
deepdrivemd

Release 0.0.2

Alexander Brace, Hyungro Lee, Heng Ma, Anda Trifan, Matteo Turci

May 17, 2022

CONTENTS

1	Summary	3
2	Getting involved	5
3	Installing DeepDriveMD	7
3.1	pip	7
3.1.1	API	7
4	Indices and tables	169
	Python Module Index	171
	Index	173

DeepDriveMD: Deep-Learning Driven Adaptive Molecular Simulations.

Release 0.0.2

Date May 17, 2022

SUMMARY

deepdrivemd is a Python package for coupling molecular dynamics ensemble simulations to sampling agents guided by machine learning.

DeepDriveMD can support two modes of execution, one is synchronous and runs MD simulations, aggregation, training, and inference stages in a pipeline where each stage blocks the others and the stages communicate via the filesystem (**DeepDriveMD-F**).

The second, and more optimal, mode of execution is asynchronous where each of the previously mentioned stages run continuously as independent components communicating via [adios2](#) to stream data between concurrently running workflow components, enabling efficient feedback between simulations and learning algorithms (**DeepDriveMD-S**).

Both modes of execution are implemented using [RADICAL-Ensemble Toolkit](#) to enable support for large scale runs on high-performance computing platforms.

Additional information can be found on our [website](#).

GETTING INVOLVED

Please report **bugs** or **enhancement requests** through the [Issue Tracker](#).

INSTALLING DEEPPDRIVEMD

To install the latest release, clone the code from the [main branch](#) and use pip to install the package.

3.1 pip

Installation with *pip* and a *minimal set of dependencies*:

```
git clone https://github.com/DeepDriveMD/DeepDriveMD-pipeline
cd deepdrivemd
pip install -e .
```

3.1.1 API

deepdrivemd

deepdrivemd

Modules

deepdrivemd.agents

deepdrivemd.aggregation

deepdrivemd.config

Schema of the YAML experiment file

deepdrivemd.data

deepdrivemd.deepdrivemd

deepdrivemd.deepdrivemd_stream

deepdrivemd.models

deepdrivemd.selection

deepdrivemd.sim

deepdrivemd.utils

deepdrivemd.agents

Modules

deepdrivemd.agents.lof

deepdrivemd.agents.stream

deepdrivemd.agents.lof

Modules

deepdrivemd.agents.lof.config

deepdrivemd.agents.lof.config**pydantic settings** deepdrivemd.agents.lof.config.OutlierDetectionConfig

Outlier detection algorithm configuration.

```

{
  "title": "OutlierDetectionConfig",
  "description": "Outlier detection algorithm configuration.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'experiment_directory'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    "task_idx": {
      "title": "Task Idx",
      "default": 0,
      "env_names": "'task_idx'",
      "type": "integer"
    },
    "output_path": {
      "title": "Output Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'output_path'",
      "type": "string",
      "format": "path"
    },
    "node_local_path": {
      "title": "Node Local Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'node_local_path'",
      "type": "string",
      "format": "path"
    },
    "num_intrinsic_outliers": {
      "title": "Num Intrinsic Outliers",
      "default": 100,
      "env_names": "'num_intrinsic_outliers'",
      "type": "integer"
    },
    "num_extrinsic_outliers": {
      "title": "Num Extrinsic Outliers",
      "default": 100,
      "env_names": "'num_extrinsic_outliers'",

```

(continues on next page)

(continued from previous page)

```

        "type": "integer"
    },
    "intrinsic_score": {
        "title": "Intrinsic Score",
        "default": "lof",
        "env_names": "'intrinsic_score'",
        "type": "string"
    },
    "extrinsic_score": {
        "title": "Extrinsic Score",
        "env_names": "'extrinsic_score'",
        "type": "string"
    },
    "n_traj_frames": {
        "title": "N Traj Frames",
        "default": 200,
        "env_names": "'n_traj_frames'",
        "type": "integer"
    },
    "n_most_recent_h5_files": {
        "title": "N Most Recent H5 Files",
        "default": 10,
        "env_names": "'n_most_recent_h5_files'",
        "type": "integer"
    },
    "k_random_old_h5_files": {
        "title": "K Random Old H5 Files",
        "default": 0,
        "env_names": "'k_random_old_h5_files'",
        "type": "integer"
    },
    "sklearn_num_jobs": {
        "title": "Sklearn Num Jobs",
        "default": -1,
        "env_names": "'sklearn_num_jobs'",
        "type": "integer"
    },
    "model_type": {
        "title": "Model Type",
        "default": "AAE3d",
        "env_names": "'model_type'",
        "type": "string"
    },
    "inference_batch_size": {
        "title": "Inference Batch Size",
        "default": 128,
        "env_names": "'inference_batch_size'",
        "type": "integer"
    }
}

```

Config

- **extra:** *str = allow*

Fields

- *extrinsic_score* (*Optional[str]*)
- *inference_batch_size* (*int*)
- *intrinsic_score* (*Optional[str]*)
- *k_random_old_h5_files* (*int*)
- *model_type* (*str*)
- *n_most_recent_h5_files* (*int*)
- *n_traj_frames* (*int*)
- *num_extrinsic_outliers* (*int*)
- *num_intrinsic_outliers* (*int*)
- *sklearn_num_jobs* (*int*)

Validators

- *model_type_check* » *model_type*
- *num_outliers_check* » all fields
- *scoring_method_check* » all fields

field extrinsic_score: *Optional[str] = None*

Validated by

- *num_outliers_check*
- *scoring_method_check*

field inference_batch_size: *int = 128*

Validated by

- *num_outliers_check*
- *scoring_method_check*

field intrinsic_score: *Optional[str] = 'lof'*

Validated by

- *num_outliers_check*
- *scoring_method_check*

field k_random_old_h5_files: *int = 0*

Validated by

- *num_outliers_check*
- *scoring_method_check*

```
field model_type:  str = 'AAE3d'

    Validated by
        • num_outliers_check
        • model_type_check
        • scoring_method_check

field n_most_recent_h5_files:  int = 10

    Validated by
        • num_outliers_check
        • scoring_method_check

field n_traj_frames:  int = 200

    Validated by
        • num_outliers_check
        • scoring_method_check

field num_extrinsic_outliers:  int = 100

    Validated by
        • num_outliers_check
        • scoring_method_check

field num_intrinsic_outliers:  int = 100

    Validated by
        • num_outliers_check
        • scoring_method_check

field sklearn_num_jobs:  int = -1

    Validated by
        • num_outliers_check
        • scoring_method_check

validator model_type_check » model_type
validator num_outliers_check » all fields
validator scoring_method_check » all fields
```

deepdrivemd.agents.stream

Modules

deepdrivemd.agents.stream.config

deepdrivemd.agents.stream.config

pydantic settings `deepdrivemd.agents.stream.config.OutlierDetectionConfig`

Outlier detection algorithm configuration.

```

{
  "title": "OutlierDetectionConfig",
  "description": "Outlier detection algorithm configuration.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'experiment_directory'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    "task_idx": {
      "title": "Task Idx",
      "default": 0,
      "env_names": "'task_idx'",
      "type": "integer"
    },
    "output_path": {
      "title": "Output Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'output_path'",
      "type": "string",
      "format": "path"
    },
    "node_local_path": {
      "title": "Node Local Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'node_local_path'",
      "type": "string",
      "format": "path"
    },
    "agg_dir": {

```

(continues on next page)

(continued from previous page)

```

    "title": "Agg Dir",
    "default": ".",
    "env_names": "'agg_dir'",
    "type": "string",
    "format": "path"
  },
  "num_agg": {
    "title": "Num Agg",
    "default": 2,
    "env_names": "'num_agg'",
    "type": "integer"
  },
  "min_step_increment": {
    "title": "Min Step Increment",
    "default": 500,
    "env_names": "'min_step_increment'",
    "type": "integer"
  },
  "timeout1": {
    "title": "Timeout1",
    "default": 30,
    "env_names": "'timeout1'",
    "type": "integer"
  },
  "timeout2": {
    "title": "Timeout2",
    "default": 10,
    "env_names": "'timeout2'",
    "type": "integer"
  },
  "best_model": {
    "title": "Best Model",
    "default": ".",
    "env_names": "'best_model'",
    "type": "string",
    "format": "path"
  },
  "lastN": {
    "title": "Lastn",
    "default": 8000,
    "env_names": "'lastn'",
    "type": "integer"
  },
  "latent_dim": {
    "title": "Latent Dim",
    "default": 10,
    "env_names": "'latent_dim'",
    "type": "integer"
  },
  "conv_layers": {
    "title": "Conv Layers",
    "default": 4,

