

PR #31590 完整报告

sgl-project/sglang

Add Cosmos3 Edge and Distilled checkpoints support

合并时间: 2026-08-12 10:29

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/31590>

执行摘要

- 一句话: Cosmos3 Edge 与蒸馏变体支持, 变体自动识别
- 推荐动作: 值得精读。重点看三处设计: ① 变体检测一次性预计算 (`update_config_from_dict + lru_cache`) 避免 per-request 下载 checkpoint 配置, 这是 diffusion 多变体服务的通用模式; ② Ulysses SP 下 SDE 噪声生成器跨 rank 同步的思路, 可复用到其它扩散模型; ③ `_predict_noise_cfg` 对 CFG padding 在无 mask cross-attention 下语义的分析 (零 K/V 仍占 softmax 权重), 是少见的数值质量权衡。另外 review 中 mickqian 对防御性代码的两次拒绝 (fail-fast、显式契约) 展示了该仓库的工程偏好, 值得团队对齐。

功能与动机

PR body 明确: "All variants continue to be served by the single Cosmos3 pipeline, with the variant selected from the checkpoint config at load time." 此前 PR#24994、PR#26492、PR#27168 建立了 Cosmos3 家族基础支持, 本 PR 在此基础上扩展 Edge 与 Distilled 变体。PR 评论中 nv-dmajchrowski 补充: "after the previous approval we had to add few more changes (edge checkpoint format changed slightly after the release + we had small accuracy issues)", 说明 Edge 发布后 checkpoint 格式有微调, 本 PR 是完整适配。

实现拆解

1. 变体检测与预计算: 在 `configs/pipeline_configs/cosmos3.py` 新增 `is_edge_checkpoint` (读取 `transformer/config.json` 的 `backbone_type` 或 `hidden_act=relu2`)、`_distilled_sampler_config` (读取 scheduler config 的 `_class_name` 与 `fixed_step_sampler_config.t_list`)、`get_distilled_sigmas`, 并使用 `functools.lru_cache` 缓存; `Cosmos3Config.update_config_from_dict` 在 `model_path` 就绪后一次性算出 `is_edge` 与 `distilled_sigmas` 存入配置对象, 避免每个请求重复下载 checkpoint 子目录 (commit 3b51f64 的优化)。
2. Edge 架构支持: `configs/models/dits/cosmos3video.py` 将 `_build_cosmos3_param_names_mapping` 参数化为 `gated_mlp` 模式, dense 变体跳过 gate/up 合并规则, `up_proj/down_proj` 直通; `Cosmos3VideoArchConfig` 新增 `hidden_act`、`qk_norm_for_text`、`use_und_k_norm_for_gen` 字段。
`runtime/models/dits/cosmos3video.py` 新增 `Cosmos3DenseMLP` (`up` → ReLU → 平方 → `down`) 与 `_build_mlp` 工厂, 并按 `hidden_act` 选择 MLP 结构;

Cosmos3CausalAttention 支持可选 QK-norm 与独立的 k_norm_und_for_gen (GEN 交叉注意力单独归一化 K)。

3. 采样默认值: configs/sample/cosmos3.py 新增 _resolve_variant_defaults, 按 is_edge / is_distilled 解析; height/width 默认改为 None 由变体填充; Edge 使用 256p/480p 分辨率白名单与对应默认尺寸, T2I/T2V 分别用 guidance 7.0/5.0, T2I 默认 CFG 区间 (400, 1000); 蒸馏变体强制 guidance_scale = 1.0, num_inference_steps 固定为 sigma 表长度 (期间修复了默认 None 导致 warmup 崩溃的问题)。
4. 调度与去噪: Cosmos3TimestepPreparationStage 蒸馏路径先 set_shift(1.0) 再用 set_timesteps(sigmas=distilled_sigmas) 驱动 scheduler, 防止双重重偏移; _default_flow_shift_for_mode 对 Edge 视频模式统一返回 3.0。Cosmos3DenoisingStage 将批次 seed 构造的 torch.Generator (device 取 latents.device) 传入 scheduler step, 保证 Ulysses SP 下各 rank 噪声一致; _predict_noise_cfg 在正负 prompt 长度不等时把 CFG 两个分支拆成独立 forward, 避免零 padding 的 K/V 稀释注意力。scheduler_loader.py 新增 _supported_init_kwargs, 按构造函数签名过滤 scheduler_config 不接受的键。
5. 兼容与配套: registry.py 注册 Cosmos3OmniPipeline 并保留 Cosmos3OmniDiffusersPipeline 别名; hf_diffusers_utils.py 新增 _split_hf_subfolder 支持 namespace/repo/subfolder 局部下载; 权重加载时丢弃 *_quantizer.* ModelOpt 校准缓存; action/api.py 增加 multipart/form-data 上传支持; video_api.py 新增 form_text_value 防止 JSON prompt 被反序列化。测试 test_cosmos3.py 新增 dense 参数映射、Edge flow_shift/ 采样默认值、_class_name 双拼写注册等用例。

