PR #35362 完整报告

sgl-project/sglang

Laguna: config-driven MoE router scoring

合并时间：2026-08-19 04:42

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35362

执行摘要

- 一句话：Laguna MoE 路由评分配置化，默认值不变
- 推荐动作：值得快速浏览而非精读。亮点在于以最小侵入方式将 TopK 已有能力通过 config 暴露，注释清晰标注了取值与分支位置，是低成本扩展点的良好示范。建议关注两点：一是后续 Laguna 变体 PR 是否会补齐 moe_router_score_func 的取值校验与单元测试；二是可对照 topk.py 中 "sigmoid" / "sqrtsoftplus" 分支实现，评估非法取值时的兜底行为。

功能与动机

PR body 仅以一句话说明动机："Needed to support future planned Laguna variants."（需要支持未来规划的 Laguna 变体）。结合代码背景：TopK（topk.py）早已实现 "sigmoid" 与 "sqrtsoftplus" 两条评分分支，但 LagunaMoE 将 scoring_func 硬编码为 "sigmoid"，导致新变体无法通过模型 config 选择路由评分非线性函数。本 PR 的目标就是把这条能力通道打开，而无需改动 TopK 本身。

实现拆解

1. 配置契约扩展（python/sglang/srt/configs/laguna.py）：在 LagunaConfig.__init__ 的参数列表新增 moe_router_score_func: str = "sigmoid"，并在初始化主体中通过 self.moe_router_score_func = moe_router_score_func 写入实例属性。默认取值与原先硬编码的 "sigmoid" 完全一致，因此对既有 Laguna 检查点加载与推理零影响，属于向后兼容的纯增量扩展。
2. 消费方接入（python/sglang/srt/models/laguna.py）：LagunaMoE.__init__ 构造 TopK 时，将 scoring_func 从常量 "sigmoid" 改为 config.moe_router_score_func，并用注释标明支持 "sigmoid" 与 "sqrtsoftplus" 两个取值、分支实现位于 topk.py。该改动只影响 top-k 选路的评分非线性函数，不触及 gate、experts、shared expert 的构建逻辑。
3. 配套与验证情况：PR 未附带单元测试、文档或性能测试；仓库 CI 的 pr-test 与 pr-test-extra 两个入口均显示失败，结合 5 行改动规模和默认值兼容性，大概率与本次变更无关，但合并前应确认失败原因。

关键文件：

- python/sglang/srt/configs/laguna.py（模块 模型配置；类别 source；类型 core-logic；符号 LagunaConfig.init, moe_router_score_func）：新增 moe_router_score_func 配置参数并写入实例属性，是本次 " 配置驱动 " 的契约源头，决定了后续 Laguna 变体可用的配

置面。

- python/sglang/srt/models/laguna.py（模块 模型实现；类别 source；类型 data-contract；符号 LagunaMoE.init, TopK）：LagunaMoE 将硬编码的 scoring_func 改为读取 config，是配置的实际消费方，直接决定路由选路的评分非线性。

关键符号：LagunaConfig.init, LagunaMoE.init

关键源码片段

python/sglang/srt/configs/laguna.py

新增 moe_router_score_func 配置参数并写入实例属性，是本次 " 配置驱动 " 的契约源头，决定了后续 Laguna 变体可用的配置面。

```
class LagunaConfig(PretrainedConfig):
    # …（省略无关的类属性）…

    def __init__(
        self,
        vocab_size: int = 100352,
        hidden_size: int = 2048,
        # …（中间大量模型结构参数省略，完整签名见仓库源码）…
        num_experts: int = 256,
        num_experts_per_tok: int = 8,
        moe_intermediate_size: int = 512,
        shared_expert_intermediate_size: int = 512,
        moe_routed_scaling_factor: float = 1.0,
        moe_router_logit_softcapping: float = 0.0,
        moe_apply_router_weight_on_input: bool = False,
        # 新增配置项：路由评分非线性函数，默认 "sigmoid"，可选 "sqrtsoftplus"
        moe_router_score_func: str = "sigmoid",
        **kwargs,
    ):
        super().__init__(
            tie_word_embeddings=tie_word_embeddings,
            bos_token_id=bos_token_id,
            eos_token_id=eos_token_id,
            pad_token_id=pad_token_id,
            **kwargs,
        )

        # MoE 路由相关字段统一写入实例属性，供模型侧读取
        self.num_experts = num_experts
        self.num_experts_per_tok = num_experts_per_tok
        self.moe_intermediate_size = moe_intermediate_size
        self.shared_expert_intermediate_size = shared_expert_intermediate_size
        self.moe_routed_scaling_factor = moe_routed_scaling_factor
        self.moe_router_logit_softcapping = moe_router_logit_softcapping
        self.moe_apply_router_weight_on_input = moe_apply_router_weight_on_input
        self.moe_router_score_func = moe_router_score_func
```