```

(continues on next page)

(continued from previous page)

```

    "env_names": "{ 'conv_layers' }",
    "type": "integer"
  },
  "conv_filters": {
    "title": "Conv Filters",
    "default": [
      64,
      64,
      64,
      64
    ],
    "env_names": "{ 'conv_filters' }",
    "type": "array",
    "items": {
      "type": "integer"
    }
  },
  "conv_filter_shapes": {
    "title": "Conv Filter Shapes",
    "default": [
      [
        3,
        3
      ],
      [
        3,
        3
      ],
      [
        3,
        3
      ],
      [
        3,
        3
      ]
    ],
    "env_names": "{ 'conv_filter_shapes' }",
    "type": "array",
    "items": {
      "type": "array",
      "minItems": 2,
      "maxItems": 2,
      "items": [
        {
          "type": "integer"
        },
        {
          "type": "integer"
        }
      ]
    }
  }
}

```

(continues on next page)

(continued from previous page)

```

    },
    "conv_strides": {
      "title": "Conv Strides",
      "default": [
        [
          1,
          1
        ],
        [
          2,
          2
        ],
        [
          1,
          1
        ],
        [
          1,
          1
        ]
      ],
      "env_names": "'conv_strides'",
      "type": "array",
      "items": {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "type": "integer"
          },
          {
            "type": "integer"
          }
        ]
      }
    },
    "dense_layers": {
      "title": "Dense Layers",
      "default": 1,
      "env_names": "'dense_layers'",
      "type": "integer"
    },
    "dense_neurons": {
      "title": "Dense Neurons",
      "default": [
        128
      ],
      "env_names": "'dense_neurons'",
      "type": "array",
      "items": {
        "type": "integer"
      }
    }
  }

```

(continues on next page)

(continued from previous page)

```

    }
  },
  "dense_dropouts": {
    "title": "Dense Dropouts",
    "default": [
      0.25
    ],
    "env_names": "'dense_dropouts'",
    "type": "array",
    "items": {
      "type": "number"
    }
  },
  "outlier_count": {
    "title": "Outlier Count",
    "default": 120,
    "env_names": "'outlier_count'",
    "type": "integer"
  },
  "outlier_max": {
    "title": "Outlier Max",
    "default": 4500,
    "env_names": "'outlier_max'",
    "type": "integer"
  },
  "outlier_min": {
    "title": "Outlier Min",
    "default": 3000,
    "env_names": "'outlier_min'",
    "type": "integer"
  },
  "init_pdb_file": {
    "title": "Init Pdb File",
    "default": ".",
    "env_names": "'init_pdb_file'",
    "type": "string",
    "format": "path"
  },
  "ref_pdb_file": {
    "title": "Ref Pdb File",
    "default": ".",
    "env_names": "'ref_pdb_file'",
    "type": "string",
    "format": "path"
  },
  "init_eps": {
    "title": "Init Eps",
    "default": 1.3,
    "env_names": "'init_eps'",
    "type": "number"
  },
  "init_min_samples": {

```

(continues on next page)

(continued from previous page)

```

    "title": "Init Min Samples",
    "default": 10,
    "env_names": "{ 'init_min_samples' }",
    "type": "integer"
  },
  "adios_xml_agg": {
    "title": "Adios Xml Agg",
    "default": "",
    "env_names": "{ 'adios_xml_agg' }",
    "type": "string",
    "format": "path"
  },
  "read_batch": {
    "title": "Read Batch",
    "default": 10000,
    "env_names": "{ 'read_batch' }",
    "type": "integer"
  },
  "project_gpu": {
    "title": "Project Gpu",
    "default": false,
    "env_names": "{ 'project_gpu' }",
    "type": "boolean"
  },
  "project_lastN": {
    "title": "Project Lastn",
    "default": 8000,
    "env_names": "{ 'project_lastn' }",
    "type": "integer"
  },
  "num_sim": {
    "title": "Num Sim",
    "default": 120,
    "env_names": "{ 'num_sim' }",
    "type": "integer"
  },
  "use_outliers": {
    "title": "Use Outliers",
    "default": true,
    "env_names": "{ 'use_outliers' }",
    "type": "boolean"
  },
  "use_random_outliers": {
    "title": "Use Random Outliers",
    "default": false,
    "env_names": "{ 'use_random_outliers' }",
    "type": "boolean"
  },
  "compute_rmsd": {
    "title": "Compute Rmsd",
    "default": true,
    "env_names": "{ 'compute_rmsd' }",

```

(continues on next page)

(continued from previous page)

```

    "type": "boolean"
  },
  "compute_zcentroid": {
    "title": "Compute Zcentroid",
    "default": false,
    "env_names": "'compute_zcentroid'",
    "type": "boolean"
  },
  "final_shape": {
    "title": "Final Shape",
    "default": [
      28,
      28,
      1
    ],
    "env_names": "'final_shape'",
    "type": "array",
    "items": {
      "type": "integer"
    }
  },
  "outlier_selection": {
    "title": "Outlier Selection",
    "default": "rmsd",
    "env_names": "'outlier_selection'",
    "type": "string"
  },
  "multi_ligand_table": {
    "title": "Multi Ligand Table",
    "default": ".",
    "env_names": "'multi_ligand_table'",
    "type": "string",
    "format": "path"
  },
  "model": {
    "title": "Model",
    "default": "cvae",
    "env_names": "'model'",
    "type": "string"
  },
  "num_points": {
    "title": "Num Points",
    "default": 539,
    "env_names": "'num_points'",
    "type": "integer"
  },
  "num_features": {
    "title": "Num Features",
    "default": 0,
    "env_names": "'num_features'",
    "type": "integer"
  }
}

```

(continues on next page)

(continued from previous page)

```
}  
}
```

Config

- **extra:** *str = allow*

Fields

- *adios_xml_agg* (*pathlib.Path*)
- *agg_dir* (*pathlib.Path*)
- *best_model* (*pathlib.Path*)
- *compute_rmsd* (*bool*)
- *compute_zcentroid* (*bool*)
- *conv_filter_shapes* (*List[Tuple[int, int]]*)
- *conv_filters* (*List[int]*)
- *conv_layers* (*int*)
- *conv_strides* (*List[Tuple[int, int]]*)
- *dense_dropouts* (*List[float]*)
- *dense_layers* (*int*)
- *dense_neurons* (*List[int]*)
- *final_shape* (*List[int]*)
- *init_eps* (*float*)
- *init_min_samples* (*int*)
- *init_pdb_file* (*pathlib.Path*)
- *lastN* (*int*)
- *latent_dim* (*int*)
- *min_step_increment* (*int*)
- *model* (*str*)
- *multi_ligand_table* (*pathlib.Path*)
- *num_agg* (*int*)
- *num_features* (*int*)
- *num_points* (*int*)
- *num_sim* (*int*)
- *outlier_count* (*int*)
- *outlier_max* (*int*)
- *outlier_min* (*int*)
- *outlier_selection* (*str*)
- *project_gpu* (*bool*)

- *project_lastN* (*int*)
- *read_batch* (*int*)
- *ref_pdb_file* (*pathlib.Path*)
- *timeout1* (*int*)
- *timeout2* (*int*)
- *use_outliers* (*bool*)
- *use_random_outliers* (*bool*)

```

field adios_xml_agg:  pathlib.Path = ''
field agg_dir:  pathlib.Path = PosixPath('.')
field best_model:  pathlib.Path = PosixPath('.')
field compute_rmsd:  bool = True
field compute_zcentroid:  bool = False
field conv_filter_shapes:  List[Tuple[int, int]] = [(3, 3), (3, 3), (3, 3), (3, 3)]
field conv_filters:  List[int] = [64, 64, 64, 64]
field conv_layers:  int = 4
field conv_strides:  List[Tuple[int, int]] = [(1, 1), (2, 2), (1, 1), (1, 1)]
field dense_dropouts:  List[float] = [0.25]
field dense_layers:  int = 1
field dense_neurons:  List[int] = [128]
field final_shape:  List[int] = [28, 28, 1]
field init_eps:  float = 1.3
field init_min_samples:  int = 10
field init_pdb_file:  pathlib.Path = PosixPath('.')
field lastN:  int = 8000
field latent_dim:  int = 10
field min_step_increment:  int = 500
field model:  str = 'cvae'
field multi_ligand_table:  pathlib.Path = PosixPath('.')
field num_agg:  int = 2
field num_features:  int = 0
    encoder_bias: bool = True encoder_relu_slope: float = 0.0 encoder_filters: List[int] = [64, 128, 256, 256,
    512] encoder_kernels: List[int] = [5, 5, 3, 1, 1] decoder_bias: bool = True decoder_relu_slope: float =
    0.0 decoder_affine_widths: List[int] = [64, 128, 512, 1024] discriminator_bias: bool = True discrimina-
    tor_relu_slope: float = 0.0 discriminator_affine_widths: List[int] = [512, 128, 64]

