

PR #55095 完整报告

vllm-project/vllm

[Bugfix] Fall back to full decode graphs for noncompiled models

合并时间: 2026-09-11 16:56

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/55095>

执行摘要

- 一句话: 非编译模型回退到 full decode 图, 修复 ROCm 图模式解析失败
- 推荐动作: 值得精读, 尤其适合负责 cudagraph/ 编译配置、ROCm 平台默认值与投机解码的同学。重点关注三个设计决策: 一是「piecewise 提供者」的判定放在 runner 侧 (has_compiled_submodule) 而非配置层, 这是本 PR 能保持 CUDA 默认行为不变的关键; 二是 without_piecewise() 让降级尽量保住 full decode graph, 而不是一律退到 eager; 三是归一化在平台更新前后各跑一次的顺序契约, 以及降级到 NONE 时清理捕获尺寸。阅读时注意区分早期 revision (架构白名单 _uses_noncompiled_cudagraph_path + _piecewise_cudagraph_provider_available) 与最终落地版本 (通用条件 + LucasWilkinson 的 #56191 简化), review 评论中的报错栈对应的是已被替换的中间实现。建议后续: 补齐非编译路径的端到端精度 / 性能长跑基线, 持续跟踪 issue #52644 以决定 gfx950 eager 默认值何时撤销。

功能与动机

ROCm 出于性能考虑默认关闭 breakable CUDA graph, 而 DeepSeek V3.2、GLM DSA、DeepSeek V4 等模型路径没有活跃的 torch.compile 边界, 正常情况下依赖 breakable graph 才能做 piecewise capture, 于是 O2 下会被解析成不可用的 PIECEWISE 路径。PR body 明确指出: 「Apply the fallback only to platform-selected model paths without an active torch-compile boundary」, 并说明 「On gfx942 the fallback keeps uniform decode captured through FULL_DECODE_ONLY, while gfx950 DeepSeek V4 stays eager until its graph-pool correctness issue is resolved」。gfx950 的 eager 默认值直接对应仍处于 open 状态的 issue #52644 (MI350/MI355 上 DeepSeek V4 使用 MRV2 + FULL_DECODE_ONLY 图出现精度下降)。PR body 还列出因果链: #52861、#53155、#53183 引入了相关默认配置, #54852 暴露了这个组合问题。

实现拆解

1. 把「piecewise 捕获提供者是否可用」变成 runner 侧的事实探测 (vllm/v1/worker/gpu/model_runner.py): 在 GPUModelRunner.initialize_kv_cache 中计算 piecewise_capture_available = bool(envs.VLLM_USE_BREAKABLE_CUDAGRAPH or has_compiled_submodule(self.model)), 新增导入 has_compiled_submodule (来自 vllm/v1/worker/gpu/cudagraph_utils.py), 并把该布尔值透传给 resolve_cudagraph_mode_and_sizes。这样判据来自实际加载后的模型结构, 而不是配置

