

# Anomalib 0.7.0 API 参考文档

本文档整理了 anomalib 0.7.0 版本中常用函数和类的参数说明，按使用频率和重要性排序。

## 目录

### 1. 数据模块 (Data Modules)

- [Folder](#)
- [MVTec](#)

### 2. 异常检测模型 (Models)

- [Patchcore](#)
- [Padim](#)
- [Stfpm](#)
- [Fastflow](#)
- [Cflow](#)
- [EfficientAd](#)
- [Draem](#)
- [Ganomaly](#)
- [Dfm](#)
- [Dfkde](#)
- [ReverseDistillation](#)
- [Cfa](#)
- [Csflow](#)
- [Rkde](#)

### 3. 回调函数 (Callbacks)

- [MetricsConfigurationCallback](#)
- [PostProcessingConfigurationCallback](#)
- [ImageVisualizerCallback](#)
- [MetricVisualizerCallback](#)
- [TilerConfigurationCallback](#)
- [LoadModelCallback](#)
- [TimerCallback](#)

### 4. 枚举类型 (Enums)

- [TaskType](#)
- [NormalizationMethod](#)
- [ThresholdMethod](#)
- [TestSplitMode](#)
- [ValSplitMode](#)
- [VisualizationMode](#)

## 5. 推理工具 (Inference)

- TorchInferencer

## 6. 完整使用示例

# 1. 数据模块 (Data Modules)

## 1.1 Folder

**导入路径:** from anomalib.data.folder import Folder

**用途:** 自定义文件夹数据集，适用于自己组织的工业缺陷检测数据

### 参数列表

参数名	类型	默认值	说明
normal_dir	str   Path   Sequence	必需	正常图像目录路径。 用于训练和测试
root	str   Path   None	None	数据集根目录。如果提供，则 normal_dir 等参数为相对路径
abnormal_dir	str   Path   Sequence   None	None	异常图像目录路径。用于测试
normal_test_dir	str   Path   Sequence   None	None	测试集正常图像目录。默认从 normal_dir 分割
mask_dir	str   Path   Sequence   None	None	异常掩码目录路径。 分割任务必需
normal_split_ratio	float	0.2	正常图像划分到测试集的比例 (当测试集无正常图像时)
extensions	tuple[str]   None	None	读取的图像扩展名，如 ('.png', '.jpg')。None 表示所有常见格式
image_size	int   tuple[int, int]   None	None	输入图像尺寸。如 (256, 256) 或 256
center_crop	int   tuple[int, int]   None	None	中心裁剪尺寸
normalization	str   InputNormalizationMethod	"imagenet"	图像归一化方式: "imagenet" , "none"
train_batch_size	int	32	训练批次大小

参数名	类型	默认值	说明
eval_batch_size	int	32	评估批次大小
num_workers	int	8	数据加载工作线程数
task	TaskType	segmentation	任务类型: classification , detection , segmentation
transform_config_train	str   A.Compose   None	None	训练时的数据增强配置
transform_config_eval	str   A.Compose   None	None	评估时的数据增强配置
test_split_mode	TestSplitMode	from_dir	测试集划分模式
test_split_ratio	float	0.2	测试集划分比例
val_split_mode	ValSplitMode	from_test	验证集划分模式
val_split_ratio	float	0.5	验证集划分比例
seed	int   None	None	随机种子

## 目录结构示例

```

datasets/task_1/
├── good/          # 正常图像 (用于训练和测试)
│   ├── 000.png
│   └── ...
└── defect/        # 异常图像 (用于测试)
    ├── 000.png
    └── ...
└── mask/
    └── defect/      # 异常掩码 (与 defect/ 一一对应)
        ├── 000.png
        └── ...

