PR #2219 完整报告

radixark/miles

[feat] Add training log-prob reuse to skip the redundant forward-only pass

合并时间：2026-08-14 05:37

原文链接：http://prhub.com.cn/radixark/miles/pull/2219

执行摘要

- 一句话：单步 PPO 训练复用训练 log-probs，跳过冗余 actor 前向
- 推荐动作：值得精读。本 PR 展示了一个典型的 " 安全性能优化 " 范式：用显式 opt-in flag + fail-closed 配置清单 + 运行时断言三层防护，把 " 两个 forward 统计等价 " 这一不变量落到代码层面；policy_loss_function 中旧策略源选择与 TIS 输入解耦、compute_advantages_and_returns 的 detach 边界统一，都是值得借鉴的设计决策。review 中针对 detach 回归（#1966）的讨论也说明核心训练路径的边界必须显式维护。

功能与动机

PR body 明确说明性能动机："A single-step rollout evaluates the same actor weights twice: once to produce training log-probs and again inside the training forward. The standalone pass consumes model time without changing the old-policy baseline, and its cost repeats for every rollout." 即单步 rollout 下同一套 actor 权重被求值两次，独立 forward-only pass 不改变旧策略基线却每次重复消耗模型时间，对于千亿级模型（如 kimi-k2.5、deepseek v3.2）这是可观的训练吞吐损失。

实现拆解

实现按 4 步拆解：

1. 参数入口与 fail-closed 校验（miles/utils/arguments.py）：新增 --skip-actor-forward-only 布尔开关（默认 False），在 miles_validate_args 尾部调用新增的 validate_skip_actor_forward_only。该校验要求 train_backend == "megatron"、loss_type == "policy_loss"、compute_advantages_and_returns 为真，并列出 20 余项不兼容配置：任何引入随机性（dropout、moe input jitter、router 偏置 / 负载均衡）、改变权重版本或策略基线（--keep-old-actor、--kl-coef、--use-opd）、重放前后向的钩子（custom hooks、source patcher、--save-debug-train-data）都会被拒绝；--use-routing-replay/--use-indexer-replay 仅在配套的 --use-rollout-*-replay 变体下放行。同时校验单步约束：num_steps_per_rollout 为 None 或 1，且静态 batch 下 global_batch_size == rollout_batch_size * n_samples_per_prompt；动态 global batch 与 multi-LoRA 场景把步数校验延后到运行时。
2. actor 生命周期跳过后端（miles/backends/megatron_utils/actor.py）：MegatronTrainRayActor.train_actor 开头计算 num_optimizer_steps = len(num_microbatches)，skip 开启时断言恰好 1 步且 rollout_data 不含预计算 actor

log_probs（防止误传旧数据）；独立的 actor compute_log_prob 仅在 not skip_actor_forward_only and (not use_rollout_logprobs or get_mismatch_metrics) 时执行，reference/teacher 的 compute_log_prob（ref_/teacher_ 前缀）保持保留；rollout replay 队列（routing/indexer）的预填职责从被跳过的 standalone scoring 移交到训练 forward 内的 fill_replay_data 消费。

3. 损失与优势计算适配（miles/backends/training_utils/loss_hub/losses.py + miles/backends/training_utils/loss.py）：policy_loss_function 把训练 forward 产出的 log-probs 在 skip 模式下 detach 为 trainer_scored_log_probs，并让 PPO 旧策略源独立选择——--use-rollout-logprobs 时仍用 rollout log-probs，否则用 detached 训练 log-probs；GSPO/OPSM 的 context-parallel all-gather 路径在 skip 且不用 rollout log-probs 时直接 detach 已 gather 的 full_log_probs，避免一次额外的 all-gather；TIS/mismatch 的 train_log_probs 改指向 trainer_scored_log_probs。compute_advantages_and_returns 在 skip 且最后 pipeline stage 且缺 log-probs/values 时用 get_local_response_loss_masks 合成零 KL；detach 边界（_detach_rollout_tensor_list 对 log_probs/rollout_log_probs/ref_log_probs/teacher_log_probs）移到所有非返回路径之前无条件执行，修复 skip 路径绕过固定分数 detach 的问题。
4. 测试与 E2E 联动：新增 tests/fast/backends/training_utils/loss/test_training_logprob_reuse.py（343 行），验证复用训练 log-probs 与显式 detached 基线在 loss/grad/metrics 上逐元素等价、use_rollout_logprobs 时旧策略保留、缺 old-policy 报错路径、最后 stage 零 KL 合成、非叶子 ref/teacher 分数 detach、loss 分发器不泄漏额外 kwargs；test_shared_ppo_lifecycle.py 新增 205 行覆盖 train_actor 的 forward 调用矩阵（skip/use_rollout_logprobs/microbatches 数）、ref/teacher 保留、replay 队列消费、多步拒绝与已有 actor log-probs 拒绝；test_arguments.py 新增 180 行覆盖 flag 解析与合法 / 非法配置矩阵；test_ppo_cp_advantages.py 覆盖 CP 下合成零 KL 与单 rank 基线对齐；E2E 侧 kimi_k25_2layer、deepseek_v32_5layer_fp8、glm5_744b4layer_r3、qwen3_30B p2p 等 Megatron 模型线启用该 flag。