层猜测；影响是所有依赖 cudagraph 初始化的分支 (cudagraph_manager、speculator) 都拿到已归一化的模式。

2. 在 CUDAGraphMode 上提供降级原语 (vllm/config/compilation.py) : 新增 `without_pieewise()`, 把 `PIECEWISE` → `NONE`、`FULL_AND_PIECEWISE` → `FULL_DECODE_ONLY`, 其余模式原样返回。语义上保证「能保住 full decode graph 就不退到完全 eager」, 这是 ROCm gfx942 性能不塌的关键。
3. 解析函数新增降级分支 (vllm/config/compilation.py) : `resolve_cudagraph_mode_and_sizes` 增加 `pieewise_capture_available: bool = True` 参数 (默认值保持既有调用方行为), 在 attention 后端能力驱动的自动降级之后、full graph 硬校验之前插入 `not pieewise_capture_available and requires_pieewise_compilation()` 的 warning + 降级逻辑。
4. 配置层归一化与调用顺序 (vllm/config/vllm.py) : 把原来内联的「Overriding to NONE」代码块抽成 `_normalize_pieewise_cudagraph_mode(breakable_cudagraph_enabled=...)`, 并调用两次——`_maybe_override_dynamic_sd_cudagraph_mode()` 之后一次, `current_platform.check_and_update_config()` 之后再一次 (平台 /connector 兼容性检查可能把 full 模式换成 `PIECEWISE`)。降级到 `NONE` 时顺带清空 `max_cudagraph_capture_size` 与 `cudagraph_capture_sizes`, 避免 `post_init_cudagraph_sizes()` 继续为不存在的图预留缓冲。PR body 里描述的「O2/O3 默认 `FULL_DECODE_ONLY`、O0/O1 默认 `NONE`」正是该机制的副产品: O2/O3 默认 `FULL_AND_PIECEWISE` 降级为 `FULL_DECODE_ONLY`, O0/O1 默认 `PIECEWISE` 降级为 `NONE`。
5. 平台默认值 (vllm/platforms/rocm.py) : `RocmPlatform.apply_config_platform_defaults` 在满足 `cudagraph_mode is None + on_gfx950()` + `use_v2_model_runner` + 架构属于 `DeepseekV4ForCausalLM / DeepseekV4ForConditionalGeneration` 时默认设为 `NONE`, 注释直接引用 issue #52644; 显式传入的模式不被覆盖。
6. 自适应验证语义修正 (vllm/config/vllm.py + vllm/v1/worker/gpu/spec_decode/adaptive_verification.py + vllm/v1/worker/gpu/model_runner.py) : `_validate_adaptive_verification` 由「`cudagraph_mode == NONE` 才报错」改为 `not has_full_cudagraphs()` 才报错 (接受 `FULL / FULL_DECODE_ONLY / FULL_AND_PIECEWISE`, 拒绝 `NONE / PIECEWISE`) ; 新增 `resolve_adaptive_cudagraph_mode(mode, *, pieewise_capture_available)`, 让 runner 不再无条件写死 `FULL_AND_PIECEWISE`, `pieewise` 不可用时退到 `FULL_DECODE_ONLY`, 并断言输入模式确实带 full graph。
7. 测试与验证配套: `tests/test_config.py` +122 行, 新增 4 个测试 (`test_rocm_gfx950_deepseek_v4_cudagraph_default` 覆盖 5 组架构 /runner/ 显式模式组合、`test_resolve_cudagraph_mode_uses_loaded_pieewise_provider` 覆盖 4 组降级矩阵、`test_late_pieewise_restrictions_without_compilation` 用 `EngineArgs` 做端到端断言并检查捕获尺寸被清空、`test_adaptive_verification_requires_full_cudagraphs` 覆盖 4 组模式) ; `tests/v1/spec_decode/test_adaptive_verification.py` 新增 `test_resolve_adaptive_cudagraph_mode` 5 组用例。作者报告最终合并前配置 / 自适应验证 / 图管理器等相关套件 332 通过 7 跳过, 并在 4xMI300X 上跑通 DeepSeek V4 Vision 检查点、7 项语义探针与 1319 题 GSM8K (0.942381) 。