```

```
field num_points:  int = 539
field num_sim:     int = 120
field outlier_count: int = 120
field outlier_max:  int = 4500
field outlier_min:  int = 3000
field outlier_selection: str = 'rmsd'
field project_gpu:  bool = False
field project_lastN: int = 8000
field read_batch:   int = 10000
field ref_pdb_file: pathlib.Path = PosixPath('.')
field timeout1:     int = 30
field timeout2:     int = 10
field use_outliers:  bool = True
field use_random_outliers: bool = False
```

deepdrivemd.aggregation

Modules

deepdrivemd.aggregation.basic

deepdrivemd.aggregation.stream

deepdrivemd.aggregation.basic

Modules

deepdrivemd.aggregation.basic.aggregate

deepdrivemd.aggregation.basic.config

deepdrivemd.aggregation.basic.aggregate**Functions**

`concatenate_last_n_h5(cfg)`

`deepdrivemd.aggregation.basic.aggregate.concatenate_last_n_h5(cfg: deepdrivemd.aggregation.basic.config.BasicAggegation)`
 → None

deepdrivemd.aggregation.basic.config

pydantic settings `deepdrivemd.aggregation.basic.config.BasicAggegation`

```
{
  "title": "BasicAggegation",
  "description": "Base class for specific aggregation configs to inherit.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'experiment_directory'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    "task_idx": {
      "title": "Task Idx",
      "default": 0,
      "env_names": "'task_idx'",
      "type": "integer"
    },
    "output_path": {
      "title": "Output Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'output_path'",
      "type": "string",
      "format": "path"
    },
    "node_local_path": {
      "title": "Node Local Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'node_local_path'",

```

(continues on next page)

(continued from previous page)

```

        "type": "string",
        "format": "path"
    },
    "rmsd": {
        "title": "Rmsd",
        "default": true,
        "env_names": "'rmsd'",
        "type": "boolean"
    },
    "fnc": {
        "title": "Fnc",
        "default": false,
        "env_names": "'fnc'",
        "type": "boolean"
    },
    "contact_map": {
        "title": "Contact Map",
        "default": false,
        "env_names": "'contact_map'",
        "type": "boolean"
    },
    "point_cloud": {
        "title": "Point Cloud",
        "default": true,
        "env_names": "'point_cloud'",
        "type": "boolean"
    },
    "verbose": {
        "title": "Verbose",
        "default": true,
        "env_names": "'verbose'",
        "type": "boolean"
    },
    "last_n_h5_files": {
        "title": "Last N H5 Files",
        "env_names": "'last_n_h5_files'",
        "type": "integer"
    }
}

```

Config

- **extra:** *str = allow*

Fields

- *contact_map* (bool)
- *fnc* (bool)
- *last_n_h5_files* (Optional[int])
- *point_cloud* (bool)
- *rmsd* (bool)

- *verbose* (*bool*)

field `contact_map`: `bool = False`

field `fnc`: `bool = False`

field `last_n_h5_files`: `Optional[int] = PydanticUndefined`

field `point_cloud`: `bool = True`

field `rmsd`: `bool = True`

field `verbose`: `bool = True`

deepdrivemd.aggregation.stream

Modules

deepdrivemd.aggregation.stream.aggregator

deepdrivemd.aggregation.stream.config

deepdrivemd.aggregation.stream.aggregator

Functions

<i>aggregate</i> (<i>cfg</i> , <i>connections</i> , ...)	Read adios streams from a subset of simulations handled by this aggregator and write them to adios file to be used by machine learning and outlier search.
<i>connect_to_input</i> (<i>cfg</i> , <i>bpfiles</i>)	Open adios streams for reading.
<i>find_input</i> (<i>cfg</i>)	Find adios streams to which simulations write.

`deepdrivemd.aggregation.stream.aggregator.aggregate`(*cfg*: `deepdrivemd.aggregation.stream.config.StreamAggregation`, *connections*: `Dict[int, Tuple[adios2.adios2.ADIOS, adios2.adios2.IO, adios2.adios2.Engine]]`, *aggregator_stream*: `adios2.adios2.Engine`, *aggregator_stream_4ml*: `adios2.adios2.Engine`)

Read adios streams from a subset of simulations handled by this aggregator and write them to adios file to be used by machine learning and outlier search.

Parameters

- **cfg** (*StreamAggregation*)
- **connections** (`Dict[int, Tuple[adios2.adios2.ADIOS, adios2.adios2.IO, adios2.adios2.Engine]]`) – key - task id, value - a tuple of adios objects
- **aggregator_stream** (*adios2.adios2.Engine*) – an adios stream of aggregated file to write to.

Note: If we do not need to save the data for postproduction, we can get rid of the aggregated adios file and replace it by SST stream.

`deepdrivemd.aggregation.stream.aggregator.connect_to_input`(*cfg*: `deepdrivemd.aggregation.stream.config.StreamAggregation`, *bpfiles*: `List[pathlib.Path]`) → `Dict[int, Tuple[adios2.adios2.ADIOS, adios2.adios2.IO, adios2.adios2.Engine]]`

Open adios streams for reading.

Parameters

- *cfg* (`StreamAggregation`)
- *bpfiles* (`List[Path]`)

Returns `Dict[int, Tuple[adios2.adios2.ADIOS, adios2.adios2.IO, adios2.adios2.Engine]]` – key - simulation task id, value - tuple of the corresponding adios objects.

`deepdrivemd.aggregation.stream.aggregator.find_input`(*cfg*: `deepdrivemd.aggregation.stream.config.StreamAggregation`) → `List[str]`

Find adios streams to which simulations write.

Parameters *cfg* (`StreamAggregation`)

Returns `List[str]` – a list of sst files associated with simulations

deepdrivemd.aggregation.stream.config

pydantic settings `deepdrivemd.aggregation.stream.config.StreamAggregation`

```
{
  "title": "StreamAggregation",
  "description": "Base class for specific aggregation configs to inherit.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'experiment_directory'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    "task_idx": {
      "title": "Task Idx",
```

(continues on next page)

(continued from previous page)

```

    "default": 0,
    "env_names": "'task_idx'",
    "type": "integer"
  },
  "output_path": {
    "title": "Output Path",
    "default": "set_by_deepdrivemd",
    "env_names": "'output_path'",
    "type": "string",
    "format": "path"
  },
  "node_local_path": {
    "title": "Node Local Path",
    "default": "set_by_deepdrivemd",
    "env_names": "'node_local_path'",
    "type": "string",
    "format": "path"
  },
  "n_sim": {
    "title": "N Sim",
    "default": 12,
    "env_names": "'n_sim'",
    "type": "integer"
  },
  "sleeptime_bpfiles": {
    "title": "Sleeptime Bpfiles",
    "default": 30,
    "env_names": "'sleeptime_bpfiles'",
    "type": "integer"
  },
  "num_tasks": {
    "title": "Num Tasks",
    "default": 2,
    "env_names": "'num_tasks'",
    "type": "integer"
  },
  "adios_xml_agg": {
    "title": "Adios Xml Agg",
    "default": ".",
    "env_names": "'adios_xml_agg'",
    "type": "string",
    "format": "path"
  },
  "compute_rmsd": {
    "title": "Compute Rmsd",
    "default": true,
    "env_names": "'compute_rmsd'",
    "type": "boolean"
  },
  "compute_zcentroid": {
    "title": "Compute Zcentroid",
    "default": false,

```

(continues on next page)

(continued from previous page)

```

        "env_names": "{ 'compute_zcentroid' }",
        "type": "boolean"
    },
    "multi_ligand_table": {
        "title": "Multi Ligand Table",
        "default": ".",
        "env_names": "{ 'multi_ligand_table' }",
        "type": "string",
        "format": "path"
    },
    "model": {
        "title": "Model",
        "default": "cvae",
        "env_names": "{ 'model' }",
        "type": "string"
    }
}

```

Config

- **extra:** *str = allow*

Fields

- *adios_xml_agg* (*pathlib.Path*)
- *compute_rmsd* (*bool*)
- *compute_zcentroid* (*bool*)
- *model* (*str*)
- *multi_ligand_table* (*pathlib.Path*)
- *n_sim* (*int*)
- *num_tasks* (*int*)
- *sleeptime_bpfiles* (*int*)

```
field adios_xml_agg:  pathlib.Path = PosixPath('.')

