PR #2706 完整报告

radixark/miles

DeepSeek-V4 support-v2: native megatron implementation as option

合并时间：2026-09-01 04:21

原文链接：http://prhub.com.cn/radixark/miles/pull/2706

执行摘要

- 一句话：DSv4 双后端可选，对齐 Megatron 权重契约，恢复训练
- 推荐动作：值得精读。该 PR 展示了三类可借鉴的设计决策：一是用显式 raise 替代静默 fallback，杜绝用户不知情的实现漂移；二是通过统一 checkpoint 命名契约让两种实现共享同一数据源，并用 assert_checkpoint_is_current 做防御性拒绝；三是利用已有的原子更新组把分散参数在导出时重新聚合（_packed_alphas 的 bucket 传递），是分布式 checkpoint 转换中优雅的处理方式。PR body 中关于 tid2eid、SwiGLU clamp、MTP replay 等调试细节也极具参考价值。

功能与动机

PR body 明确说明动机：'Bring DeepSeek-V4 back on the bumped Megatron, and make the training implementation an explicit choice. Second of the three bump PRs; stacked on #2673 (base bump), which disabled DSv4.' 同时指出旧行为被退役的原因：'June's MILES_DSV4_ATTENTION_BACKEND silently forced miles on TP>1 and that behaviour is retired.' 即不再让实现选择被环境变量静默决定，而是由用户在配置阶段显式声明；两个实现的权重也必须统一命名，否则 checkpoint 只能在写入它的实现上加载。

实现拆解

变更分五个步骤落地：

1. 参数选择与校验入口：新增 miles_plugins/models/deepseek_v4/arguments.py，声明 --dsv4-impl（默认值最终为 megatron，与 PR body 表格中 miles 为默认的描述不一致，是合入前的最终调整）。normalize_dsv4_args 必须在 core_transformer_config_from_args 之前运行，把 impl 选择翻译成 Megatron 的 experimental_attention_variant（megatron→dsv4_hybrid、miles→dsv4）并统一开启 enable_hyper_connections。_validate_impl 对三组非法组合直接 raise：megatron impl 要求 TP=1、megatron 拒绝 tilelang kernel、miles 拒绝 cudnn kernel。assert_checkpoint_is_current 读取 torch_dist 元数据，检测到旧命名（self_attention.wq_a.）即拒绝加载。
2. 权重契约统一：miles_plugins/models/deepseek_v4/deepseek_v4.py 的 DeepSeekV4Attention 全面改用 Megatron 命名（linear_q_down_proj/q_layernorm/linear_q_up_proj/linear_kv_proj/kv_layernorm/linear_o_group_proj/linear_proj），attn_sink、compressor、indexer 移到 core_attention 子模块下，wo_a 从 ColumnParallelLinear 改为裸参数并显式声明 TP 分片，fp32 参数改用 mark_keep_in_fp32。

miles_plugins/mbridge/deepseekv4.py 改名为 deepseek_v4.py 并重写全部映射键；miles_plugins/models/deepseek_v4/ops/hyper_connection.py 删除 217 行 TileKernels mHC 包装，统一走 Megatron 原生超连模块。

3. 导出与量化链路适配：miles/backends/megatron_utils/megatron_to_hf/deepseekv4.py 的 convert_deepseekv4_to_hf 跟进全部新路径，并新增 _packed_alphas：把 Megatron 按 segment 分开的三个 alpha 标量通过原子更新组 bucket 重新打包成 checkpoint 的 [pre, post, res] 单张量。FP8/MXFP8 量化器白名单同步更新（曾因旧命名导致 wkv/wq_a 等未量化直接进引擎，由 commit daed3c02 修复）。
4. 训练脚本与并行配置：scripts/run_deepseek_v4.py 把 --dsv4-impl、--dsa-kernel-backend 同时透传给转换与训练；megatron impl 走 --qkv-format thd、动态 batch、TP/CP 秩转 DP 的 TP1 并行配置；4-layer 转换改为单 GPU（PP1），规避新 Megatron 对 hash-MoE + PP>1 的显式布局断言。miles/utils/debug_utils/run_megatron/worker/main.py 与 checkpoint 转换入口补收插件参数。
5. 测试与 CI 配套：新增 tests/fast/backends/megatron_utils/test_dsv4_arguments.py 覆盖三组非法组合的解析期报错；新增 4-layer megatron impl e2e（dsa_kernel_backend=none，因 CI 镜像无 flash_mla/cudnn-frontend）；command_utils.py 为 convert_checkpoint/fp8_cast_bf16 增加 fcntl 文件锁，串行化同一主机上共享缓存目录的 prepare 竞争。