关键文件:

- python/sglang/multimodal_gen/configs/pipeline_configs/cosmos3.py (模块 管线配置; 类别 source; 类型 core-logic; 符号 is_edge_checkpoint, _distilled_sampler_config, is_distilled_checkpoint, get_distilled_sigmas): 变体检测核心: is_edge_checkpoint / _distilled_sampler_config / get_distilled_sigmas, 并在 update_config_from_dict 中一次性预计算, per-request 路径都从这里读。
- python/sglang/multimodal_gen/runtime/models/dits/cosmos3video.py (模块 模型层; 类别 source; 类型 data-contract; 符号 Cosmos3DenseMLP, _build_mlp, forward, init): Edge dense 骨干的模型侧实现: 新增 Cosmos3DenseMLP 与 _build_mlp 工厂, Cosmos3CausalAttention 支持可选 QK-norm 与 GEN 侧独立 K 归一化。
- python/sglang/multimodal_gen/configs/sample/cosmos3.py (模块 采样配置; 类别 source; 类型 core-logic; 符号 _guidance_is_explicit, _resolve_variant_defaults): 变体采样默认值 (分辨率、guidance、CFG 区间、步数) 在此解析, 直接影响每个请求的采样行为。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/cosmos3.py (模块 管线阶段; 类别 source; 类型 data-contract; 符号 _inject_caption_metadata, _default_flow_shift_for_mode, _predict_noise_cfg, _cfg_combine): 调度与去噪主路径: distilled sigma 驱动 scheduler、SP 同步 generator、CFG 分支拆分、结构化 caption 元数据注入。

- python/sglang/multimodal_gen/configs/models/dits/cosmos3video.py (模块 模型配置; 类别 source; 类型 data-contract; 符号 _build_cosmos3_param_names_mapping, gated_mlp) : 权重映射契约: _build_cosmos3_param_names_mapping 参数化 gated_mlp, 决定 gate/up 是否合并。
- python/sglang/multimodal_gen/test/unit/test_cosmos3.py (模块 单元测试; 类别 test; 类型 test-coverage; 符号 TestCosmos3DenseParamNamesMapping, test_edge_flow_shift_default, test_und_mlp_up_proj_unmerged, test_gen_mlp_down_proj_unmerged) : 测试覆盖 dense 权重映射、Edge 采样默认值、_class_name 新旧注册、flow_shift 默认值。
- python/sglang/multimodal_gen/runtime/loader/component_loaders/scheduler_loader.py (模块 组件加载; 类别 source; 类型 core-logic; 符号 _supported_init_kwargs) : _supported_init_kwargs 按构造函数签名过滤 scheduler_config 键, 蒸馏 checkpoint 才能被加载。
- python/sglang/multimodal_gen/runtime/utils/hf_diffusers_utils.py (模块 工具层; 类别 source; 类型 core-logic; 符号 _split_hf_subfolder) : _split_hf_subfolder 支持 namespace/repo/subfolder 形式的组件路径下载, 解决新格式 checkpoint 的子目录加载。
- python/sglang/multimodal_gen/runtime/entrypoints/action/api.py (模块 入口服务; 类别 source; 类型 entrypoint; 符号 _multipart_action_payload, _parse_form_value, _is_form_upload, _save_action_upload) : 附带新增 multipart/form-data 支持 (kediwu0331 提交), 让 action 端点能处理视频 / 图片上传。
- python/sglang/multimodal_gen/runtime/entrypoints/openai/video_api.py (模块 入口服务; 类别 source; 类型 entrypoint; 符号 form_text_value) : form_text_value 避免结构化 JSON prompt 被 form 解析反序列化, Edge 的 JSON caption 才能通过请求校验。