```

```
field compute_rmsd:  bool = True

```

```
field compute_zcentroid:  bool = False

```

```
field model:  str = 'cvae'

```

```
field multi_ligand_table:  pathlib.Path = PosixPath('.')

```

```
field n_sim:  int = 12

```

```
field num_tasks:  int = 2

```

```
field sleeptime_bpfiles:  int = 30

```

deepdrivemd.config

Schema of the YAML experiment file

Functions

`generate_sample_config()`

pydantic settings `deepdrivemd.config.AgentStageConfig`

Global agent configuration (written one per experiment).

```

{
  "title": "AgentStageConfig",
  "description": "Global agent configuration (written one per experiment).",
  "type": "object",
  "properties": {
    "pre_exec": {
      "title": "Pre Exec",
      "default": [],
      "env_names": "'pre_exec'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "executable": {
      "title": "Executable",
      "default": "",
      "env_names": "'executable'",
      "type": "string"
    },
    "arguments": {
      "title": "Arguments",
      "default": [],
      "env_names": "'arguments'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "cpu_reqs": {
      "title": "Cpu Reqs",
      "default": {
        "processes": 1,
        "process_type": null,
        "threads_per_process": 1,
        "thread_type": null
      },
      "env_names": "'cpu_reqs'",
      "allOf": [

```

(continues on next page)

(continued from previous page)

```

        {
            "$ref": "#/definitions/CPUReqs"
        }
    ]
},
"gpu_reqs": {
    "title": "Gpu Reqs",
    "default": {
        "processes": 0,
        "process_type": null,
        "threads_per_process": 0,
        "thread_type": null
    },
    "env_names": "'gpu_reqs'",
    "allOf": [
        {
            "$ref": "#/definitions/GPUReqs"
        }
    ]
},
"task_config": {
    "title": "Task Config",
    "env_names": "'task_config'",
    "allOf": [
        {
            "$ref": "#/definitions/AgentTaskConfig"
        }
    ]
}
},
"required": [
    "task_config"
],
"additionalProperties": false,
"definitions": {
    "CPUReqs": {
        "title": "CPUReqs",
        "description": "radical.entk task.cpu_reqs parameters.",
        "type": "object",
        "properties": {
            "processes": {
                "title": "Processes",
                "default": 1,
                "env_names": "'processes'",
                "type": "integer"
            },
            "process_type": {
                "title": "Process Type",
                "env_names": "'process_type'",
                "type": "string"
            },
            "threads_per_process": {

```

(continues on next page)

(continued from previous page)

```

        "title": "Threads Per Process",
        "default": 1,
        "env_names": "{ 'threads_per_process' }",
        "type": "integer"
    },
    "thread_type": {
        "title": "Thread Type",
        "env_names": "{ 'thread_type' }",
        "type": "string"
    }
},
"additionalProperties": false
},
"GPUReqs": {
    "title": "GPUReqs",
    "description": "radical.entk task.gpu_reqs parameters.",
    "type": "object",
    "properties": {
        "processes": {
            "title": "Processes",
            "default": 0,
            "env_names": "{ 'processes' }",
            "type": "integer"
        },
        "process_type": {
            "title": "Process Type",
            "env_names": "{ 'process_type' }",
            "type": "string"
        },
        "threads_per_process": {
            "title": "Threads Per Process",
            "default": 0,
            "env_names": "{ 'threads_per_process' }",
            "type": "integer"
        },
        "thread_type": {
            "title": "Thread Type",
            "env_names": "{ 'thread_type' }",
            "type": "string"
        }
    },
    "additionalProperties": false
},
"AgentTaskConfig": {
    "title": "AgentTaskConfig",
    "description": "Base class for specific agent configs to inherit.",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",
            "default": "set_by_deepdrivemd",
            "env_names": "{ 'experiment_directory' }",

```

(continues on next page)

(continued from previous page)

```

        "type": "string",
        "format": "path"
    },
    "stage_idx": {
        "title": "Stage Idx",
        "default": 0,
        "env_names": "'stage_idx'",
        "type": "integer"
    },
    "task_idx": {
        "title": "Task Idx",
        "default": 0,
        "env_names": "'task_idx'",
        "type": "integer"
    },
    "output_path": {
        "title": "Output Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'output_path'",
        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'node_local_path'",
        "type": "string",
        "format": "path"
    }
}
}
}
}

```

Fields

- *task_config* (*deepdrivemd.config.AgentTaskConfig*)

field **task_config**: *deepdrivemd.config.AgentTaskConfig* [Required]

pydantic settings *deepdrivemd.config.AgentTaskConfig*

Base class for specific agent configs to inherit.

```

{
    "title": "AgentTaskConfig",
    "description": "Base class for specific agent configs to inherit.",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",
            "default": "set_by_deepdrivemd",
            "env_names": "'experiment_directory'",

```

(continues on next page)

(continued from previous page)

```

        "type": "string",
        "format": "path"
    },
    "stage_idx": {
        "title": "Stage Idx",
        "default": 0,
        "env_names": "'stage_idx'",
        "type": "integer"
    },
    "task_idx": {
        "title": "Task Idx",
        "default": 0,
        "env_names": "'task_idx'",
        "type": "integer"
    },
    "output_path": {
        "title": "Output Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'output_path'",
        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'node_local_path'",
        "type": "string",
        "format": "path"
    }
}

```

Config

- **extra:** *str = allow*

Fields

- *experiment_directory* (*pathlib.Path*)
- *node_local_path* (*Optional[pathlib.Path]*)
- *output_path* (*pathlib.Path*)
- *stage_idx* (*int*)
- *task_idx* (*int*)

```

field experiment_directory: pathlib.Path = PosixPath('set_by_deepdrivemd')
field node_local_path: Optional[pathlib.Path] = PosixPath('set_by_deepdrivemd')
field output_path: pathlib.Path = PosixPath('set_by_deepdrivemd')
field stage_idx: int = 0

```

```
field task_idx: int = 0
```

pydantic settings `deepdrivemd.config.AggregationStageConfig`

Global aggregation configuration (written one per experiment).

```
{
  "title": "AggregationStageConfig",
  "description": "Global aggregation configuration (written one per experiment).",
  "type": "object",
  "properties": {
    "pre_exec": {
      "title": "Pre Exec",
      "default": [],
      "env_names": "'pre_exec'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "executable": {
      "title": "Executable",
      "default": "",
      "env_names": "'executable'",
      "type": "string"
    },
    "arguments": {
      "title": "Arguments",
      "default": [],
      "env_names": "'arguments'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "cpu_reqs": {
      "title": "Cpu Reqs",
      "default": {
        "processes": 1,
        "process_type": null,
        "threads_per_process": 1,
        "thread_type": null
      },
      "env_names": "'cpu_reqs'",
      "allOf": [
        {
          "$ref": "#/definitions/CPUReqs"
        }
      ]
    },
    "gpu_reqs": {
      "title": "Gpu Reqs",
      "default": {
        "processes": 0,
```

(continues on next page)

(continued from previous page)

```

        "process_type": null,
        "threads_per_process": 0,
        "thread_type": null
    },
    "env_names": "'gpu_reqs'",
    "allOf": [
        {
            "$ref": "#/definitions/GPUReqs"
        }
    ]
},
"skip_aggregation": {
    "title": "Skip Aggregation",
    "default": false,
    "env_names": "'skip_aggregation'",
    "type": "boolean"
},
"task_config": {
    "title": "Task Config",
    "env_names": "'task_config'",
    "allOf": [
        {
            "$ref": "#/definitions/AggregationTaskConfig"
        }
    ]
}
},
"required": [
    "task_config"
],
"additionalProperties": false,
"definitions": {
    "CPUReqs": {
        "title": "CPUReqs",
        "description": "radical.entk task.cpu_reqs parameters.",
        "type": "object",
        "properties": {
            "processes": {
                "title": "Processes",
                "default": 1,
                "env_names": "'processes'",
                "type": "integer"
            },
            "process_type": {
                "title": "Process Type",
                "env_names": "'process_type'",
                "type": "string"
            },
            "threads_per_process": {
                "title": "Threads Per Process",
                "default": 1,
                "env_names": "'threads_per_process'",

```

(continues on next page)