关键文件：

- miles_plugins/models/deepseek_v4/arguments.py（模块 参数解析；类别 source；类型 data-contract；符号 is_dsv4_model, add_dsv4_arguments, normalize_dsv4_args, _validate_impl）：新增 --dsv4-impl 参数面与三组硬校验，是双后端选择的唯一入口；normalize_dsv4_args 必须在 config 构建前运行，决定了 attention variant 与 post-init 契约。
- miles_plugins/mbridge/deepseek_v4.py（模块 桥接层；类别 source；类型 dependency-wiring；符号 DeepseekV4Bridge, _weight_name_mapping_mcore_to_hf, _weight_to_mcore_format, _build_config）：从 deepseekv4.py 改名并重写全部权重映射键，使 mbridge 能按 Megatron 命名读取 HF checkpoint 并拆分打包的 alpha 张量，是统一权重契约的关键一环。
- miles_plugins/models/deepseek_v4/deepseek_v4.py（模块 模型插件；类别 source；类型 data-contract；符号 DeepSeekV4Attention, DeepSeekV4Attention.init, DeepSeekV4Attention.sharded_state_dict）：DeepSeekV4Attention 按 Megatron 布局整体改名，attn_sink/compressor/indexer 移入 core_attention，wo_a 变裸参数并显式声明分片，fp32 改用 mark_keep_in_fp32，是权重契约统一的核心实现。
- miles_plugins/models/deepseek_v4/ops/hyper_connection.py（模块 超连模块；类别 source；类型 deletion；符号 HCHeadParams, DeepSeekV4HyperConnectionUtil, hc_pre_raw, hc_post_raw）：删除 217 行 TileKernels 超连包装（HCHeadParams、DeepSeekV4HyperConnectionUtil），两个 impl 统一走 Megatron 原生超连模块，是本次架构收敛的标志性删除。
- miles/backends/megatron_utils/megatron_to_hf/deepseekv4.py（模块 导出转换；类别 source；类型 core-logic；符号 get_deepseek_v4_atomic_update_groups,

convert_deepseekv4_to_hf, _packed_alphas）：导出转换器跟随全部新权重名，并新增 _packed_alphas 从原子更新组 bucket 重新打包三 alpha，保证 rollout 权重导出与 HF 布局一致。

- scripts/run_deepseek_v4.py（模块 训练脚本；类别 source；类型 core-logic；符号 ScriptArgs, _prepare_spmd, _get_parallel_config, _train）：训练脚本把 --dsv4-impl 与 kernel backend 透传给转换和训练，并为 megatron impl 提供 TP1/DP 化的并行配置与 thd 动态 batch，是双实现落地的操作入口。
- tests/fast/backends/megatron_utils/test_dsv4_arguments.py（模块 单元测试；类别 test；类型 test-coverage；符号 _parse, test_only_the_dsv4_spec_triggers_normalization, test_impl_selects_the_attention_variant, test_unsupported_combinations_fail_at_parse_time）：覆盖 normalize_dsv4_args 的三种非法组合解析期报错，以及 impl 对 attention variant 的选择，是参数契约的回归保障。
- tests/e2e/megatron/model_scripts/test_deepseek_v4_flash_4layer_megatron_impl_ci.py（模块 E2E 测试；类别 test；类型 test-coverage；符号 _args, prepare, execute）：恢复被 #2673 禁用的 4-layer e2e 并新增 megatron impl 覆盖；因 CI 镜像缺 flash_mla 使用 PyTorch fallback，明确标注 fused-kernel 路径未覆盖。
- miles/utils/external_utils/command_utils.py（模块 工具函数；类别 source；类型 core-logic；符号 convert_checkpoint, fp8_cast_bf16, _exclusive_path_lock）：为共享缓存目录的 prepare 步骤（转换、fp8 降精度）增加 fcntl 排他文件锁，解决多 runner 同主机竞争同一路径的竞态，属于本次大批量转换场景的配套修复。

关键符号：normalize_dsv4_args, _validate_impl, assert_checkpoint_is_current, is_dsv4_model, DeepseekV4Bridge._build_config, DeepseekV4Bridge._weight_to_mcore_format, convert_deepseekv4_to_hf, _packed_alphas, DeepSeekV4Attention.init, DeepSeekV4Attention.sharded_state_dict, _exclusive_path_lock

