PR #35679 完整报告

sgl-project/sglang

[diffusion] Refresh eager optimization skills and benchmark safeguards

合并时间：2026-08-20 22:03

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35679

执行摘要

- 一句话：刷新 diffusion 优化技能：eager 默认基准加缓存清理保障
- 推荐动作：值得略读。对参与 SGLang Diffusion 性能优化或基准工作流的工程师有直接参考价值，重点可关注三点设计：eager 作为 ground truth、compile 仅作受控对照的定位；task-owned 缓存 + marker + ledger 的安全清理模式；以及通过依赖轻量单测守护 nightly 对齐的做法。不涉及生产代码，无需进入核心代码审查。

功能与动机

PR body 指出：现有 skill 仍假设 compile 是默认快速路径，且未覆盖近期新增的多个模型家族，这让 eager 内核机会发现更难，重复的模型 profiling 也容易耗尽受限主机的磁盘。关联调研 issue #21 的 H100/H200/B300 数据显示：多数模型 eager 与 compile 相当甚至更快，compile 经常因 warmup 超时（>180 秒）或数值漂移而无数据。因此把 eager 作为 ground truth、compile 作为显式对照组，并保证每次新建的 per-model 下载缓存可在成功、失败或受控中断后删除；SGLang JIT 缓存则故意不重定向、不删除。

实现拆解

1. 基准预设刷新：在 bench_diffusion_denoise.py 中为 LongCat-Image、SANA-Video、LingBot Video MoE、Cosmos3 Edge/distilled、LTX-2.5 及 diffusion decoder 新增 preset，并新增结构化 LINGBOT_VIDEO_PROMPT；NIGHTLY_PRESET_ORDER 末尾追加 minimax-h3-t2va，与 CI 的 comparison_configs.json 保持对齐。
2. eager 默认化：run_benchmark_once / build_sglang_cmd 默认不追加 --enable-torch-compile，--torch-compile 成为显式标注的对照组；--no-torch-compile 保留兼容但不再必需；MiniMax-H3 即使在请求 compile 时也强制 eager 一致性模式。文档中同时把 torch.compile 从 " 默认加速 " 重新定位为 " 需验证的 measured comparator"。
3. 任务专属缓存隔离与清理：新增 MODEL_CACHE_MARKER（.sglang-diffusion-benchmark-cache）和 MODEL_WEIGHT_SUFFIXES；_prepare_model_cache 创建每任务 run 目录并拒绝复用已存在的目录；_model_cache_env 将 HF_HOME、HF_XET_CACHE、TRANSFORMERS_CACHE、MODELSCOPE_CACHE 重定向到该目录；_cleanup_model_cache 只删除带 marker 的 run 目录，并把前后字节数、权重文件数写入 JSONL ledger；清理在 finally 中执行，覆盖成功、失败与 KeyboardInterrupt，且不重定向 SGLANG_CACHE_DIR。

4. 文档同步：benchmark-and-profile.md 更新用法、预设表格和缓存清理说明；existing-fast-paths.md 新增 "Recent Model Audit Boundaries"，明确 LongCat 拆分 QKV、SANA-Video 复用 bit-exact 辅助、LingBot 路由与 srt/layers/moe/topk.py::biased_grouped_topk 的兼容性、LTX-2.5 NATTEN 解码器、Cosmos3 Edge 不要重复关闭的 BCG 方向；sglang-diffusion-performance/SKILL.md 与 add-model skill 同步缓存清理要求。
5. 测试配套：新增 test_diffusion_benchmark_skill.py（185 行、6 个测试），通过 fake diffusion_skill_env 模块直接加载真实脚本，不下载权重、不使用 GPU，覆盖夜间对齐（validate_nightly_alignment）、新预设 eager 默认、缓存清理 ledger、拒绝复用目录、中断清理和失败记录。