关键文件：

- vllm/config/vllm.py (模块 配置归一化；类别 source；类型 core-logic；符号 `_normalize_pieewise_cudagraph_mode`, `_validate_adaptive_verification`)：配置层归一化枢纽：新增 `_normalize_pieewise_cudagraph_mode` 并在平台更新前后各调用一次，同时把自适应验证校验改为 `has_full_cudagraphs()`；review 中关于强制 `mode=NONE` 副作用与 `VLLM_COMPILE` 误伤的讨论都落在这个文件。
- vllm/config/compilation.py (模块 图模式定义；类别 source；类型 core-logic；符号 `CUDAGraphMode.without_pieewise`, `resolve_cudagraph_mode_and_sizes`)：降级机制的原语与入口：新增 `CUDAGraphMode.without_pieewise()`，并在 `resolve_cudagraph_mode_and_sizes` 中新增 `pieewise_capture_available` 参数与降级分支，是本 PR 所有回退行为的基础。
- vllm/platforms/rocm.py (模块 平台默认值；类别 source；类型 configuration；符号 `RocmPlatform.apply_config_platform_defaults`)：新增的 ROCm 平台默认值：gfx950 + V2 runner + DeepSeek V4 架构在未显式指定 `cudagraph_mode` 时默认 `NONE`，直接对应 open 状态的 issue #52644，是本 PR 中唯一改变默认运行时行为的平台侧改动。
- vllm/v1/worker/gpu/model_runner.py (模块 模型运行器；类别 source；类型 dependency-wiring；符号 `GPUModelRunner.initialize_kv_cache`, `has_compiled_submodule`, `resolve_adaptive_cudagraph_mode`)：提供判据来源与自适应验证接线：`initialize_kv_cache` 计算 `pieewise_capture_available` (`envs.VLLM_USE_BREAKABLE_CUDAGRAPH` 或 `has_compiled_submodule`)，并把自适应验证从无条件写 `FULL_AND_PIECEWISE` 改为调用 `resolve_adaptive_cudagraph_mode`，是 Rohan138 review 提问的落点。
- vllm/v1/worker/gpu/spec_decode/adaptive_verification.py (模块 自适应验证；类别 source；类型 core-logic；符号 `resolve_adaptive_cudagraph_mode`)：新增 `resolve_adaptive_cudagraph_mode`：把「自适应验证在有 full graph 前提下应选哪种模式」的决策从 runner 抽成可测试的纯函数，`pieewise` 不可用时退到 `FULL_DECODE_ONLY`，避免覆盖配置层的回退结果。
- tests/test_config.py (模块 配置测试；类别 test；类型 test-coverage；符号 `test_rocm_gfx950_deepseek_v4_cudagraph_default`, `test_resolve_cudagraph_mode_uses_loaded_pieewise_provider`, `test_late_pieewise_restrictions_without_compilation`, `test_adaptive_verification_requires_full_cudagraphs`)：配置回归核心测试：新增 gfx950 DeepSeek V4 默认值、provider-aware 解析矩阵、late pieewise 限制的端到端断言（含捕获尺寸清空）与自适应验证校验矩阵，是本 PR 防止回退的主要安全网。
- tests/v1/spec_decode/test_adaptive_verification.py (模块 投机解码测试；类别 test；类型 test-coverage；符号 `test_resolve_adaptive_cudagraph_mode`)：为新增的 `resolve_adaptive_cudagraph_mode` 提供 5 组模式 / 可用性矩阵测试，覆盖 review 中 CodeRabbit 指出的 `FULL_AND_PIECEWISE` + 可用捕获组合。

关键符号：`_normalize_pieewise_cudagraph_mode`,

`CUDAGraphMode.without_pieewise`, `resolve_cudagraph_mode_and_sizes`,

`resolve_adaptive_cudagraph_mode`, `RocmPlatform.apply_config_platform_defaults`,

VllmConfig._validate_adaptive_verification, GPUModelRunner.initialize_kv_cache, has_compiler_submodule

关键源码片段

vllm/config/vllm.py

配置层归一化枢纽：新增 `_normalize_pieewise_cudagraph_mode` 并在平台更新前后各调用一次，同时把自适应验证校验改为 `has_full_cudagraphs()`；review 中关于强制 `mode=NONE` 副作用与 `VLLM_COMPILE` 误伤的讨论都落在这个文件。