关键符号: is_edge_checkpoint, is_distilled_checkpoint, get_distilled_sigmas, update_config_from_dict, Cosmos3DenseMLP, _build_mlp, _build_cosmos3_param_names_mapping, _resolve_variant_defaults, _guidance_is_explicit, _inject_caption_metadata, _default_flow_shift_for_mode, _predict_noise_cfg, _supported_init_kwargs, _split_hf_subfolder, _multipart_action_payload, form_text_value

关键源码片段

python/sglang/multimodal_gen/configs/pipeline_configs/cosmos3.py

变体检测核心: is_edge_checkpoint / _distilled_sampler_config / get_distilled_sigmas, 并在 update_config_from_dict 中一次性预计算, per-request 路径都从这里读。

```
# Cosmos3 变体检测: Edge 与蒸馏 checkpoints 的识别全部基于
# checkpoint 自身配置, 而不是硬编码模型名, 保证同一管线可服务整个家族。
# 函数带 lru_cache, model_path 首次解析后结果常驻内存。
```

```
@functools.lru_cache(maxsize=None)
def is_edge_checkpoint(model_path: str) -> bool:
    """判断 checkpoint 是否为 Edge (dense 骨干) 变体。
```

直接从 transformer/config.json 读取，早于权重加载，
因此客户端进程在解析采样默认值时无需把权重搬上设备。

```
"""
from sglang.multimodal_gen.runtime.utils.hf_diffusers_utils import (
    get_diffusers_component_config,
)

config = get_diffusers_component_config(
    component_path=os.path.join(model_path, "transformer")
)
# Edge 的 dense 骨干可能以 backbone_type 或 hidden_act=relu2 之一标记
return (
    config.get("backbone_type") == COSMOS3_EDGE_BACKBONE_TYPE
    or config.get("hidden_act") == "relu2"
)
```

```
@functools.lru_cache(maxsize=None)
def _distilled_sampler_config(model_path: str) -> dict | None:
    """读取蒸馏 checkpoint 的固定步 sigma 表，非蒸馏返回 None。
```

蒸馏只体现在 scheduler: checkpoint 自带
FlowMatchEulerDiscreteScheduler 与 fixed_step_sampler_config.t_list,
而其余变体使用多步 FlowUniPC。

```
"""
from sglang.multimodal_gen.runtime.utils.hf_diffusers_utils import (
    get_diffusers_component_config,
)

config = get_diffusers_component_config(
    component_path=os.path.join(model_path, "scheduler")
)
# 只有显式声明了固定步表才判定为蒸馏变体
if config.get("_class_name") != "FlowMatchEulerDiscreteScheduler":
    return None
sampler = config.get("fixed_step_sampler_config")
if not sampler or not sampler.get("t_list"):
    return None
return sampler
```

```
def get_distilled_sigmas(model_path: str) -> list[float] | None:
    """向调度阶段暴露蒸馏 sigma 表；非蒸馏 checkpoint 返回 None。"""
    sampler = _distilled_sampler_config(model_path)
    return list(sampler["t_list"]) if sampler is not None else None
```

[python/sglang/multimodal_gen/runtime/models/dits/cosmos3video.py](#)

Edge dense 骨干的模型侧实现：新增 Cosmos3DenseMLP 与 _build_mlp 工厂，Cosmos3CausalAttention 支持可选 QK-norm 与 GEN 侧独立 K 归一化。

```
class Cosmos3DenseMLP(nn.Module):  
    """dense MLP: down(relu(up(x)) ** 2), Edge 变体的非门控结构。
```

