

PR #35613 完整报告

sgl-project/sglang

[diffusion] refactor: scope model-specific API parameters

合并时间: 2026-08-28 19:08

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/35613>

执行摘要

- 一句话: 扩散模型 API 参数按模型作用域收敛, 杜绝跨模型字段泄漏
- 推荐动作: 值得精读。核心价值在于:
 1. 如何在通用 OpenAI 兼容端点和模型专属参数之间建立‘声明式契约’, 避免字段泄漏;
 2. request_extra_value 统一了多种 transport 容器 (extra_body / extra_json / extra_args / extra_params) 的读取优先级;
 3. 模型默认输出格式也通过 default_image_output_format 下放到子类。对需要扩展 SGLang diffusion API 的工程师, 这份契约设计可作为参考模板。

功能与动机

PR body 明确指出: *Model-specific diffusion controls should not accumulate in the shared OpenAI request or base sampling schemas*。旧实现把 Cosmos3、ERNIE-Image、Ideogram 字段硬编码在通用端点里, 而文档中 LTX-2.5 的在线控制字段 (auto_duration、use_diffusion_decoder) 根本没有被转发; 同时 LongCat-Image 缺少公开的 cookbook 与导航覆盖。

实现拆解

实现分四步:

1. 在基础 SamplingParams (python/sglang/multimodal_gen/configs/sample/sampling_params.py) 中新增类方法 image_request_extra_fields / video_request_extra_fields / default_image_output_format / default_image_response_format, 作为模型字段声明的契约入口; 并把 LTX-2.5 的 use_diffusion_decoder、auto_duration 等字段从基类移回子类。
2. 在 openai/utils.py 中新增传输层兼容工具: request_extra_value (统一读取根级字段、extra_body、extra_json、extra_args、extra_params, 并保持根级字段优先)、get_declared_request_extra_fields (返回模型声明的字段集合, 含传输别名)、get_sampling_request_extra_fields (过滤出能作为 dataclass 构造参数初始化的字段)、resolve_sampling_params_cls (集中解析当前 server 激活的 SamplingParams 子类), 并用 functools.cache 缓存。
3. 重构 image_api.py 与 video_api.py 两个入口: 删除原先硬编码的 is_cosmos3 判断、Cosmos3 字段读取和 _MULTIPART_EXTRA_FORM_FIELDS 中的模型专属项, 改为先

resolve_sampling_params_cls 再通过 _image_request_model_kwargs / _video_request_model_kwargs 按声明字段提取。

4. 在 cosmos3.py 等模型配置子类中把字段声明与 lowering 逻辑收拢：

Cosmos3SamplingParams 新增 use_duration_template 等字段并实现 image/video_request_extra_fields，把 guardrails 别名、sound_duration 推导和 action 归一化下沉到 lower_video_request_kwargs；ltx_2_5.py 补 video_request_extra_fields。测试覆盖了协议隔离、激活模型过滤、别名、根级字段、嵌套兼容与优先级。

关键文件：

- python/sglang/multimodal_gen/runtime/entrypoints/openai/video_api.py（模块 视频端点；类别 source；类型 entrypoint；符号 _video_request_model_kwargs, _video_sampling_params_cls, _is_cosmos3_server, _normalize_optional_string）：视频端点是本次重构的核心入口：删除了 203 行硬编码的 Cosmos3 专属字段处理与 multipart 表单字段，改为通过 get_sampling_request_extra_fields / request_extra_value / resolve_sampling_params_cls 动态提取激活模型声明的字段，通用端点不再感知具体模型。
- python/sglang/multimodal_gen/runtime/entrypoints/openai/utils.py（模块 API 工具层；类别 source；类型 dependency-wiring；符号 _parse_request_extra_container, request_extra_value, get_declared_request_extra_fields, get_sampling_request_extra_fields）：新增的 request_extra_value、get_declared_request_extra_fields、get_sampling_request_extra_fields、resolve_sampling_params_cls 是本次重构的契约基础设施，统一了 extra_body / extra_json / extra_args / extra_params 四种传输容器的读取与优先级。
- python/sglang/multimodal_gen/configs/sample/cosmos3.py（模块 模型配置；类别 source；类型 dependency-wiring；符号 _parse_request_value, _optional_int_list, video_request_extra_fields, image_request_extra_fields）：Cosmos3 是本次重构受影响的最大的模型：新增 use_duration_template 等字段、补全 image/video_request_extra_fields 声明、把 guardrails 别名、sound_duration 推导与 action 字段归一化下沉到子类 lowering，是模型侧契约落地的示范样本。
- python/sglang/multimodal_gen/configs/sample/sampling_params.py（模块 采样参数；类别 source；类型 core-logic；符号 image_request_extra_fields, default_image_output_format, default_image_response_format）：基础 SamplingParams 是本次契约变更的‘宪法’：文档注释明确新增字段必须为跨模型共享、模型专属字段必须走子类声明；同时把 use_diffusion_decoder、auto_duration、sound_duration、LongCat 和 ErnieImage 的字段从基类移除。
- python/sglang/multimodal_gen/runtime/entrypoints/openai/image_api.py（模块 图像端点；类别 source；类型 entrypoint；符号 _image_request_model_kwargs, _parse_extra_container）：图像端点删除了 is_cosmos3 字符串匹配和 _parse_extra_container 本地实现，改为 resolve_sampling_params_cls + _image_request_model_kwargs + default_image_output_format，是通用端点去模型化的另一个入口。
- python/sglang/multimodal_gen/test/unit/test_openai_image_api.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_longcat_image_fields_remain_model_specific,

