PR #6318 完整报告

verl-project/verl

[megatron, cfg] feat: add Qwen3.5-35B Megatron-Bridge launch script on Ascend

合并时间：2026-05-19 20:20

原文链接：http://prhub.com.cn/verl-project/verl/pull/6318

执行摘要

- 一句话：新增 Qwen3.5-35B Megatron 训练脚本与 checkpoint strict 配置
- 推荐动作：建议合并此 PR，重点审查 checkpoint strict 默认值（True）是否适合当前 Bridge 版本，以及 NPU 端 expandable_segments 的异常处理是否足够安全。其他配置冗余问题已解决，代码质量可接受。

功能与动机

用户需要在 Ascend 硬件上使用 Megatron Bridge 对 Qwen3.5-35B 模型进行 GRPO 训练；现有配置不支持模型特定序列化需求和 NPU 内存设置。PR body 展示了 TensorBoard 训练曲线但未关联具体 issue。

实现拆解

1. 新增训练脚本：在 examples/grpo_trainer/run_qwen3_5_35b_megatron.sh 中添加 Ascend NPU 自动检测、并行配置（TP/PP/EP 等）、Megatron 覆盖配置（如 MoE 融合、优化器卸载、checkpoint strict=False）等。
2. 新增 strict 配置字段：在 CheckpointConfig 中加入 strict: bool = True，用于控制 bridge.save_hf_weights 时的严格键匹配。
3. 修复 transformer 配置序列化：在 megatron_checkpoint_manager.py 的 json.dump 中添加 default 参数，处理包含 to_dict 方法的自定义类（如 Qwen3.5 视觉模块）。
4. NPU 动态内存支持：在 device.py 的 set_expandable_segments 中添加 torch.npu.memory._set_allocator_settings 调用并捕获异常，避免内存配置失败导致训练崩溃。
5. 同步更新配置模板：在 actor.yaml、_generated_ppo_megatron_trainer.yaml、_generated_ppo_torchtitan_trainer.yaml 等配置文件中添加 strict: true 条目。

关键文件：

- examples/grpo_trainer/run_qwen3_5_35b_megatron.sh（模块 示例脚本；类别 other；类型 core-logic）：核心训练脚本，支持 Qwen3.5-35B 在 Ascend 和 GPU 上的 GRPO 训练，包含自动设备检测、并行配置和 MoE 优化设置。
- verl/utils/checkpoint/megatron_checkpoint_manager.py（模块 检查点；类别 source；类型 core-logic；符号 save_checkpoint）：核心 checkpoint 保存逻辑修改：添加 strict 参数传递和自定义 json 序列化处理。

- verl/utils/device.py（模块 设备层；类别 source；类型 core-logic；符号 set_expandable_segments）：添加 NPU expandable_segments 支持，确保 Ascend 场景下动态内存配置正确。
- verl/trainer/config/config.py（模块 配置层；类别 source；类型 core-logic；符号 CheckpointConfig）：在 CheckpointConfig 中新增 strict 字段定义。
- verl/trainer/config/actor/actor.yaml（模块 配置层；类别 config；类型 configuration）：在 actor 配置的 checkpoint 部分添加 strict 参数。
- verl/trainer/config/_generated_ppo_megatron_trainer.yaml（模块 配置层；类别 config；类型 configuration）：自动生成的 Megatron 配置模板同步添加 strict 配置。
- verl/trainer/config/_generated_ppo_torchtitan_trainer.yaml（模块 配置层；类别 config；类型 configuration）：负责 TorchTitan 后端的配置模板同步更新。

关键符号：save_checkpoint, set_expandable_segments, CheckpointConfig

关键源码片段

examples/grpo_trainer/run_qwen3_5_35b_megatron.sh

核心训练脚本，支持 Qwen3.5-35B 在 Ascend 和 GPU 上的 GRPO 训练，包含自动设备检测、并行配置和 MoE 优化设置。

```
# ...（脚本头部注释）
# 自动检测设备类型
DEVICE=${DEVICE:-$(python3 -c 'import torch_npu' 2>/dev/null && echo npu || echo gpu)}
case "${DEVICE}" in
  gpu)
    TP=${TP:-2}; PP=${PP:-1}; EP=${EP:-8}
    n_devices_per_node=${NDEVICES_PER_NODE:-8}
    ;;
  npu)
    TP=${TP:-2}; PP=${PP:-2}; EP=${EP:-8}
    n_devices_per_node=${NDEVICES_PER_NODE:-16} # 8 physical cards * 2 dies
    ;;
esac

# Megatron 覆盖配置（针对 Ascend 优化）
ACTOR+=(
  actor_rollout_ref.actor.megatron.vanilla_mbridge=False
  actor_rollout_ref.actor.checkpoint.strict=False # 避免桥接权重键不匹配导致保存失败
  # MoE 与通信优化
  actor_rollout_ref.actor.megatron.override_transformer_config.moe_permute_fusion=True
  actor_rollout_ref.actor.megatron.override_transformer_config.moe_grouped_gemm=True
)
```

