PR #30463 完整报告

sgl-project/sglang

[Bugfix] Map reasoning_effort=low to Nemotron-3 Super low_effort + warn on unsupported levels

合并时间：2026-07-09 03:19

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/30463

执行摘要

- 一句话：映射 Nemotron-3 Super low_effort 并警告不支持级别
- 推荐动作：建议阅读本 PR，特别是 protocol.py 中 normalize_reasoning_inputs 的重构，以及 template_detection.py 中如何通过配置字段实现模型特定的 effort 映射。这些设计模式值得应用到其他需要差异化的模板参数中。

功能与动机

Nemotron-3 Super 的 chat template 暴露了 low_effort 布尔参数，但 SGLang 未将其与 OpenAI 的 reasoning_effort 字段关联，导致 low 级别静默无效，其他级别也无声失败。下游用户（如 CoreWeave）请求看起来被接受但从未生效。该 PR 解决了这一缺失映射。

实现拆解

1. 新增 effort_kwarg 配置字段（template_detection.py）：在 ReasoningToggleConfig 中加入 effort_kwarg: Optional[str]，用于存储模型特定的 effort 模板参数名（如 low_effort）。
2. 检测 Nemotron-3 Super（template_detection.py）：新增 nemotron_3_super_low_effort 检测规则，通过模板中同时存在 low_effort 和 truncate_history_thinking 特征词来识别 Super 变体，并设置 effort_kwarg='low_effort'。同时修改 _is_nemotron_3 函数，使用字段比较而非严格相等判断，使携带 effort_kwarg 的 Super 配置仍能被识别为 nemotron_3 解析器。
3. 统一 thinking 开关逻辑（protocol.py）：重构 normalize_reasoning_inputs validator，使用局部变量 thinking 统一处理 reasoning 字典和顶层 reasoning_effort 字段，最后一次性设置 chat_template_kwargs['thinking'] 和 chat_template_kwargs['enable_thinking']，消除重复分支。
4. 在 _apply_jinja_template 中应用 effort_kwarg（serving_chat.py）：构建模板参数时，若 reasoning_config.effort_kwarg 存在，则 reasoning_effort='low' 时通过 setdefault 设置对应 kwarg；若为 medium/high/max 则输出警告日志。none 继续由 #28860 的 normalize_reasoning_inputs 处理。
5. 新增测试：test_serving_chat.py 中增加 _setup_nemotron_super、_run_jinja_with_effort 辅助方法和三个测试用例（low 映射、high 警告、Nano 无影响）。test_protocol.py 增加 test_chat_completion_reasoning_effort_high_enables_thinking

和 test_chat_completion_reasoning_effort_none_overrides_enabled。test_template_manager.py 增加 test_nemotron_super_detects_low_effort_kwarg。

关键文件：

- python/sglang/srt/entrypoints/openai/protocol.py（模块 协议层；类别 source；类型 core-logic；符号 normalize_reasoning_inputs）：重构了 normalize_reasoning_inputs，统一 thinking 开关逻辑，影响所有模型的 reasoning_effort 处理。
- python/sglang/srt/entrypoints/openai/serving_chat.py（模块 对话服务；类别 source；类型 core-logic；符号 _apply_jinja_template）：在 _apply_jinja_template 中新增了 effort_kwarg 处理逻辑，实现对 Nemotron-3 Super 低努力级别的映射和警告。
- python/sglang/srt/managers/template_detection.py（模块 模板检测；类别 source；类型 core-logic；符号 ReasoningToggleConfig, _is_nemotron_3, DETECTION_RULES）：增加了 effort_kwarg 字段和 Nemotron-3 Super 检测规则，修改了 _is_nemotron_3 比较方式。
- test/registered/unit/entrypoints/openai/test_serving_chat.py（模块 对话测试；类别 test；类型 test-coverage；符号 _setup_nemotron_super, _run_jinja_with_effort, test_nemotron_super_low_effort_mapped_to_kwarg, test_nemotron_super_high_effort_warns_without_kwarg）：包含了 Nemotron-3 Super 和 Nano 的 effort 映射测试用例。
- test/registered/unit/entrypoints/openai/test_protocol.py（模块 协议测试；类别 test；类型 test-coverage；符号 test_chat_completion_reasoning_effort_high_enables_thinking, test_chat_completion_reasoning_effort_none_overrides_enabled）：新增了顶层 reasoning_effort 启用 thinking 和 none 覆盖 enabled 的测试。
- test/registered/unit/managers/test_template_manager.py（模块 模板测试；类别 test；类型 test-coverage；符号 test_nemotron_super_detects_low_effort_kwarg）：验证 Nemotron-3 Super 模板检测及 effort_kwarg 配置。

关键符号：normalize_reasoning_inputs, _apply_jinja_template, _is_nemotron_3, _setup_nemotron_super, _run_jinja_with_effort

