PR #22289 完整报告

sgl-project/sglang

[Bugfix] multimodal_gen(hunyuan3d): honor config precisions for delight/paint

合并时间：2026-05-20 10:38

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22289

执行摘要

- 一句话：修复 Hunyuan3D 精度配置和负提示词
- 推荐动作：值得精读，尤其是理解如何在现有代码中通过 PRECISION_TO_TYPE 和简单回退逻辑实现精度配置的兼容性修复。对于类似的多模态生成模块有参考价值。

功能与动机

PR body 指出，Hunyuan3D Paint 之前硬编码 fp16，忽略了管道配置中的 dit_precision 和 vae_precision，在 CPU/MPS 上因缺乏半精度支持而崩溃，且不同管道间的精度行为不一致。此外，delight 阶段忽略了 delight_negative_prompt。

实现拆解

1. 导入 PRECISION_TO_TYPE 映射：在 python/sglang/multimodal_gen/runtime/pipelines_core/stages/hunyuan3d_paint.py 中添加导入，用于将配置字符串（如 "fp16"）转换为 torch.dtype。
2. 修复 Delight 管道加载：在 _load_delight_model 中，从配置中读取 dit_precision（默认 "fp16"），通过 PRECISION_TO_TYPE 获取对应 dtype，若设备为 CPU/MPS 且 dtype 为半精度则回退到 fp32；随后使用该 dtype 加载和移动管道。
3. 修复 VAE 和 UNet 加载：在 _do_load_paint 中，分别处理 VAE 的 vae_precision（默认 "fp32"）和 UNet 的 dit_precision（默认 "fp16"），同样添加 CPU/MPS 回退逻辑。
4. 传递 delight_negative_prompt：在 _run_delight 的管道调用中，新增 negative_prompt 参数，其值从 self.config.delight_negative_prompt 获取（默认空字符串）。

关键文件：

- python/sglang/multimodal_gen/runtime/pipelines_core/stages/hunyuan3d_paint.py（模块 生成管道；类别 source；类型 dependency-wiring；符号 _load_delight_model, _do_load_paint, _run_delight）：唯一变更文件，修改了 Delight 管道加载、VAE/UNet 加载以及推理时的负提示词传递

关键符号：_load_delight_model, _do_load_paint, _run_delight

关键源码片段

python/sglang/multimodal_gen/runtime/pipelines_core/stages/hunyuan3d_paint.py

唯一变更文件，修改了 Delight 管道加载、VAE/UNet 加载以及推理时的负提示词传递

```
# python/sglang/multimodal_gen/runtime/pipelines_core/stages/hunyuan3d_paint.py

# 关键变更：在文件顶部的 import 中添加了 PRECISION_TO_TYPE
from sglang.multimodal_gen.utils import PRECISION_TO_TYPE

# 在 _load_delight_model 中：
if local_path and os.path.exists(local_path):
    # 从配置中解析 dit_precision，默认 "fp16"
    dit_dtype = PRECISION_TO_TYPE.get(
        getattr(self.config, "dit_precision", "fp16"), torch.float16
    )
    # CPU/MPS 上回退到 fp32，因为半精度可能不受支持
    if self.device.type in ("cpu", "mps") and dit_dtype in (
        torch.float16,
        torch.bfloat16,
    ):
        dit_dtype = torch.float32
    pipeline = StableDiffusionInstructPix2PixPipeline.from_pretrained(
        local_path,
        torch_dtype=dit_dtype, # 以前是 torch.float16
        safety_checker=None,
    )
    # ...
    self._delight_pipeline = pipeline.to(self.device, dit_dtype) # 以前是 torch.float16

# 在 _do_load_paint 中：
# VAE: 使用 vae_precision，默认 "fp32"
vae_dtype = PRECISION_TO_TYPE.get(
    getattr(self.config, "vae_precision", "fp32"), torch.float32
)
if self.device.type in ("cpu", "mps") and vae_dtype in (torch.float16, torch.bfloat16):
    vae_dtype = torch.float32
self.vae = self.vae.to(device=self.device, dtype=vae_dtype).eval()

# UNet: 使用 dit_precision，默认 "fp16"
dit_dtype = PRECISION_TO_TYPE.get(
    getattr(self.config, "dit_precision", "fp16"), torch.float16
)
if self.device.type in ("cpu", "mps") and dit_dtype in (torch.float16, torch.bfloat16):
    dit_dtype = torch.float32
self.transformer = UNet2p5DConditionModel.from_pretrained(
    os.path.join(local_path, "unet"),
    torch_dtype=dit_dtype, # 以前是 torch.float16
).to(self.device)
```

```
# 在 _run_delight 中：
image = self._delight_pipeline(
    prompt=self.config.delight_prompt,
    negative_prompt=getattr(self.config, "delight_negative_prompt", ""), # 新增：传递负提示词
    image=image,
    generator=torch.manual_seed(42),
    height=512,
    # ...
)
```

评论区精华

PR 没有 review 评论，只有批准。作者在 issue 评论中提供了本地验证用的单元测试代码（未包含在 PR 内），并解释了精度回退的安全性。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。改动集中在精度配置的读取与使用，且保留了默认行为（CUDA 下默认 fp16/bf16 不变）。CPU/MPS 回退到 fp32 是安全降级。未覆盖测试可能引入回归，但变更逻辑简单直接。
- 影响：影响范围限于 Hunyuan3D 的 paint 功能。对 CUDA/ROCm 用户无行为变化；对 CPU/MPS 用户，原先因半精度而崩溃的用例现在可正常运行（回退到 fp32）。此外，delight_negative_prompt 现在生效。
- 风险标记：缺少测试覆盖

关联脉络

- 暂无明显关联 PR