verl/utils/checkpoint/megatron_checkpoint_manager.py

核心 checkpoint 保存逻辑修改：添加 strict 参数传递和自定义 json 序列化处理。

```
def save_checkpoint(self, local_path: str, ...):
```

```
        # ... 其他逻辑
        if self.should_save_model and self.use_hf_checkpoint:
            # 使用 Megatron Bridge 保存 HF 格式权重
            if self.vanilla_bridge:
                # ...
            else:
                if self.peft_cls is not None:
                    # 保存 adapter
                else:
                    # 关键修改：传递 strict 参数（从 CheckpointConfig 读取）
                    self.bridge.save_hf_weights(
                        self.model, hf_ckpt_path,
                        strict=self.checkpoint_config.strict
                    )
        # 保存 transformer 配置（用于推理加载）
        if self.should_save_extra:
            # ... 清理不可序列化字段
            transformer_config_path = get_transformer_config_checkpoint_path(local_path)
            # NOTE: 使用 Megatron-Bridge 时，transformers >= 5.4.0 会出现循环导入问题
            with open(transformer_config_path, "w") as f:
                json.dump(
                    transformer_config_dict,
                    f,
                    indent=2,
                    default=lambda o: o.to_dict() if hasattr(o, "to_dict") else o # 适配 Qwen3Vision
                    自定义类
                )
```

verl/utils/device.py

添加 NPU expandable_segments 支持，确保 Ascend 场景下动态内存配置正确。

```
def set_expandable_segments(enable: bool) -> None:
    """
    启用或禁用 PyTorch 内存分配器的 expandable_segments 特性。
    此设置有助于减少显存碎片，在长序列训练中效果明显。
    """
    if is_cuda_available:
        torch.cuda.memory._set_allocator_settings(f"expandable_segments:{enable}")
        return # CUDA 分支会自动映射到 NPU（仅限 PyTorch 旧版本）

    # NPU 需要独立设置，因为 NPU 场景下 is_cuda_available 为 False
    if is_npu_available:
        try:
            torch.npu.memory._set_allocator_settings(
                f"expandable_segments:{enable}"
            )
        except Exception:
            # 静默捕获，避免设置失败导致训练中断
            # 若未正确关闭可能导致严重内存泄漏，建议监控
```

```
        pass
```

评论区精华

- device.py 分支必要性：wucong25 指出 torch.cuda 会自动调用 torch.npu 的 allocator 设置，建议删除 NPU 分支；Zhang1Sheng 解释 NPU 场景下 is_cuda_available 为 False，必须独立添加，最终保留。
- 删除 27B 脚本：wuxibin89 要求移除 run_qwen3_5_27b_megatron.sh，认为不需要为每个模型单独保留脚本，作者确认删除。
- 默认配置冗余：wucong25 建议删除 enable_chunked_prefill、use_flash_attn 等已默认开启的配置；作者说明 NPU 端需显式设置以传递给 Mindspeed 底层，保留必要项。
- 内存泄漏风险：ZLiao097 警告 expandable_segments 未正确关闭可能导致严重内存泄漏，作者添加 try-except 保护并保留日志警告。
  - device.py NPU expandable_segments 分支必要性 (design): 保留 NPU 分支，因为 CUDA 判断不适用于 NPU 场景。
  - 删除 Qwen3.5-27B 训练脚本 (other): 已删除 27B 脚本，仅保留 35B 脚本。
  - checkpoint 配置序列化自定义 to_dict (design): 添加 default lambda 处理具有 to_dict 方法的对象，解决序列化问题。
  - NPU 配置项默认值冗余 (correctness): 保留 NPU 专属的 override 配置，删除 GPU 端冗余项。
  - expandable_segments 内存泄漏风险 (performance): 使用 try-except 静默捕获，待后续增加明确警告日志。

风险与影响

- 风险：
  - strict 配置影响：默认 strict=True 可能使保存权重时因 Bridge 模型缺少某些键而失败，影响现有 GPU 训练流程（建议根据实际需要设为 False）。
  - 序列化兼容性：json.dump 的 default 函数假设对象有 to_dict 方法，若其他自定义类无此方法将抛出异常，缺乏兜底。
  - NPU 内存配置：expandable_segments 的异常被静默捕获可能隐藏配置错误，导致训练内存泄漏或性能下降。
  - 回归风险：修改了 megatron_checkpoint_manager.py 的保存逻辑，可能影响所有使用 Megatron Bridge 的 checkpoint 保存。
- 影响：
  - 用户：可直接在 Ascend NPU 上使用 Qwen3.5-35B 进行 GRPO 训练，降低新模型适配成本；但需注意 strict 配置可能对旧权重兼容性产生影响。
  - 系统：NPU 动态内存设置有助于稳定大规模训练，但需监控内存使用。
  - 团队：新增脚本维护成本，但遵循了“不保留每个模型单独脚本”的约定（27B 脚本已删除）。

- 风险标记：默认 strict=True 可能破坏现有保存，JSON 序列化依赖 to_dict 方法，NPU 内存异常被静默捕获

关联脉络

- PR #6346 [trainer] feat: add set_expandable_segments support for npu: 同样涉及 NPU expandable_segments 配置，但 #6346 是环境变量迁移，本 PR 直接调用 API，二者互补。
- PR #6323 [veomni] feat: add veomni qwen3-30b and fix ep: 类似的新模型训练脚本添加（Qwen3-30B VeOmni 后端），展现 verl 在多硬件后端上支持新模型的模式。