PR #6791 完整报告

verl-project/verl

[megatron,doc] feat: support DeepseekV4, GLM5, KimiK2.5 via Megatron Lite

合并时间：2026-06-18 08:12

原文链接：http://prhub.com.cn/verl-project/verl/pull/6791

执行摘要

- 一句话：新增 Megatron Lite 后端示例脚本与文档
- 推荐动作：建议：对使用 Megatron Lite 进行大规模 RL 训练的团队，精读文档和启动脚本，特别是网格配置和优化器选择（fsdp2 vs dist_opt）。命名检查的修改展示了如何优雅扩展多 token 后端后缀，值得参考。

功能与动机

支持 DeepSeek-V4、GLM5、KimiK2.5 等模型通过 Megatron Lite（Megatron 的实验性轻量路径）进行训练，提供可直接运行的示例启动器和文档，降低用户集成门槛。

实现拆解

1. 在 docs/advance/megatron_lite_backend.rst 中添加 Megatron Lite 集成文档，解释 megatron.lite 与 verl_mlite 的安装、环境变量配置和自定义扩展方法。
2. 新增四个启动脚本：examples/sft/gsm8k/run_deepseek_v4_megatron_lite.sh（SFT）、examples/grpo_trainer/run_deepseek_v4_megatron_lite.sh（GRPO/DAPO）、examples/grpo_trainer/run_kimi_k2_6_glm5_1_megatron_lite.sh（256-GPU GRPO）、examples/grpo_trainer/run_qwen3_5_35b_megatron_lite.sh（Qwen3.5 GRPO）。每个脚本通过分组的环境变量（MLITE_ROOT、MLITE_VERL_ROOT、MODEL_VARIANT、OPTIMIZER、网格参数等）控制 mlite 专属行为，并包含网格有效性检查。
3. 修改 tests/special_sanity/check_example_naming.py：在 ALLOWED_BACKENDS 中添加 'megatron_lite'，并新增 _has_allowed_backend 函数，使用 stem.endswith 代替 tokens[-1] in ALLOWED_BACKENDS，以正确匹配包含下划线的 multi-token 后端后缀。
4. 在 tests/special_sanity/test_check_example_naming.py 中新增 test_multi_token_train_backend_accepted，验证 run_deepseek_v4_megatron_lite.sh 通过命名检查。
5. 将四个新启动脚本路径添加到 check_example_naming.py 的 DEFAULT_IGNORE_FILES 元组中（虽然已支持 _megatron_lite 后缀，但保留忽略列表以避免与其他检查冲突）。
6. 更新 examples/README.md 和 docs/index.rst 添加文档链接。

关键文件：

- examples/grpo_trainer/run_kimi_k2_6_glm5_1_megatron_lite.sh（模块 GRPO 示例；类别 other；类型 core-logic）：最复杂的启动脚本，展示 256-GPU 大规模 GRPO 训练配置

，包含网格验证和模型选择。

- *examples/grpo_trainer/run_deepseek_v4_megatron_lite.sh*（模块 GRPO 示例；类别 other；类型 core-logic）：DeepSeek-V4 GRPO 启动脚本，支持 flash/pro 变体，固定 TP1/ETP1，并说明 DSA 依赖。
- *examples/grpo_trainer/run_qwen3_5_35b_megatron_lite.sh*（模块 GRPO 示例；类别 other；类型 core-logic）：Qwen3.5-35B GRPO 启动脚本，展示 allgather CP 路径和 FLA 5.0 依赖。
- *examples/sft/gsm8k/run_deepseek_v4_megatron_lite.sh*（模块 SFT 示例；类别 other；类型 core-logic）：SFT 启动脚本，展示 DeepSeek-V4 在 Megatron Lite 上的 SFT 训练配置。
- tests/special_sanity/check_example_naming.py（模块 命名检查；类别 test；类型 test-coverage；符号 _has_allowed_backend）：核心命名检查逻辑修改，新增 _has_allowed_backend 函数支持 multi-token 后端后缀。
- tests/special_sanity/test_check_example_naming.py（模块 命名检查；类别 test；类型 test-coverage；符号 test_multi_token_train_backend_accepted）：新增测试用例 test_multi_token_train_backend_accepted 验证 multi-token 后端通过检查。
- *docs/advance/megatron_lite_backend.rst*（模块 高级文档；类别 docs；类型 documentation）：提供 Megatron Lite 后端集成文档，包括安装、配置和扩展指南。
- *examples/README.md*（模块 示例说明；类别 docs；类型 documentation）：更新示例说明文档，添加 Megatron Lite 后端相关描述。
- *docs/index.rst*（模块 文档索引；类别 docs；类型 documentation）：更新文档索引，链接新添加的 Megatron Lite 后端文档。

关键符号：_has_allowed_backend, check_filename, test_multi_token_train_backend_accepted

关键源码片段

tests/special_sanity/check_example_naming.py

核心命名检查逻辑修改，新增 _has_allowed_backend 函数支持 multi-token 后端后缀。

```
# 允许的训练后端列表，包含新增的 megatron_lite
ALLOWED_BACKENDS = (
    "fsdp",
    "fsdp2",
    "megatron",
    "megatron_lite",
    "mindspeed",
    "automodel",
    "veomni",
)


def _has_allowed_backend(stem: str) -> bool:
    """判断 stem 是否以允许的后缀结尾。"""
```

```
    # 使用 endswith 代替分割，以正确匹配多 token 后端（如 megatron_lite）
    return any(stem.endswith(f"_{backend}") for backend in ALLOWED_BACKENDS)


def check_filename(path: Path, display: str | None = None) -> list[str]:
    # ... 其他逻辑 ...
    # 原来的检查改为调用 _has_allowed_backend
    if not tokens or not _has_allowed_backend(path.stem):
        errors.append(...)
    # ...
```

评论区精华

在 Review 中，gemini-code-assist[bot] 指出 examples/sft/gsm8k/run_deepseek_v4_megatron_lite.sh 未加入 DEFAULT_IGNORE_FILES，可能导致 CI 命名检查失败。开发者已在后续提交中将该文件加入忽略列表，问题已解决。

- 将 SFT 启动器加入忽略列表 (testing): 开发者在后续提交中将该文件加入 DEFAULT_IGNORE_FILES，问题已解决。

风险与影响

- 风险：风险：
  1. 对外部依赖：启动脚本依赖外部 megatron.lite 和 verl_mlite 仓库，用户需自行安装且版本敏感。
  2. 缺乏 CI 覆盖：启动脚本未在持续集成中运行，可能因环境差异（如路径、依赖版本、硬件）而失效。
  3. 配置复杂性：用户需理解 MLITE_ROOT、网格参数等概念，错误配置可能导致运行失败。
- 影响：影响：
  1. 用户：获得了 Megatron Lite 训练的参考实现，尤其对 DeepSeek-V4、Kimi K2.6、GLM 5.1 等大型模型的多节点训练提供了开箱即用的配置。
  2. 系统：未修改 verl 核心代码，仅添加外部示例，对现有功能无影响。
  3. 团队：需维护文档与外部集成的一致性，未来可能将 mlite 后端整合进主代码库。 - 风险标记：依赖外部 mlite 仓库，启动脚本未集成 CI 测试

关联脉络

- 暂无明显关联 PR