(continued from previous page)

```

        "type": "integer"
    },
    "thread_type": {
        "title": "Thread Type",
        "env_names": "{ 'thread_type' }",
        "type": "string"
    }
},
"additionalProperties": false
},
"GPUReqs": {
    "title": "GPUReqs",
    "description": "radical.entk task.gpu_reqs parameters.",
    "type": "object",
    "properties": {
        "processes": {
            "title": "Processes",
            "default": 0,
            "env_names": "{ 'processes' }",
            "type": "integer"
        },
        "process_type": {
            "title": "Process Type",
            "env_names": "{ 'process_type' }",
            "type": "string"
        },
        "threads_per_process": {
            "title": "Threads Per Process",
            "default": 0,
            "env_names": "{ 'threads_per_process' }",
            "type": "integer"
        },
        "thread_type": {
            "title": "Thread Type",
            "env_names": "{ 'thread_type' }",
            "type": "string"
        }
    },
    "additionalProperties": false
},
"AggregationTaskConfig": {
    "title": "AggregationTaskConfig",
    "description": "Base class for specific aggregation configs to inherit.",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",
            "default": "set_by_deepdrivemd",
            "env_names": "{ 'experiment_directory' }",
            "type": "string",
            "format": "path"
        }
    },

```

(continues on next page)

(continued from previous page)

```

    "stage_idx": {
        "title": "Stage Idx",
        "default": 0,
        "env_names": "'stage_idx'",
        "type": "integer"
    },
    "task_idx": {
        "title": "Task Idx",
        "default": 0,
        "env_names": "'task_idx'",
        "type": "integer"
    },
    "output_path": {
        "title": "Output Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'output_path'",
        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'node_local_path'",
        "type": "string",
        "format": "path"
    }
}
}
}
}
}

```

Fields

- *skip_aggregation* (bool)
- *task_config* (deepdrivemd.config.AggregationTaskConfig)

field skip_aggregation: bool = False

field task_config: deepdrivemd.config.AggregationTaskConfig [Required]

pydantic settings deepdrivemd.config.AggregationTaskConfig

Base class for specific aggregation configs to inherit.

```

{
    "title": "AggregationTaskConfig",
    "description": "Base class for specific aggregation configs to inherit.",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",
            "default": "set_by_deepdrivemd",
            "env_names": "'experiment_directory'",

```

(continues on next page)

(continued from previous page)

```

        "type": "string",
        "format": "path"
    },
    "stage_idx": {
        "title": "Stage Idx",
        "default": 0,
        "env_names": "'stage_idx'",
        "type": "integer"
    },
    "task_idx": {
        "title": "Task Idx",
        "default": 0,
        "env_names": "'task_idx'",
        "type": "integer"
    },
    "output_path": {
        "title": "Output Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'output_path'",
        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'node_local_path'",
        "type": "string",
        "format": "path"
    }
}

```

Config

- **extra:** *str = allow*

Fields

- *experiment_directory* (*pathlib.Path*)
- *node_local_path* (*Optional[pathlib.Path]*)
- *output_path* (*pathlib.Path*)
- *stage_idx* (*int*)
- *task_idx* (*int*)

```
field experiment_directory: pathlib.Path = PosixPath('set_by_deepdrivemd')
```

```
field node_local_path: Optional[pathlib.Path] = PosixPath('set_by_deepdrivemd')
```

```
field output_path: pathlib.Path = PosixPath('set_by_deepdrivemd')
```

```
field stage_idx: int = 0
```

```
field task_idx: int = 0
```

pydantic settings deepdrivemd.config.BaseSettings

```
{
  "title": "BaseSettings",
  "description": "Base class for settings, allowing values to be overridden by
  ↪environment variables.\n\nThis is useful in production for secrets you do not
  ↪wish to save in code, it plays nicely with docker(-compose),\nHeroku and any 12
  ↪factor app design.",
  "type": "object",
  "properties": {},
  "additionalProperties": false
}
```

```
dump_yaml(cfg_path: Union[str, pathlib.Path]) → None
```

```
classmethod from_yaml(filename: Union[str, pathlib.Path]) → deepdrivemd.config._T
```

pydantic settings deepdrivemd.config.BaseStageConfig

Base configuration for all StageConfig objects.

```
{
  "title": "BaseStageConfig",
  "description": "Base configuration for all StageConfig objects.",
  "type": "object",
  "properties": {
    "pre_exec": {
      "title": "Pre Exec",
      "default": [],
      "env_names": "'pre_exec'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "executable": {
      "title": "Executable",
      "default": "",
      "env_names": "'executable'",
      "type": "string"
    },
    "arguments": {
      "title": "Arguments",
      "default": [],
      "env_names": "'arguments'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "cpu_reqs": {
      "title": "Cpu Reqs",
      "default": {
```

(continues on next page)

(continued from previous page)

```

        "processes": 1,
        "process_type": null,
        "threads_per_process": 1,
        "thread_type": null
    },
    "env_names": "'cpu_reqs'",
    "allOf": [
        {
            "$ref": "#/definitions/CPUReqs"
        }
    ]
},
"gpu_reqs": {
    "title": "Gpu Reqs",
    "default": {
        "processes": 0,
        "process_type": null,
        "threads_per_process": 0,
        "thread_type": null
    },
    "env_names": "'gpu_reqs'",
    "allOf": [
        {
            "$ref": "#/definitions/GPUReqs"
        }
    ]
}
},
"additionalProperties": false,
"definitions": {
    "CPUReqs": {
        "title": "CPUReqs",
        "description": "radical.entk task.cpu_reqs parameters.",
        "type": "object",
        "properties": {
            "processes": {
                "title": "Processes",
                "default": 1,
                "env_names": "'processes'",
                "type": "integer"
            },
            "process_type": {
                "title": "Process Type",
                "env_names": "'process_type'",
                "type": "string"
            },
            "threads_per_process": {
                "title": "Threads Per Process",
                "default": 1,
                "env_names": "'threads_per_process'",
                "type": "integer"
            }
        }
    },

```

(continues on next page)

(continued from previous page)

```

        "thread_type": {
            "title": "Thread Type",
            "env_names": "{ 'thread_type' }",
            "type": "string"
        }
    },
    "additionalProperties": false
},
"GPUReqs": {
    "title": "GPUReqs",
    "description": "radical.entk task.gpu_reqs parameters.",
    "type": "object",
    "properties": {
        "processes": {
            "title": "Processes",
            "default": 0,
            "env_names": "{ 'processes' }",
            "type": "integer"
        },
        "process_type": {
            "title": "Process Type",
            "env_names": "{ 'process_type' }",
            "type": "string"
        },
        "threads_per_process": {
            "title": "Threads Per Process",
            "default": 0,
            "env_names": "{ 'threads_per_process' }",
            "type": "integer"
        },
        "thread_type": {
            "title": "Thread Type",
            "env_names": "{ 'thread_type' }",
            "type": "string"
        }
    },
    "additionalProperties": false
}
}
}

```

Fields

- *arguments* (*List[str]*)
- *cpu_reqs* (*deepdrivemd.config.CPUReqs*)
- *executable* (*str*)
- *gpu_reqs* (*deepdrivemd.config.GPUReqs*)
- *pre_exec* (*List[str]*)

field arguments: `List[str] = []`

```

field cpu_reqs: deepdrivemd.config.CPUReqs = CPUReqs(processes=1,
process_type=None, threads_per_process=1, thread_type=None)

field executable: str = ''

field gpu_reqs: deepdrivemd.config.GPUReqs = GPUReqs(processes=0,
process_type=None, threads_per_process=0, thread_type=None)

field pre_exec: List[str] = []

```

pydantic settings *deepdrivemd.config.BaseTaskConfig*

Base configuration for all TaskConfig objects.

```

{
  "title": "BaseTaskConfig",
  "description": "Base configuration for all TaskConfig objects.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'experiment_directory'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    "task_idx": {
      "title": "Task Idx",
      "default": 0,
      "env_names": "'task_idx'",
      "type": "integer"
    },
    "output_path": {
      "title": "Output Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'output_path'",
      "type": "string",
      "format": "path"
    },
    "node_local_path": {
      "title": "Node Local Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'node_local_path'",
      "type": "string",
      "format": "path"
    }
  }
}