关键文件：

- miles/backends/training_utils/loss_hub/losses.py（模块 损失函数；类别 source；类型 core-logic；符号 policy_loss_function）：policy_loss_function 的核心改造：skip 模式下训练 forward 的 log-probs detach 后充当旧策略基线，实现训练 log-prob 复用；GSPO/OPSM 的 CP all-gather 路径也据此避免额外 gather。
- miles/backends/megatron_utils/actor.py（模块 训练循环；类别 source；类型 core-logic；符号 train_actor）：MegatronTrainRayActor.train_actor 是 " 跳过 standalone forward" 的执行点：运行时断言单步与无预计算 log-probs，条件化调用 compute_log_prob，并把 replay 队列消费职责移交训练 forward。
- miles/utils/arguments.py（模块 参数解析；类别 source；类型 core-logic；符号 validate_skip_actor_forward_only）：新增 --skip-actor-forward-only 参数与 validate_skip_actor_forward_only fail-closed 校验清单，是整个优化安全性的第一道防线；同时放宽 get_mismatch_metrics 提示逻辑。
- miles/backends/training_utils/loss.py（模块 优势计算；类别 source；类型 core-logic；符号 compute_advantages_and_returns）：compute_advantages_and_returns 在 skip

场景下允许最后 pipeline stage 缺失 actor log-probs，并用局部 loss mask 合成零 KL；detach 边界移到所有非返回路径之前，修复 review 指出的 #1966 回归风险。

- tests/fast/backends/training_utils/loss/test_training_logprob_reuse.py（模块 单元测试；类别 test；类型 test-coverage；符号 process_group, _run_policy_loss, test_reused_training_log_probs_match_an_explicit_detached_baseline, test_skip_actor_forward_only_preserves_rollout_log_probs_as_old_policy）：新增 343 行核心测试：验证复用训练 log-probs 与显式 detached 基线在 loss/grad/metrics 上逐元素等价，覆盖 GRPO/GSPO/TIS、rollout log-probs 保留、缺旧策略报错、零 KL 合成、detach 边界与 loss 分发器行为。
- tests/fast/backends/megatron_utils/test_shared_ppo_lifecycle.py（模块 训练生命周期；类别 test；类型 test-coverage；符号 _actor_train_args, _actor_reuse_worker, _patch_actor_reuse_dependencies, test_actor_logprob_forward_is_explicit_single_step_opt_in）：新增 205 行 actor 生命周期测试：验证 train_actor 的 compute_log_prob 调用矩阵（skip 与 use_rollout_logprobs 组合）、ref/teacher forward 保留、replay 队列消费移交、多步与已有 log-probs 的拒绝路径。
- tests/fast/utils/test_arguments.py（模块 参数测试；类别 test；类型 test-coverage；符号 test_skip_actor_forward_only_flag_is_parsed, test_skip_actor_forward_only_is_gated_during_miles_validation, _make_skip_actor_forward_only_args, TestValidateSkipActorForwardOnly）：新增 180 行参数校验测试：flag 解析、miles_validate_args 门控、合法配置（TIS/rollout logprobs/dumper/dump-details/rollout replay）与 20 余项不兼容配置的拒绝矩阵。
- tests/fast/backends/training_utils/test_ppo_cp_advantages.py（模块 单元测试；类别 test；类型 test-coverage；符号 _worker_reused_zero_kl, test_reused_zero_kl_matches_single_rank_baseline_with_context_parallelism）：验证 context parallelism 下合成零 KL 的 advantages/returns 与单 rank 基线逐元素对齐，覆盖真实 CP 多进程场景。
- tests/e2e/megatron/model_scripts/test_kimi_k25_2layer_ci.py（模块 端到端测试；类别 test；类型 test-coverage）：Kimi-K2.5 2 层 Megatron E2E 启用 --skip-actor-forward-only，为跳过路径提供真实训练验证。
- tests/e2e/megatron/model_scripts/test_deepseek_v32_5layer_fp8.py（模块 端到端测试；类别 test；类型 test-coverage）：DeepSeek-V3.2 5 层 FP8 E2E 启用 skip flag，同时覆盖 fp8 量化与 rollout routing replay 场景。

