# PR #29393 完整报告

sgl-project/sglang

[Deps] Bump transformers to 5.12.1

合并时间：2026-06-30 15:54

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29393

## 执行摘要

- 一句话：升级 transformers 至 5.12.1 并清理兼容层
- 推荐动作：建议精读 mistral_utils.py 中的 adapter 模式，它展示了如何通过 monkey-patch 平滑适配上游 API 变化而不影响核心代码。同时 config.py 和 tokenizer.py 的清理贯彻了“移除过时 workaround”的好实践。对于计划升级依赖的开发者，本 PR 提供了完整的兼容性管理案例。

## 功能与动机

升级 transformers 从 5.8.1 至 5.12.1，使 nvidia/GLM-5.2-NVFP4 模型能开箱使用（PR body 原话）。上游已合入相关修正（transformers PR #46338），因此可以移除旧版本引入的临时补丁，降低维护成本。

## 实现拆解

1. 升级版本约束：在 pyproject.toml 中将 transformers 版本下限从 >=5.8.1 改为 >=5.12.1。
2. 移除 GLM MoE DSA 兼容代码：
   - config.py：删除 _is_legacy_glm_moe_dsa_layer_types_error 和 _load_glm_moe_dsa_config_without_legacy_layer_types 函数，以及 HfModelConfigParser.parse 中的 try-except 回退和 GlmMoeDsaConfig 字段恢复代码。直接调用 AutoConfig.from_pretrained。
   - tokenizer.py：删除 _retry_auto_tokenizer_with_glm_moe_dsa_config 及其在 _auto_tokenizer_from_pretrained 和 _resolve_tokenizers_backend 中的调用，简化异常处理。
3. 增强 MistralCommon tokenizer 适配（mistral_utils.py）：
   - 新增 _safe_add_special_tokens：当只设置 pad_token 时直接赋值，绕过新版 add_special_tokens 的校验。
   - 新增 _text_to_ids_with_pixtral_markers：手动扫描 [IMG]、[IMG_BREAK]、[IMG_END] 标记并分段调用 _text_to_ids，解决新版无法直接识别的问题。
   - 将旧的 _safe_apply_chat_template 拆分为 _adapt_placeholder_content_for_mistral_common 和 _adapt_placeholder_messages_for_mistral_common，以兼容新版 chat template 的 content 结构。

4. 修复 VL 模型 rotary 调用：在 mimo_vl.py、moss_vl.py、qwen2_5_vl.py 中，将原来直接传入 max_grid_size（Tensor）改为先转为 int，再用 torch.arange 在正确设备上创建 position_ids，适配 transformers 5.12 的新接口要求。
5. 其他适配：调整 multimodal_gen 下编码器的少量导入和配置（minicpmo.py 导入调整，qwen3vl.py、mistral_3.py、qwen2_5vl.py 配置键变更）。
6. 测试配套：更新 engine.py 中的导入和断言，匹配新版 transformers 行为。CI 配置触发了 run-ci 和 run-ci-extra 覆盖测试。

关键文件：

- python/sglang/srt/utils/hf_transformers/mistral_utils.py（模块 Mistral 工具；类别 source；类型 core-logic；符号 _safe_add_special_tokens, _text_to_ids_with_pixtral_markers, _adapt_placeholder_content_for_mistral_common, _adapt_placeholder_messages_for_mistral_common）：核心兼容性修复文件，新增两个关键补丁函数并重写 chat template 适配，确保 MistralCommon tokenizer 在 transformers 5.12 下正常工作。
- python/sglang/srt/utils/hf_transformers/config.py（模块 配置解析；类别 source；类型 dependency-wiring；符号 _is_legacy_glm_moe_dsa_layer_types_error, _load_glm_moe_dsa_config_without_legacy_layer_types）：移除 GLM MoE DSA 配置回退代码，简化 HfModelConfigParser 实现，体现上游修复后的清理。
- python/sglang/srt/utils/hf_transformers/tokenizer.py（模块 分词器加载；类别 source；类型 dependency-wiring；符号 _retry_auto_tokenizer_with_glm_moe_dsa_config）：移除 _retry_auto_tokenizer_with_glm_moe_dsa_config 及相关异常捕获，简化 tokenizer 加载逻辑。
- python/sglang/srt/models/mimo_vl.py（模块 多模态模型；类别 source；类型 data-contract；符号 rot_pos_emb）：调整 rot_pos_emb 方法以适配 transformers 5.12 的 rotary 接口变化，代表多个 VL 模型的同步修改。
- python/pyproject.toml（模块 项目配置；类别 config；类型 configuration）：版本依赖约束变更入口，触发整个升级。

关键符号：_safe_add_special_tokens, _text_to_ids_with_pixtral_markers, _adapt_placeholder_content_for_mistral_common, _adapt_placeholder_messages_for_mistral_common, rot_pos_emb