```

## 使用示例

```
from anomalib.data.folder import Folder
from anomalib.data.task_type import TaskType

datamodule = Folder(
    normal_dir="./datasets/task_1/good",
    abnormal_dir="./datasets/task_1/defect",
    mask_dir="./datasets/task_1/mask/defect",
    image_size=(1024, 1024),
    train_batch_size=1,
    eval_batch_size=1,
    num_workers=8,
    task=TaskType.SEGMENTATION,
)
datamodule.setup()
```

## 1.2 MVTec

导入路径: `from anomalib.data.mvtec import MVTec`

用途: MVTec AD 标准数据集

### 参数列表

参数名	类型	默认值	说明
root	Path   str	必需	MVTec 数据集根目录
category	str	必需	类别名称, 如 "bottle", "cable" 等
image_size	int   tuple[int, int]   None	None	输入图像尺寸
center_crop	int   tuple[int, int]   None	None	中心裁剪尺寸
normalization	str   InputNormalizationMethod	"imagenet"	图像归一化方式
train_batch_size	int	32	训练批次大小
eval_batch_size	int	32	评估批次大小
num_workers	int	8	数据加载线程数
task	TaskType	segmentation	任务类型
transform_config_train	str   A.Compose   None	None	训练数据增强
transform_config_eval	str   A.Compose   None	None	评估数据增强

参数名	类型	默认值	说明
test_split_mode	TestSplitMode	from_dir	测试集划分模式
test_split_ratio	float	0.2	测试集比例
val_split_mode	ValSplitMode	same_as_test	验证集划分模式
val_split_ratio	float	0.5	验证集比例
seed	int   None	None	随机种子

## 2. 异常检测模型 (Models)

### 2.1 Patchcore ★ 推荐

导入路径: `from anomalib.models.patchcore import Patchcore`

特点:

- 基于记忆库的异常检测方法
- 只需要 1 个 epoch 训练
- 适合高精度要求的场景
- **推荐用于工业缺陷检测**

### 参数列表

参数名	类型	默认值	说明
input_size	tuple[int, int]	必需	输入图像尺寸, 如 (1024, 1024)
backbone	str	必需	骨干网络名称, 如 "resnet18", "wide_resnet50_2"
layers	list[str]	必需	提取特征的层, 如 ["layer2", "layer3"]
pre_trained	bool	True	是否使用预训练权重
coreset_sampling_ratio	float	0.1	核心集采样比例。较小值减少内存, 但可能降低性能
num_neighbors	int	9	K近邻数量。影响异常分数计算

### 可用骨干网络

- resnet18 - 轻量级, 速度快
- resnet50 - 平衡速度和精度
- wide\_resnet50\_2 - 高精度, 内存占用大

## 可用特征层

- layer1 - 浅层特征，局部细节
- layer2 - 中层特征
- layer3 - 深层特征，语义信息
- layer4 - 最深层特征

## 使用示例

```
from anomalib.models.patchcore import Patchcore

model = Patchcore(
    input_size=(1024, 1024),
    backbone="resnet18",
    layers=["layer2", "layer3"],
    pre_trained=True,
    coreset_sampling_ratio=0.01, # 减少内存占用
    num_neighbors=9,
)
```

## 2.2 Padim

导入路径: `from anomalib.models.padim import Padim`

特点:

- 基于统计分布建模
- 无需训练，只需特征提取
- 内存占用较小

## 参数列表

参数名	类型	默认值	说明
layers	list[str]	必需	提取特征的层
input_size	tuple[int, int]	必需	输入图像尺寸
backbone	str	必需	骨干网络
pre_trained	bool	True	是否使用预训练权重
n_features	int   None	None	降维后的特征数。resnet18 默认 100, wide_resnet50_2 默认 550

## 使用示例

```
from anomalib.models.padim import Padim

model = Padim(
    layers=["layer1", "layer2", "layer3"],
    input_size=(256, 256),
    backbone="resnet18",
    pre_trained=True,
    n_features=100,
)
```

## 2.3 Stfpm

导入路径: `from anomalib.models.stfpmp import Stfpmp`

特点:

- 学生-教师特征金字塔匹配
- 需要多轮训练
- 适合追求速度的场景

## 参数列表

参数名	类型	默认值	说明
input_size	tuple[int, int]	必需	输入图像尺寸
backbone	str	必需	骨干网络
layers	list[str]	必需	提取特征的层

## 2.4 Fastflow

导入路径: `from anomalib.models.fastflow import Fastflow`

特点:

- 基于归一化流的方法
- 推理速度快
- 适合实时检测场景

## 参数列表

参数名	类型	默认值	说明
input_size	tuple[int, int]	必需	输入图像尺寸
backbone	str	必需	骨干网络
pre_trained	bool	True	是否使用预训练权重
flow_steps	int	8	流模型步数。步数越多模型越强大，但训练越慢
conv3x3_only	bool	False	是否只使用 3x3 卷积
hidden_ratio	float	1.0	隐藏通道数比例

## 2.5 Cflow

导入路径: `from anomalib.models.cflow import Cflow`

特点:

- 条件归一化流
- 高检测精度
- 训练时间较长

## 参数列表

参数名	类型	默认值	说明
input_size	tuple[int, int]	必需	输入图像尺寸
backbone	str	必需	骨干网络
layers	list[str]	必需	提取特征的层
pre_trained	bool	True	是否使用预训练权重
fiber_batch_size	int	64	纤维批次大小
decoder	str	"freia-cflow"	解码器类型
condition_vector	int	128	条件向量维度
coupling_blocks	int	8	耦合块数量
clamp_alpha	float	1.9	钳制系数
permute_soft	bool	False	是否使用软置换
lr	float	0.0001	学习率

## 2.6 EfficientAd

导入路径: `from anomalib.models.efficient_ad import EfficientAd`

特点:

- 轻量级模型
- 适合边缘部署
- 推理速度极快

### 参数列表

参数名	类型	默认值	说明
teacher_out_channels	int	必需	教师网络输出通道数
image_size	tuple[int, int]	必需	输入图像尺寸
model_size	EfficientAdModelSize	small	模型大小: small 或 medium
lr	float	0.0001	学习率
weight_decay	float	1e-05	权重衰减
padding	bool	False	是否在卷积层使用填充
pad_maps	bool	True	是否填充输出异常图
batch_size	int	1	ImageNet 数据加载批次大小

## 2.7 Draem

导入路径: `from anomalib.models.draem import Draem`

特点:

- 使用异常合成进行训练
- 不需要异常样本
- 适合无异常数据的场景

### 参数列表

参数名	类型	默认值	说明
enable_sspcab	bool	False	是否启用 SSPCAB 模块
sspcab_lambda	float	0.1	SSPCAB 损失权重

参数名	类型	默认值	说明
anomaly_source_path	str   None	None	异常纹理图像路径（用于合成异常）

## 2.8 Ganomaly

导入路径: `from anomalib.models.ganomaly import Ganomaly`

特点:

- 基于 GAN 的方法
- 需要多轮训练
- 适合分类任务

## 参数列表

参数名	类型	默认值	说明
batch_size	int	必需	批次大小
input_size	tuple[int, int]	必需	输入图像尺寸
n_features	int	必需	CNN 特征层数量
latent_vec_size	int	必需	自编码器潜在向量大小
extra_layers	int	0	编码器/解码器额外层数
add_final_conv_layer	bool	True	是否添加最终卷积层
wadv	int	1	对抗损失权重
wcon	int	50	图像重建损失权重
wenc	int	1	潜在向量编码器损失权重
lr	float	0.0002	学习率
beta1	float	0.5	Adam 优化器 $\beta_1$
beta2	float	0.999	Adam 优化器 $\beta_2$

## 2.9 Dfm

导入路径: `from anomalib.models.dfm import Dfm`

特点:

- 深度特征建模
- 支持分类和分割任务
- 无需训练

## 参数列表

参数名	类型	默认值	说明
backbone	str	必需	骨干网络
layer	str	必需	提取特征的层
input_size	tuple[int, int]	必需	输入图像尺寸
pre_trained	bool	True	是否使用预训练权重
pooling_kernel_size	int	4	池化核大小
pca_level	float	0.97	PCA 保留方差比例
score_type	str	"fre"	评分类型: "fre" (特征重建误差) 或 "nll" (负对数似然)