```

Config

- **extra:** *str* = *allow*

Fields

- *experiment_directory* (*pathlib.Path*)
- *node_local_path* (*Optional[pathlib.Path]*)
- *output_path* (*pathlib.Path*)
- *stage_idx* (*int*)
- *task_idx* (*int*)

field *experiment_directory*: *pathlib.Path* = *PosixPath('set_by_deepdrivemd')*

field *node_local_path*: *Optional[pathlib.Path]* = *PosixPath('set_by_deepdrivemd')*

field *output_path*: *pathlib.Path* = *PosixPath('set_by_deepdrivemd')*

field *stage_idx*: *int* = 0

field *task_idx*: *int* = 0

pydantic settings *deepdrivemd.config.CPUReqs*

radical.entk task.cpu_reqs parameters.

```
{
  "title": "CPUReqs",
  "description": "radical.entk task.cpu_reqs parameters.",
  "type": "object",
  "properties": {
    "processes": {
      "title": "Processes",
      "default": 1,
      "env_names": "'processes'",
      "type": "integer"
    },
    "process_type": {
      "title": "Process Type",
      "env_names": "'process_type'",
      "type": "string"
    },
    "threads_per_process": {
      "title": "Threads Per Process",
      "default": 1,
      "env_names": "'threads_per_process'",
      "type": "integer"
    },
    "thread_type": {
      "title": "Thread Type",
      "env_names": "'thread_type'",
      "type": "string"
    }
  },
  "additionalProperties": false
}
```

Fields

- *process_type* (Optional[str])
- *processes* (int)
- *thread_type* (Optional[str])
- *threads_per_process* (int)

Validators

- *process_type_check* » *process_type*
- *thread_type_check* » *thread_type*

field *process_type*: Optional[str] = PydanticUndefined

Validated by

- *process_type_check*

field *processes*: int = 1

field *thread_type*: Optional[str] = PydanticUndefined

Validated by

- *thread_type_check*

field *threads_per_process*: int = 1

validator *process_type_check* » *process_type*

validator *thread_type_check* » *thread_type*

pydantic settings deepdrivemd.config.ExperimentConfig

Main configuration.

```
{
  "title": "ExperimentConfig",
  "description": "Main configuration.",
  "type": "object",
  "properties": {
    "title": {
      "title": "Title",
      "env_names": "'title'",
      "type": "string"
    },
    "resource": {
      "title": "Resource",
      "env_names": "'resource'",
      "type": "string"
    },
    "queue": {
      "title": "Queue",
      "env_names": "'queue'",
      "type": "string"
    },
    "schema_": {
```

(continues on next page)

(continued from previous page)

```

    "title": "Schema ",
    "env_names": "{ 'schema_' }",
    "type": "string"
  },
  "project": {
    "title": "Project",
    "env_names": "{ 'project' }",
    "type": "string"
  },
  "walltime_min": {
    "title": "Walltime Min",
    "env_names": "{ 'walltime_min' }",
    "type": "integer"
  },
  "max_iteration": {
    "title": "Max Iteration",
    "env_names": "{ 'max_iteration' }",
    "type": "integer"
  },
  "cpus_per_node": {
    "title": "Cpus Per Node",
    "env_names": "{ 'cpus_per_node' }",
    "type": "integer"
  },
  "gpus_per_node": {
    "title": "Gpus Per Node",
    "env_names": "{ 'gpus_per_node' }",
    "type": "integer"
  },
  "hardware_threads_per_cpu": {
    "title": "Hardware Threads Per Cpu",
    "env_names": "{ 'hardware_threads_per_cpu' }",
    "type": "integer"
  },
  "experiment_directory": {
    "title": "Experiment Directory",
    "env_names": "{ 'experiment_directory' }",
    "type": "string",
    "format": "path"
  },
  "node_local_path": {
    "title": "Node Local Path",
    "env_names": "{ 'node_local_path' }",
    "type": "string",
    "format": "path"
  },
  "molecular_dynamics_stage": {
    "title": "Molecular Dynamics Stage",
    "env_names": "{ 'molecular_dynamics_stage' }",
    "allOf": [
      {
        "$ref": "#/definitions/MolecularDynamicsStageConfig"
      }
    ]
  }
}

```

(continues on next page)

(continued from previous page)

```

    }
  ]
},
"aggregation_stage": {
  "title": "Aggregation Stage",
  "env_names": "'aggregation_stage'",
  "allOf": [
    {
      "$ref": "#/definitions/AggregationStageConfig"
    }
  ]
},
"machine_learning_stage": {
  "title": "Machine Learning Stage",
  "env_names": "'machine_learning_stage'",
  "allOf": [
    {
      "$ref": "#/definitions/MachineLearningStageConfig"
    }
  ]
},
"model_selection_stage": {
  "title": "Model Selection Stage",
  "env_names": "'model_selection_stage'",
  "allOf": [
    {
      "$ref": "#/definitions/ModelSelectionStageConfig"
    }
  ]
},
"agent_stage": {
  "title": "Agent Stage",
  "env_names": "'agent_stage'",
  "allOf": [
    {
      "$ref": "#/definitions/AgentStageConfig"
    }
  ]
}
},
"required": [
  "title",
  "resource",
  "queue",
  "schema_",
  "project",
  "walltime_min",
  "max_iteration",
  "cpus_per_node",
  "gpus_per_node",
  "hardware_threads_per_cpu",
  "experiment_directory",

```

(continues on next page)

(continued from previous page)

```

    "molecular_dynamics_stage",
    "aggregation_stage",
    "machine_learning_stage",
    "model_selection_stage",
    "agent_stage"
  ],
  "additionalProperties": false,
  "definitions": {
    "CPUReqs": {
      "title": "CPUReqs",
      "description": "radical.entk task.cpu_reqs parameters.",
      "type": "object",
      "properties": {
        "processes": {
          "title": "Processes",
          "default": 1,
          "env_names": "{ 'processes' }",
          "type": "integer"
        },
        "process_type": {
          "title": "Process Type",
          "env_names": "{ 'process_type' }",
          "type": "string"
        },
        "threads_per_process": {
          "title": "Threads Per Process",
          "default": 1,
          "env_names": "{ 'threads_per_process' }",
          "type": "integer"
        },
        "thread_type": {
          "title": "Thread Type",
          "env_names": "{ 'thread_type' }",
          "type": "string"
        }
      }
    },
    "additionalProperties": false
  },
  "GPUReqs": {
    "title": "GPUReqs",
    "description": "radical.entk task.gpu_reqs parameters.",
    "type": "object",
    "properties": {
      "processes": {
        "title": "Processes",
        "default": 0,
        "env_names": "{ 'processes' }",
        "type": "integer"
      },
      "process_type": {
        "title": "Process Type",
        "env_names": "{ 'process_type' }",

```

(continues on next page)

(continued from previous page)

```

        "type": "string"
    },
    "threads_per_process": {
        "title": "Threads Per Process",
        "default": 0,
        "env_names": "{ 'threads_per_process' }",
        "type": "integer"
    },
    "thread_type": {
        "title": "Thread Type",
        "env_names": "{ 'thread_type' }",
        "type": "string"
    }
},
"additionalProperties": false
},
"MolecularDynamicsTaskConfig": {
    "title": "MolecularDynamicsTaskConfig",
    "description": "Auto-generates configuration file for MD tasks.",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",
            "default": "set_by_deepdrivemd",
            "env_names": "{ 'experiment_directory' }",
            "type": "string",
            "format": "path"
        },
        "stage_idx": {
            "title": "Stage Idx",
            "default": 0,
            "env_names": "{ 'stage_idx' }",
            "type": "integer"
        },
        "task_idx": {
            "title": "Task Idx",
            "default": 0,
            "env_names": "{ 'task_idx' }",
            "type": "integer"
        },
        "output_path": {
            "title": "Output Path",
            "default": "set_by_deepdrivemd",
            "env_names": "{ 'output_path' }",
            "type": "string",
            "format": "path"
        },
        "node_local_path": {
            "title": "Node Local Path",
            "default": "set_by_deepdrivemd",
            "env_names": "{ 'node_local_path' }",
            "type": "string",

```

(continues on next page)

(continued from previous page)

```

        "format": "path"
    },
    "pdb_file": {
        "title": "Pdb File",
        "default": "set_by_deepdrivemd",
        "env_names": "{ 'pdb_file' }",
        "type": "string",
        "format": "path"
    },
    "initial_pdb_dir": {
        "title": "Initial Pdb Dir",
        "env_names": "{ 'initial_pdb_dir' }",
        "type": "string",
        "format": "path"
    }
},
"required": [
    "initial_pdb_dir"
]
},
"MolecularDynamicsStageConfig": {
    "title": "MolecularDynamicsStageConfig",
    "description": "Global MD configuration (written one per experiment).",
    "type": "object",
    "properties": {
        "pre_exec": {
            "title": "Pre Exec",
            "default": [],
            "env_names": "{ 'pre_exec' }",
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "executable": {
            "title": "Executable",
            "default": "",
            "env_names": "{ 'executable' }",
            "type": "string"
        },
        "arguments": {
            "title": "Arguments",
            "default": [],
            "env_names": "{ 'arguments' }",
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "cpu_reqs": {
            "title": "Cpu Reqs",
            "default": {

```

(continues on next page)