test_longcat_image_fields_accept_nested_extra_body, test_other_image_extensions_remain_model_specific, test_cosmos_image_guardrails_alias_is_preserved) : 新增 4 个测试直接验证本次重构的核心不变量：模型专属字段不进基类、不进请求模型、只有激活子类能提取、根级字段优先于嵌套 extra_body。

- python/sglang/multimodal_gen/configs/sample/ltx_2_5.py (模块 模型配置; 类别 source; 类型 core-logic; 符号 video_request_extra_fields) : LTX-2.5 补上 video_request_extra_fields 声明, 使文档中的 auto_duration / use_diffusion_decoder 在线控制首次真正被转发, 同时修复 LTX-2.3 默认走 VAE decode 的行为。
- python/sglang/multimodal_gen/test/unit/test_video_api_profiling.py (模块 单元测试; 类别 test; 类型 test-coverage; 符号 test_ltx25_video_extensions_remain_model_specific, test_ltx23_request_defaults_to_vae_decoder) : 为 LTX-2.5 在线扩展字段和 VAE/Diffusion decoder 默认行为补充测试覆盖, 防止重构后行为漂移。
- python/sglang/multimodal_gen/configs/sample/longcat_image.py (模块 模型配置; 类别 source; 类型 core-logic; 符号 image_request_extra_fields) : LongCat-Image 子类补 image_request_extra_fields 声明, 保证其专属字段 (enable_cfg_renorm 等) 在重构后仍能被图像端点接受。

关键符号: request_extra_value, get_declared_request_extra_fields, get_sampling_request_extra_fields, resolve_sampling_params_cls, _video_request_model_kwargs, _image_request_model_kwargs, image_request_extra_fields, video_request_extra_fields, default_image_output_format, default_image_response_format, lower_video_request_kwargs

关键源码片段

python/sglang/multimodal_gen/runtime/entrypoints/openai/video_api.py

视频端点是本次重构的核心入口：删除了 203 行硬编码的 Cosmos3 专属字段处理与 multipart 表单字段，改为通过 get_sampling_request_extra_fields / request_extra_value / resolve_sampling_params_cls 动态提取激活模型声明的字段，通用端点不再感知具体模型。

```
# python/sglang/multimodal_gen/runtime/entrypoints/openai/video_api.py
def _video_request_model_kwargs(
    request: VideoGenerationsRequest,
    sampling_params_cls: type[SamplingParams],
) -> dict[str, Any]:
    """Extract fields owned and declared by the active model contract."""

    kwargs = {}
    # 只取当前激活模型在 SamplingParams 子类上声明的视频扩展字段,
    # 避免 Cosmos3 / ERNIE-Image 等模型的字段泄漏进通用端点。
    for field_name in get_sampling_request_extra_fields(sampling_params_cls, "video"):
        value = _extra_value(request, field_name)
        if value is not None:
            kwargs[field_name] = value
    return kwargs
```

```

# multipart 表单中仅保留服务端控制的通用字段；模型专属字段
# （如 use_guardrails、use_duration_template）已下沉到模型子类声明。
_MULTIPART_EXTRA_FORM_FIELDS = (
    "attention_backend_override",
    "cache_dit_params",
    "cfg_gate_step",
    "enable_cache_dit",
    "quality",
)

def _multipart_extra_form_keys(
    sampling_params_cls: type[SamplingParams],
) -> tuple[str, ...]:
    # 把模型子类声明的 video 扩展字段并入 multipart 表单允许键，
    # 使表单上传路径与 JSON 路径对模型字段的接受行为保持一致。
    return tuple(
        dict.fromkeys(
            (
                *VideoGenerationsRequest.model_fields,
                *_MULTIPART_EXTRA_FORM_FIELDS,
                *sorted(
                    get_declared_request_extra_fields(sampling_params_cls, "video")
                ),
            )
        )
    )