关键源码片段

miles_plugins/models/deepseek_v4/arguments.py

新增 --dsv4-impl 参数面与三组硬校验，是双后端选择的唯一入口；normalize_dsv4_args 必须在 config 构建前运行，决定了 attention variant 与 post-init 契约。

```
"""DeepSeek-V4 命令行参数。

模型形状通过 Megatron 自己的 flags 传入（--csa-window-size、--o-groups、
--num-residual-streams 等）；插件只声明 Megatron 无法知道的部分：
用哪个实现训练模型。参数解析发生在 tilelang 加载之前，因此本文件
不导入 megatron 或插件内核。
"""

from argparse import ArgumentParser, Namespace

DSV4_SPEC_MODULE = "miles_plugins.models.deepseek_v4.deepseek_v4"

def is_dsv4_model(args: Namespace) -> bool:
```

```
    """是否从 DeepSeek-V4 插件 spec 构建层。"""
    spec = getattr(args, "spec", None)
    return bool(spec) and spec[0] == DSV4_SPEC_MODULE


def add_dsv4_arguments(parser: ArgumentParser) -> ArgumentParser:
    """声明 DeepSeek-V4 参数。"""
    group = parser.add_argument_group(title="deepseek-v4")
    group.add_argument(
        "--dsv4-impl",
        type=str,
        choices=["miles", "megatron"],
        default="megatron", # 最终提交把默认从 miles 切到了 megatron
        help=(
            "选择训练用的 DeepSeek-V4 注意力实现。'miles' 是插件路径"
            "（BSHD、稀疏上下文并行、tilelang 内核、miles 超连），是唯一支持"
            "张量并行的实现；'megatron' 是 Megatron 原生 dsv4_hybrid 路径"
            "（THD、cuDNN 或 unfused 内核、原生超连）。两者读取同一个"
            " HuggingFace checkpoint，但 torch_dist checkpoint 不互通。"
        ),
    )
    return parser


def normalize_dsv4_args(args: Namespace) -> None:
    """把 --dsv4-impl 解析成 Megatron 选择实现的字段。

    必须在 core_transformer_config_from_args 之前运行：attention variant
    决定了 Megatron 在 config 后置初始化时执行哪一套契约检查。
    """
    _validate_impl(args)
    # 两个实现都从 Megatron 自己的模块取 hyper-connections。
    args.enable_hyper_connections = True
    args.experimental_attention_variant = (
        "dsv4_hybrid" if args.dsv4_impl == "megatron" else "dsv4"
    )


def _validate_impl(args: Namespace) -> None:
    """非法组合直接 raise，绝不静默 fallback。

    旧版 MILES_DSV4_ATTENTION_BACKEND 在 TP>1 时静默强制 miles 路径，
    该行为已被移除：宁可启动失败，也不让训练跑在用户不知情的实现上。
    """
    kernel_backend = getattr(args, "dsa_kernel_backend", None)
    if args.dsv4_impl == "megatron":
        # 原生 dsv4_hybrid 尚未支持张量并行，TP 秩只能转成 DP。
        if args.tensor_model_parallel_size > 1:
            raise ValueError(
```

```
                f"--dsv4-impl megatron requires tensor-model-parallel-size 1, got "
                f"{args.tensor_model_parallel_size}. Use --dsv4-impl miles for tensor parallelism."
            )
        if kernel_backend == "tilelang":
            raise ValueError(
                "--dsv4-impl megatron does not support --dsa-kernel-backend tilelang; "
                "use 'cudnn' for fused kernels or 'none' for the PyTorch fallback."
            )
    elif kernel_backend == "cudnn":
        # miles 路径自带 tilelang 内核，会忽略 cuDNN 配置。
        raise ValueError(
            "--dsv4-impl miles runs its own tilelang kernels and ignores cuDNN; "
            "drop --dsa-kernel-backend or switch to --dsv4-impl megatron."
        )
```