关键文件：

- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/scripts/bench_diffusion_denoise.py（模块 基准脚本；类别 source；类型 core-logic；符号 _safe_cache_component, _prepare_model_cache, _model_cache_env, _cache_stats）：本 PR 的核心脚本：默认切换为 eager、新增任务专属缓存隔离与 JSONL ledger 清理，并补充 7 个新模型预设和 LingBot 结构化 prompt，同时将 MiniMax-H3 加入夜间顺序。
- python/sglang/multimodal_gen/test/unit/test_diffusion_benchmark_skill.py（模块 单元测试；类别 test；类型 test-coverage；符号 _load_benchmark_module, TestDiffusionBenchmarkSkill, test_nightly_presets_remain_aligned, test_recent_model_presets_are_eager_by_default）：新增 6 个不依赖权重和 GPU 的单元测试：验证夜间对齐、新预设 eager 默认、缓存清理 ledger、拒绝复用目录、中断和失败清理，是本次保障设计的可执行契约。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/benchmark-and-profile.md（模块 技能文档；类别 docs；类型 documentation）：权威工作流文档，同步 eager 默认、--torch-compile 显式对照和缓存清理用法，并更新预设表格中的 DiT 驻留说明。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/existing-fast-paths.md（模块 优化指引；类别 docs；类型 documentation）：新增 Recent Model Audit Boundaries，为 LongCat-Image、SANA-Video、LingBot Video MoE、LTX-2.5、Cosmos3 Edge 指明既有融合路径与审计边界，是后续内核优化的关键指引。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/SKILL.md（模块 技能入口；类别 docs；类型 documentation）：技能入口文档，更新主引用列表、eager 默认说明与缓存清理使用要求。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-performance/SKILL.md（模块 性能技能；类别 docs；类型 documentation）：性能技能文档把 torch.compile 重新定位为受控对照而非默认加速，并补充任务缓存清理要求。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-add-model/SKILL.md（模块 加模技能；类别 docs；类型 documentation）：add-model 技能同步补充缓存清理提示，保证新增模型流程同样遵守任务缓存纪律。

关键符号：_safe_cache_component, _prepare_model_cache, _model_cache_env, _cache_stats, _cleanup_model_cache, run_benchmark_once,

_run_benchmark_once_impl, build_sglang_cmd,
validate_nightly_alignment,_load_benchmark_module

关键源码片段

python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/scripts/bench_diffusion_denoise.py

本 PR 的核心脚本：默认切换为 eager、新增任务专属缓存隔离与 JSONL ledger 清理，并补充 7 个新模型预设和 LingBot 结构化 prompt，同时将 MiniMax-H3 加入夜间顺序。

```
#!/usr/bin/env python3
"""
End-to-end denoise-stage benchmark presets for SGLang Diffusion.

本脚本是 SGLang Diffusion 官方 denoise 基准助手（skill 内嵌）。
默认以 eager 为 ground truth，torch.compile 只是显式标注的对照组；
依据来自关联调研 issue #21：多数预设下 eager 与 compile 相当甚至更快，
compile 还经常 warmup 超时或数值漂移。

新增用法契约（写入模块 docstring）：
    python3 .../bench_diffusion_denoise.py --model flux
        # 默认 eager 运行
    python3 .../bench_diffusion_denoise.py --model flux --torch-compile
        # 显式请求 compile 对照
    python3 .../bench_diffusion_denoise.py --model longcat-image \
        --model-cache-root /task/model-caches --cleanup-model-cache
        # 使用任务专属权重缓存，结束后无论成败都会清理
"""

# 缓存清理标记：只有带此标记的 run 目录才会被 --cleanup-model-cache 删除，
# 避免误删共享 Hugging Face / ModelScope 缓存。
MODEL_CACHE_MARKER = ".sglang-diffusion-benchmark-cache"

# 用于统计缓存内权重文件数量的后缀集，区分权重与普通缓存文件。
MODEL_WEIGHT_SUFFIXES = {
    ".bin",
    ".ckpt",
    ".gguf",
    ".pt",
    ".pth",
    ".safetensors",
}

# 夜间基准顺序新增 MiniMax-H3 T2VA，保持与
# scripts/ci/utils/diffusion/comparison_configs.json 对齐。
NIGHTLY_PRESET_ORDER = (
    "flux",
    "flux2",
```

```
    "qwen",
    "qwen-edit",
    "zimage",
    "wan-t2v",
    "wan-ti2v",
    "ltx23-ti2v-two-stage",
    "ideogram4-fp8",
    "cosmos3-super-t2v",
    "wan-i2v",
    "minimax-h3-t2va",
)
```

