

PR #56554 完整报告

vllm-project/vllm

[DSV4.1] Remove compressor-aware image sentinel token padding

合并时间: 2026-09-12 15:07

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56554>

执行摘要

- 一句话: 移除 DSV4.1 误移植的 compressor 对齐 padding, token 流对齐 HF 参考
- 推荐动作: 值得精读。该 PR 是“参考实现比对驱动删除”的小而清晰案例: 核心看点是 `_apply_token_matches_with_placeholders` 覆写的删除动机——pad 依赖拼接位置导致 prompt updates 无法缓存, 删除后不仅 token 流对齐官方, 还顺带恢复基类默认路径。对多模态模型接入团队有参考价值: 跨模型版本移植预处理逻辑前应逐行比对官方 `image_processor.py`。建议后续补充一条针对 DSV4.1 图片 token 流 bit-match 的回归测试, 并核查 `mm_prefix_span_leading_pad_modulus` 是否还有其他读取方。

功能与动机

PR body 明确指出 Image sentinel padding only lives in deepseek-v4-vision-exp, not deepseek-v4.1, 并以官方仓库 `image_processor.py` 的具体行号作为证据 (v4-vision-exp 的 L135-L155 与 v4.1 的 L140-L173)。vLLM 此前从 v4-vision-exp 移植了 compressor-alignment padding 到 v4.1, 这属于跨模型版本的误共享: v4.1 参考实现允许图片 token 自由跨越 compressor 组边界, 不做任何相位对齐。保留该 pad 会让 vLLM 产出的图片 token 流与官方预处理不一致, 进而影响路由、engram 掩码等依赖图片 sentinel 位置的下游逻辑。

实现拆解

1. 常量与掩码精简 (`common/mm_preprocess.py`, +10/-110): 删除 `COMPRESS_PAD_TO = 2`、`IMAGE_PAD_ID = 129265`、`IMAGE_PAD_TOKEN_NAME` 三个常量; `image_sentinel_mask` 从双 id 比较退化为 `token_ids == IMAGE_SENTINEL_BASE_ID` 单 id 比较; `validate_image_sentinel_ids` 删除对 pad token 的 id 校验。这保证后续以 sentinel 掩码为基准的逻辑 (MoE router 图片 token 识别、engram 掩码) 不再感知 pad。
2. splice 行为恢复基类: 删除自定义 `_apply_token_matches_with_placeholders` 覆写 (约 67 行)。原覆写在图片块拼接时按 `COMPRESS_PAD_TO - 1 - start_idx % COMPRESS_PAD_TO` 前插 pad, 且因 pad 依赖块的最终位置而无法缓存 prompt updates; 删除后恢复基类 `_plan_prompt_updates` 默认拼接, `PlaceholderFeaturesInfo` 的 token 位置与无 pad 内容天然对齐。
3. 预算与 token 上限恢复: `safe_resize` 不再预留 `COMPRESS_PAD_TO - 1` 个 token; `get_mm_max_tokens_per_item` 直接返回 `vision_max_n_token` 而非加 pad 余量;

`get_image_size_with_most_features` 的预算同样恢复为 `vision_max_n_token`。最大允许图片 token 数回落为配置值，预算临界处的图片尺寸选择与官方一致。

4. 模型侧清理 (`amd/vl_model.py`、`nvidia/vl_model.py`，各 -8)：两文件同步删除 `IMAGE_PAD_ID`、`IMAGE_SENTINEL_BASE_ID` 导入，以及 `embed_input_ids` 中把 `IMAGE_PAD_ID` 通过 `masked_fill` 映射为 `IMAGE_SENTINEL_BASE_ID` 的兜底逻辑。该兜底原先用于让 pad 位置以普通图片 token 嵌入 (v4.1 无 `image_pad` 向量)；pad 不复存在后 `input_ids` 直接送入 `embed_input_ids`。
5. 配置属性清理 (`vllm/transformers_utils/configs/deepseek_v41.py`，-6)：删除 `mm_prefix_span_leading_pad_modulus = 2` 属性。该属性描述 mm-prefix 可见区间前导 pad 的周期；删除后 `is_mm_prefix_lm` 与 `mm_prefix_clamp_sliding_window` 继续保留，mm-prefix 管道行为不受影响。
6. 测试与配套：本次没有测试文件变更，PR 正文 Test Plan 与 Test Result 均为 TODO；无配置模板、部署或文档配套改动。回归保障依赖现有 DSV4.1 相关 CI 用例与维护者人工核对。