vllm/config/vllm.py: 把「需要 `pieewise capture` 但没有提供者」的图模式降级为可执行模式。
该函数在 `__post_init__` 中被调用两次：平台配置更新之前一次，之后再一次。

```
def _normalize_pieewise_cudagraph_mode(
    self, *, breakable_cudagraph_enabled: bool
) -> None:
    compilation_config = self.compilation_config
    # 三个条件同时成立才降级：
    # 1) 目标模式确实需要 pieewise capture (PIECEWISE / FULL_AND_PIECEWISE) ;
    # 2) 编译模式不是 VLLM_COMPILE，即没有 torch.compile 产生的分段捕获；
    # 3) breakable CUDA graph 未启用，说明 runner 侧也没有捕获提供者。
    if (
        compilation_config.cudagraph_mode.requires_pieewise_compilation()
        and compilation_config.mode != CompilationMode.VLLM_COMPILE
        and not breakable_cudagraph_enabled
    ):
        # 不再一律退到 NONE： FULL_AND_PIECEWISE -> FULL_DECODE_ONLY，
        # 这样 gfx942 上 uniform decode 仍能走图，避免性能大幅回退；
        # O0/O1 的默认 PIECEWISE 则自然退到 NONE。
        fallback_mode = compilation_config.cudagraph_mode.without_pieewise()
        logger.info_once(
            "Cudagraph mode %s is not compatible with compilation mode %s. "
            "Overriding to %s.",
            compilation_config.cudagraph_mode,
            compilation_config.mode,
            fallback_mode,
        )
        compilation_config.cudagraph_mode = fallback_mode
        # 退到 NONE 时必须清理陈旧的捕获尺寸，否则后续 post_init_cudagraph_sizes()
        # 仍会按已经不存在的图预留缓冲。
        if fallback_mode == CUDAGraphMode.NONE:
            compilation_config.max_cudagraph_capture_size = 0
            compilation_config.cudagraph_capture_sizes = []

# 调用点 1：动态投机解码相关的图模式改写之后
self._maybe_disable_dynamic_sd_for_data_parallel()
self._maybe_override_dynamic_sd_cudagraph_mode()
```

```

self._normalize_pieewise_cudagraph_mode(
    breakable_cudagraph_enabled=breakable_cudagraph_enabled
)

# 调用点 2: 平台 check_and_update_config() 之后必须再归一化一次,
# 因为平台或 connector 的兼容性检查可能把 full 模式换成 PIECEWISE,
# 且归一化必须发生在自适应验证等消费者读取 cudagraph_mode 之前。
current_platform.check_and_update_config(self)
self._normalize_pieewise_cudagraph_mode(
    breakable_cudagraph_enabled=breakable_cudagraph_enabled
)

# 下游消费者: 只接受真正的 full CUDA graph 模式
self._validate_batch_sharded_sampling()
self._validate_adaptive_verification()

```

vllm/config/compilation.py

降级机制的原语与入口: 新增 `CUDAGraphMode.without_pieewise()`, 并在 `resolve_cudagraph_mode_and_sizes` 中新增 `pieewise_capture_available` 参数与降级分支, 是本 PR 所有回退行为的基础。

```

class CUDAGraphMode(enum.Enum):
    NONE = 0
    PIECEWISE = 1
    FULL = 2
    FULL_DECODE_ONLY = (FULL, NONE)
    FULL_AND_PIECEWISE = (FULL, PIECEWISE)

    def requires_pieewise_compilation(self) -> bool:
        return self.has_mode(CUDAGraphMode.PIECEWISE)

    def without_pieewise(self) -> "CUDAGraphMode":
        # 本次新增: 把需要 pieewise capture 的模式映射为不需要它的等价模式。
        # PIECEWISE 本身没有 full graph 可保留, 只能退到 NONE;
        # FULL_AND_PIECEWISE 可以只保留 decode full graph, 性能损失更小。
        if self == CUDAGraphMode.PIECEWISE:
            return CUDAGraphMode.NONE
        if self == CUDAGraphMode.FULL_AND_PIECEWISE:
            return CUDAGraphMode.FULL_DECODE_ONLY
        return self # NONE / FULL / FULL_DECODE_ONLY 原样返回

def resolve_cudagraph_mode_and_sizes(
    self,
    min_cg_support: "AttentionCGSupport",
    min_cg_attn_backend: str | None,
    uniform_decode_query_len: int = 1,
    use_v2_model_runner: bool = False,
    tensor_parallel_size: int = 1,

