

# THUDM/slime 第 17 周周报 (2026-04-20 至 04-26)

THUDM/slime

周期: 2026-04-20 至 2026-04-26

来源 PR: 7 • 重点 PR: 7 • 自动生成

原文链接: <http://prhub.com.cn/THUDM/slime/reports/2026-04-20-to-2026-04-26>

## 执行摘要

本周 (2026-04-20 至 04-26), THUDM/slime 共合并 7 个 PR, 核心观察是 PPO 训练架构的重构 (#1856) 和 基础设施清理与扩展 (sglang 补丁精简、DGX Spark 支持)。训练系统模块经历重要调整, actor-critic 解耦、分布式通信方式变更, 同时多项 bug 修复与配置优化并行推进。整体趋势是系统架构走向灵活化与解耦, 同时为新一代硬件做好适配。

## 本周重点变化

1. PPO 训练架构重构 (#1856): 引入独立 critic YAML 配置, 移除 actor-critic 间的 NCCL 同步, 改为 Ray 对象传递; 调整训练循环, 支持自定义优势函数钩子。此举显著降低 actor-critic 耦合, 但移除了原有同步保证, 需关注训练稳定性与性能。
2. slang 补丁清理 (#1859): 将补丁文件从 1536 行精简至 71 行, 仅保留关键兼容修改, 为 slang 版本升级扫清障碍。同步更新测试脚本, 加入 --colocate 参数, 可能成为后续训练标配。
3. DGX Spark 硬件支持 (#1835): 新增基于 arm64 架构的 Docker 构建, 解决 15 个阻塞问题, 并通过端到端 GRPO 训练验证。项目硬件兼容性得以扩展, 但依赖特定版本, 需持续跟踪。
4. 熵计算 bug 修复 (#1846): 修复 allgather-CP 路径下熵返回 None 的隐患, 提升分布式训练鲁棒性, 同时简化代码结构。
5. 序列长度分区后备机制回滚 (#1848): 移除内存安全屏障, 简化分区逻辑, 但可能增加内存溢出风险, 需关注后续替代方案。
6. 其他调整: 模型脚本路径迁移 (#1849) 统一目录结构, 但缺乏联动更新; 构建脚本版本修复 (#1827) 确保 cuda-python 与 CUDA 12.9 对齐。

## 模块与主题趋势

- 训练系统: PPO 相关变更 (#1856、#1846) 集中, 架构演进与 bug 修复并行, 数据流与通信模式重塑。
- 基础设施: Docker 与补丁管理活跃 (#1859、#1835), 精简与扩展并举, 强调可维护性与硬件兼容性。
- 配置与工具: 脚本标准化 (#1849)、构建脚本修复 (#1827) 体现环境一致性诉求, 但测试覆盖不足仍是普遍短板。
- 风险模式: " 核心路径变更 " 与 " 缺少测试覆盖 " 高频出现, 反映重构与安全之间的张力, 建议加强回归测试。

风险观察

- 1. PPO 重构移除 NCCL 同步：改用 Ray 对象传递可能引入新的通信瓶颈或数据竞争，需在实  
际训练任务中观测吞吐与收敛。
- 2. sglang 补丁精简：删除 1465 行补丁可能影响对旧版本或特殊模型的兼容性，需监控升级  
后的回归。
- 3. 熵修复类型风险：\_extract\_per\_sample 的返回类型未与运行时对齐，可能调用方期待  
None 时收到空张量。
- 4. 分区安全保障缺失：回滚后备机制后，动态批处理路径缺乏内存上限保护，极端负载下可能  
OOM。
- 5. 新硬件依赖特定版本：GB10 支持依赖特定容器和 CUDA 版本，未来升级需对应适配。

重点 PR 速览

| PR<br>编号  | 标题                                                              | 重<br>要<br>性 | 关键信息                                                                |
|-----------|-----------------------------------------------------------------|-------------|---------------------------------------------------------------------|
| #18<br>56 | refactor/ppo                                                    | 8.7         | 核心架构重构，解耦配置与通信，引入 critic YAML。<br>精读建议：关注训练循环改动和 Ray 数据传递的设计。       |
| #18<br>59 | [docker] cleanup<br>sglang patch                                | 6.2         | 大幅清理补丁，为升级铺路。值得关注删除内容对现<br>有模型的影响，以及 <code>--colocate</code> 参数的推广。 |
| #18<br>35 | Add support for<br>NVIDIA DGX Spark                             | 6.2         | 跨平台 Docker 构建，解决 arm64 内核编译问题。适<br>合在新硬件上部署的用户。                     |
| #18<br>46 | fix entropy bug and<br>update code                              | 6.9         | 分布式训练边界条件修复，类型系统一致性值得参考。<br>关注空切片处理方式。                              |
| #18<br>48 | Revert "Add fallback<br>for get_seqlen_balan<br>ced_partitions" | 6.7         | 简化分区逻辑但移除保护，需评估内存风险。可结合<br>训练规模对比。                                  |
| #18<br>27 | Fix cuda-python pin<br>in build_conda.sh                        | 2.6         | 小修复但体现环境一致性维护。                                                      |

后续建议

- 加强测试覆盖：针对本次重构的核心路径（PPO 训练、数据分区），补充单元与集成测试，  
尤其是边界情况。
- 监控 PPO 训练效果：建议在多种模型规模上运行收敛测试，对比新旧配置吞吐与 loss 曲线。
- 完善脚本目录迁移：#1849 的路径变更应同步更新文档、CI 引用，避免构建失败。
- 持续关注硬件支持：DGX Spark 的后续维护需与上游容器对齐，关注 arm64 生态发展。
- 建立补丁精简规范：对于 sglang 等第三方补丁，制定清理节奏与兼容性验证流程。