PR #7422 完整报告

verl-project/verl

[rollout] fix: preserve dummy load_format in disaggregated rollout

合并时间：2026-08-17 12:51

原文链接：http://prhub.com.cn/verl-project/verl/pull/7422

执行摘要

- 一句话：移除 dummy 格式覆盖，修复分离式 rollout 权重广播
- 推荐动作：值得精读。该 PR 展示了一个跨后端（vLLM/SGLang/TRT-LLM）共有的配置契约：load_format=dummy 在分离式训练中代表“跳过加载、等广播”。建议关注： 1) 三个服务器如何共用相同 guard 逻辑（横向对比）； 2) 移除 guard 后如何通过测试和示例守住契约边界； 3) FP8 例外被删除的后续验证。

功能与动机

PR body 明确指出：disaggregated（async）训练中 rollout worker 有意以 load_format=dummy 启动，跳过启动时的权重加载，稍后通过权重广播（NCCL 或 Gloo）从 trainer 接收权重；而 SGLangHttpServer.__init__ 中的 guard 会在非 hybrid 模式下把 dummy 静默改成 auto，导致 SGLang 回退到从磁盘加载权重，所有 rollout worker 同时读完整模型，使 dummy 格式失去意义。该行为在 vLLM 与 TRT-LLM 对应服务器中同样存在，一并移除。

实现拆解

1. 移除三个异步服务器 __init__ 中的 load_format 改写 guard：
   - verl/workers/rollout/sglang_rollout/async_sglang_server.py：删除 if self.rollout_mode != RolloutMode.HYBRID and self.config.load_format == 'dummy' 分支。
   - verl/workers/rollout/trtllm_rollout/trtllm_async_server.py：删除同样分支，连同 FP8 例外条件一并删除。
   - verl/workers/rollout/vllm_rollout/vllm_async_server.py：删除同样分支。 原因：dummy 在分离式训练中代表“跳过本地加载、等待广播”，改写会破坏该契约。
2. 为 standalone 测试显式设置 load_format=auto： 涉及 tests/workers/rollout/rollout_vllm/test_vllm_generation_determinism.py、tests/experimental/agent_loop/test_standalone_rollout.py、tests/experimental/reward_loop/test_agent_reward_loop_standalone.py、tests/workers/rollout/rollout_trtllm/test_adapter.py、tests/workers/rollout/rollout_trtllm/test_inter_node_rollout.py、tests/workers/rollout/rollout_vllm/test_vllm_abort.py。 原因：这些测试没有 trainer 负

责广播权重，必须从磁盘加载，否则会拿随机权重比较。

3. 更新生成示例与教程： examples/generation/run_deepseek_llm_7b.sh 增加注释并显式传 actor_rollout_ref.rollout.load_format=auto；examples/tutorial/agent_loop_get_started/agent_loop_tutorial.ipynb 同步补充说明。
4. 提交演进：作者提交初始修复，merge 者 wuxibin89 追加"keep load_format=dummy for disaggregated rollout"与"fix ci"两个提交，用于保证 dummy 在分离式场景保留并修复 CI 测试。

关键文件：

- verl/workers/rollout/sglang_rollout/async_sglang_server.py（模块 异步服务；类别 source；类型 core-logic；符号 SGLangHttpServer.init）：PR 核心修复对象：移除 __init__中把 load_format=dummy 改写为 auto 的 guard，让分离式训练下权重广播生效。
- verl/workers/rollout/trtllm_rollout/trtllm_async_server.py（模块 异步服务；类别 source；类型 core-logic；符号 TRTLLMHttpServer.init）：与 SGLang 相同的 guard 被删除，且连带移除 FP8 例外条件，属于本次行为闭环的一部分。
- verl/workers/rollout/vllm_rollout/vllm_async_server.py（模块 异步服务；类别 source；类型 core-logic；符号 VLLMAsyncServer.init）：vLLM 异步服务器同样移除该 guard，保证分离式训练下 vLLM 权重由广播提供。
- tests/workers/rollout/rollout_vllm/test_vllm_generation_determinism.py（模块 确定性测试；类别 test；类型 test-coverage；符号 _make_rollout_config）：代表 standalone 场景测试：因为没有 trainer 广播，显式设置 load_format=auto，保证测试比较真实权重而非随机权重。
- tests/experimental/agent_loop/test_standalone_rollout.py（模块 独立测试；类别 test；类型 test-coverage；符号 init_config, test_standalone_rollout）：初始化 fixture 补充 load_format=auto，覆盖 agent_loop 的 standalone rollout 用法。
- tests/experimental/reward_loop/test_agent_reward_loop_standalone.py（模块 奖励环测试；类别 test；类型 test-coverage；符号 test_agent_reward_loop_standalone）：reward loop standalone 测试同样显式 auto，避免移除 guard 后权重缺失。
- examples/generation/run_deepseek_llm_7b.sh（模块 示例脚本；类别 other；类型 configuration）：纯生成场景没有 trainer，示例脚本显式传 load_format=auto 并注释说明，帮助用户避免踩坑。