```

```

kv_cache_config: "KVCacheConfig | None" = None,
max_num_reqs: int | None = None,
is_profiling: bool = False,
piecewise_capture_available: bool = True, # 新增, 默认值保持既有调用方行为
) -> CUDAGraphMode:
    # ... (前面是按 attention 后端能力做的自动降级) ...
    # 关键新增: 内存里加载的模型既没有 compiled submodule, 也没有启用
    # breakable CUDA graph 时, 任何仍要求 piecewise capture 的模式都必须降级,
    # 否则捕获阶段会失败或产生静默错算。
    if (
        not piecewise_capture_available
        and cudagraph_mode.requires_pieewise_compilation()
    ):
        fallback_mode = cudagraph_mode.without_pieewise()
        logger.warning_once(
            "Cudagraph mode %s requires pieewise capture, but the loaded "
            "model provides neither a compiled submodule nor breakable CUDA "
            "graphs. Overriding to %s.",
            cudagraph_mode,
            fallback_mode,
        )
        cudagraph_mode = fallback_mode

    # 降级之后仍然保留硬校验: 要求 full graph 但 attention 后端完全不支持时直接报错,
    # 保证降级只发生在真正安全的方向上。
    if (
        cudagraph_mode.has_full_cudagraphs()
        and min_cg_support == AttentionCGSupport.NEVER
    ):
        raise ValueError(
            f"CUDAGraphMode.{cudagraph_mode.name} is not "
            f"supported with {min_cg_attn_backend} backend"
        )
    # ... (后续是 MRV1 的 spec-decode 捕获尺寸调整, 本次未改动) ...

```

vllm/platforms/rocm.py

新增的 ROCm 平台默认值: gfx950 + V2 runner + DeepSeek V4 架构在未显式指定 cudagraph_mode 时默认 NONE, 直接对应 open 状态的 issue #52644, 是本 PR 中唯一改变默认运行时行为的平台侧改动。

```

@classmethod
def apply_config_platform_defaults(cls, vllm_config: "VllmConfig") -> None:
    from vllm._aiter_ops import rocm_aiter_ops
    from vllm.config.compilation import CUDAGraphMode

    compilation_config = vllm_config.compilation_config
    model_config = vllm_config.model_config

    # gfx950 (MI350/MI355) 上, DeepSeek V4 走 V2 model runner 时的

```

```

# FULL_DECODE_ONLY 图存在尚未解决的精度回归 (issue #52644) ,
# 因此只在用户没有显式指定 cudagraph_mode 时默认改为 eager。
# 显式传入的模式 (例如 FULL_DECODE_ONLY) 仍然被保留,
# 由用户自行承担该已知风险; 矩阵由
# test_rocm_gfx950_deepseek_v4_cudagraph_default 覆盖。
if (
    compilation_config.cudagraph_mode is None
    and model_config is not None
    and on_gfx950()
    and vllm_config.use_v2_model_runner
    and model_config.architecture
    in {
        "DeepseekV4ForCausalLM",
        "DeepseekV4ForConditionalGeneration",
    }
):
    # Default to eager after reported gfx950/MRV2 accuracy regressions:
    # https://github.com/vllm-project/vllm/issues/52644
    compilation_config.cudagraph_mode = CUDAGraphMode.NONE

# 以下 AITER 相关默认值 (+quant_fp8 / grouped_topk / sparse_attn_indexer)
# 在本 PR 之前就已存在, 此处保留以标明新增默认值的插入位置。
use_aiter_fused_moe = rocm_aiter_ops.is_fused_moe_enabled()
use_aiter_fp8_linear = rocm_aiter_ops.is_linear_fp8_enabled()
use_aiter_fused_se = rocm_aiter_ops.is_fusion_moe_shared_experts_enabled()
# ... (后续为原有的 custom_ops 调整逻辑) ...

```

评论区精华

讨论集中在四个点: (1) okorzh-amd 指出新增的 **VLLM_COMPILE** 校验会击穿 **NgramProposerGPU.__init__** 内部构造的辅助 **VllmConfig** (硬编码 **VLLM_COMPILE + NONE**), 作者把校验收窄到「cudagraph_mode 为 None 或需要 piecewise」, 最终实现里该路径只做降级与日志、不再抛错; (2) 同一位 reviewer 指出早期版本强制 **mode = NONE** 会抢先于「mode is None → VLLM_COMPILE」默认值, 静默改变 ROCm 内核分派 (custom_ops none→all、ir_enable_torch_wrap、IR 优先级、AITER rms_norm 丢失), 作者删除该赋值并补充回归测试与 MI300X O2 探针; (3) liulanze 指出自适应验证需要的是 full CUDA graph 而非 piecewise provider, 旧校验既会误杀本 PR 新增的 **NONE + FULL_DECODE_ONLY** 回退, 又会放过 **VLLM_COMPILE + PIECEWISE**, 作者改用 **has_full_cudagraphs()** 并同步修正 runner; (4) 维护者 njhill / LucasWilkinson 认为实现过于绕, 最终把 LucasWilkinson 的简化提案 (#56191) 作为 commit 59e5b407 直接合入本 PR。Fangzhou-Ai 独立复测确认该改动在 MI355X + DeepSeek-V4-Pro FP4 上「by design inert」且性能持平。

