PR #42654 完整报告

vllm-project/vllm

[Model] Openvla support

合并时间：2026-05-19 23:17

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/42654

执行摘要

- 一句话：新增 OpenVLA 模型支持
- 推荐动作：值得精读 openvla.py 和 processors/openvla.py，理解如何处理无法直接复用 HF remote code 的模型移植。关注 PrismaticVisionBackbone 中 timm 模型的加载方式以及 weight loading 的适配。通过此 PR 可学习 vLLM 多模态模型的接入模式（ProcessingInfo、PromptInsertion、TensorSchema 等）。

功能与动机

OpenVLA 是一个重要的机器人基础模型，但它的 Hugging Face remote code 依赖老版本 transformers/timm，无法直接在 vLLM 中运行。本 PR 作为 issue #42100 的子目标，从头实现 OpenVLA 全链路，使 vLLM 能够支持该模型进行动作预测。

实现拆解

1. 添加配置文件 vllm/transformers_utils/configs/openvla.py，定义 OpenVLAConfig 类继承 PretrainedConfig，处理嵌套的 text_config 字典，避免执行 Hugging Face remote code。
2. 实现处理器 vllm/transformers_utils/processors/openvla.py，提供 to_rgb_image、preprocess_openvla_image、OpenVLAImageProcessor 和 OpenVLAProcessor，完成 OpenVLA 特有的 6 通道图像预处理（分别用 ImageNet 和 SigLIP 归一化后拼接）。
3. 构建模型架构：PrismaticVisionBackbone（fused DINOv2+SigLIP，输出 2176 维）、PrismaticProjector（3 层 MLP，映射到 4096 维）、OpenVLAForActionPrediction 主模型集成 Llama-2-7B，通过 embed_multimodal 将图像特征插入到 BOS 后，执行语言模型前向得到动作 token。
4. 注册模型：在 vllm/model_executor/models/registry.py 中添加映射；在 vllm/transformers_utils/configs/__init__.py 和 processors/__init__.py 中 hook 配置类和处理器类。
5. 添加单元测试：tests/models/multimodal/processing/test_openvla.py 覆盖预处理、配置转换、处理器输出形状等，标记为 CPU 测试。
6. 更新文档：在 docs/models/supported_models.md 中添加 OpenVLA 条目。

关键文件：

- vllm/model_executor/models/openvla.py（模块 模型层；类别 source；类型 core-logic；符号 _get_num_image_tokens, OpenVLAImagePixelInputs, PrismaticVisionBackbone, init）：OpenVLA 主模型文件，包含完整的模型架构定义、多模态嵌入逻辑和权重加载实现。
- vllm/transformers_utils/processors/openvla.py（模块 处理器；类别 source；类型 dependency-wiring；符号 to_rgb_image, preprocess_openvla_image, OpenVLAImageProcessor, init）：独立的图像处理器，实现 OpenVLA 特有的 6 通道预处理（分别执行 DINOv2 和 SigLIP 归一化）。
- vllm/transformers_utils/configs/openvla.py（模块 配置器；类别 source；类型 core-logic；符号 OpenVLAConfig, init）：定义 OpenVLA 配置类，避免执行 Hugging Face remote code。
- vllm/model_executor/models/registry.py（模块 注册表；类别 source；类型 data-contract）：将 OpenVLAForActionPrediction 注册到模型注册表，使 vLLM 能识别该模型。
- tests/models/multimodal/processing/test_openvla.py（模块 测试；类别 test；类型 test-coverage；符号 _FakeTokenizer, encode, call, _FakeProcessingInfo）：提供 OpenVLA 处理器和配置的单元测试，覆盖预处理、配置转换和处理器输出。
- vllm/transformers_utils/configs/__init__.py（模块 配置器；类别 source；类型 core-logic）：导出 OpenVLAConfig，使配置系统能够找到并实例化该类。

关键符号：_get_num_image_tokens, to_rgb_image, preprocess_openvla_image, OpenVLAImageProcessor.call, OpenVLAProcessor.init, PrismaticVisionBackbone.init, PrismaticVisionBackbone.forward, PrismaticProjector.init, PrismaticProjector.forward, OpenVLAForActionPrediction.init, OpenVLAForActionPrediction.forward, OpenVLAForActionPrediction.load_weights, OpenVLAMultiModalProcessor.init, OpenVLAProcessingInfo.get_hf_config