**注意:** 使用 `nll` 时只支持分类任务, `fre` 支持分割任务

## 2.10 Dfkde

**导入路径:** `from anomalib.models.dfkde import Dfkde`

**特点:**

- 深度特征核密度估计
- 适合分类任务
- 无需训练

## 参数列表

参数名	类型	默认值	说明
layers	list[str]	必需	提取特征的层
backbone	str	必需	骨干网络
pre_trained	bool	True	是否使用预训练权重
n_pca_components	int	16	PCA 组件数量
feature_scaling_method	FeatureScalingMethod	scale	特征缩放方法: "norm" 或 "scale"

参数名	类型	默认值	说明
max_training_points	int	40000	KDE 拟合的最大训练点数

## 2.11 ReverseDistillation

导入路径: `from anomalib.models.reverse_distillation import ReverseDistillation`

特点:

- 反向蒸馏方法
- 高检测精度
- 需要多轮训练

## 参数列表

参数名	类型	默认值	说明
input_size	tuple[int, int]	必需	输入图像尺寸
backbone	str	必需	骨干网络
layers	list[str]	必需	提取特征的层
anomaly_map_mode	AnomalyMapGenerationMode	必需	异常图生成模式
lr	float	必需	学习率
beta1	float	必需	Adam 优化器 $\beta_1$
beta2	float	必需	Adam 优化器 $\beta_2$
pre_trained	bool	True	是否使用预训练权重

## 2.12 Cfa

导入路径: `from anomalib.models.cfa import Cfa`

特点:

- 耦合双曲流异常检测
- 高检测精度

## 参数列表

参数名	类型	默认值	说明
input_size	tuple[int, int]	必需	输入图像尺寸
backbone	str	必需	骨干网络
gamma_c	int	1	中心损失权重
gamma_d	int	1	距离损失权重
num_nearest_neighbors	int	3	K近邻数量
num_hard_negative_features	int	3	难负样本特征数
radius	float	1e-05	球半径

## 2.13 Csflow

导入路径: `from anomalib.models.csflow import Csflow`

特点:

- 交叉尺度归一化流
- 多尺度特征融合

## 参数列表

参数名	类型	默认值	说明
input_size	tuple[int, int]	必需	输入图像尺寸
cross_conv_hidden_channels	int	必需	交叉卷积隐藏通道数
n_coupling_blocks	int	必需	耦合块数量
clamp	int	必需	钳制值
num_channels	int	必需	通道数

## 2.14 Rkde

导入路径: `from anomalib.models.rkde import Rkde`

特点:

- 基于区域的核密度估计

- 适合检测任务

## 参数列表

参数名	类型	默认值	说明
roi_stage	RoiStage	rcnn	ROI 提取阶段
roi_score_threshold	float	0.001	ROI 分数阈值
min_box_size	int	25	最小边界框大小
iou_threshold	float	0.3	IoU 阈值
max_detections_per_image	int	100	每张图像最大检测数
n_pca_components	int	16	PCA 组件数
feature_scaling_method	FeatureScalingMethod	scale	特征缩放方法
max_training_points	int	40000	最大训练点数

## 3. 回调函数 (Callbacks)

### 3.1 MetricsConfigurationCallback

导入路径: `from anomalib.utils.callbacks import MetricsConfigurationCallback`

用途: 配置评估指标

## 参数列表

参数名	类型	默认值	说明
task	TaskType	segmentation	任务类型
image_metrics	list[str]   None	None	图像级别指标列表
pixel_metrics	list[str]   None	None	像素级别指标列表 (仅分割任务)

## 可用指标

- "AUROC" - 曲线下面积 (最常用)
- "F1Score" - F1 分数
- "AUPR" - 精确率-召回率曲线下面积
- "Precision" - 精确率
- "Recall" - 召回率

