

# Base Datamodules

## Contents

- `AnomalibDataModule`
- `collate_fn()`

Anomalib datamodule base class.

```
class anomalib.data.base.datamodule.AnomalibDataModule(train_batch_size,  
eval_batch_size, num_workers, val_split_mode, val_split_ratio,  
test_split_mode=None, test_split_ratio=None, image_size=None,  
transform=None, train_transform=None, eval_transform=None, seed=None)
```

Bases: `LightningDataModule`, `ABC`

Base Anomalib data module.

### Parameters:

- **train\_batch\_size** (*int*) – Batch size used by the train dataloader.
- **eval\_batch\_size** (*int*) – Batch size used by the val and test dataloaders.
- **num\_workers** (*int*) – Number of workers used by the train, val and test dataloaders.
- **val\_split\_mode** ([ValSplitMode](#)) – Determines how the validation split is obtained. Options: [none, same\_as\_test, from\_test, synthetic]
- **val\_split\_ratio** (*float*) – Fraction of the train or test images held out for validation.
- **test\_split\_mode** (*Optional*[[TestSplitMode](#)], *optional*) – Determines how the test split is obtained. Options: [none, from\_dir, synthetic]. Defaults to `None`.
- **test\_split\_ratio** (*float*) – Fraction of the train images held out for testing. Defaults to `None`.
- **image\_size** (*tuple*[*int*, *int*], *optional*) – Size to which input images should be resized. Defaults to `None`.
- **transform** (*Transform*, *optional*) – Transforms that should be applied to the input images. Defaults to `None`.
- **train\_transform** (*Transform*, *optional*) – Transforms that should be applied to the input images during training. Defaults to `None`.
- **eval\_transform** (*Transform*, *optional*) – Transforms that should be applied to the input images during evaluation. Defaults to `None`.
- **seed** (*int* | *None*, *optional*) – Seed used during random subset splitting. Defaults to `None`.

**property** **category:** *str*

Get the category of the datamodule.

**property** **eval\_transform:** *Transform*

Get the transform that will be passed to the val/test/predict datasets.

If the eval\_transform is not set, the engine will request the transform from the model.

**property** **name:** *str*

Name of the datamodule.

### `predict_dataloader()`

Use the test dataloader for inference unless overridden.

**Return type:**

`Any`

### `setup(stage=None)`

Set up train, validation and test data.

**Parameters:**

**stage** (`str` | `None`) – str | None: Train/Val/Test stages. Defaults to `None`.

**Return type:**

`None`

### `test_dataloader()`

Get test dataloader.

**Return type:**

`Any`

### `train_dataloader()`

Get train dataloader.

**Return type:**

`Any`

### *property* `train_transform: Transform`

Get the transforms that will be passed to the train dataset.

If the `train_transform` is not set, the engine will request the transform from the model.

**property `transform`: *Transform***

Property that returns the user-specified transform for the datamodule, if any.

This property is accessed by the engine to set the transform for the model. The `eval_transform` takes precedence over the `train_transform`, because the transform that we store in the model is the one that should be used during inference.

**`val_dataloader()`**

Get validation dataloader.

**Return type:**

Any

**`anomalib.data.base.datamodule.collate_fn(batch)`**

Collate bounding boxes as lists.

Bounding boxes are collated as a list of tensors, while the default collate function is used for all other entries.

**Parameters:**

**batch** (*List*) – list of items in the batch where `len(batch)` is equal to the batch size.

**Returns:**

Dictionary containing the collated batch information.

**Return type:**

`dict[str, Any]`