关键源码片段

vllm/model_executor/models/openvla.py

OpenVLA 主模型文件，包含完整的模型架构定义、多模态嵌入逻辑和权重加载实现。

```
# SPDX-License-Identifier: Apache-2.0

class PrismaticVisionBackbone(nn.Module):
    """OpenVLA 的 fused DINOv2 + SigLIP 视觉主干。"""

    def __init__(
        self,
        *,
        image_sizes: Sequence[int],
        timm_model_ids: Sequence[str],
        timm_override_act_layers: Sequence[str | None],
        use_fused_vision_backbone: bool,
    ) -> None:
        super().__init__()
```

```
# 当前仅支持 fused backbone
if not use_fused_vision_backbone:
    raise ValueError(
        "OpenVLA currently supports only the fused DINOv2 + SigLIP "
        "vision backbone."
    )
# 图像尺寸固定为 224x224
if tuple(image_sizes) != _OPENVLA_IMAGE_SIZES:
    raise ValueError(
        "OpenVLA currently supports only 224x224 image inputs, "
        f"got image_sizes={list(image_sizes)}."
    )
# timm 模型 ID 固定
if tuple(timm_model_ids) != _OPENVLA_TIMM_MODEL_IDS:
    raise ValueError(
        "Only dinosiglip-vit-so-224px backbone is supported."
    )
# 激活层覆盖固定为 None
if tuple(timm_override_act_layers) != _OPENVLA_TIMM_OVERRIDE_ACT_LAYERS:
    raise ValueError(
        "Only default timm activation layers are supported."
    )

self.image_size = image_sizes[0]
self.use_fused_vision_backbone = use_fused_vision_backbone
# 融合后的视觉特征维度
self.embed_dim = 2176 if use_fused_vision_backbone else 1024

try:
    import timm
except ImportError:
    raise ImportError(
        "Please install timm to use OpenVLA. OpenVLA verification "
        "used timm==0.9.10."
    )

# 加载 DINOv2 和 SigLIP 模型（不加载预训练权重，由 weight loader 后续加载）
self.dinov2_featurizer = timm.create_model(
    timm_model_ids[0],
    pretrained=False,
    num_classes=0,
    img_size=self.image_size,
    act_layer=timm_override_act_layers[0],
)
self.siglip_featurizer = (
    timm.create_model(
        timm_model_ids[1],
        pretrained=False,
        num_classes=0,
```

```
                img_size=self.image_size,
                act_layer=timm_override_act_layers[1],
            )
        )

    def forward(self, pixel_values: torch.Tensor) -> torch.Tensor:
        # pixel_values: [batch, 6, 224, 224] — 前 3 通道 DINOv2 归一化，后 3 通道 SigLIP 归一化
        dinov2_input = pixel_values[:, :3, :, :]
        siglip_input = pixel_values[:, 3:6, :, :]

        # DINOv2 输出包含 prefix tokens（1 CLS + 4 register），需要切片移除
        dinov2_features = self.dinov2_featurizer(dinov2_input)
        # 假设 dinov2_features 形状为 [batch, 261, 1024]（14x14+5），取 patch tokens
        dinov2_features = dinov2_features[:, 5:, :] # 现在为 [batch, 256, 1024]

        # SigLIP 输出已是 patch tokens
        siglip_features = self.siglip_featurizer(siglip_input)
        # siglip_features 形状为 [batch, 256, 1152]

        fused = torch.cat([dinov2_features, siglip_features], dim=-1) # [batch, 256, 2176]
        return fused
```