```

python/sglang/multimodal_gen/runtime/entrypoints/openai/utils.py

新增的 request_extra_value、get_declared_request_extra_fields、get_sampling_request_extra_fields、resolve_sampling_params_cls 是本次重构的契约基础设施，统一了 extra_body / extra_json / extra_args / extra_params 四种传输容器的读取与优先级。

```

# python/sglang/multimodal_gen/runtime/entrypoints/openai/utils.py

# 兼容的传输容器：OpenAI SDK 的 extra_body、旧版 extra_json / extra_args、
# vLLM-Omni 风格的 extra_params。
_REQUEST_EXTRA_CONTAINERS = (
    "extra_body",
    "extra_json",
    "extra_args",
    "extra_params",
)

```

```

def request_extra_value(request: Any, field_name: str) -> Any:
    """Read an extension field while preserving top-level precedence.

```

This function only handles transport compatibility. Callers must first use the active SamplingParams subclass to decide which model-owned fields are valid; transport helpers must not introduce per-model allowlists.

"""

```
extra = dict(getattr(request, "model_extra", None) or {})
# 根级直接字段优先于任何嵌套容器。
direct = {
    key: value
    for key, value in extra.items()
    if key not in _REQUEST_EXTRA_CONTAINERS
}
direct = flatten_extra_params(direct)
if field_name in direct and direct[field_name] is not None:
    return direct[field_name]

# 按容器顺序回退查找, extra_body 优先。
for container_name in _REQUEST_EXTRA_CONTAINERS:
    nested = _parse_request_extra_container(extra.get(container_name))
    if field_name in nested and nested[field_name] is not None:
        return nested[field_name]
return None
```

@cache

```
def get_sampling_request_extra_fields(
    sampling_params_cls: type[SamplingParams],
    api: Literal["image", "video"],
) -> frozenset[str]:
    """Return declared extension fields that can initialize SamplingParams.
```

A video declaration may also contain transport-only aliases. Those remain on the request for the model's lowering hook instead of being passed to the dataclass constructor.

"""

```
declared = get_declared_request_extra_fields(sampling_params_cls, api)
init_fields = {
    field.name for field in dataclasses.fields(sampling_params_cls) if field.init
}
# 只有能作为 dataclass 构造参数的声明字段才会被端点提取;
# 纯传输别名留给模型的 lowering hook 处理。
return declared & init_fields
```

[python/sglang/multimodal_gen/configs/sample/cosmos3.py](#)

Cosmos3 是本次重构受影响最大的模型：新增 use_duration_template 等字段、补全 image/video_request_extra_fields 声明、把 guardrails 别名、sound_duration 推导与 action 字段归一化下沉到子类 lowering，是模型侧契约落地的示范样本。

```

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

def _parse_request_value(value: Any) -> Any:
    # multipart 表单传入的字段是字符串，尝试按 JSON 解析回原始类型。
    if not isinstance(value, str):
        return value
    try:
        return json.loads(value)
    except (json.JSONDecodeError, TypeError, ValueError):
        return value

def _optional_int_list(value: Any) -> list[int] | None:
    # 兼容 JSON 字符串、单个整数和整数列表三种写法。
    value = _parse_request_value(value)
    if value is None or (isinstance(value, str) and not value.strip()):
        return None
    if isinstance(value, (list, tuple)):
        return [int(item) for item in value]
    return [int(value)]

@dataclass
class Cosmos3SamplingParams(SamplingParams):
    # 这些字段之前在通用端点和基类里硬编码，现在收归模型子类持有。
    use_duration_template: bool | None = None
    use_resolution_template: bool | None = None
    use_system_prompt: bool | None = None
    use_guardrails: bool | None = None
    sound_duration: float = 0.0
    # ...

    @classmethod
    def image_request_extra_fields(cls) -> frozenset[str]:
        return frozenset(
            {
                "guidance_interval",
                "use_duration_template",
                "use_guardrails",
                "use_resolution_template",
                "use_system_prompt",
            }
        )

    @classmethod
    def video_request_extra_fields(cls) -> frozenset[str]:
        # 图像字段之外追加视频与 action 控制字段。
        return cls.image_request_extra_fields() | frozenset(
            {