与 SwiGLU 的 gate_up_proj 合并权重布局不同，dense 变体只有独立的 up_proj / down_proj，权重加载走未合并的直通过程。

```
    """
```

```
def __init__(  
    self,  
    hidden_size: int,  
    intermediate_size: int,  
    prefix: str = "",  
    quant_config: QuantizationConfig | None = None,  
):
```

```
    super().__init__()  
    # up_proj 不合并 gate，直接展开放大中间维度  
    self.up_proj = ColumnParallelLinear(  
        hidden_size,  
        intermediate_size,  
        bias=False,  
        gather_output=False,  
        quant_config=quant_config,  
        prefix=add_prefix("up_proj", prefix),  
    )
```

```
    self.down_proj = RowParallelLinear(  
        intermediate_size,  
        hidden_size,  
        bias=False,  
        input_is_parallel=True,  
        quant_config=quant_config,  
        prefix=add_prefix("down_proj", prefix),  
    )
```

```
def forward(self, x: torch.Tensor) -> torch.Tensor:  
    up, _ = self.up_proj(x)  
    up = F.relu(up)  
    out, _ = self.down_proj(up * up) # squared-ReLU: relu(x) ** 2  
    return out
```

```
def _build_mlp(  
    hidden_act: str,  
    hidden_size: int,  
    intermediate_size: int,  
    prefix: str,  
    quant_config: QuantizationConfig | None,
```

```

) -> nn.Module:
    """按 hidden_act 选择 MLP 结构: relu2 走 dense, 其余走 SwiGLU 门控。"""
    mlp_cls = Cosmos3DenseMLP if hidden_act == "relu2" else Cosmos3GatedMLP
    return mlp_cls(
        hidden_size=hidden_size,
        intermediate_size=intermediate_size,
        prefix=prefix,
        quant_config=quant_config,
    )

```

python/sglang/multimodal_gen/configs/sample/cosmos3.py

变体采样默认值（分辨率、guidance、CFG 区间、步数）在此解析，直接影响每个请求的采样行为。

```

def _resolve_variant_defaults(self, is_edge: bool, is_distilled: bool = False) -> None:
    """按变体补齐未显式指定的分辨率与 guidance 默认值。

```

Edge 只在 256p/480p 分辨率上训练过，更大尺寸会把空间 mRoPE 网格推到训练范围外导致画面破碎，因此必须收窄分辨率白名单并给出对应的默认尺寸。

```

"""

```

```

is_t2i = self.num_frames == 1
if is_distilled:
    # 蒸馏模型把 guidance 烘焙进权重，推理只需单次 forward,
    # 关闭 CFG 并强制 guidance_scale = 1.0
    self.guidance_scale = 1.0
elif is_edge and not self._guidance_is_explicit():
    # 用户显式传入 guidance_scale 时尊重显式值，避免被默认覆盖
    self.guidance_scale = (
        COSMOS3_EDGE_T2I_GUIDANCE_SCALE
        if is_t2i
        else COSMOS3_EDGE_T2V_GUIDANCE_SCALE
    )
# T2I 只在噪声高的 [400, 1000] 区间施加 CFG;
# 低噪声步的 guidance 已被证明会损害采样质量
if is_t2i and not is_distilled and self.guidance_interval is None:
    self.guidance_interval = COSMOS3_T2I_GUIDANCE_INTERVAL
if is_edge:
    self.supported_resolutions = COSMOS3_EDGE_SUPPORTED_RESOLUTIONS
    if self.height is None and self.width is None:
        if is_t2i:
            self.width = self.height = COSMOS3_EDGE_T2I_SIZE
        else:
            self.width, self.height = (
                COSMOS3_EDGE_T2V_WIDTH,
                COSMOS3_EDGE_T2V_HEIGHT,
            )

```

评论区精华

围绕 5 个话题展开，其中 3 个被采纳修复、2 个被维护者明确拒绝。核心交锋：① gemini-code-assist[bot] 指出蒸馏 sigma 双重 shift 问题，mickqian 确认在 commit 78ce372 修复；② kediwu0331 发现 `guidance_scale` 默认 `None` 导致 warmup 崩溃，作者随即修复；③ mickqian 两次拒绝 bot 的防御性建议（`model_path` 空值防护、`inspect.signature` try-except），理由是 fail-fast 优于静默选错变体、过滤契约必须可检查；④ bot 建议 generator 用 `latents.device`，已被采纳。