vllm/transformers_utils/processors/openvla.py

独立的图像处理器，实现 OpenVLA 特有的 6 通道预处理（分别执行 DINOv2 和 SigLIP 归一化）。

```
# SPDX-License-Identifier: Apache-2.0

import numpy as np
import torch
from PIL import Image

# 归一化参数：DINOv2 使用 ImageNet 统计，SigLIP 使用 [0.5, 0.5, 0.5]
IMAGENET_MEAN = np.array([0.484375, 0.455078125, 0.40625], dtype=np.float32)
IMAGENET_STD = np.array([0.228515625, 0.2236328125, 0.224609375], dtype=np.float32)
SIGLIP_MEAN = np.array([0.5, 0.5, 0.5], dtype=np.float32)
SIGLIP_STD = np.array([0.5, 0.5, 0.5], dtype=np.float32)


def preprocess_openvla_image(image: Any, image_size: int) -> torch.Tensor:
    """将输入图像处理为 6 通道张量：前 3 通道为 DINOv2 归一化，后 3 通道为 SigLIP 归一化。"""
    rgb_image = to_rgb_image(image)
    rgb_image = rgb_image.resize(
        (image_size, image_size),
        Image.Resampling.BICUBIC,
    )

    raw = np.asarray(rgb_image, dtype=np.float32) / 255.0
    dinov2_pixels = ((raw - IMAGENET_MEAN) / IMAGENET_STD).transpose(2, 0, 1)
```

```
    siglip_pixels = ((raw - SIGLIP_MEAN) / SIGLIP_STD).transpose(2, 0, 1)
    pixel_values = np.concatenate([dinov2_pixels, siglip_pixels], axis=0)
    return torch.from_numpy(pixel_values)


class OpenVLAImageProcessor:
    """轻量级图像处理器，支持批量输入。"""

    def __init__(self, *, image_size: int) -> None:
        self.image_size = image_size

    def __call__(
        self,
        images: Any | None = None,
        **kwargs: object,
    ) -> dict[str, object]:
        if images is None:
            return {}
        if not isinstance(images, Sequence) or isinstance(images, (str, bytes)):
            images = [images]
        if len(images) == 0:
            return {}

        pixel_values = torch.stack(
            [
                preprocess_openvla_image(image, image_size=self.image_size)
                for image in images
            ],
            dim=0,
        )
        return {"pixel_values": pixel_values}
```

评论区精华

- Gemini Code Assist 指出了三个关键实现错误：DINOv2 输出未去除 prefix tokens（导致序列长度不匹配）；Prismatic 投影器的 intermediate_dim 应为 text_dim 而非 4 * vision_dim（否则权重形状不匹配）；forward 方法缺少 kv_caches 和 attn_metadata 参数，会导致运行时出错。这些已在作者后续提交中修复。
- DarkLight1337 建议将处理器代码从模型文件移动到 transformers_utils.processors，作者已执行；还建议使用 torchvision 预处理，作者参考 QwenVLProcessor 重构。
- 作者询问是否可以将 OpenVLAImagePixelInputs 和解析方法内联，DarkLight1337 解释保留类定义用于输入验证测试自动执行，因此保留原设计。
- 关于 tests/models/registry.py 中 _HfExamplesInfo 条目，作者最初加入但后来在 DarkLight1337 建议下移除，因为 CI 不会执行（依赖过时 transformers）。
- DINOv2 输出未去除 prefix tokens (correctness): 作者在后续提交中修复。
- Prismatic projector intermediate_dim 错误 (correctness): 作者修正。

- forward 方法缺少 kv_caches 和 attn_metadata (correctness): 作者修正。
- 将处理器代码移动到 transformers_utils.processors (design): 作者已执行重构。
- 保留 OpenVLAImagePixelInputs 类用于输入验证 (design): 保留原设计。
- 移除 registry 中 HfExamplesInfo 条目 (testing): 作者已移除。

风险与影响

- 风险：依赖风险：timm 作为运行时可选依赖，但版本需严格锁定为 0.9.10，否则视觉 tower 结果可能不一致。未在 CI 中自动测试。 数值稳定性：测试发现视觉 tower 前向存在数值漂移（mean diff 约 0.05），这可能影响生成 token 的完全复现（仅 6/10 完全匹配）。 模型限制：当前仅支持 fused DINOv2+SigLIP 变体，且图像尺寸固定 224x224，对于其他 OpenVLA 配置会直接报错。 兼容性：HF *remote code* 被完全绕过，未来 OpenVLA 上游更改可能需要同步更新 shim。
- 影响：用户影响：新增模型支持，用户可加载 openvla/openvla-7b 进行推理，但需自行安装 timm==0.9.10。 系统影响：注册表增加一条映射，无侵入核心路径；timm 为 lazy import，不增加默认依赖。 团队影响：多模态模型维护扩展，对机器人领域用户友好。
- 风险标记：timm 运行时差异，仅 fused backbone，HF *remote code* 不可用，生成 token 部分可复现

关联脉络

- 暂无明显关联 PR