PR #35664 完整报告

sgl-project/sglang

[diffusion] feat: warn on an unverified short edge instead of rejecting it for minimax-h3

合并时间：2026-08-20 16:52

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35664

执行摘要

- 一句话：放宽 MiniMax-H3 短边限制，非 768 仅告警不拒绝
- 推荐动作：值得精读。该 PR 展示了“默认值不等于硬限制”的产品化取舍，以及用 lru_cache 实现按值去重告警的轻量做法，配套测试覆盖了边界与告警行为。建议阅读时重点关注 request_validation.py 与 resolved_plan.py 两处告警调用是否一致，以及如何为“未验证配置”设计既透明又不打扰用户的反馈机制。

功能与动机

PR body 明确指出 target.short_edge 此前必须恰好为 768，否则请求会被两处拒绝（request_validation.py 的 HTTP gate 与 resolved_plan.py 的形状策略），而形状数学本身并不需要这个限制。MiniMax 官方 model card 说明 768 只是 default 而非 limit，社区已有用户在 352p / 416p 生成且 prompt 遵从度更好（huggingface discussions/65）。作者希望在小显存卡上把短边作为调节显存与耗时的杠杆。

实现拆解

实现分为四步：

1. 新增推荐短边常量与去重告警设施（python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/minimax_h3/constants.py）：新增 MINIMAX_H3_RECOMMENDED_SHORT_EDGE = 768，并定义以 lru_cache(maxsize=None) 装饰的 warn_unverified_short_edge(short_edge)，保证同一取值进程内只告警一次，避免高频请求刷屏。
2. 放宽形状策略校验（resolved_plan.py）：_validate_base_short_edge 从“必须是 768”改为“必须为正整数”，非 768 时调用 warn_unverified_short_edge 告警。解析函数 minimax_h3_resolve_spatial_shape 的逻辑不变，任意正整数短边都能按宽高比缩放、落入 768×1344 像素预算并取整到 32px 画布网格。
3. 放宽 HTTP 层校验（request_validation.py）：_validate_target 中的 target.short_edge 从“必须等于 768”改为“必须为正整数”，并同样调用告警函数。注意该调用点未像 resolved_plan.py 那样先判断是否等于 768，存在 768 也会触发告警的缺陷（详见风险分析）。
4. 测试配套（新增 test_minimax_h3_short_edge.py，11 个用例）：覆盖 352/384/416/512/768 均解析到 32px 对齐且不超像素预算；短边减半画布面积约四分之

一；超大短边回退到 size_mode="area"；0、负数、小数抛 ValueError；768 不告警；非 768 值重复调用只告警一次。测试通过 autouse fixture 清理 lru_cache，保证用例隔离。

关键文件：

- python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/minimax_h3/constants.py（模块 H3 形状策略；类别 source；类型 data-contract；符号 warn_unverified_short_edge, MINIMAX_H3_RECOMMENDED_SHORT_EDGE）：新增推荐短边常量与按值去重的告警函数，是整个放宽策略的基础设施与数据契约核心。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/minimax_h3/resolved_plan.py（模块 H3 形状策略；类别 source；类型 data-contract；符号 _validate_base_short_edge, minimax_h3_resolve_spatial_shape）：形状策略的入口校验函数从硬编码 768 改为正整数 + 告警，是核心数据契约变更。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/minimax_h3/request_validation.py（模块 H3 请求校验；类别 source；类型 data-contract；符号 _validate_target）：HTTP 层入口校验同步放宽，但告警调用缺少 768 判断，存在误告警缺陷。
- python/sglang/multimodal_gen/test/unit/test_minimax_h3_short_edge.py（模块 H3 单元测试；类别 test；类型 test-coverage；符号 _reset_warn_cache, TestResolveSpatialShape, test_a_smaller_short_edge_resolves_to_an_aligned_canvas, test_halving_the_short_edge_quarters_the_canvas）：新增 11 个用例覆盖分辨率解析、像素预算、非法值拒绝与告警去重行为，是验证本次放宽正确性的关键配套。

关键符号：warn_unverified_short_edge, _validate_base_short_edge, _validate_target, minimax_h3_resolve_spatial_shape

关键源码片段

python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/minimax_h3/constants.py

新增推荐短边常量与按值去重的告警函数，是整个放宽策略的基础设施与数据契约核心。

```
# SPDX-License-Identifier: Apache-2.0
from __future__ import annotations

from functools import lru_cache

from sglang.multimodal_gen.runtime.utils.logging_utils import init_logger

logger = init_logger(__name__)

# 官方所有 recipe 与参考输出使用的短边值。
# 形状数学本身是通用的：按宽高比缩放、受像素预算约束并取整到画布网格，
# 因此其他正值也能解析出合法画布，只是不在官方调优与评测范围内。
MINIMAX_H3_RECOMMENDED_SHORT_EDGE = 768

@lru_cache(maxsize=None)
```

```
def warn_unverified_short_edge(short_edge: int) -> None:
    """按不同取值各警告一次；请求会重复，告警不应刷屏。"""
    logger.warning(
        "MiniMax H3 target.short_edge=%d is outside the verified configuration: "
        "%d is the only short edge MiniMax publishes recipes and reference "
        "outputs for. Smaller edges cost proportionally less memory and time; "
        "quality and prompt adherence at this size are not covered by any "
        "MiniMax or SGLang measurement.",
        short_edge,
        MINIMAX_H3_RECOMMENDED_SHORT_EDGE,
    )
```

python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/minimax_h3/request_validation.py