- 蒸馏 checkpoint 的 sigma 双重 shift (correctness): mickqian 确认已在 commit 78ce372 修复：蒸馏路径先 `set_shift(1.0)` 再应用 checkpoint 提供的 sigma 表。
- `guidance_scale` 默认 `None` 导致 warmup 崩溃 (correctness): nv-dmajchrowski 已推送修复，恢复有值默认并配合 `_guidance_is_explicit` 判断。
- `is_edge_checkpoint` 是否需要空 `model_path` 防护 (question): 不添加防护，保持 fail-fast。
- `inspect.signature` 是否加 try-except 兜底 (design): 不添加兜底。
- 随机噪声 generator 的 device 选择 (correctness): mickqian 确认已在 commit 78ce372 改为 `latents.device`。

风险与影响

- 风险：
 1. 调度器配置键过滤：`scheduler_loader._supported_init_kwargs` 依赖 `inspect.signature` 过滤配置键，若调度器类不可检查（如动态生成类），当前实现会直接抛异常；mickqian 明确拒绝加 try-except 兜底，该假设是硬性的，未来接入新调度器时需要留意。
 2. checkpoint 配置读取崩溃：`is_edge_checkpoint` 与 `_distilled_sampler_config` 在 `model_path` 为空或路径非法时 `os.path.join` 会抛 `TypeError`，且 `get_diffusers_component_config` 读取失败会中断服务启动——这是有意的 fail-fast，但对间接调用路径（如测试 mock 或非 Cosmos3 配置）可能引入意外崩溃。
 3. CFG 分支性能回退：`_predict_noise_cfg` 在正负 prompt 长度不一致时拆成两次独立 forward，Edge 默认 `negative_prompt=""` 与长 prompt 之间几乎必然触发该路径，推理耗时约为合并批次的两倍（但避免了 padding 造成的质量损失）。
 4. 蒸馏步数契约：`distilled` 路径忽略用户传入的 `num_inference_steps`，步数固定为 sigma 表长度，用户期望与实际行为可能不一致。
 5. SP 噪声行为改变：SP 下 SDE 噪声从各 rank 独立随机改为种子一致，修复 shard-boundary artifact 的同时改变了既有输出的随机性分布，属于有意的行为变更。
 6. multipart form 新逻辑：action 端点的 multipart 解析、上传文件落盘与临时目录清理是新增入口逻辑，文件路径拼接使用 `request_id + filename`，存在路径注入与文件清理风险，需要验证。
 - 影响：功能影响：Cosmos3 全家族（Nano/Super/Edge，普通与蒸馏）由一个管线统一服务，checkpoint 变体在加载时自动识别；旧 checkpoint 的 `_class_name` 拼写（`Cosmos3OmniDiffusersPipeline`）继续可用，无需用户迁移。系统影响：扩散管线的时间步准备与噪声生成路径改动影响所有 Cosmos3 请求；蒸馏请求默认走固定 sigma 表，不再受 `num_inference_steps` 控制。团队影响：diffusion 维护者需要理解变体检测的缓存时机与 CFG 拆分语义；后续新增变体只需扩展检测函数与默认

值表。用户影响：Edge 分辨率被限制在 256p/480p，显式请求更高分辨率会被拒绝或警告，属于有意的产品约束，需要文档说明。 - 风险标记：核心路径变更，checkpoint 格式兼容，蒸馏调度契约，CFG 分支性能回退，SP 噪声一致性

关联脉络

- PR #24994 Add Cosmos3 model family support: PR body 引用的 Cosmos3 基础支持，本 PR 是对该家族的变体扩展。
- PR #26492 Cosmos3 follow-ups: PR body 引用的 Cosmos3 后续支持，与本 PR 同属 Cosmos3 功能线。
- PR #27168 Cosmos3 extensions: PR body 引用的 Cosmos3 扩展，与本 PR 同属 Cosmos3 功能线。
- PR #30569 New model format support: PR body 明确说明 copied changes from #30569 to make it work with new model format。
- PR #34249 [diffusion] move DiT execution capabilities to runtime models: 同属 diffusion 架构向 runtime 模型层集中的演进脉络，涉及相同文件 runtime/models/dits 与 configs/models/dits。
- PR #34401 Fix model-driven DiT layerwise offload auto policy: 同属 diffusion 模块的 configs/models/dits 与 runtime/platforms 演进，涉及 DiT 配置与运行时一致性。