python/sglang/multimodal_gen/test/unit/test_diffusion_benchmark_skill.py

新增 6 个不依赖权重和 GPU 的单元测试：验证夜间对齐、新预设 eager 默认、缓存清理 ledger、拒绝复用目录、中断和失败清理，是本次保障设计的可执行契约。

```
import importlib.util
import sys
import types
from pathlib import Path
from unittest.mock import patch


def _load_benchmark_module(temp_root: Path):
    # 直接从 skill 目录加载真实脚本，不经过安装入口。
    multimodal_gen_root = Path(__file__).resolve().parents[2]
    script_path = (
        multimodal_gen_root
        / ".claude"
        / "skills"
        / "sglang-diffusion-benchmark-profile"
        / "scripts"
        / "bench_diffusion_denoise.py"
    )
    # 用轻量 fake 模块替代 diffusion_skill_env，所有路径都落到临时目录，
    # 让单测不依赖真实仓库状态、不下载任何权重。
    fake_env = types.ModuleType("diffusion_skill_env")
    fake_env.ensure_dir = lambda path: (
        Path(path).mkdir(parents=True, exist_ok=True) or Path(path)
    )
    fake_env.get_assets_dir = lambda _root: temp_root / "assets"
    fake_env.get_output_dir = lambda _kind, _root: temp_root / "outputs"
    fake_env.get_repo_root = lambda: temp_root / "repo"
    fake_env.pick_idle_gpus = lambda count: list(range(count))

    spec = importlib.util.spec_from_file_location(
        "test_bench_diffusion_denoise", script_path
    )
    assert spec is not None and spec.loader is not None
```

```
    module = importlib.util.module_from_spec(spec)
    with patch.dict(sys.modules, {"diffusion_skill_env": fake_env}):
        spec.loader.exec_module(module)
    return module
```

评论区精华

本 PR 无 review 评论（review_comments_count = 0），核心设计权衡体现在 PR body 和代码实现中：一是把 *eager* 定为 ground truth，compile 只作为显式对照组，依据来自 issue #21 的全模型性能对比；二是缓存清理只针对带 marker 的 per-run 目录且拒绝复用，明确警告不要指向共享 Hugging Face/ModelScope 缓存；三是 SGLang JIT 缓存被刻意排除在清理之外，避免误删编译产物。这些决策通过单元测试固化为可验证的契约。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 默认行为变更：bench 脚本从 compile 默认变为 *eager* 默认，依赖旧行为的自动化脚本或文档命令需要适配；已保留 --no-torch-compile 兼容，并提供 --torch-compile 显式对照。
  2. 缓存误删风险：若用户将 --model-cache-root 指向共享 HF/ModelScope 缓存，清理逻辑可能删除非任务文件；实现通过 marker 文件、拒绝复用已有 run 目录和文档警告降低风险，但删除操作本身不可逆。
  3. 测试与真实环境脱节：单元测试用 fake env 加载模块，只验证命令字符串与清理逻辑，不下载权重、不跑 GPU；新预设的实际 sglang *generate* 参数是否正确只能靠后续 CI/夜间任务暴露。
  4. 夜间对齐强耦合：test_nightly_presets_remain_aligned 与 scripts/ci/utils/diffusion/comparison_configs.json 强绑定，CI 配置更新而 skill 未同步时测试会失败；这是有意的守护，但也增加跨 PR 维护成本。
  5. 磁盘残留：SGLANG_CACHE_DIR 不重定向、不清理，长时间 profiling 仍可能累积 JIT 缓存占用任务磁盘。 - 影响：本 PR 不触及任何生产推理路径，对最终用户无影响。影响范围集中在团队内部的 diffusion 优化工作流：统一了 *eager* 基线方法论，新增模型预设便于后续内核优化机会发现；任务专属缓存隔离与 JSONL ledger 显著降低受限主机上的磁盘风险和 profiling 污染；夜间基准与 CI 配置通过测试形成一致性约束。对参与 diffusion 性能优化、夜间基准维护的工程师影响较大，属于中等强度的工具链变更。 - 风险标记：默认行为变更（*eager*），缓存清理误删风险，新预设未真机验证，夜间对齐强耦合

关联脉络

- PR #35688 [diffusion] feat: let every layerwise component be configurable: 同属 diffusion 性能优化技能与配置工作线，layerwise 组件可配置化与本 PR 的 *eager* 基线方法论互补。

- PR #35418 [Diffusion] Support MiniMax-H3 pruned safetensors checkpoints: MiniMax-H3 权重加载支持是本 PR 将 minimax-h3-t2va 加入夜间预设顺序的前置依赖。
- PR #35511 [diffusion] CI: add minimax-h3 ref2va audio consistency coverage and guard peak vram: 同为夜间基准 /CI 对齐体系，本 PR 的 nightly 预设对齐测试与 CI 覆盖直接相关。
- PR #35668 [diffusion] feat: add weight source reader: 权重读取后端重构为后续模型预设的下载与缓存行为提供基础，与本 PR 的缓存隔离设计相关。
- PR #35664 [diffusion] feat: warn on an unverified short edge instead of rejecting it for minimax-h3: MiniMax-H3 预设校验线的一部分，与本 PR 的预设刷新和维护方向一致。