基础设施，新增的 prefix-cache 与 seed 独立性测试可提前拦截图片 KV 复用和 processor 初始化导致的不确定性回归，并为 Kimi K3 (PR#32541) 的 media-kwags 契约提供测试锚点。- 风险标记：断言修复可能暴露 VLM 套件失败，全局 RNG 重置的副作用，PD 路由守卫放宽并发边界，依赖 tokenizer vocab 顺序

关联脉络

- PR #33025 [Kimi K3] Add reasoning, tool-call, and OpenAI serving support: 同一 Kimi K3 功能线; 本 PR 的 KimiK3Processor media-kwarg 契约测试为后续集成铺路
- PR #32890 feat(kernels): port standalone Kimi K3 kernels: Kimi K3 内核移植与数据集契约测试属于同一模型接入 workflow
- PR #33143 Replace Kimi K3 DeepGEMM patch with 0.1.5.post1: Kimi K3 依赖升级, 与本 PR 的基准测试契约共用同一模型接入上下文