miles_plugins/mbridge/deepseek_v4.py

从 deepseekv4.py 改名并重写全部权重映射键，使 mbridge 能按 Megatron 命名读取 HF checkpoint 并拆分打包的 alpha 张量，是统一权重契约的关键一环。

```
import torch
from megatron.core.transformer.enums import AttnBackend

from mbridge.core import register_model
from mbridge.models import DeepseekV3Bridge


@register_model("deepseek_v4")
class DeepseekV4Bridge(DeepseekV3Bridge):
    # HF checkpoint 把三个超连 alpha 打包成一个 [pre, post, res] 张量，
    # 而模型按段各存一个参数（与原生模块一致），这里记录每段的下标。
    _ALPHA_SEGMENTS = {"alpha_pre": 0, "alpha_post": 1, "alpha_res": 2}

    def _weight_to_mcore_format(self, mcore_weights_name: str, hf_weights: list[torch.Tensor]) -
    > torch.Tensor:
        # 从打包张量中切出当前参数对应的那一段，并转 float32。
        for suffix, segment in self._ALPHA_SEGMENTS.items():
            if mcore_weights_name.endswith(suffix):
                return hf_weights[0].reshape(-1)[segment : segment + 1].float()

        # V4 有若干参数保持 fp32（attn_sink、compressor.ape、hc_* 参数，
        # 均标记 _keep_fp32）。基类桥会把所有加载权重降成 self.dtype
        #  （bf16），那会在到达 fp32 的 mcore 参数前静默舍入；这里对
        # fp32 源权重临时关闭 dtype 降级，只保留基类 reshape。
        if len(hf_weights) == 1 and hf_weights[0].dtype == torch.float32:
            saved_dtype = getattr(self, "dtype", None)
            self.dtype = None
            try:
                return super()._weight_to_mcore_format(mcore_weights_name, hf_weights)
            finally:
```

```
                self.dtype = saved_dtype
        return super()._weight_to_mcore_format(mcore_weights_name, hf_weights)
```

miles/backends/megatron_utils/megatron_to_hf/deepseekv4.py

导出转换器跟随全部新权重名，并新增 _packed_alphas 从原子更新组 bucket 重新打包三 alpha，保证 rollout 权重导出与 HF 布局一致。

```
# Megatron 把每个超连位置的三个 alpha 存成独立标量，分别对应一个 bias 段；
# checkpoint 以 [pre, post, res] 单张量保存。三个参数因共享同一个原子更新组
#  （AtomicUpdateGroup）而同时在 bucket 中出现，因此这里能直接聚合成完整张量。

def _packed_alphas(name: str, param, bucket):
    """把一个超连位置的三个 alpha 按 checkpoint 布局打包。"""
    import torch

    prefix = name.rsplit(".", 1)[0]
    return torch.cat(
        [bucket[f"{prefix}.alpha_{seg}"].reshape(1) for seg in ("pre", "post", "res")]
    )


def get_deepseek_v4_atomic_update_groups():
    # 原子更新组保证同一组参数在更新时一起发出；alpha 三段必须同组，
    # 否则 _packed_alphas 取不到完整的 bucket。
    return [
        AtomicUpdateGroup(key, suffixes)
        for key, suffixes in [
            ("wqkv_a", (".self_attention.linear_q_down_proj.weight", ".self_attention.linear_kv_proj.
            weight")),
            (
                "self_attention_hc_alphas",
                (
                    ".self_attention_hyper_connection.alpha_pre",
                    ".self_attention_hyper_connection.alpha_post",
                    ".self_attention_hyper_connection.alpha_res",
                ),
            ),
            (
                "mlp_hc_alphas",
                (
                    ".mlp_hyper_connection.alpha_pre",
                    ".mlp_hyper_connection.alpha_post",
                    ".mlp_hyper_connection.alpha_res",
                ),
            ),
            (
                "compressor_wkv_gate",
                (
                    ".self_attention.core_attention.compressor.linear_wkv.weight",
```

```
                ".self_attention.core_attention.compressor.linear_wgate.weight",
            ),
        ),
        (
            "indexer_compressor_wkv_gate",
            (
                ".self_attention.core_attention.indexer.compressor.linear_wkv.weight",
                ".self_attention.core_attention.indexer.compressor.linear_wgate.weight",
            ),
        ),
    ]
]
```

```
def convert_deepseekv4_to_hf(args, name, param, bucket=None):
    # 转换器按参数逐个被调用；只有 alpha_pre 会触发 _packed_alphas 打包，
    # alpha_post / alpha_res 返回空列表，避免重复写出同名的 hc_*_scale。
    ...
    if rest == "self_attention_hyper_connection.alpha_pre":
        return [(f"model.layers.{layer_idx}.hc_attn_scale", _packed_alphas(name, param, bucket))]
    elif rest.startswith("self_attention_hyper_connection.alpha_"):
        return []
    ...
```

