

# PR #24187 完整报告

sgl-project/sglang

[Fix] Qwen3-ASR config: set thinker\_config before super().\_\_init\_\_

合并时间: 2026-05-12 11:52

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/24187>

## 执行摘要

- 一句话: 修复 Qwen3-ASR 配置初始化顺序崩溃
- 推荐动作: 值得立即合入。这是一个精确的构造顺序修复, 属于经典 Python 初始化陷阱, 值得团队在自定义配置类中注意此类依赖顺序。

## 功能与动机

PR body 指出: Launching any Qwen3-ASR model crashes during config init under huggingface\_hub >= 1.12。原因是 huggingface\_hub.dataclasses 的 init\_with\_validate 在 base \_\_init\_\_ 期间运行 validate\_token\_ids → get\_text\_config(decoder=True), 而 get\_text\_config 读取 self.thinker\_config.text\_config, 但 self.thinker\_config 只在 super().\_\_init\_\_ 返回后才设置。该问题由 #22073 引入, 并在 huggingface\_hub 1.12+ 后暴露。

## 实现拆解

1. 在 python/sglang/srt/configs/qwen3\_asr.py 的 Qwen3ASRConfig.\_\_init\_\_ 中, 将 super().\_\_init\_\_(\*\*kwargs) 一行从函数体开头移至末尾。
2. 这样在调用父类初始化时, self.thinker\_config 已经设置完毕 (由之前的 if isinstance(thinker\_config, dict): self.thinker\_config = Qwen3ASRThinkerConfig(...) 等逻辑处理), get\_text\_config 可以正常访问。
3. 无其他配置、测试或文档变更, 仅一行代码顺序调整。

关键文件:

- python/sglang/srt/configs/qwen3\_asr.py (模块 配置层; 类别 source; 类型 core-logic; 符号 Qwen3ASRConfig.init): 唯一变更文件, 将 super().\_\_init\_\_ 移到设置 thinker\_config 之后, 修复初始化崩溃

关键符号: Qwen3ASRConfig.init

## 关键源码片段

[python/sglang/srt/configs/qwen3\\_asr.py](#)

唯一变更文件, 将 super().\_\_init\_\_ 移到设置 thinker\_config 之后, 修复初始化崩溃

# 修复前: super().\_\_init\_\_ 在开头, get\_text\_config 访问 thinker\_config 时属性未设置

```

class Qwen3ASRConfig(PretrainedConfig):
    def __init__(self, thinker_config=None, **kwargs):
        super().__init__(**kwargs) # 此行会触发 validate_token_ids → get_text_config
        if thinker_config is None:
            thinker_config = {}
        if isinstance(thinker_config, dict):
            self.thinker_config = Qwen3ASRThinkerConfig(**thinker_config)
        else:
            self.thinker_config = thinker_config

    def get_text_config(self, decoder=False) -> PretrainedConfig:
        return self.thinker_config.text_config # 此时 self.thinker_config 还未定义!

# 修复后: 先设置 thinker_config, 再调用 super().__init__
class Qwen3ASRConfig(PretrainedConfig):
    def __init__(self, thinker_config=None, **kwargs):
        if thinker_config is None:
            thinker_config = {}
        if isinstance(thinker_config, dict):
            self.thinker_config = Qwen3ASRThinkerConfig(**thinker_config)
        else:
            self.thinker_config = thinker_config # 确保父类验证前属性已就绪
        super().__init__(**kwargs) # 此时 get_text_config 可以正常访问

    def get_text_config(self, decoder=False) -> PretrainedConfig:
        return self.thinker_config.text_config

```

## 评论区精华

无 review 讨论。PR body 中详细说明了崩溃原因和验证方式。

- 暂无高价值评论线程

## 风险与影响

- 风险：风险极低。仅一行代码顺序调整，逻辑等价于在父类初始化前准备好依赖属性。对 huggingface\_hub < 1.12 的用户无影响，对 >= 1.12 的用户修复了启动崩溃。无回归风险，所有现有测试（如 test/manual/models/test\_qwen3\_asr.py）已验证通过。
- 影响：影响范围仅限于 Qwen3-ASR 模型配置初始化。对所有使用 huggingface\_hub >= 1.12 启动 Qwen3-ASR 的用户是必要的修复；否则服务器无法启动。影响程度中等（阻止了关键功能）。
- 风险标记：依赖升级暴露问题，仅一行变更

## 关联脉络

- PR #22073 Qwen3-ASR support: 引入 Qwen3-ASR 功能，导致初始配置顺序问题
- PR #22731 Registered unittest for Qwen3-ASR: 为 Qwen3-ASR 添加注册单元测试，与本 PR 的验证相关