关键符号：SGLangHttpServer.init, TRTLLMHttpServer.init, VLLMAsyncServer.init, _make_rollout_config, init_config, test_standalone_rollout, test_agent_reward_loop_standalone

关键源码片段

verl/workers/rollout/sglang_rollout/async_sglang_server.py

PR 核心修复对象：移除 __init__中把 load_format=dummy 改写为 auto 的 guard，让分离式训练下权重广播生效。

```
# verl/workers/rollout/sglang_rollout/async_sglang_server.py（head 版本关键片段）
class SGLangHttpServer:
```

```
def __init__(self, config, model_config, rollout_mode, workers, replica_rank,
             node_rank, nnodes, base_gpu_id, cuda_visible_devices, ...):
    # 将 OmegaConf 配置转换为 dataclass，便于后续访问
    self.config: RolloutConfig = omega_conf_to_dataclass(config)
    self.model_config: HFModelConfig = omega_conf_to_dataclass(model_config, dataclass_
    type=HFModelConfig)

    # 校验 max_model_len，未设置时回退到模型的最大位置编码
    max_position_embeddings = get_max_position_embeddings(self.model_config.hf_config)
    if self.config.max_model_len is None:
        self.config.max_model_len = max_position_embeddings
    elif self.config.max_model_len > max_position_embeddings:
        raise ValueError(
            f'max_model_len ({self.config.max_model_len}) should be <= '
            f'max_position_embeddings ({max_position_embeddings})'
        )

    self.rollout_mode = rollout_mode
    self.workers = workers
    self.replica_rank = replica_rank
    self.node_rank = node_rank
    self.nnodes = nnodes
    self.base_gpu_id = base_gpu_id
    # 权重版本号，由 ServerAdapter 在每次权重更新后递增
    self.global_steps = None

    # 以下两行用于 PD 分离部署，启动后由 SGLangPDReplica.set_pd_peer 填充
    self._pd_decode_peers: list[ActorHandle] = []
    self._pd_bootstrap_host: Optional[str] = None

    # 关键行为（本次 PR 的改动核心）：
    # 此处不再根据 rollout_mode 改写 load_format。
    # 在 disaggregated（async）训练中，rollout worker 故意以 load_format='dummy'
    # 启动，跳过本地权重读取，后续通过 NCCL/Gloo 广播接收 trainer 权重；
    # 若按旧逻辑把 dummy 改成 'auto'，每个 worker 都会同时从磁盘读完整模型，
    # 广播机制就失去了意义。standalone 场景必须由调用方显式设置
    # load_format='auto'，因为那里没有 trainer 来同步权重。

    # 用于 HTTP 服务的地址与端口
    self._server_address = ray.util.get_node_ip_address().strip('[]')
    self._server_port = None
    # 多节点时通过 master 地址协调 NCCL 初始化
    self._master_address = None
    self._master_port = None
    self._master_sock = None
    if self.nnodes > 1 and self.node_rank == 0:
        self._master_address = self._server_address
        self._master_port, self._master_sock = get_free_port(self._server_address, with_alive_
        sock=True)
```