关键文件：

- `vllm/models/deepseek_v4_1/common/mm_preprocess.py` (模块 多模态预处理；类别 source；类型 data-contract；符号 `_apply_token_matches_with_placeholders`, `image_sentinel_mask`, `validate_image_sentinel_ids`, `safe_resize`)：核心文件：删除 `COMPRESS_PAD_TO`、`IMAGE_PAD_ID` 常量、`IMAGE_PAD_ID` 掩码及自定义 `_apply_token_matches_with_placeholders` 覆写，图片 token 流与 HF 参考逐位对齐 (+10/-110)。
- `vllm/models/deepseek_v4_1/amd/vl_model.py` (模块 视觉模型；类别 source；类型 data-contract；符号 `embed_input_ids`)：AMD 视觉路径同步清理：移除 `IMAGE_PAD_ID` 导入与 `embed_input_ids` 中的 `masked_fill` 兜底 (-8 行)。
- `vllm/models/deepseek_v4_1/nvidia/vl_model.py` (模块 视觉模型；类别 source；类型 data-contract；符号 `embed_input_ids`)：NVIDIA 视觉路径与 AMD 侧完全同步的清理，删除同一段 `masked_fill` 兜底与导入 (-8 行)。
- `vllm/transformers_utils/configs/deepseek_v41.py` (模块 模型配置；类别 source；类型 core-logic；符号 `DeepseekV41Config`)：配置类删除 `mm_prefix_span_leading_pad_modulus` 属性，mm-prefix 可见区间不再描述前导 pad 周期 (-6 行)。

关键符号：`_apply_token_matches_with_placeholders`, `image_sentinel_mask`, `validate_image_sentinel_ids`, `safe_resize`, `get_mm_max_tokens_per_item`, `get_image_size_with_most_features`, `embed_input_ids`

关键源码片段

`vllm/models/deepseek_v4_1/common/mm_preprocess.py`

核心文件：删除 `COMPRESS_PAD_TO`、`IMAGE_PAD_ID` 常量、`IMAGE_PAD_ID` 掩码及自定义 `_apply_token_matches_with_placeholders` 覆写，图片 token 流与 HF 参考逐位对齐 (+10/-110)。

```
# DeepSeek-V4.1 视觉输入的 sentinel 相关常量与校验 (head 状态) 。
# 移除 compressor-alignment padding 后, token 流与官方 image_processor.py
# 完全一致: 官方参考实现允许图片 token 跨 compressor 组边界自由拼接,
# 不做相位对齐; 对齐 pad 是 deepseek-v4-vision-exp 才有的逻辑。
```

```
IMAGE_START, IMAGE, IMAGE_NEW_LINE, IMAGE_END = range(4)
```

```
IMAGE_PLACEHOLDER = "< | deepseek_image | >"
```

```
# checkpoint tokenizer 中 IMAGE_PLACEHOLDER 的 id (即配置里的 image_token_id) 。
# 每个 image-span 位置都携带该 id, MoE router 以区间
# [IMAGE_SENTINEL_BASE_ID, IMAGE_SENTINEL_BASE_ID + 5) 识别图片 token。
# 原实现曾额外引入 IMAGE_PAD_ID = 129265 (<|place_holder_mm_span_0436|>)
# 作为对齐 pad 的占位 id, 本 PR 已删除。
IMAGE_SENTINEL_BASE_ID = 129264
```

```
def image_sentinel_mask(token_ids: torch.Tensor) -> torch.Tensor:
```

```
    """返回 image-span 位置的布尔掩码。"""
    # 不再需要叠加 IMAGE_PAD_ID 判断, 掩码退化为单 id 比较,
    # 路由与 engram 掩码路径的语义随之简化。
    return token_ids == IMAGE_SENTINEL_BASE_ID
```

```
def validate_image_sentinel_ids(tokenizer) -> None:
```

```
    """校验图片 token id 与 tokenizer 是否一致。"""
    # 原先还会校验 IMAGE_PAD_TOKEN_NAME 对应的 id,
    # padding 移除后该项校验一并删除, 只保留图片占位符校验。
    image_id = tokenizer.convert_tokens_to_ids(IMAGE_PLACEHOLDER)
    if image_id != IMAGE_SENTINEL_BASE_ID:
        raise ValueError(
            f"Image placeholder {IMAGE_PLACEHOLDER!r} has id {image_id}, "
            f"expected {IMAGE_SENTINEL_BASE_ID} (the config's "
            "image_token_id); the DeepSeek-V4.1 vision path keys image "
            "routing and engram masking off this id."
        )
```

```
def safe_resize(
```

```
    height, width, best_height, best_width, patch_size, downsample_ratio, max_n_token
):
```

```
    """收缩图片尺寸, 直到图片 LLM token 消耗不超过 max_n_token。"""
    # 原先会先扣除 COMPRESS_PAD_TO - 1 个预留 token 再比较,
    # 现已移除, 预算判定与官方参考完全一致。
    n_llm_h, n_llm_w = llm_grid(best_height, best_width, patch_size, downsample_ratio)
    if num_image_tokens(n_llm_h, n_llm_w) > max_n_token:
        best_height, best_width = solve_resize_ratio(
            height, width, patch_size, downsample_ratio, max_n_token
        )
```

```

n_llm_h, n_llm_w = llm_grid(
    best_height, best_width, patch_size, downsample_ratio
)
assert num_image_tokens(n_llm_h, n_llm_w) <= max_n_token
return n_llm_h, n_llm_w, best_height, best_width