- 新增的 **VLLM_COMPILE** 校验会击穿 **NgramProposerGPU** 的辅助 **VllmConfig** (correctness): 作者把校验收窄为「cudagraph_mode 为 None 或 requires_piecewise_compilation() 才拒绝」, 并说明显式非 piecewise 模式 (**NONE / FULL / FULL_DECODE_ONLY**) 保持有效; 在最终简化版本中该路径只做

without_pieewise() 降级并打日志，不再抛错。

- 强制 mode=NONE 会静默改变 ROCm 内核分派与 AITER rms_norm 优先级 (performance): 作者删除该赋值，让未设置的编译模式走正常解析器；回归测试断言 compilation mode 保持未设置，并用 MI300X O2 探针确认 AITER 仍是选中的 RMSNorm 实现。
- 自适应验证应校验 full CUDA graph，而不是 pieewise provider (design): 作者采纳：校验改为 not has_full_cudagraphs() 才报错，并同步修正 model_runner 侧——保留 FULL_DECODE_ONLY，pieewise 不可用时退到它，而不是无条件写 FULL_AND_PIECEWISE。
- 为何要在 model_runner 里改写自适应验证的 cudagraph_mode (question): 作者解释该回退位于共享配置代码，因此也覆盖 CUDA 上显式 VLLM_USE_BREAKABLE_CUDAGRAPH=0 的 DeepSeek V4 场景：配置层已选 FULL_DECODE_ONLY，若 runner 仍无条件写 FULL_AND_PIECEWISE 会撤销回退并触发捕获提供者检查；默认启用 breakable graph 的 CUDA 路径不受影响。
- 共享配置层的 provider 检查是否会波及默认 CUDA 路径 (question): 结论是默认 CUDA 行为不变，CUDA 仅在显式关闭 breakable graph 时受影响；该检查保留在通用路径以保证一致性。
- 维护者认为实现过于绕，要求先合入 LucasWilkinson 的简化提案 (design): LucasWilkinson 的 #56191 提案以 commit 59e5b407 直接合成到本 PR：删除架构白名单 (_uses_noncompiled_cudagraph_path、_pieewise_cudagraph_provider_available、NON_COMPILED_CUDAGRAPH_FALLBACK_ARCHITECTURES)，统一为 requires_pieewise_compilation() + mode != VLLM_COMPILE + not breakable_cudagraph_enabled，并引入 without_pieewise(); 作者仍承诺后续可继续收敛复杂度。
- 自适应验证测试矩阵缺少 FULL_AND_PIECEWISE + 捕获可用组合 (testing): 最终测试参数表已包含 (FULL_AND_PIECEWISE, True → FULL_AND_PIECEWISE) 用例，该 nitpick 被吸收。
- CUDAGraphMode 在配置文件中被重复 lazy import (style): 属于清理性建议，最终代码未在可见片段中确认是否已改为顶层导入，可视为遗留的后续清理项。

风险与影响

- 风险：
 1. 核心配置路径回归面扩大：_normalize_pieewise_cudagraph_mode 位于 VllmConfig.__post_init__ 的通用路径，CUDA 默认不再受影响的前提是「CUDA 自动启用 breakable graph」，但用户显式设置 VLLM_USE_BREAKABLE_CUDAGRAPH=0 时行为会从 NONE 变成 FULL_DECODE_ONLY（至少是行为变化）。
 2. 降级判据依赖编译模式语义：条件中的 mode != CompilationMode.VLLM_COMPILE 把「能否 pieewise capture」与编译模式绑定，早期版本正是因为直接改写 mode 而影响了 AITER rms_norm 选择与 IR 优先级；虽然该赋值已被删除，但任何未来在平台 /connector 层改写 compilation mode 的改动都可能重新引入不一致。

