PR #26530 完整报告

sgl-project/sglang

[diffusion] CI: Infer diffusion test sampling params from task type

合并时间：2026-06-01 11:57

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26530

执行摘要

- 一句话：扩散测试采样参数自动推断，移除冗余参数
- 推荐动作：建议扩散测试相关开发者精读 testcase_configs.py 中的新增函数，理解推断逻辑。该 PR 的设计模式（通过 task type 自动选择参数模板）值得在其他类似需要多配置的测试场景中复制。同时注意后续新增模型时检查推断映射是否覆盖。

功能与动机

简化 diffusion 测试用例的编写和维护，避免因人工指定错误的 sampling_params 模板（如 LTX 之前误用 T2I 模板）导致测试参数错误。PR body 明确指出：允许 test case 省略 sampling_params，从 ModelTaskType 推断共享的测试采样模板。

实现拆解

1. sampling_params 可选化：在 DiffusionTestCase 中将 sampling_params 字段类型改为 DiffusionSamplingParams | None，并在 __post_init__ 中若为 None 则调用 get_default_sampling_params_for_server_args 自动填充。
2. 新增推断函数：
   - _get_extra_arg_value(extras, option_name)：解析 server_args.extras 中的键值对。
   - get_model_task_type_for_server_args(server_args)：先检查 --pipeline-class-name，若存在则通过 get_pipeline_config_classes 获取 task type；否则 fallback 到 model registry 中的 pipeline_config_cls.task_type。
   - get_default_sampling_params_for_model_task(task_type)：根据 task type（T2I, I2I/TI2I, T2V, I2V/TI2V, I2M 等）返回对应预定义的采样参数。
   - get_default_sampling_params_for_server_args(server_args)：组合上述两步。
3. 清理 gpu_cases.py：移除约 35 行显式传递的 T2I_sampling_params、TI2I_sampling_params、T2V_sampling_params 等参数，仅保留需要覆盖默认值的 case 显式传递（如 qwen_image_t2i_cache_dit_scm_config_diffusers_1gpu 通过 replace 修改参数）。同时移除不再需要的导入符号。
4. 更新 perf baseline：根据新的 LTX 2.3 HQ 采样参数重新运行测试，更新 perf_baselines.json 中的 ltx_2_3_hq_pipeline 性能基线数据（stages_ms、denoise_step_ms、expected_e2e_ms 等均有变化）。

5. 锁定 ci-data 版本：在 test_utils.py 中将 SGL_TEST_FILES_CI_DATA_REVISION 更新为对应 LTX 2.3 HQ 新 GT 的提交 SHA。 无额外配置或部署改动。

关键文件：

- python/sglang/multimodal_gen/test/server/testcase_configs.py（模块 测试配置；类别 test；类型 test-coverage；符号 _get_extra_arg_value, get_model_task_type_for_server_args, get_default_sampling_params_for_model_task, get_default_sampling_params_for_server_args）：核心变更文件：添加了 sampling_params 可选化和自动推断逻辑
- python/sglang/multimodal_gen/test/server/perf_baselines.json（模块 性能基线；类别 test；类型 test-coverage）：更新了 LTX 2.3 HQ 的性能基线，反映采样参数变化后的实际性能
- python/sglang/multimodal_gen/test/server/gpu_cases.py（模块 GPU 用例；类别 test；类型 test-coverage）：移除了大量冗余的 sampling_params 参数，简化测试用例定义
- python/sglang/multimodal_gen/test/test_utils.py（模块 测试工具；类别 test；类型 test-coverage）：更新 ci-data 引用版本，确保新 GT 被使用

关键符号：_get_extra_arg_value, get_model_task_type_for_server_args, get_default_sampling_params_for_model_task, get_default_sampling_params_for_server_args

关键源码片段

python/sglang/multimodal_gen/test/server/testcase_configs.py

核心变更文件：添加了 sampling_params 可选化和自动推断逻辑

```
# 根据 server_args 推断任务类型
def get_model_task_type_for_server_args(server_args: DiffusionServerArgs) -> ModelTaskType:
    # 首先尝试从 extras 中解析 --pipeline-class-name
    pipeline_class_name = _get_extra_arg_value(
        server_args.extras, "--pipeline-class-name"
    )
    if pipeline_class_name:
        config_classes = get_pipeline_config_classes(pipeline_class_name)
        if config_classes is not None:
            pipeline_config_cls, _ = config_classes
            return pipeline_config_cls.task_type
    # 回退到从 model registry 获取模型信息
    model_info = get_model_info(server_args.model_path)
    if model_info is None:
        raise ValueError(
            f"Could not resolve model info for {server_args.model_path!r}"
        )
    return model_info.pipeline_config_cls.task_type


# 根据任务类型返回默认采样参数
```

```
def get_default_sampling_params_for_model_task(
    task_type: ModelTaskType,
) -> DiffusionSamplingParams:
    if task_type == ModelTaskType.T2I:
        return T2I_sampling_params
    if task_type in (ModelTaskType.I2I, ModelTaskType.TI2I):
        return TI2I_sampling_params
    if task_type == ModelTaskType.T2V:
        return T2V_sampling_params
    if task_type in (ModelTaskType.I2V, ModelTaskType.TI2V):
        return TI2V_sampling_params
    if task_type == ModelTaskType.I2M:
        return HUNYUAN3D_SHAPE_sampling_params
    # 默认使用 T2I 参数
    return T2I_sampling_params
```

评论区精华

无 Review 评论。从提交历史可见渐进式演进：先实现核心推断逻辑，然后移除冗余参数，接着添加 fallback 到 model task 以支持自定义 pipeline case，最后调整 LTX baseline。整体开发思路清晰，未出现重大争议。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 推断映射不完整风险：get_default_sampling_params_for_model_task 中定义的 task type 映射可能未覆盖所有模型变体或新增任务类型，导致新模型测试时获得不正确的默认参数。
  2. 自定义 pipeline 解析风险：get_model_task_type_for_server_args 先检查 --pipeline-class-name，但若 get_pipeline_config_classes 返回 None 则会 fallback 到 model registry，此时可能得到错误的 task type。
  3. 基线更新风险：新基线依赖于 ci-data 更新，若 ci-data 未被正确拉取，测试可能因基线不匹配而失败。
  4. 回归风险：移除了大量显式参数，但覆盖了 DiffusionTestCase 构造函数的默认行为，若某些 case 依赖的默认参数与之前显式传递的不一致，可能导致测试失败。从 source_diff 来看，所有移除的 case 的 sampling_params 都正好是推断出的默认值，因此风险较低。 - 影响：影响范围：仅限于 diffusion 测试模块（ python/sglang/multimodal_gen/test/server/），影响开发者添加和运行测试的方式。开发者现在只需指定 server_args 即可自动获得正确的采样参数，减少样板代码和潜在错误。对用户和系统的运行无影响。对团队协作的影响：标准化了测试参数配置，降低了新模型接入测试的门槛。 - 风险标记：推断映射可能遗漏模型类型，基线依赖外部 ci-data

关联脉络

- 暂无明显关联 PR