(continued from previous page)

```

        "processes": 1,
        "process_type": null,
        "threads_per_process": 1,
        "thread_type": null
    },
    "env_names": "'cpu_reqs'",
    "allOf": [
        {
            "$ref": "#/definitions/CPUReqs"
        }
    ]
},
"gpu_reqs": {
    "title": "Gpu Reqs",
    "default": {
        "processes": 0,
        "process_type": null,
        "threads_per_process": 0,
        "thread_type": null
    },
    "env_names": "'gpu_reqs'",
    "allOf": [
        {
            "$ref": "#/definitions/GPUReqs"
        }
    ]
},
"num_tasks": {
    "title": "Num Tasks",
    "default": 1,
    "env_names": "'num_tasks'",
    "type": "integer"
},
"task_config": {
    "title": "Task Config",
    "env_names": "'task_config'",
    "allOf": [
        {
            "$ref": "#/definitions/MolecularDynamicsTaskConfig"
        }
    ]
}
},
"required": [
    "task_config"
],
"additionalProperties": false
},
"AggregationTaskConfig": {
    "title": "AggregationTaskConfig",
    "description": "Base class for specific aggregation configs to inherit.",
    "type": "object",

```

(continues on next page)

(continued from previous page)

```

"properties": {
  "experiment_directory": {
    "title": "Experiment Directory",
    "default": "set_by_deepdrivemd",
    "env_names": "'experiment_directory'",
    "type": "string",
    "format": "path"
  },
  "stage_idx": {
    "title": "Stage Idx",
    "default": 0,
    "env_names": "'stage_idx'",
    "type": "integer"
  },
  "task_idx": {
    "title": "Task Idx",
    "default": 0,
    "env_names": "'task_idx'",
    "type": "integer"
  },
  "output_path": {
    "title": "Output Path",
    "default": "set_by_deepdrivemd",
    "env_names": "'output_path'",
    "type": "string",
    "format": "path"
  },
  "node_local_path": {
    "title": "Node Local Path",
    "default": "set_by_deepdrivemd",
    "env_names": "'node_local_path'",
    "type": "string",
    "format": "path"
  }
},
"AggregationStageConfig": {
  "title": "AggregationStageConfig",
  "description": "Global aggregation configuration (written one per_
↪ experiment).",
  "type": "object",
  "properties": {
    "pre_exec": {
      "title": "Pre Exec",
      "default": [],
      "env_names": "'pre_exec'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "executable": {

```

(continues on next page)

(continued from previous page)

```

        "title": "Executable",
        "default": "",
        "env_names": "'executable'",
        "type": "string"
    },
    "arguments": {
        "title": "Arguments",
        "default": [],
        "env_names": "'arguments'",
        "type": "array",
        "items": {
            "type": "string"
        }
    },
    "cpu_reqs": {
        "title": "Cpu Reqs",
        "default": {
            "processes": 1,
            "process_type": null,
            "threads_per_process": 1,
            "thread_type": null
        },
        "env_names": "'cpu_reqs'",
        "allOf": [
            {
                "$ref": "#/definitions/CPUReqs"
            }
        ]
    },
    "gpu_reqs": {
        "title": "Gpu Reqs",
        "default": {
            "processes": 0,
            "process_type": null,
            "threads_per_process": 0,
            "thread_type": null
        },
        "env_names": "'gpu_reqs'",
        "allOf": [
            {
                "$ref": "#/definitions/GPUReqs"
            }
        ]
    },
    "skip_aggregation": {
        "title": "Skip Aggregation",
        "default": false,
        "env_names": "'skip_aggregation'",
        "type": "boolean"
    },
    "task_config": {
        "title": "Task Config",

```

(continues on next page)

(continued from previous page)

```

        "env_names": "{ 'task_config' }",
        "allOf": [
            {
                "$ref": "#/definitions/AggregationTaskConfig"
            }
        ]
    },
    "required": [
        "task_config"
    ],
    "additionalProperties": false
},
"MachineLearningTaskConfig": {
    "title": "MachineLearningTaskConfig",
    "description": "Base class for specific model configs to inherit.",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",
            "default": "set_by_deepdrivemd",
            "env_names": "{ 'experiment_directory' }",
            "type": "string",
            "format": "path"
        },
        "stage_idx": {
            "title": "Stage Idx",
            "default": 0,
            "env_names": "{ 'stage_idx' }",
            "type": "integer"
        },
        "task_idx": {
            "title": "Task Idx",
            "default": 0,
            "env_names": "{ 'task_idx' }",
            "type": "integer"
        },
        "output_path": {
            "title": "Output Path",
            "default": "set_by_deepdrivemd",
            "env_names": "{ 'output_path' }",
            "type": "string",
            "format": "path"
        },
        "node_local_path": {
            "title": "Node Local Path",
            "default": "set_by_deepdrivemd",
            "env_names": "{ 'node_local_path' }",
            "type": "string",
            "format": "path"
        },
        "model_tag": {

```

(continues on next page)

(continued from previous page)

```

        "title": "Model Tag",
        "default": "set_by_deepdrivemd",
        "env_names": "{ 'model_tag' }",
        "type": "string"
    },
    "init_weights_path": {
        "title": "Init Weights Path",
        "env_names": "{ 'init_weights_path' }",
        "type": "string",
        "format": "path"
    }
},
"MachineLearningStageConfig": {
    "title": "MachineLearningStageConfig",
    "description": "Global ML configuration (written one per experiment).",
    "type": "object",
    "properties": {
        "pre_exec": {
            "title": "Pre Exec",
            "default": [],
            "env_names": "{ 'pre_exec' }",
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "executable": {
            "title": "Executable",
            "default": "",
            "env_names": "{ 'executable' }",
            "type": "string"
        },
        "arguments": {
            "title": "Arguments",
            "default": [],
            "env_names": "{ 'arguments' }",
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "cpu_reqs": {
            "title": "Cpu Reqs",
            "default": {
                "processes": 1,
                "process_type": null,
                "threads_per_process": 1,
                "thread_type": null
            },
            "env_names": "{ 'cpu_reqs' }",
            "allOf": [

```

(continues on next page)

(continued from previous page)

```

        {
            "$ref": "#/definitions/CPUReqs"
        }
    ],
    },
    "gpu_reqs": {
        "title": "Gpu Reqs",
        "default": {
            "processes": 0,
            "process_type": null,
            "threads_per_process": 0,
            "thread_type": null
        },
        "env_names": "'gpu_reqs'",
        "allOf": [
            {
                "$ref": "#/definitions/GPUReqs"
            }
        ]
    },
    },
    "retrain_freq": {
        "title": "Retrain Freq",
        "default": 1,
        "env_names": "'retrain_freq'",
        "type": "integer"
    },
    },
    "task_config": {
        "title": "Task Config",
        "env_names": "'task_config'",
        "allOf": [
            {
                "$ref": "#/definitions/MachineLearningTaskConfig"
            }
        ]
    }
    ],
    },
    "required": [
        "task_config"
    ],
    "additionalProperties": false
},
"ModelSelectionTaskConfig": {
    "title": "ModelSelectionTaskConfig",
    "description": "Base class for specific model selection configs to inherit.
↪",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",
            "default": "set_by_deepdrivemd",
            "env_names": "'experiment_directory'",
            "type": "string",

```

(continues on next page)

(continued from previous page)

```

        "format": "path"
    },
    "stage_idx": {
        "title": "Stage Idx",
        "default": 0,
        "env_names": "'stage_idx'",
        "type": "integer"
    },
    "task_idx": {
        "title": "Task Idx",
        "default": 0,
        "env_names": "'task_idx'",
        "type": "integer"
    },
    "output_path": {
        "title": "Output Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'output_path'",
        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'node_local_path'",
        "type": "string",
        "format": "path"
    }
},
"ModelSelectionStageConfig": {
    "title": "ModelSelectionStageConfig",
    "description": "Global ML configuration (written one per experiment).",
    "type": "object",
    "properties": {
        "pre_exec": {
            "title": "Pre Exec",
            "default": [],
            "env_names": "'pre_exec'",
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "executable": {
            "title": "Executable",
            "default": "",
            "env_names": "'executable'",
            "type": "string"
        },
        "arguments": {
            "title": "Arguments",

```

(continues on next page)