关键源码片段

python/sglang/srt/entrypoints/openai/protocol.py

重构了 normalize_reasoning_inputs，统一 thinking 开关逻辑，影响所有模型的 reasoning_effort 处理。

```
@model_validator(mode="before")
@classmethod
def normalize_reasoning_inputs(cls, values: Dict):
    r = values.get("reasoning")
    thinking = None # 统一 thinking 状态，  None 表示未设置

    if r is not None and isinstance(r, dict):
        effort = r.get("effort") or r.get("reasoning_effort")
        if effort in {"none", "low", "medium", "high"}:
```

```
            values["reasoning_effort"] = effort

        enabled = (
            r.get("enabled")
            if r.get("enabled") is not None
            else r.get("enable", False)
        )
        if isinstance(enabled, str):
            enabled = enabled.strip().lower() in {"1", "true", "yes", "y", "on"}
        if enabled:
            thinking = True # 从 reasoning dict 启用 thinking

    # 顶层 reasoning_effort 字段覆盖 thinking 状态
    effort = values.get("reasoning_effort")
    if effort is not None:
        thinking = effort != "none"

    if thinking is not None:
        ctk = values.get("chat_template_kwargs")
        if not isinstance(ctk, dict):
            ctk = {}
        # 同时设置两个键以兼容不同模型
        ctk.setdefault("thinking", thinking)
        ctk.setdefault("enable_thinking", thinking)
        values["chat_template_kwargs"] = ctk

    return values
```

python/sglang/srt/entrypoints/openai/serving_chat.py

在 _apply_jinja_template 中新增了 effort_kwarg 处理逻辑，实现对 Nemotron-3 Super 低努力级别的映射和警告。

```
# 在构建 extra_template_kwargs 之后（已有 reasoning_effort 和 chat_template_kwargs）
rc = self.template_manager.reasoning_config
if rc is not None and rc.effort_kwarg is not None:
    if request.reasoning_effort == "low":
        # 映射 low -> 设置 low_effort=True； 使用 setdefault 保留用户显式值
        extra_template_kwargs.setdefault(rc.effort_kwarg, True)
    elif request.reasoning_effort in ("medium", "high", "max"):
        # 模型仅支持 low， 其他级别回退默认 thinking 并发出警告
        logger.warning(
            "Model '%s' supports only 'low' reasoning effort; "
            "requested '%s' treated as default thinking",
            self.tokenizer_manager.server_args.served_model_name,
            request.reasoning_effort,
        )
```

python/sglang/srt/managers/template_detection.py

增加了 effort_kwarg 字段和 Nemotron-3 Super 检测规则，修改了 _is_nemotron_3 比较方式。

```
class ReasoningToggleConfig:
    toggle_param: Optional[str] = None
    default_enabled: Optional[bool] = None
    special_case: Optional[str] = None
    effort_kwarg: Optional[str] = None # 新增：模型特定的 effort 模板参数名

    @property
    def always_on(self) -> bool:
        return self.default_enabled is True and self.effort_kwarg is None

# 在检测规则列表中添加
DetectionRule(
    name="nemotron_3_super_low_effort",
    value=ReasoningToggleConfig(
        toggle_param="enable_thinking",
        default_enabled=True,
        effort_kwarg="low_effort", # 设置 effort 参数名
    ),
    predicate=lambda ctx: ctx.has_text("low_effort")
    and ctx.has_text("truncate_history_thinking"),
),

# 修改 _is_nemotron_3 函数，允许额外字段
def _is_nemotron_3(ctx):
    return ctx.has_text("truncate_history_thinking") and (
        ctx.reasoning_config is not None
        and ctx.reasoning_config.toggle_param == "enable_thinking"
        and ctx.reasoning_config.default_enabled is True
    )
```

评论区精华

PR 获得两位审阅者（b8zhong、Fridge003）的批准，无其他讨论或争议。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险：normalize_reasoning_inputs 的重构可能影响其他依赖此逻辑的模型（如 Qwen3、GLM4-5），但重构后的逻辑等价于原有行为（测试已覆盖顶层 effort 和 reasoning 字典两种路径）。另外，_is_nemotron_3 的字段比较放宽了严格匹配，任何 toggle_param='enable_thinking' 且 default_enabled=True 的高层配置都可能被识别为 nemotron_3，但这不是新风险，因为之前也是通过相等比较，现在只是允许额外字段存在。总体来说风险较低。

- 影响：对用户：Nemotron-3 Super 用户现在可以正确使用 reasoning_effort='low'，其他级别会收到明确警告。Nemotron-3 Nano 用户无影响。对其他模型：thinking 开关逻辑重构为统一路径，行为一致，不应有副作用。对系统：增加一条条件判断和一个警告日志，性能无负面影响。
- 风险标记：核心路径变更，模型特定配置

关联脉络

- PR #28860 [Bugfix] Fix reasoning_effort to thinking-toggle mapping: 本 PR 构建于 #28860 之上，在通用 thinking-toggle 修复的基础上添加 Nemotron-3 Super 特定的 effort 映射。