## 使用示例

```
from anomalib.utils.callbacks import MetricsConfigurationCallback
from anomalib.data.task_type import TaskType

callback = MetricsConfigurationCallback(
    task=TaskType.SEGMENTATION,
    image_metrics=["AUROC", "F1Score"],
    pixel_metrics=["AUROC", "F1Score"],
)
```

## 3.2 PostProcessingConfigurationCallback

导入路径: `from anomalib.utils.callbacks import PostProcessingConfigurationCallback`

用途: 配置后处理参数（归一化和阈值）

### 参数列表

参数名	类型	默认值	说明
normalization_method	NormalizationMethod	min_max	异常分数归一化方法
threshold_method	ThresholdMethod	adaptive	阈值确定方法
manual_image_threshold	float   None	None	手动图像阈值（仅当 threshold_method=manual）
manual_pixel_threshold	float   None	None	手动像素阈值（仅当 threshold_method=manual）

## 使用示例

```
from anomalib.utils.callbacks import PostProcessingConfigurationCallback
from anomalib.post_processing import NormalizationMethod, ThresholdMethod

callback = PostProcessingConfigurationCallback(
    normalization_method=NormalizationMethod.MIN_MAX,
    threshold_method=ThresholdMethod.ADAPTIVE,
)
```

## 3.3 ImageVisualizerCallback

导入路径: `from anomalib.utils.callbacks import ImageVisualizerCallback`

**用途:** 可视化推理结果图像

## 参数列表

参数名	类型	默认值	说明
task	TaskType	必需	任务类型
mode	VisualizationMode	必需	可视化模式: full 或 simple
image_save_path	str	必需	图像保存路径
inputs_are_normalized	bool	True	输入是否已归一化
show_images	bool	False	是否显示图像
log_images	bool	True	是否记录到日志
save_images	bool	True	是否保存图像

## 3.4 MetricVisualizerCallback

**导入路径:** from anomalib.utils.callbacks import MetricVisualizerCallback

**用途:** 可视化评估指标曲线

## 参数列表

参数名	类型	默认值	说明
task	TaskType	必需	任务类型
mode	VisualizationMode	必需	可视化模式
image_save_path	str	必需	图像保存路径
inputs_are_normalized	bool	True	输入是否已归一化
show_images	bool	False	是否显示图像
log_images	bool	True	是否记录到日志
save_images	bool	True	是否保存图像

## 3.5 TilerConfigurationCallback

**导入路径:** from anomalib.utils.callbacks import TilerConfigurationCallback

**用途:** 配置图像分块处理（用于处理大尺寸图像）

## 参数列表

参数名	类型	默认值	说明
enable	bool	False	是否启用分块
tile_size	int   Sequence	256	分块大小
stride	int   Sequence   None	None	分块步长。None 表示与 tile_size 相同
remove_border_count	int	0	移除边界像素数
mode	ImageUpscaleMode	padding	上采样模式
tile_count	int	4	分块数量

## 3.6 LoadModelCallback

**导入路径:** from anomalib.utils.callbacks import LoadModelCallback

**用途:** 加载预训练模型权重

## 参数列表

参数名	类型	默认值	说明
weights_path	Any	必需	模型权重文件路径

## 3.7 TimerCallback

**导入路径:** from anomalib.utils.callbacks import TimerCallback

**用途:** 测量训练和测试时间

## 参数列表

无参数

## 4. 枚举类型 (Enums)

### 4.1 TaskType

导入路径: `from anomalib.data.task_type import TaskType`

值	说明
<code>TaskType.CLASSIFICATION</code>	分类任务 - 判断图像是否异常
<code>TaskType.DETECTION</code>	检测任务 - 定位异常区域（边界框）
<code>TaskType.SEGMENTATION</code>	分割任务 - 像素级异常定位