```

原 _apply_token_matches_with_placeholders 覆写（已整体删除）会在拼接图片块时
前插 COMPRESS_PAD_TO - 1 - start_idx % COMPRESS_PAD_TO 个 IMAGE_PAD_ID;
因 v4.1 官方 image_processor.py 无此 pad, 现恢复基类 _plan_prompt_updates 的
默认拼接, PlaceholderFeaturesInfo 的 token 位置与无 pad 内容天然对齐。

vllm/models/deepseek_v4_1/amd/vl_model.py

AMD 视觉路径同步清理：移除 IMAGE_PAD_ID 导入与 embed_input_ids 中的 masked_fill 兜底（-8 行）。

```

def embed_input_ids(
    self,
    input_ids: torch.Tensor,
    multimodal_embeddings: MultiModalEmbeddings | None = None,
    *,
    is_multimodal: torch.Tensor | None = None,
) -> torch.Tensor:
    from vllm.model_executor.models.utils import _merge_multimodal_embeddings

    # v4.1 参考实现不插入 compressor-alignment pad, token 流与 HF 完全一致,
    # 因此不再需要把 IMAGE_PAD_ID 掩码成 IMAGE_SENTINEL_BASE_ID 的兜底;
    # 原先的 masked_fill 是 Branch-free 的编译区安全写法, 现已整体移除。
    inputs_embeds = self.language_model.embed_input_ids(input_ids)

    if multimodal_embeddings is None or len(multimodal_embeddings) == 0:
        return inputs_embeds

    assert is_multimodal is not None
    return _merge_multimodal_embeddings(
        inputs_embeds=inputs_embeds,
        multimodal_embeddings=multimodal_embeddings,
        is_multimodal=is_multimodal,
    )

```

vllm/transformers_utils/configs/deepseek_v41.py

配置类删除 mm_prefix_span_leading_pad_modulus 属性, mm-prefix 可见区间不再描述前导 pad 周期（-6 行）。

```

# DeepseekV41Config.__init__ 尾部（head 状态）。
# mm-prefix 相关属性中, 原实现额外设置了
# mm_prefix_span_leading_pad_modulus = 2 用于描述 compressor-alignment pad
# 的周期; padding 移除后该属性不复存在, 其余 mm-prefix 属性保留。
vision_n_layers = vision_config.get("num_hidden_layers", 0)
self.vision_n_layers = vision_n_layers