评论区精华

7 条 review 评论集中在 5 个话题：

- indexer replay 的未来取舍（Zhichenzzz）：'maybe later we should opt this indexer replay, since now it is easy to cause host oom / and long-latency. and this will be very help for the following models, e.g., glm, qwen'。作者以 no-op seam 方式合入（manager 默认禁用，stream_idx=layer_id 预留），未在本次 PR 中解决。
- mbridge 文件改名（Zhichenzzz 建议 deepseekv4.py → deepseek_v4.py），已通过 commit 11b5421d 落实。
- 原子更新组是否伤害异步训练（Zhichenzzz 转述 claude 判断并持异议）：'claude code says this will harm async train, but i dont agree (maybe quick check'。作者回复 'just checked, this should not harm the training'，结论为不影响。
- FP8 量化器残留旧名字（Zhichenzzz 问 'old names?'），作者回复 'good catch' 并随后修复 indexer wk 路径。
- AMD 脚本是否需要跟进（Zhichenzzz 问 'do we also need to modify amd script into the latest impl?'），通过 commit 19e3f8d6 处理：ROCm 无 cudnn 路径，pin --dsv4-impl miles。
- indexer replay seam 的后续取舍 (performance): 作者以 no-op/transparent seam 方式合入（manager 默认禁用，仅注册 stream_idx），未在本 PR 改变默认行为；完整 record/replay-overlap 验证仍需 RL 运行。

- mbridge 文件命名 (style): 已通过 commit 11b5421d 将文件改名为 deepseek_v4.py 并 isort 整理导入顺序。
- 原子更新组改动是否影响异步训练 (performance): 作者回复 'just checked, this should not harm the training'，确认不影响异步训练。
- FP8 量化器白名单残留旧名 (correctness): 作者回复 'good catch'，随后 commit 3d71ce062 更新 v4 indexer wk 的新路径；commit daed3c02 也修复了 wkv/wq_a/wo_b 与 indexer 投影未量化进 FP8 引擎的问题。
- AMD/ROCm 脚本是否跟进新 impl (question): commit 19e3f8d6 处理：AMD 无 cudnn 路径，pin --dsv4-impl miles，并刷新过期 snapshot。

风险与影响

- 风险：主要风险集中在四点：
- Checkpoint 契约破坏：权重命名全面切换后，旧 torch_dist checkpoint 无法加载（assert_checkpoint_is_current 会显式拒绝），存量 miles impl 用户必须从 HF checkpoint 重新转换，存在迁移成本。
- 默认值不一致：add_dsv4_arguments 的 default="megatron" 与 PR body 表格中声称的 miles 默认不符，属于合入前的默认切换（commit bed70932），但 body 未同步，存在文档误导风险。
- 验证缺口：cudnn kernel 后端（镜像缺 flash_mla）、TP>1（设计上被拒）、megatron impl 多步长稳运行均未覆盖；PR body 自述 4-layer e2e 中所有样本 reward=0，GRPO advantage 与 grad_norm 数值未真正验证。
- 并发锁适用范围：command_utils.py 的 fcntl.flock 只对同一主机上的进程有效，多节点共享缓存目录时锁不成立；另外 FP8/MXFP8 白名单本次曾漏更新，说明权重再次改名时量化链路存在静默跳过风险。
- 影响：对用户：DSv4 训练默认切换到 megatron 原生实现，TP>1 用户必须显式指定 --dsv4-impl miles；旧 checkpoint 需重新转换。对系统：权重契约统一后，一个 HF checkpoint 可同时服务两种实现，且导出转换保持 impl 无关；删除 217 行自研超连代码，减少内核维护面；indexer replay seam 为 GLM、Qwen 等后续模型复用铺路。对性能：8xGB300 50-step A/B 显示 megatron impl 相对 miles 全周期仅慢 1.7%（200.5s vs 197.0s），训练阶段反而快 4.4%，无回归迹象。
- 风险标记：checkpoint 命名契约破坏，默认实现切换为 megatron，TP>1 组合受限，量化白名单曾漏改，cudnn 后端未覆盖

关联脉络

- PR #2673 Megatron base bump: PR body 明确说明本 PR 是 'Second of the three bump PRs; stacked on #2673 (base bump), which disabled DSv4'。#2673 升级 Megatron 并禁用 DSv4 4-layer e2e，本 PR 在其之上恢复 DSv4 并新增原生实现选项。
- PR #2779 test(dsv4): accept ValueError from reasoning-effort validation: 同为 DeepSeek-V4 支持线的测试适配，说明 DSv4 功能在 bump 期间存在多处联动修复，本 PR 则集中恢复其训练主链路。