### 4.2 NormalizationMethod

导入路径: `from anomalib.post_processing import NormalizationMethod`

值	说明
<code>NormalizationMethod.MIN_MAX</code>	最小-最大归一化，映射到 [0, 1]
<code>NormalizationMethod.CDF</code>	累积分布函数归一化
<code>NormalizationMethod.NONE</code>	不进行归一化

### 4.3 ThresholdMethod

导入路径: `from anomalib.post_processing import ThresholdMethod`

值	说明
<code>ThresholdMethod.ADAPTIVE</code>	自适应阈值 - 根据验证集自动确定
<code>ThresholdMethod.MANUAL</code>	手动阈值 - 需要指定阈值值

### 4.4 TestSplitMode

导入路径: `from anomalib.data.base.datamodule import TestSplitMode`

值	说明
TestSplitMode.NONE	不划分测试集
TestSplitMode.FROM_DIR	从目录读取测试集
TestSplitMode.SYNTHETIC	合成测试集

## 4.5 ValSplitMode

导入路径: `from anomalib.data.base.datamodule import ValSplitMode`

值	说明
ValSplitMode.NONE	不使用验证集
ValSplitMode.SAME_AS_TEST	验证集与测试集相同
ValSplitMode.FROM_TEST	从测试集划分验证集
ValSplitMode.SYNTHETIC	合成验证集

## 4.6 VisualizationMode

导入路径: `from anomalib.post_processing.visualizer import VisualizationMode`

值	说明
VisualizationMode.FULL	完整可视化 - 显示所有信息
VisualizationMode.SIMPLE	简单可视化 - 仅显示基本信息

# 5. 推理工具 (Inference)

## 5.1 TorchInferencer

导入路径: `from anomalib.deploy import TorchInferencer`

用途: 使用训练好的模型进行推理

## 参数列表

参数名	类型	默认值	说明
path	str   Path	必需	模型权重文件路径 (.ckpt 或 .pt)
device	str	"auto"	推理设备: "auto" , "cpu" , "cuda"

## 使用示例

```
from anomalib.deploy import TorchInferencer

# 加载模型
inferencer = TorchInferencer(
    path="results/task_1/weights/model.ckpt",
    device="auto",
)

# 推理单张图像
from PIL import Image
image = Image.open("test_image.png")
result = inferencer.predict(image)

# 获取结果
anomaly_map = result.anomaly_map # 异常热力图
pred_score = result.pred_score # 图像级异常分数
pred_label = result.pred_label # 预测标签 (0: 正常, 1: 异常)
pred_mask = result.pred_mask # 预测掩码 (分割任务)
```