```

```
# 视觉变体需要 mm-prefix 管道（原子 image-span prefill + 稀疏 SWA 扩窗），  
# 基础架构配置转换器读取该配置级属性。  
self.is_mm_prefix_lm = vision_n_layers > 0  
# 稀疏 SWA 索引内核会在 image-span 内原地扩窗，因此超过 sliding_window  
# 的 mm-prefix 区间必须保留（仅这些内核消费它们）。  
self.mm_prefix_clamp_sliding_window = vision_n_layers > 0  
# mm_prefix_span_leading_pad_modulus 已删除：v4.1 参考实现不插入前导 pad。
```

评论区精华

该 PR 没有实质 review 讨论线程。claude[bot] 自动评论指出 This pull request is from a fork — automated review is disabled，即 fork PR 未触发自动评审；维护者 ywang96 直接 APPROVE 且未留文字评论。PR body 中对 HF 官方 image_processor.py 行号（v4-vision-exp L135-L155 与 v4.1 L140-L173）的引用是唯一设计论据，表明这是一次基于参考实现比对的“删除误移植逻辑”决策。未解决的疑虑主要是测试证据缺口：无配套测试、Test Plan 为 TODO，以及 `mm_prefix_span_leading_pad_modulus` 属性删除后没有显式说明是否还有其他消费方。

- fork PR 自动评审被禁用 (other): 未触发人工 bot 评审，无实际技术讨论产生。
- 维护者批准与测试证据缺失 (testing): 维护者基于 HF 参考代码行号证据直接批准；测试证据缺口未被讨论，遗留回归风险。

风险与影响

- 风险：1) 行为回归：移除 pad 后图片 token 流与上一版 vLLM 行为不同，每个图片块最多少 1 个 pad token；若模型实际依赖相位对齐将影响生成质量——依据 HF 参考代码判断官方不做 pad，方向风险较低但未被自动化测试显式验证。2) 配置属性兼容性：`mm_prefix_span_leading_pad_modulus` 被删除，若 vllm/v1 的 mm-prefix 或 SWA 相关模块直接读取该属性，将触发 `AttributeError`；本 PR 未改动其他消费者，存在隐含耦合风险。3) 测试缺失：`image_sentinel_mask` 与 `validate_image_sentinel_ids` 语义变化（掩码范围缩小）会影响 MoE router 图片 token 识别与 engram 掩码路径，缺少对应回归测试。4) 双路径一致性：AMD 与 NVIDIA 两份 `vl_model.py` 逻辑需保持同步，本次同步删除，后续单侧修改容易漂移。
- 影响：对用户：使用 DeepSeek-V4.1 视觉能力的输出 token 序列与旧版 vLLM 不同（更贴近官方参考），图片 token 计数与生成长度可能变化，属预期内的保真度修正。对系统：影响 DSV4.1 的 mm-prefix 拼接、图片 sentinel 掩码、MoE 路由与 engram 掩码链路，同时覆盖 AMD 与 NVIDIA 两条硬件路径。对团队：消除了 v4-vision-exp 逻辑被误共享到 v4.1 的隐患，为后续 DSV4.1 视觉能力迭代提供正确基线；但没有配套测试，后续改动需自行补回归。整体影响集中在单模型家族（DeepSeek-V4.1），程度中等。
- 风险标记：token 流行为变更，无配套测试，配置属性删除需核查消费者，AMD/NVIDIA 双路径同步改动

关联脉络

- PR #56503 [ROCm][DSV4.1][Perf] Use AITER mHC for the delayed pre block: 同属 DeepSeek-V4.1 模型家族开发线，该 PR 在 ROCm 侧推进 V4.1 内核性能，本 PR 修正 V4.1 视觉 token 流，两者共同构成 V4.1 支持闭环。
- PR #56463 [XPU][CI] skip DeepSeek-V4.1-Flash in test_tensor_schema.py: 同为 DeepSeek-V4.1-Flash 相关配套改动，反映 V4.1 系列在 CI 与模型路径上的持续适配。
- PR #56153 [ROCm][Kernel][DSV4] Remove tl.constexpr to avoid cold-compile churn in indexer gather kernel: 同属 DSV4/DSV4.1 架构演进，关注图片 token 路由 (indexer) 路径的执行效率，与本 PR 的 sentinel 路由语义调整相邻。