PR #36024 完整报告

sgl-project/sglang

[diffusion] Speed up LingBot high-quality VAE decode

合并时间：2026-08-24 14:13

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/36024

执行摘要

- 一句话：LingBot 高质量请求 VAE 解码切 BF16，端到端提速约 3.7%
- 推荐动作：值得精读，尤其适合关注 diffusion 管线精度 / 性能权衡的工程师。重点看三点：resolve_decode_precision 的回退链设计如何保证向后兼容；「FP32 权重常驻 + 请求级 BF16 decode」如何同时满足 lossless byte-exact 与 high 提速；以及 ABBA + SSIM/PSNR + byte-exact 的三重验证方法，可作为同类精度类 PR 的验收模板。

功能与动机

作者在 PR body 中明确：denoise 时间不变、性能收益完全来自 VAE decode，说明 VAE 解码是高质量 LingBot 请求的耗时瓶颈。同时需要兼顾精度：lossless 请求必须保持 FP32 权重，避免 BF16 舍入污染输出，因此设计为「权重常驻 FP32、仅 high 请求在解码时切 BF16」。base.py 中新增字段注释也点明了动机：The loader keeps the reference decode dtype resident so lossless requests never consume pre-rounded weights。

实现拆解

1. 配置层：python/sglang/multimodal_gen/configs/pipeline_configs/base.py 的 PipelineConfig 新增可选字段 vae_decode_precision_high: str | None = None，作为请求级解码精度覆盖；python/sglang/multimodal_gen/configs/pipeline_configs/lingbot_world.py 为 LingBotWorldCausalDMDConfig 与 LingBotWorldV2CausalDMDConfig 设置 vae_decode_precision="fp32"、vae_decode_precision_high="bf16"，形成「lossless 用 FP32、high 用 BF16」的默认行为。
2. 决策层：python/sglang/multimodal_gen/runtime/utils/precision.py 的 resolve_decode_precision 新增只读 quality: str | None = None 关键字参数；当 quality == "high" 且配置了 vae_decode_precision_high 时优先返回该 dtype，否则按 vae_decode_precision → vae_precision 回退；audio_vae/vocoder 分支不受影响。
3. 调用层：python/sglang/multimodal_gen/runtime/pipelines_core/stages/decoding.py 的 DecodingStage.forward 将 batch.sampling_params.quality 传入 resolve_decode_precision，使解码 dtype 与请求质量绑定，并与既有 use_vae_fast_path(vae, quality == "high") 的开关保持一致。
4. Realtime 透传：python/sglang/multimodal_gen/runtime/entrypoints/openai/realtime/realtime_adapter.py 增加 1 行，将 RealtimeVideoGenerationsRequest.quality 写入 sampling params，补上实时请求路径的传递缺口。

5. 测试与文档：test_precision_consistency.py 扩展了 high/lossless 分支与非法精度值抛错用例；test_lingbot_causal_denoising.py 新增断言确认 v1/v2 配置默认值；test_video_api_profiling.py 新增 realtime 请求 quality 转发测试；docs/cookbook/diffusion/LingBot-World/LingBot-World.mdx 与 LingBot-World-2.0.mdx 补充 realtime quality 模式说明。