# 6. 完整使用示例

## 基础训练脚本

```
#!/usr/bin/env python3
"""anomalib 0.7.0 完整训练示例"""

from pathlib import Path
from pytorch_lightning import Trainer
from pytorch_lightning.callbacks import ModelCheckpoint
from pytorch_lightning.loggers import TensorBoardLogger

from anomalib.data.folder import Folder
from anomalib.data.task_type import TaskType
from anomalib.models.patchcore import Patchcore
from anomalib.post_processing import NormalizationMethod, ThresholdMethod
from anomalib.utils.callbacks import (
    MetricsConfigurationCallback,
    PostProcessingConfigurationCallback,
)
)

def main():
    # 1. 配置数据模块
    dataset_root = Path("./datasets/task_1")

    datamodule = Folder(
        normal_dir=dataset_root / "good",
        abnormal_dir=dataset_root / "defect",
        mask_dir=dataset_root / "mask" / "defect",
        image_size=(1024, 1024),
        train_batch_size=1,
        eval_batch_size=1,
        num_workers=8,
        task=TaskType.SEGMENTATION,
    )
    datamodule.setup()

    # 2. 配置模型
    model = Patchcore(
        input_size=(1024, 1024),
        backbone="resnet18",
        layers=["layer2", "layer3"],
        pre_trained=True,
        coreset_sampling_ratio=0.01,
        num_neighbors=9,
    )

    # 3. 配置回调
```

```
output_dir = Path("results/task_1")
output_dir.mkdir(parents=True, exist_ok=True)

callbacks = [
    MetricsConfigurationCallback(
        task=TaskType.SEGMENTATION,
        image_metrics=["AUROC", "F1Score"],
        pixel_metrics=["AUROC", "F1Score"],
    ),
    PostProcessingConfigurationCallback(
        normalization_method=NormalizationMethod.MIN_MAX,
        threshold_method=ThresholdMethod.ADAPTIVE,
    ),
    ModelCheckpoint(
        dirpath=output_dir / "weights",
        filename="model",
        monitor="pixel_AUROC",
        mode="max",
        save_last=True,
    ),
]
]

# 4. 配置训练器
trainer = Trainer(
    max_epochs=1, # Patchcore 只需1个epoch
    accelerator="gpu",
    devices=1,
    default_root_dir=output_dir,
    logger=TensorBoardLogger(save_dir="logs/", name="task_1_log"),
    callbacks=callbacks,
    num_sanity_val_steps=0,
)

# 5. 训练
trainer.fit(model=model, datamodule=datamodule)

# 6. 测试
test_results = trainer.test(model=model, datamodule=datamodule)

print("测试结果: ")
for key, value in test_results[0].items():
    if isinstance(value, float):
        print(f" {key}: {value:.4f}")

if __name__ == "__main__":
    main()
```

## 附录：所有可用模型一览

模型名	导入路径	特点	推荐场景
<b>Patchcore</b>	anomalib.models.patchcore	高精度、1 epoch	工业缺陷检测 <span style="color: yellow;">★</span>
<b>Padim</b>	anomalib.models.padim	无训练、内存小	快速部署
<b>Stfpm</b>	anomalib.models.stfpm	学生-教师、速度快	实时检测
<b>Fastflow</b>	anomalib.models.fastflow	归一化流、快速推理	实时检测
<b>Cflow</b>	anomalib.models.cflow	条件流、高精度	追求精度
<b>EfficientAd</b>	anomalib.models.efficient_ad	轻量级、极快推理	边缘部署
<b>Draem</b>	anomalib.models.draem	异常合成	无异常样本
<b>Ganomaly</b>	anomalib.models.ganomaly	GAN 方法	分类任务
<b>Dfm</b>	anomalib.models.dfm	特征建模	快速部署
<b>Dfkde</b>	anomalib.models.dfkde	核密度估计	分类任务
<b>ReverseDistillation</b>	anomalib.models.reverse_distillation	反向蒸馏	高精度
<b>Cfa</b>	anomalib.models.cfa	耦合双曲流	高精度
<b>Csflow</b>	anomalib.models.csflow	交叉尺度流	多尺度检测
<b>Rkde</b>	anomalib.models.rkde	区域核密度	检测任务
<b>AiVad</b>	anomalib.models.ai_vad	视频异常检测	视频数据

## 附录：所有可用数据模块一览

数据模块	导入路径	用途
<b>Folder</b>	anomalib.data.folder	自定义文件夹数据集 <span style="color: yellow;">★</span>
<b>MVTec</b>	anomalib.data.mvtac	MVTec AD 数据集
<b>BTech</b>	anomalib.data.btech	BTech 数据集
<b>Visa</b>	anomalib.data.visa	VisA 数据集
<b>Folder3D</b>	anomalib.data.folder_3d	3D 文件夹数据集
<b>MVTec3D</b>	anomalib.data.mvtac_3d	MVTec 3D 数据集
<b>Avenue</b>	anomalib.data.avenue	Avenue 视频数据集

数据模块	导入路径	用途
<b>ShanghaiTech</b>	<code>anomalib.data.shanghaitech</code>	ShanghaiTech 视频数据集
<b>UCSDped</b>	<code>anomalib.data.ucsd_ped</code>	UCSD Ped 视频数据集

文档生成日期: 2026年1月31日

适用版本: *anomalib 0.7.0*