关键符号：validate_skip_actor_forward_only, compute_advantages_and_returns, policy_loss_function, train_actor

关键源码片段

miles/backends/training_utils/loss_hub/losses.py

policy_loss_function 的核心改造：skip 模式下训练 forward 的 log-probs detach 后充当旧策略基线，实现训练 log-prob 复用；GSPO/OPSM 的 CP all-gather 路径也据此避免额外 gather。

```
# miles/backends/training_utils/loss_hub/losses.py
```

```
# policy_loss_function 中的旧策略源选择（核心改动）
def policy_loss_function(args, batch, logits, sum_of_sample_mean):
    # 旧策略基线来源一：rollout 引擎产出的 log-probs（--use-rollout-logprobs 时）
    rollout_old_log_probs = (
        [log_prob.detach() for log_prob in batch["rollout_log_probs"]]
        if batch.get("rollout_log_probs") is not None
        else None
    )
    if args.use_rollout_logprobs:
        assert rollout_old_log_probs is not None, "rollout_log_probs must be provided"
    elif not args.skip_actor_forward_only and batch.get("log_probs") is None:
        # 非 skip 路径仍然强制要求预计算的 actor log-probs，保持 fail-fast
        raise ValueError("policy loss requires old-policy log-probs")

    # 训练 forward 算出的当前策略 log-probs 与 entropy
    log_probs_and_entropy = get_log_probs_and_entropy(
        logits, args=args, unconcat_tokens=batch["unconcat_tokens"],
        total_lengths=total_lengths, response_lengths=response_lengths,
        with_entropy=calculate_entropy, max_seq_lens=max_seq_lens,
    )
    log_probs = log_probs_and_entropy["log_probs"]

    if args.skip_actor_forward_only:
        # 关键设计：单步确定性训练中，训练 forward 的 log-probs detach 后
        # 直接充当旧策略基线，PPO importance log-ratio 恒为 0；
        # 当前（可微分）log-probs 仍正常参与梯度计算。
        trainer_scored_log_probs = [log_prob.detach() for log_prob in log_probs]
    else:
        trainer_scored_log_probs = (
            [log_prob.detach() for log_prob in batch["log_probs"]]
            if batch.get("log_probs") is not None
            else None
        )
    # 旧策略源独立于训练分数选择：--use-rollout-logprobs 优先使用 rollout 分数
    old_log_probs = rollout_old_log_probs if args.use_rollout_logprobs else trainer_scored_log_
    probs

    # GSPO/OPSM 需要全序列 log-probs：skip 且不用 rollout 分数时，直接 detach
    # 已 gather 的 full_log_probs，避免对同一批 log-probs 再做一次 all-gather。
    if need_full_log_probs:
        full_log_probs = [
            all_gather_with_cp(log_prob, total_length, response_length)
            for log_prob, total_length, response_length in zip(
                log_probs, total_lengths, response_lengths, strict=False
            )
        ]
        if args.skip_actor_forward_only and not args.use_rollout_logprobs:
            full_old_log_probs = [full_log_prob.detach() for full_log_prob in full_log_probs]
        else:
```

```
            full_old_log_probs = [
                all_gather_with_cp(old_log_prob, total_length, response_length)
                for old_log_prob, total_length, response_length in zip(
                    old_log_probs, total_lengths, response_lengths, strict=False
                )
            ]
```