关键文件：

- python/sglang/multimodal_gen/runtime/utils/precision.py（模块 精度解析；类别 source；类型 core-logic；符号 resolve_decode_precision）：核心决策逻辑：resolve_decode_precision 新增 quality 参数并引入 vae_decode_precision_high 优先分支，是本次请求级精度切换的枢纽。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/decoding.py（模块 解码阶段；类别 source；类型 core-logic；符号 DecodingStage.forward）：解码阶段入口：将 sampling_params.quality 传入 resolve_decode_precision，使解码 dtype 与请求质量绑定，并与 use_vae_fast_path 开关保持一致。
- python/sglang/multimodal_gen/configs/pipeline_configs/base.py（模块 流水线配置；类别 source；类型 core-logic；符号 PipelineConfig）：PipelineConfig 新增 vae_decode_precision_high 字段，是请求级精度覆盖的配置载体，默认 None 保证向后兼容。
- python/sglang/multimodal_gen/configs/pipeline_configs/lingbot_world.py（模块 模型配置；类别 source；类型 core-logic；符号 LingBotWorldCausalDMDConfig, LingBotWorldV2CausalDMDConfig）：LingBotWorld v1/v2 配置落点：设置 vae_decode_precision=fp32 与 vae_decode_precision_high=bf16，是默认行为的来源。
- python/sglang/multimodal_gen/runtime/entrypoints/openai/realtime/realtime_adapter.py（模块 实时适配；类别 source；类型 core-logic；符号 build_realtime_sampling_params）：补上实时请求 quality 透传到 sampling params 的缺口，使 realtime 路径同样能触发请求级解码精度。
- python/sglang/multimodal_gen/test/unit/test_video_api_profiling.py（模块 API 测试；类别 test；类型 test-coverage；符号 test_realtime_video_api_forwards_sampling_quality）：新增 realtime 请求 quality 转发测试，验证 build_realtime_sampling_params 能正确携带 quality=high。
- python/sglang/multimodal_gen/test/unit/test_precision_consistency.py（模块 精度测试；类别 test；类型 test-coverage）：覆盖 resolve_decode_precision 的 high/lossless 分支与非法精度值抛错，是决策逻辑的关键测试。
- python/sglang/multimodal_gen/test/unit/realtime/test_lingbot_causal_denoising.py（模块 去噪测试；类别 test；类型 test-coverage；符号 test_lingbot_quality_high_uses_bf16_vae_decode_only）：断言 LingBotWorld v1/v2 配置默认值：vae_decode_precision=fp32 且 vae_decode_precision_high=bf16，锁定默认行为。
- docs/cookbook/diffusion/LingBot-World/LingBot-World-2.0.mdx（模块 文档；类别 docs；类型 documentation）：文档同步：说明 LingBotWorld v2 的 realtime quality 模式与 BF16 VAE decode 行为。

- docs/cookbook/diffusion/LingBot-World/LingBot-World.mdx（模块 文档；类别 docs；类型 documentation）：文档同步：说明 LingBotWorld v1 的 realtime quality 模式与 BF16 VAE decode 行为。

关键符号：resolve_decode_precision, DecodingStage.forward, build_realtime_sampling_params, test_realtime_video_api_forwards_sampling_quality, test_lingbot_quality_high_uses_bf16_vae_decode_only

关键源码片段

python/sglang/multimodal_gen/runtime/utils/precision.py

核心决策逻辑：resolve_decode_precision 新增 quality 参数并引入 vae_decode_precision_high 优先分支，是本次请求级精度切换的枢纽。

```
def resolve_decode_precision(
    server_args,
    component_name: str = "vae",
    *,
    quality: str | None = None,
) -> torch.dtype:
    """按组件与请求质量解析解码精度。

    quality = "high" 时优先使用 vae_decode_precision_high（如 BF16），
    以便高质量请求获得更快的 VAE decode；其它质量或未配置时回退到
    vae_decode_precision，再回退到组件默认精度 vae_precision。
    音频组件（audio_vae / vocoder）不走此逻辑，保持原有处理。
    """
    pipeline_config = server_args.pipeline_config

    # 音频组件单独走 audio_vae_precision，与视频 / 图像 VAE 解耦。
    if component_name in ("audio_vae", "vocoder"):
        return resolve_precision(
            server_args,
            component_name,
            precision_attr="audio_vae_precision",
        )

    # 仅当请求明确为 high 且配置了独立的高质量解码精度时才切换，
    # 保证 lossless 请求始终使用未经过舍入的权重。
    if quality == "high":
        high_precision = getattr(pipeline_config, "vae_decode_precision_high", None)
        if high_precision is not None:
            return precision_to_dtype(high_precision, "vae_decode_precision_high")

    # 回退链：vae_decode_precision → vae_precision，保持向后兼容。
    decode_precision = getattr(pipeline_config, "vae_decode_precision", None)
    if decode_precision is not None:
        return precision_to_dtype(decode_precision, "vae_decode_precision")
    return resolve_precision(
```

```
        server_args,
        component_name,
        precision_attr="vae_precision",
    )
```