## 关键源码片段

### python/sglang/srt/utils/hf_transformers/mistral_utils.py

核心兼容性修复文件，新增两个关键补丁函数并重写 chat template 适配，确保 MistralCommon tokenizer 在 transformers 5.12 下正常工作。

```
# Keep the old no-op pad add working on transformers 5.12 MistralCommon.
_orig_add_special_tokens = tokenizer.add_special_tokens

def _safe_add_special_tokens(special_tokens_dict, *args, **kwargs):
    """如果字典仅包含 'pad_token'，直接设置并返回 0，避免底层校验失败。"""
```

```
    if set(special_tokens_dict) == {"pad_token"}:
        tokenizer.pad_token = special_tokens_dict["pad_token"]
        return 0
    return _orig_add_special_tokens(special_tokens_dict, *args, **kwargs)

tokenizer.add_special_tokens = _safe_add_special_tokens

if hasattr(tokenizer, "_text_to_ids"):
    _orig_text_to_ids = tokenizer._text_to_ids
    # 缓存 marker 对应的 token id，避免重复查询。
    marker_to_id = {
        "[IMG]": tokenizer.convert_tokens_to_ids("[IMG]"),
        "[IMG_BREAK]": tokenizer.convert_tokens_to_ids("[IMG_BREAK]"),
        "[IMG_END]": tokenizer.convert_tokens_to_ids("[IMG_END]"),
    }

    def _text_to_ids_with_pixtral_markers(text, add_special_tokens):
        """手动扫描 Pixtral marker 并分段调用 _orig_text_to_ids。
        transformers 5.12 的 MistralCommon 无法直接处理这些 marker，
        因此需要先按 marker 位置拆分文本，依次转换后拼接。
        """
        if not isinstance(text, str) or not any(
            marker in text for marker in marker_to_id
        ):
            return _orig_text_to_ids(text, add_special_tokens)

        ids = []
        pos = 0
        while pos < len(text):
            # 寻找当前剩余文本中最早出现的 marker。
            next_marker = None
            next_idx = len(text)
            for marker in marker_to_id:
                marker_idx = text.find(marker, pos)
                if marker_idx != -1 and marker_idx < next_idx:
                    next_marker = marker
                    next_idx = marker_idx

            if next_marker is None:
                # 无更多 marker，处理剩余纯文本。
                ids.extend(_orig_text_to_ids(text[pos:], False))
                break
            if next_idx > pos:
                # 处理 marker 前的文本。
                ids.extend(_orig_text_to_ids(text[pos:next_idx], False))
            ids.append(marker_to_id[next_marker])
            pos = next_idx + len(next_marker)

        if add_special_tokens:
```

```
            return tokenizer.build_inputs_with_special_tokens(ids)
        return ids

    tokenizer._text_to_ids = _text_to_ids_with_pixtral_markers
```

### python/sglang/srt/utils/hf_transformers/config.py

移除 GLM MoE DSA 配置回退代码，简化 HfModelConfigParser 实现，体现上游修复后的清理。

```
@register_model_config_parser("hf")
class HfModelConfigParser(ModelConfigParserBase):
    def parse(self, model, trust_remote_code, revision=None, **kwargs):
        # transformers >= 5.10 已修复 GlmMoeDsaConfig 的兼容性问题，
        # 因此直接调用 AutoConfig.from_pretrained，无需 try-except 回退。
        config = AutoConfig.from_pretrained(
            model,
            trust_remote_code=trust_remote_code,
            revision=revision,
            **kwargs,
        )

        # 以下为遗留的模型特定修复（Phi4MM、Longcat 等），保持不变。
        if config.architectures is not None and config.architectures[0] == "Phi4MMForCausalLM":
            from transformers import SiglipVisionConfig
            config.vision_config = SiglipVisionConfig(
                hidden_size=1152, image_size=448, intermediate_size=4304,
                model_type="siglip_vision_model", num_attention_heads=16,
                num_hidden_layers=26, patch_size=14,
            )
        # ... 其他 overrides ...
        return config
```

## 评论区精华

本 PR 无公开 review 讨论。维护者 Fridge003 仅批准并注明“Can be merged after passing CI”。整体决策明确直接，无争议。

- 暂无高价值评论线程

## 风险与影响

- 风险：
  1. 依赖升级导致其他不兼容：transformers 5.12 可能引入未被覆盖的 API 变更，影响未被测试覆盖的模型。
  2. 移除回退逻辑的回归风险：若 upstream 修复不完整，某些 GLM MoE DSA 模型加载可能失败。
  3. MistralCommon 补丁覆盖不全：新补丁可能未处理音频 / 视频 placeholder 等 edge case。

4. VL 模型 rotary 变更：若设备推断错误可能导致 CUDA 错误。
5. 多文件同步修改：回归面较广，需依赖 CI 测试保障。 - 影响：用户：升级后所有用户自动使用 transformers 5.12.1，可开箱使用 GLM-5.2-NVFP4 模型；Pixtral/VL 用户受益于更完善的 tokenizer 适配。系统：依赖版本提升，需更新环境；未来升级需持续关注上游变更。团队：清理了历史 workaround，降低维护负担；但 mistral_utils.py 中的补丁仍为临时方案，需跟踪 upstream 是否原生支持。 - 风险标记：核心依赖升级，兼容层重构，多模型耦合

## 关联脉络

- PR #29686 Update GLM tests to 5.2 and delete redundant tests: 同样涉及 GLM 模型支持，PR 29686 将 GLM 测试升级到 5.2，与本 PR 的 transformers 升级共同支撑 GLM-5.2-NVFP4 的开箱可用。
- PR #29376 Fix test_type_based_dispatcher after TokenizedGenerateReqInput field changes: 同仓库中的测试修复 PR，说明近期有较多的字段变更，本 PR 的部分模型修改可能需要类似适配。