miles/backends/megatron_utils/actor.py

MegatronTrainRayActor.train_actor 是 " 跳过 standalone forward" 的执行点：运行时断言单步与无预计算 log-probs，条件化调用 compute_log_prob，并把 replay 队列消费职责移交训练 forward。

```
# miles/backends/megatron_utils/actor.py
# MegatronTrainRayActor.train_actor 中新增的 skip 逻辑
# 创建 data iterator 时同时得到本次 rollout 对应的 microbatches 数量，
# 它直接决定 optimizer step 次数。
data_iterator, num_microbatches = get_data_iterator(self.args, self.model, rollout_data)
num_optimizer_steps = len(num_microbatches)
skip_actor_forward_only = self.args.skip_actor_forward_only

if skip_actor_forward_only:
    # 运行时兜底断言：静态校验覆盖不到的动态 batch / multi-LoRA 场景，
    # 在这里保证 standalone scorer 只能在 " 恰好一步 " 时被安全跳过。
    option = "--skip-actor-forward-only"
    assert num_optimizer_steps == 1, f"{option} requires 1 optimizer step, got {num_optimizer_
    steps}"
    # 拒绝携带旧 actor log-probs 的 rollout 数据，防止把过期基线混入训练
    assert rollout_data.get("log_probs") is None, f"{option} requires rollout data without actor log
    probs"

# ... 省略 replay manager 预填与 weights_backuper 切换 ...

self._switch_model("old_actor" if self.args.keep_old_actor else "actor")

# 仅当没有跳过 standalone pass 且确实需要 actor 分数时才执行独立 forward：
# 1) 不使用 rollout log-probs 时需要它提供旧策略基线；
# 2) 开启 mismatch 指标时需要用训练引擎重新计算 log-probs。
# 跳过时，旧策略基线改由训练 forward 内 detach 的 log-probs 提供。
if not skip_actor_forward_only and (
    not self.args.use_rollout_logprobs or self.args.get_mismatch_metrics
):
    for m in all_replay_managers:
        if m.enabled:
            if self._use_rollout_replay(m):
                ... # 预填 rollout replay 队列（standalone scoring 路径）
    self.compute_log_prob(...) # actor scorer：store_prefix 为 ""

# ref/teacher forward 不受 skip 影响，始终执行（它们针对不同的权重备份）
if "ref" in self.weights_backuper.backup_tags:
```

```
        self.compute_log_prob(..., store_prefix="ref_")
    if "teacher" in self.weights_backuper.backup_tags:
        self.compute_log_prob(..., store_prefix="teacher_")

    # 训练 forward 负责消费跳过 standalone scoring 后剩余的预填 replay 队列
    num_rollouts = get_num_rollouts(self.args, rollout_data, num_optimizer_steps)
    self._set_replay_stage("replay_backward")
    with timer("actor_train"):
        train_step_outcome = train(
            self.args, self.model, self.optimizer, self.opt_param_scheduler,
            data_iterator, num_microbatches, num_rollouts,
            witness_info=witness_info, attempt=attempt,
            ft_test_action_executor=self._ft_test_action_executor,
        )
```