python/sglang/multimodal_gen/runtime/pipelines_core/stages/decoding.py

解码阶段入口：将 sampling_params.quality 传入 resolve_decode_precision，使解码 dtype 与请求质量绑定，并与 use_vae_fast_path 开关保持一致。

```
# DecodingStage.forward 中，解码精度与请求质量绑定：
# 之前只按全局配置解析 dtype，现在把 sampling_params 里的 quality 传进
# resolve_decode_precision，使 high 请求自动落到 BF16 decode 路径。
vae_dtype = resolve_decode_precision(
    server_args,
    self.component_name,
    quality=batch.sampling_params.quality,
)

# fast path 开关同样以 quality == "high" 为条件，二者保持一致；
# 这样 high 请求既能走更快的解码分支，也使用更低的 decode 精度。
with self.use_declared_component(
    component_name=self.component_name,
    module=self.vae,
) as vae:
    assert vae is not None
    self.vae = vae

    with use_vae_fast_path(vae, batch.sampling_params.quality == "high"):
        frames = self.decode(batch.latents, server_args, vae_dtype=vae_dtype)
```

python/sglang/multimodal_gen/configs/pipeline_configs/base.py

PipelineConfig 新增 vae_decode_precision_high 字段，是请求级精度覆盖的配置载体，默认 None 保证向后兼容。

```
# VAE 配置区：vae_precision 是权重加载精度（默认 fp32，常驻内存），
# vae_decode_precision 是全局解码精度覆盖，None 表示跟随 vae_precision。
# 新增的 vae_decode_precision_high 是请求级覆盖：quality == "high" 时
# 优先使用它（LingBot 场景为 bf16），lossless 请求则始终走 fp32，
# 避免 pre-rounded 权重被 BF16 解码污染。
vae_config: VAEConfig = field(default_factory=VAEConfig)
vae_precision: str = "fp32"
vae_decode_precision: str | None = None
vae_decode_precision_high: str | None = None
```

评论区精华

本 PR 没有 review 评论或讨论线程，核心权衡由作者在 PR body 中自证：性能提升仅来自 VAE decode（denoise 不变），并通过 SSIM/PSNR 与 lossless byte-exact 双重验证来支撑「high 用 BF16、lossless 用 FP32」的取舍。作者强调 candidate 输出跨多次运行稳定，且

lossless ABBA 在经历 high 请求后仍保持四份 MP4 字节一致，说明请求间无精度污染。

- 暂无高价值评论线程

风险与影响

- 风险：
  - resolve_decode_precision 仅对 quality == "high" 特判，其它取值（如 None、未来可能的 medium）会静默回退到 FP32，正确性安全但需注意 quality 值域演进时该函数是否同步扩展。
  - decoding.py 依赖 batch.sampling_params.quality；若上游（非 realtime 或旧客户端）未填充 quality，high 请求不会触发 BF16，仅是性能目标不达成，无正确性风险。
  - realtime_adapter.py 只有 1 行透传，单测仅覆盖 type="init" 的请求，其它实时消息类型或构造路径可能存在透传缝隙，需要关注后续用例覆盖。
  - 质量风险：BF16 decode 相比 FP32，v1 PSNR 46.83 dB、v2 42.65 dB，总体高位但非无损，极端帧（文字、细纹理）可能肉眼可辨。
  - 兼容性：新增字段默认 None，非 LingBot 模型不感知变化，回退链完整。
- 影响：
  - 用户：LingBotWorld v1/v2 且请求 quality=high 的用户获得约 3.7% 端到端加速（832x480x9、4 步场景）；lossless 用户输出与之前完全一致（byte-exact）。
  - 系统：VAE 权重继续以 FP32 常驻显存，仅在解码时按请求切换 dtype，显存占用与加载逻辑不变；新增 1 个配置字段和 1 条请求级决策分支。
  - 团队：确立了「权重加载精度与解码计算精度分离 + 请求级覆盖」的模式，为后续扩展更多 quality 档位提供了可复制的范式；测试、文档、CI（含 AMD ROCm 7.2）同步到位。
  - 风险标记：请求级精度动态切换，核心解码路径变更，新增配置字段，低精度解码质量退化

关联脉络

- PR #35969 [diffusion] Accelerate LingBot Video RMSNorm in quality=high: 同属 LingBot 高质量链路优化，同样以 quality=high 为条件提速，涉及 denoising 与 LingBot Video 内核，与本 PR 的 VAE decode 提速形成互补。
- PR #36084 [Diffusion] Add per-component quantization overrides: 同为精度 / 量化决策链的配置覆盖机制，按组件粒度覆盖精度，与本 PR 的请求级精度覆盖思路一脉相承。
- PR #36062 [diffusion] cache LoRA-merged weights in files the page cache can hold: 同属 diffusion 运行时性能 / 内存优化系列，与本 PR 一样在保证输出一致性的前提下优化运行时开销。