python/sglang/srt/models/laguna.py

LagunaMoE 将硬编码的 scoring_func 改为读取 config，是配置的实际消费方，直接决定路由选路的评分非线性。

```
class LagunaMoE(nn.Module):
    """Laguna 稀疏 MoE 层：gate 路由 + 专家前馈 + shared expert。"""

    def __init__(self, config, layer_id, quant_config=None, prefix=""):
        # 签名按仓库源码为准；此处聚焦路由构建路径
        super().__init__()
        self.tp_size = get_parallel().tp_size
        self.routed_scaling_factor = config.moe_routed_scaling_factor
        self.router_logit_softcapping = getattr(config, "moe_router_logit_softcapping", 0.0)

        if self.tp_size > config.num_experts:
            raise ValueError(f"TP size {self.tp_size} > num_experts {config.num_experts}.")

        # 门控网络：输出 expert logits 与 e_score_correction_bias
        self.gate = LagunaMoEGate(config, prefix=add_prefix("gate", prefix))

        # EP 下可挂冗余专家；reduce_results=False 由上层统一归约
        self.experts = get_moe_impl_class(quant_config)(
            num_experts=config.num_experts + get_exec().moe.ep_num_redundant_experts,
            top_k=config.num_experts_per_tok,
            layer_id=layer_id,
            hidden_size=config.hidden_size,
            intermediate_size=config.moe_intermediate_size,
            quant_config=quant_config,
            reduce_results=False,
            apply_router_weight_on_input=bool(config.moe_apply_router_weight_on_input),
            prefix=add_prefix("experts", prefix),
        )

        # TopK 选路：scoring_func 由硬编码改为配置驱动（本次变更）
        self.topk = TopK(
            top_k=config.num_experts_per_tok,
            layer_id=layer_id,
            renormalize=True,
            use_grouped_topk=False,
            # 默认 "sigmoid"，可选 "sqrtsoftplus"；分支逻辑在 topk.py 中
            scoring_func=config.moe_router_score_func,
            correction_bias=self.gate.e_score_correction_bias,
        )
```

评论区精华

本 PR 没有任何 review 评论或讨论线程：唯一审核人 kpham-sgl 直接给出 APPROVED 且未附加说明，PR 的 comments_count 与 review_comments_count 均为 0。因此不存在可供提

炼的评审交锋；合并判断主要依赖改动本身的自明性（5 行、默认兼容）。若未来补充配置枚举校验与测试，建议在评审中明确校验契约。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 配置取值缺少枚举校验：moe_router_score_func 只做透传，若未来变体传入拼写错误或未支持的取值，将直接进入 TopK（topk.py）分支，可能静默回退默认行为或产生未定义结果；当前没有任何防御性校验。
  - 缺少测试覆盖：没有针对新配置项的单元测试，既未验证默认 "sigmoid" 与旧硬编码行为一致，也未验证 "sqrtsoftplus" 从 config 到 TopK 的传递链路。
  - CI 状态：pr-test 与 pr-test-extra 两个入口均为失败状态，且 PR 未更新任何结论；改动仅 5 行且默认兼容，基本可排除本 PR 引发，但需人工确认 CI 失败根因，避免掩盖环境问题。
  - 兼容性风险：极低。新参数带默认值且位于参数列表尾部，不会破坏既有构造调用。
- 影响：
  - 对存量用户：无行为变化，Laguna 模型推理结果与性能完全一致（默认 "sigmoid" 不变）。
  - 对系统：影响面严格限定在 Laguna 的 MoE 路由路径（models/laguna.py 的 LagunaMoE），不涉及共享调度、缓存或其它模型。
  - 对未来开发：为规划中的 Laguna 变体提供配置切换点，后续只需在模型 config 中声明 moe_router_score_func="sqrtsoftplus" 即可启用另一种评分非线性，扩展成本极低。
  - 对团队协作：变更小、自明性强，一次性批准通过；但缺少测试与校验意味着后续维护者需要依赖 topk.py 的行为契约来完成验证。
  - 风险标记：缺少测试覆盖，配置项缺少枚举校验，CI 失败未确认原因

关联脉络

- PR #31370 [AMD] feat(moe): fold padded-topk_ids fill into fused shared-experts append+remap: 该 PR 修改了 python/sglang/srt/layers/moe/topk.py，与本次变更依赖的评分函数分支实现同源，可作为 TopK 支撑逻辑的上下文参考；文件层面无直接交集（弱关联）。
- PR #32099 [AMD] MiniMax-M3：Fuse QKV+index proj for block-fp8: 同为 MoE 模型以最小代价优化 / 扩展能力的演进案例，展示了 SGLang MoE 模型侧改动模式，与本 PR 思路相似但无文件交集（弱关联）。