3. 探测为假阴性时静默降级: `piecewise_capture_available` 只认 `VLLM_USE_BREAKABLE_CUDAGRAPH` 或 `has_compiled_submodule(self.model)`; 若某条模型路径通过其他方式提供分段捕获, 会被无提示地降为 `FULL_DECODE_ONLY` (性能损失而非正确性损失)。
4. gfx950 默认 eager 是性能换正确性: `vllm/platforms/rocm.py` 让 DeepSeek V4 MRV2 在 MI350/MI355 上默认退到 eager, 直到 issue #52644 解决; 期间该配置在 MI350/MI355 上会损失 uniform decode 的图收益。
5. 自适应验证仍存在跨层不一致风险: `config` 校验 (`has_full_cudagraphs()`) 与 `runner` 内 `resolve_adaptive_cudagraph_mode` 的 `assert` 分处两层, 若后续在两者之间再次改写模式, 可能出现 `AssertionError` 而不是友好的配置报错。
6. 辅助配置构造模式脆弱: `NgramProposerGPU` 之类在库内部构造第二个 `VllmConfig` 的代码, 暴露了配置校验被误伤的隐患; 本 PR 修好了一处, 但没有通用防线。
 - 影响: 用户侧: ROCm (尤其 gfx942) 用户在无 `torch.compile` 边界的模型路径上不再启动失败或静默错算, uniform decode 仍由图捕获; gfx950 上的 DeepSeek V4 MRV2 用户默认退到 eager, 吞吐可能下降但规避已知精度回归; CUDA 默认路径不变, 仅在显式关闭 breakable graph 时行为变化。系统侧: cudagraph 模式解析新增一层「provider 可用性」判据与一个可复用的 `without_pieewise()` 原语, 配置归一化被提升为平台更新前后各执行一次的双保险, MRV2 的 `GPUModelRunner.initialize_kv_cache` 与 `adaptive verification` 接线随之调整。团队侧: 该区域 (cudagraph 模式解析 + ROCm 平台默认值) 已成为多个 PR 的交汇点, 维护者明确要求后续继续简化, 说明此处复杂度仍在累积。
 - 风险标记: 核心配置路径变更, ROCm 默认图模式变更, 依赖未关闭的精度 Issue #52644, 非编译路径缺少端到端性能基线, 辅助 `VllmConfig` 构造易被新校验误伤

关联脉络

- PR #56191 [Proposal] Simplify noncompiled cudagraph fallback: LucasWilkinson 的简化提案以 commit 59e5b407 直接合入本 PR, 替换了早期基于架构白名单 (`_uses_noncompiled_cudagraph_path / _piecewise_cudagraph_provider_available`) 的复杂判定, 最终形成 `_normalize_pieewise_cudagraph_mode + without_pieewise` 的组合。
- PR #56379 [EC] Automatically enable embedding inputs on EC/KV consumers: 本 PR 最后一次 merge main (commit 3a10c2a0) 时适配了该改动带来的 `_resolve_mm_embedding_inputs` 重命名与 EC/KV 消费者自动 embedding 输入, 两者都修改 `vllm/config/vllm.py` 的 `__post_init__` 流程。
- PR #56214 [Model] Support DeepSeek-V4.1-Flash: 同属 DeepSeek V4 在 ROCm / MRV2 上的落地序列, 同样改动 `GPUModelRunner` 的初始化路径与 cudagraph 相关配置, 与本 PR 的 `piecewise_capture_available` 计算位置相邻。
- PR #54852 (PR body 引用: 暴露该 compile/cudagraph 组合问题的变更): PR body 明确写道该 PR 「exposed the combination」, 即暴露了非编译模型路径被解析为 PIECEWISE 的失效组合, 是本 PR 的直接触发点。
- PR #52861 (PR body 引用: 因果变更之一): PR body 将其列为 causal changes 之一 (另见 #53155、#53183), 它们引入了需要 `piecewise capture` 的默认配置, 构成本次回

退逻辑的上游背景。

- PR #54782 (review 讨论中引用：记录同类无效状态会产生乱码输出)：作者在讨论 CUDA 默认路径是否受影响时引用该 PR：它记录了相同的无效状态在 NVIDIA DeepSeek V4 开发期间产生 garbled output，佐证该检查应保留在通用配置路径。