```

```

        "action",
        "action_fps",
        "action_mode",
        "action_normalization",
        "condition_frame_indexes",
        "control_path",
        "generate_sound",
        "guardrails",
        "sound_duration",
        # ...
    }
)

@classmethod
def default_image_output_format(cls) -> str:
    # Cosmos3 图像输出默认 PNG，由模型自己声明，不再由通用端点特判。
    return "png"

```

评论区精华

该 PR 的 review 讨论为 0 条，comments 仅有一条作者触发的 `/tag-run-ci-label` 指令。相关设计权衡体现在代码注释中：`request_extra_value` 只负责传输兼容，模型字段合法性必须由激活的 `SamplingParams` 子类决定，传输辅助函数不得引入逐模型的 `allowlist`。

- PR 无 review 讨论，唯一评论是触发 CI 标签指令 (other): 无设计争议记录，设计权衡体现在代码注释与 PR body 中。

风险与影响

- 风险:
 1. 兼容性回归风险：虽然保留了根级字段、`extra_body`、`extra_params`、`extra_args` 的兼容，但 multi-part 表单的 `_MULTIPART_EXTRA_FORM_FIELDS` 从旧的 Cosmos3 专属字段（`use_duration_template` 等）替换为 `attention_backend_override`、`cache_dit_params`、`cfg_gate_step`、`enable_cache_dit`，如果已有客户端以 multipart 表单直接传 Cosmos3 的 `use_guardrails` 等字段，可能不再被识别。
 2. 字段声明遗漏风险：`get_sampling_request_extra_fields` 只返回声明集合与 `dataclass` `init` 字段的交集，如果某个模型子类声明的字段名与 `dataclass` 字段名不一致（如 `transport alias`），会静默丢弃该字段。
 3. 行为默认值变更风险：`Base SamplingParams` 中移除了 `use_duration_template` 等字段后，非 Cosmos3 模型如果请求里带了这些字段且未在子类声明，会被忽略；LTX 的 `use_diffusion_decoder` 默认值从基类移除后，LTX-2.3 等模型默认走 VAE decode，需要在子类显式声明才能切换。
 4. CI 覆盖风险：PR Test (Extra) 标注为 :x: 失败，AMD ROCm 7.2 测试还在运行中，需要确认失败原因是否与本次重构相关。
 5. 测试覆盖范围：单测补充集中在 image API 的协议隔离，video API 的 multipart 兼容路径覆盖较少。

- 影响：影响范围：
 - 用户侧：使用 Cosmos3、ERNIE-Image、Ideogram 4、LTX-2.5 在线 API 的开发者，请求字段不再依赖硬编码的通用端点逻辑，LTX-2.5 的 `auto_duration` / `use_diffusion_decoder` 首次真正被转发；新增 LongCat-Image 部署 cookbook。
 - 系统侧：image/video 端点的扩展字段处理从‘通用端点内硬编码映射’变为‘模型子类声明 + 通用端点动态解析’，新增模型时可以只改子类声明，不需要改入口层，降低后续模型接入成本。
 - 团队侧：为后续模型的 API 参数接入确立了‘模型持有字段声明’的契约模式，后续新增模型必须遵循 `image_request_extra_fields` / `video_request_extra_fields` 声明。
 - 风险标记：multipart 表单字段集变更，基类字段移除可能影响旧请求，CI Extra 测试失败，transport alias 静默丢弃风险

关联脉络

- PR #36504 [diffusion][kernel] support transposed residual-gate add: 同为 diffusion 模块的 API/ 内核能力增强，表明 sglang diffusion 子系统的活跃演进；本 PR 的 LTX-2.5 `auto_duration` / `use_diffusion_decoder` 转发与其同属 diffusion 在线能力建设。
- PR #36521 [diffusion][kernel] avoid 4D scale-shift autotuning: 同样是 diffusion 模块的运行时优化与测试配套，且同时维护 `python/sglang/multimodal_gen` 与 `test/registered/kernels` 下的 diffusion 测试基础设施，与本 PR 的 diffusion API 契约演进同属一个功能域。