HTTP 层入口校验同步放宽，但告警调用缺少 768 判断，存在误告警缺陷。

```
def _validate_target(target: Any, *, profile: MiniMaxH3TaskProfile) -> dict[str, Any]:
    path = "target"
    # 规范化 target 只保留少量字段，这里仅校验并输出 short_edge 与 aspect_ratio。
    short_edge = _require_int(target.get("short_edge"), f"{path}.short_edge")
    if short_edge <= 0:
        raise ValueError(f"{path}.short_edge must be positive, got {short_edge}")
    # 注意：这里没有像 resolved_plan.py 那样先判断是否为 768，
    # 因此合法的 768 请求也会触发一次“未验证配置”告警，与 PR 意图不符。
    warn_unverified_short_edge(short_edge)
    # ... 后续 aspect_ratio 与 duration_seconds 校验保持不变 ...
```

python/sglang/multimodal_gen/test/unit/test_minimax_h3_short_edge.py

新增 11 个用例覆盖分辨率解析、像素预算、非法值拒绝与告警去重行为，是验证本次放宽正确性的关键配套。

```
@pytest.fixture(autouse=True)
def _reset_warn_cache():
    # 每个用例前后清理 lru_cache，避免告警去重状态跨用例泄漏。
    constants.warn_unverified_short_edge.cache_clear()
    yield
    constants.warn_unverified_short_edge.cache_clear()


class TestResolveSpatialShape:
    @pytest.mark.parametrize("short_edge", [352, 384, 416, 512, 768])
    def test_a_smaller_short_edge_resolves_to_an_aligned_canvas(self, short_edge):
        shape = minimax_h3_resolve_spatial_shape(
            width=16, height=9, base_short_edge=short_edge
        )
        # 宽高都应落在 32px 画布网格上，且不超过像素预算。
        assert shape["width"] % MINIMAX_H3_CANVAS_MULTIPLE == 0
        assert shape["height"] % MINIMAX_H3_CANVAS_MULTIPLE == 0
        assert shape["width"] * shape["height"] <= MINIMAX_H3_MAX_PIXELS
```

```
        # 短边应落在请求值附近（允许 32px 网格取整误差）。
        assert (
            abs(shape["effective_short_edge"] - short_edge) < MINIMAX_H3_CANVAS_MULTIPLE
        )


class TestWarning:
    def test_an_unverified_short_edge_warns_once_per_value(self, caplog):
        with caplog.at_level("WARNING"):
            for _ in range(3):
                minimax_h3_resolve_spatial_shape(
                    width=16, height=9, base_short_edge=384
                )
        # 同一取值重复调用只告警一次。
        assert caplog.text.count("outside the verified configuration") == 1
        assert "768" in caplog.text
```

评论区精华

本 PR 没有实质性的 review 讨论线程，唯一的 Issue 评论是作者触发的 /tag-and-rerun-ci，属于 CI 流程操作。核心设计权衡（default 不等于 limit、警告代替拒绝）已在 PR body 中充分说明，未引发其他争论。

- 暂无高价值评论线程

风险与影响

- 风险：存在以下风险：

1. HTTP gate 路径 768 也会误告警：request_validation.py 的 _validate_target 中直接无条件调用 warn_unverified_short_edge(short_edge)，而该函数内部不检查是否等于 768。因此合法的 768 请求在 HTTP 层也会收到“outside the verified configuration”的告警，与 PR body 宣称的“No change at 768 … does not warn”不一致。测试 test_the_recommended_short_edge_does_not_warn 只覆盖了 resolved_plan.py 路径，未覆盖 request_validation.py 路径。
2. 非 768 分辨率质量无背书：告警文案明确表示小于 768 的短边质量与 prompt 遵从度不在任何 MiniMax 或 SGLang 测量范围内，用户可能误以为所有尺寸都得到同等保障。
3. 进程级全局缓存状态：lru_cache(maxsize=None) 使告警状态成为进程级全局状态，在长驻服务中不同取值第一次出现后不再重复告警；若后续想调整告警文案或改回拒绝策略，需要清理缓存或重启进程。
4. 极小短边可能超出质量下限：虽然形状数学接受任意正整数，但极小分辨率（如 64）可能生成无意义内容，当前没有设置下界。 - 影响：对用户而言，MiniMax-H3 在 API 层不再拒绝非 768 短边，小显存卡（如 2×RTX 4090）可以用 384 等分辨率换取约 4.9 倍的生成加速（57.69s → 11.69s），并成比例降低显存占用；对大尺寸短边则会自动被像素预算约束。对系统而言，改动集中在 multimodal_gen 的 MiniMax-H3 请求校验与形状解析路径，默认行为（768）不变，但告警日志在非 768 使用时首次出现会提示用户风险。对团队而言，该 PR 确立了一个可复用的“未验证配置仅告警”的模式（类似 constants.py 中按值去重的告警

函数），后续其他模型门禁可以借鉴。 - 风险标记：HTTP gate 路径 768 也会误告警，非 768 短边质量未经官方验证，进程级 lru_cache 全局状态，极小短边无质量下限

关联脉络

- PR #35511 [diffusion] CI: add minimax-h3 ref2va audio consistency coverage and guard peak vram: 同属 MiniMax-H3 的质量保障与 CI 覆盖线，本 PR 放宽分辨率后需要这些一致性测试来守门。
- PR #35370 [diffusion] feat: load GGUF transformer checkpoints (MiniMax-H3): MiniMax-H3 的加载与运行链路扩展，本 PR 允许更多分辨率后与该加载路径协同影响显存占用。
- PR #35626 [diffusion] fix: keep large vocab tables in host memory under layerwise offload: 同为降低 MiniMax-H3 显存占用的方向，短边缩小与 layerwise offload 是互补的两种手段。
- PR #35618 [diffusion] UX: report where a component's weights are: 同属 MiniMax-H3 / diffusion 的 UX 改进方向，本 PR 的告警提示与组件驻留报告共同改善可观测性。