miles/utils/arguments.py

新增 --skip-actor-forward-only 参数与 validate_skip_actor_forward_only fail-closed 校验清单，是整个优化安全性的第一道防线；同时放宽 get_mismatch_metrics 提示逻辑。

```
    # miles/utils/arguments.py
    # --skip-actor-forward-only 的 fail-closed 配置校验：任何会让 " 独立 forward"
    # 与 " 训练 forward" 在统计上不等价或产生额外 forward 的配置都被拒绝，
    # 保证 detached 训练 log-probs 可以安全充当 PPO 旧策略基线。
    def validate_skip_actor_forward_only(args) -> None:
        option = "--skip-actor-forward-only"
        assert args.train_backend == "megatron", f"{option} only supports --train-backend megatron"
        assert args.loss_type == "policy_loss", f"{option} only supports --loss-type policy_loss"
        assert args.compute_advantages_and_returns, f"{option} requires actor advantage
        computation"

        # 不兼容清单：随机化（dropout / moe jitter / router 偏置与负载均衡）、
        # 权重版本或策略基线变化（keep_old_actor / kl-coef / OPD）、重放前后向的
        # 钩子（custom hooks / dumper source patcher / save-debug-train-data）全部拒绝；
        # 仅 rollout 侧 replay（use_rollout_routing/indexer_replay）被放行，因为
        # 它们的预填职责可以移交训练 forward 消费。
        incompatible_options = [
            name
            for name, enabled in (
                ("--keep-old-actor", args.keep_old_actor),
                ("--kl-coef", args.kl_coef != 0),
                ("--use-opd", args.use_opd),
                ("--hidden-dropout", args.hidden_dropout != 0),
                ("--attention-dropout", args.attention_dropout != 0),
                ("--lora-dropout", args.lora_dropout != 0),
                ("--moe-input-jitter-eps", args.moe_input_jitter_eps not in (None, 0)),
                ("--moe-router-force-load-balancing", args.moe_router_force_load_balancing),
                ("--moe-router-force-biased", args.moe_router_force_biased is not None),
                ("--moe-router-load-balancing-type sinkhorn", "sinkhorn" in args.moe_router_load_
                balancing_type),
```

```
            ("--use-rollout-entropy", args.use_rollout_entropy),
            ("--true-on-policy-mode", args.true_on_policy_mode),
            ("--log-correct-samples", args.log_correct_samples),
            ("--rollout-data-postprocess-path", args.rollout_data_postprocess_path is not None),
            ("--custom-megatron-before-log-prob-hook-path", args.custom_megatron_before_log_
            prob_hook_path is not None),
            ("--custom-megatron-before-train-step-hook-path", args.custom_megatron_before_train_
            step_hook_path is not None),
            ("--custom-model-provider-path", args.custom_model_provider_path is not None),
            ("--dumper-source-patcher-config-train", args.dumper_source_patcher_config_train is
            not None),
            ("--save-debug-train-data", args.save_debug_train_data is not None and args.dump_
            details is None),
            ("--use-routing-replay", args.use_routing_replay and not args.use_rollout_routing_
            replay),
            ("--use-indexer-replay", args.use_indexer_replay and not args.use_rollout_indexer_
            replay),
        )
        if enabled
    ]
    assert not incompatible_options, f"{option} is incompatible with: {', '.join(incompatible_
    options)}"

    # 单步约束：多步训练会让权重在 rollout 内更新，训练 log-probs 不再能代表旧策略
    assert args.num_steps_per_rollout in (None, 1), (
        f"{option} requires exactly one optimizer step per rollout; "
        f"got --num-steps-per-rollout {args.num_steps_per_rollout}"
    )
    # 静态 batch 下，一次 rollout 的样本数必须恰好等于一个 global batch（一步）
    if not args.use_dynamic_global_batch_size and not args.multi_lora:
        samples_per_rollout = args.rollout_batch_size * args.n_samples_per_prompt
        assert args.global_batch_size == samples_per_rollout, (
            f"{option} requires exactly one optimizer step for {samples_per_rollout} rollout samples;
            "
            f"got --global-batch-size {args.global_batch_size}"
        )
```

评论区精华

3 条 review 评论形成 2 个核心讨论线程。

- detach 边界回归风险：yueming-yuan 指出当使用 KL loss 且 allow_missing_log_prob 生效时，_detach_rollout_tensor_list 会被跳过，ref_log_probs 不再 detach，"the bug in #1966 exists again"。这是最关键的 correctness 讨论：跳过路径不能绕过既有的固定分数持久化边界。作者最终通过 commit b8e73736 将 detach 提升到所有非返回路径之前无条件执行，并补充非叶子（sin/cos 派生）ref/teacher tensor 的 detach 测试。
- dumper / dump-details 门控是否过严：yueming-yuan 质疑为什么 --dump-details 和 dumper 参数不允许："For (1), I think it should not be blocked theoretically, just

some field in the dump details might be missing"。guapisolo 回应已放宽：commit 12cc66f1 移除 --dump-details 限制（DumpReader 把 actor log-probs 视为可选列，policy-loss debug dump 仍会记录训练 forward 与所选旧策略 log-probs），commit 7edbd954 启用 dumper 参数；仅保留 standalone --save-debug-train-data 门控，因为训练 source patching 会重放被跳过的 forward。