tests/workers/rollout/rollout_vllm/test_vllm_generation_determinism.py

代表 standalone 场景测试：因为没有 trainer 广播，显式设置 load_format=auto，保证测试比较真实权重而非随机权重。

```
# tests/workers/rollout/rollout_vllm/test_vllm_generation_determinism.py（head 关键片段）
def _make_rollout_config(model_path, seed):
    with initialize_config_dir(config_dir=_get_config_dir(), version_base=None):
        config = compose(config_name='ppo_trainer')
    config.trainer.n_gpus_per_node = 1
    config.trainer.nnodes = 1
    config.actor_rollout_ref.model.path = model_path
    config.actor_rollout_ref.model.trust_remote_code = True
    config.actor_rollout_ref.rollout.name = 'vllm'
    config.actor_rollout_ref.rollout.mode = 'async'
    config.actor_rollout_ref.rollout.tensor_model_parallel_size = 1
    config.actor_rollout_ref.rollout.prompt_length = 128
    config.actor_rollout_ref.rollout.response_length = 256
    config.actor_rollout_ref.rollout.max_model_len = 512
    config.actor_rollout_ref.rollout.max_num_seqs = 8
    config.actor_rollout_ref.rollout.gpu_memory_utilization = 0.4
    config.actor_rollout_ref.rollout.enforce_eager = True
    config.actor_rollout_ref.rollout.full_determinism = True
    config.actor_rollout_ref.rollout.seed = seed
    config.actor_rollout_ref.rollout.scheduling_policy = 'priority'
    # standalone 服务器没有 trainer 来广播权重；若不显式从磁盘加载，
    # 测试会拿随机初始化模型的 logprobs 做比较，导致结果失真。
    config.actor_rollout_ref.rollout.load_format = 'auto'
    return config
```

评论区精华

该 PR 没有 review 评论，wuxibin89 直接 APPROVE。从提交历史可见两轮补充调整：

- 15ecd75 “keep load_format=dummy for disaggregated rollout”：强化保留 dummy 的设计意图。
- 106e771 “fix ci”：说明移除 guard 后测试需要显式 auto 才能通过。

> 结论：无公开讨论争议，功能修复与测试适配均由提交闭环完成。

- 移除 guard 后的 CI/ 测试适配 (testing): 以测试与示例显式配置 load_format=auto 收尾，功能修复保持不变。

风险与影响

- 风险：主要风险： 1) 移除 guard 后，任何在非 hybrid 模式下继续依赖旧行为（dummy 被自动转 auto）的配置会保留 dummy，若调用方没有 trainer 广播且未显式设置 auto，模型可能在未加载权重状态下启动。官方示例与测试已同步，但用户自定义脚本可能遗漏。 2) TRT-LLM 原 guard 对 FP8 有例外，本次一并删除，需确认 FP8 量化模式下 dummy 初始化与后续广播填充是否兼容。 3) 三个后端对 dummy 的语义支持程度不同，后续若某后端

不支持 dummy，可能出现回归。 4) 测试文件均显式加了 auto，但用户自定义脚本可能遗漏，需要文档提示。

- 影响：影响范围：使用 async/disaggregated 训练的 PPO/GRPO 用户受益，修复了启动阶段多 worker 并发读盘的性能问题与等待广播的语义被破坏问题；改动集中在 rollout 异步服务器初始化路径，hybrid 模式行为不变。对已经显式设置 load_format 的配置无影响；对 standalone 部署，要求用户显式 auto，官方示例已同步。影响程度：中。
- 风险标记：移除隐式配置兜底，standalone 需显式 auto，FP8 例外一并删除，多后端行为一致性

关联脉络

- PR #7434 [vllm] fix: vllm always need to resume weights before weight sync: 同属 vLLM 权重同步链路：本 PR 保证 dummy 格式下权重由广播提供，7434 修复广播前权重映射恢复，两者共同保障分离式训练权重正确性。
- PR #7413 [sglang] fix: lora sglang e2e: 同改 SGLang rollout 文件与权重同步链，涉及 LoRA 权重同步到 SGLang 的链路，与本 PR 的 load_format 广播前提相关。
- PR #7291 [ckpt] feat: Node-local multi-sender broadcast in NCCL checkpoint engine: 权重广播机制的性能演进，dummy 加载依赖该广播在训练启动后填充权重，二者构成同一架构的两个侧面。