- KL loss 场景下 detach 边界被绕过，可能复现 #1966 的固定分数 graph 泄漏 (correctness): 作者通过 commit b8e73736 将 detach 提升到所有非返回路径之前无条件执行（log_probs/rollout_log_probs/ref_log_probs/teacher_log_probs），并新增针对非叶子 ref/teacher tensor（sin/cos 派生）的 detach 测试 test_skip_actor_forward_only_detaches_fixed_scores_without_precomputed_log_probs。head 版本中该问题已修复。
- --dump-details 与 dumper 参数为何被禁止 (question): guapisolo 回应：commit 12cc66f1 移除 --dump-details 限制，DumpReader 已将 actor log_probs 视为可选列，policy-loss debug dump 仍记录训练 forward 与所选旧策略 log-probs；保留 standalone --save-debug-train-data 门控（因为训练 source patching 会重放被跳过的 forward）；dumper 参数在 commit 7edbd954 已启用。

风险与影响

- 风险：
  1. PPO 指标语义变化（losses.py）：启用 skip 后 ppo_kl 与 pg_clipfrac 在缺少 rollout log-probs 时恒为 0，dashboard 与告警对这些指标的解读会失真；PR body 也明确说明被跳过的 pass 不再产出 rollout/log_probs 指标。
  2. detach 边界回归（loss.py）：若未来放开 --kl-coef != 0 与 skip 的组合，ref_log_probs/teacher_log_probs 的 detach 必须同步验证，否则 #1966 的固定分数 graph 泄漏会复发；当前用校验清单硬性禁止该组合。
  3. dump 数据契约放宽（arguments.py 与 dump reader）：train-data dump 可能缺失 actor log_probs 列，依赖该列的旧 dump 消费者需要兼容处理；PR 已用 test_summary_and_tokens_survive_dump_without_log_probs 覆盖，但自定义下游脚本不在保护范围内。
  4. 配置与运行时两层校验的一致性：静态配置无法覆盖动态 batch 场景，单步约束靠 train_actor 运行时断言兜底，两层逻辑需同步维护。
  5. TIS 组合语义变化：skip + use_tis 时 train_log_probs 来自 detached 训练 log-probs，TIS 比值（训练与 rollout 分数差异）被抹平，实验解读需注意。 - 影响：对训练团队：启用后单步确定性 Megatron PPO 每个 rollout 省去一次完整 actor forward-only pass，对 kimi-k2.5、deepseek v3.2 这类大模型是实打实的吞吐收益；但必须满足严格的配置约束（单 optimizer step、无随机化、kl_coef=0），适合奖励密集、策略单步更新的 RL 场景。对系统：actor 训练循环中 replay 队列消费时机从 standalone scoring 后移到训练 forward，影响 fill_replay_data 与 pop_backward 的调用顺序。对团队流程：4 条 Megatron E2E 模型线默认启用该优化，后续单步训练配置可参照此模式获得性能收益，同时测试矩阵新增了训练 log-prob 复用与生命周期行为的专项覆盖，提升了核心训练路径的可回归性。 - 风险标记：核心训练路径变更，旧策略基线语义改变，

opt-in 默认关闭，运行时断言兜底，dump 数据契约放宽

关联脉络

- PR #1966 (review 中引用的 detach 边界修复 PR): yueming-yuan 在 review 中明确警告跳过路径会让 #1966 的 ref_log_probs 未 detach bug 复发，本 PR 通过无条件 detach 修复并补充回归测试。
- PR #2476 fix(dashboard): zero the trainer log-probs the loss masks out: 该 PR 修复了 dashboard dump_reader 对训练 log-probs 与 loss mask 关系的处理，与本 PR 放宽 dump-details（train-data dump 可省略 actor log_probs 列）形成 dump 数据契约的同一演进线。
- PR #2214 fix(ci): calibrate lora E2E estimates from nightly runs and halve GLM5 lora matrices: 同属 Megatron E2E CI 成本控制与校准方向，本 PR 进一步为 kimi/deepseek/glm5/qwen3 等模型线增加 --skip-actor-forward-only 以削减训练时长。