(continued from previous page)

```

        "default": [],
        "env_names": "{ 'arguments' }",
        "type": "array",
        "items": {
            "type": "string"
        }
    },
    "cpu_reqs": {
        "title": "Cpu Reqs",
        "default": {
            "processes": 1,
            "process_type": null,
            "threads_per_process": 1,
            "thread_type": null
        },
        "env_names": "{ 'cpu_reqs' }",
        "allOf": [
            {
                "$ref": "#/definitions/CPUReqs"
            }
        ]
    },
    "gpu_reqs": {
        "title": "Gpu Reqs",
        "default": {
            "processes": 0,
            "process_type": null,
            "threads_per_process": 0,
            "thread_type": null
        },
        "env_names": "{ 'gpu_reqs' }",
        "allOf": [
            {
                "$ref": "#/definitions/GPUReqs"
            }
        ]
    },
    "task_config": {
        "title": "Task Config",
        "env_names": "{ 'task_config' }",
        "allOf": [
            {
                "$ref": "#/definitions/ModelSelectionTaskConfig"
            }
        ]
    }
},
"required": [
    "task_config"
],
"additionalProperties": false
},

```

(continues on next page)

(continued from previous page)

```

"AgentTaskConfig": {
  "title": "AgentTaskConfig",
  "description": "Base class for specific agent configs to inherit.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'experiment_directory'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    "task_idx": {
      "title": "Task Idx",
      "default": 0,
      "env_names": "'task_idx'",
      "type": "integer"
    },
    "output_path": {
      "title": "Output Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'output_path'",
      "type": "string",
      "format": "path"
    },
    "node_local_path": {
      "title": "Node Local Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'node_local_path'",
      "type": "string",
      "format": "path"
    }
  }
},
"AgentStageConfig": {
  "title": "AgentStageConfig",
  "description": "Global agent configuration (written one per experiment).",
  "type": "object",
  "properties": {
    "pre_exec": {
      "title": "Pre Exec",
      "default": [],
      "env_names": "'pre_exec'",
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  }
}

```

(continues on next page)

(continued from previous page)

```

    }
  },
  "executable": {
    "title": "Executable",
    "default": "",
    "env_names": "'executable'",
    "type": "string"
  },
  "arguments": {
    "title": "Arguments",
    "default": [],
    "env_names": "'arguments'",
    "type": "array",
    "items": {
      "type": "string"
    }
  },
  "cpu_reqs": {
    "title": "Cpu Reqs",
    "default": {
      "processes": 1,
      "process_type": null,
      "threads_per_process": 1,
      "thread_type": null
    },
    "env_names": "'cpu_reqs'",
    "allOf": [
      {
        "$ref": "#/definitions/CPUReqs"
      }
    ]
  },
  "gpu_reqs": {
    "title": "Gpu Reqs",
    "default": {
      "processes": 0,
      "process_type": null,
      "threads_per_process": 0,
      "thread_type": null
    },
    "env_names": "'gpu_reqs'",
    "allOf": [
      {
        "$ref": "#/definitions/GPUReqs"
      }
    ]
  },
  "task_config": {
    "title": "Task Config",
    "env_names": "'task_config'",
    "allOf": [
      {

```

(continues on next page)

(continued from previous page)

```

        "$ref": "#/definitions/AgentTaskConfig"
    }
    ]
  },
  "required": [
    "task_config"
  ],
  "additionalProperties": false
}
}
}

```

Fields

- *agent_stage* (*deepdrivemd.config.AgentStageConfig*)
- *aggregation_stage* (*deepdrivemd.config.AggregationStageConfig*)
- *cpus_per_node* (*int*)
- *experiment_directory* (*pathlib.Path*)
- *gpus_per_node* (*int*)
- *hardware_threads_per_cpu* (*int*)
- *machine_learning_stage* (*deepdrivemd.config.MachineLearningStageConfig*)
- *max_iteration* (*int*)
- *model_selection_stage* (*deepdrivemd.config.ModelSelectionStageConfig*)
- *molecular_dynamics_stage* (*deepdrivemd.config.MolecularDynamicsStageConfig*)
- *node_local_path* (*Optional[pathlib.Path]*)
- *project* (*str*)
- *queue* (*str*)
- *resource* (*str*)
- *schema_* (*str*)
- *title* (*str*)
- *walltime_min* (*int*)

Validators

- *experiment_directory_cannot_exist* » *experiment_directory*

field *agent_stage*: *deepdrivemd.config.AgentStageConfig* [Required]

field *aggregation_stage*: *deepdrivemd.config.AggregationStageConfig* [Required]

field *cpus_per_node*: *int* [Required]

```

field experiment_directory: pathlib.Path [Required]

    Validated by
        • experiment_directory_cannot_exist

field gpus_per_node: int [Required]

field hardware_threads_per_cpu: int [Required]

field machine_learning_stage: deepdrivemd.config.MachineLearningStageConfig
[Required]

field max_iteration: int [Required]

field model_selection_stage: deepdrivemd.config.ModelSelectionStageConfig
[Required]

field molecular_dynamics_stage: deepdrivemd.config.MolecularDynamicsStageConfig
[Required]

field node_local_path: Optional[pathlib.Path] = PydanticUndefined

field project: str [Required]

field queue: str [Required]

field resource: str [Required]

field schema_: str [Required]

field title: str [Required]

field walltime_min: int [Required]

validator experiment_directory_cannot_exist » experiment_directory

```

pydantic settings *deepdrivemd.config.GPUReqs*

radical.entk task.gpu_reqs parameters.

```

{
  "title": "GPUReqs",
  "description": "radical.entk task.gpu_reqs parameters.",
  "type": "object",
  "properties": {
    "processes": {
      "title": "Processes",
      "default": 0,
      "env_names": "'processes'",
      "type": "integer"
    },
    "process_type": {
      "title": "Process Type",
      "env_names": "'process_type'",
      "type": "string"
    },
    "threads_per_process": {
      "title": "Threads Per Process",

```

(continues on next page)

(continued from previous page)

```

        "default": 0,
        "env_names": {"threads_per_process"},
        "type": "integer"
    },
    "thread_type": {
        "title": "Thread Type",
        "env_names": {"thread_type"},
        "type": "string"
    }
},
"additionalProperties": false
}

```

Fields

- *process_type* (Optional[str])
- *processes* (int)
- *thread_type* (Optional[str])
- *threads_per_process* (int)

Validators

- *process_type_check* » *process_type*
- *thread_type_check* » *thread_type*

field process_type: Optional[str] = PydanticUndefined

Validated by

- *process_type_check*

field processes: int = 0

field thread_type: Optional[str] = PydanticUndefined

Validated by

- *thread_type_check*

field threads_per_process: int = 0

validator process_type_check » *process_type*

validator thread_type_check » *thread_type*

pydantic settings deepdrivemd.config.MachineLearningStageConfig

Global ML configuration (written one per experiment).

```

{
    "title": "MachineLearningStageConfig",
    "description": "Global ML configuration (written one per experiment).",
    "type": "object",
    "properties": {
        "pre_exec": {

```

(continues on next page)

(continued from previous page)

```

    "title": "Pre Exec",
    "default": [],
    "env_names": "'pre_exec'",
    "type": "array",
    "items": {
        "type": "string"
    }
},
"executable": {
    "title": "Executable",
    "default": "",
    "env_names": "'executable'",
    "type": "string"
},
"arguments": {
    "title": "Arguments",
    "default": [],
    "env_names": "'arguments'",
    "type": "array",
    "items": {
        "type": "string"
    }
},
"cpu_reqs": {
    "title": "Cpu Reqs",
    "default": {
        "processes": 1,
        "process_type": null,
        "threads_per_process": 1,
        "thread_type": null
    },
    "env_names": "'cpu_reqs'",
    "allOf": [
        {
            "$ref": "#/definitions/CPUReqs"
        }
    ]
},
"gpu_reqs": {
    "title": "Gpu Reqs",
    "default": {
        "processes": 0,
        "process_type": null,
        "threads_per_process": 0,
        "thread_type": null
    },
    "env_names": "'gpu_reqs'",
    "allOf": [
        {
            "$ref": "#/definitions/GPUReqs"
        }
    ]
}

```

(continues on next page)

(continued from previous page)

```

    },
    "retrain_freq": {
      "title": "Retrain Freq",
      "default": 1,
      "env_names": "'retrain_freq'",
      "type": "integer"
    },
    "task_config": {
      "title": "Task Config",
      "env_names": "'task_config'",
      "allOf": [
        {
          "$ref": "#/definitions/MachineLearningTaskConfig"
        }
      ]
    }
  },
  "required": [
    "task_config"
  ],
  "additionalProperties": false,
  "definitions": {
    "CPUReqs": {
      "title": "CPUReqs",
      "description": "radical.entk task.cpu_reqs parameters.",
      "type": "object",
      "properties": {
        "processes": {
          "title": "Processes",
          "default": 1,
          "env_names": "'processes'",
          "type": "integer"
        },
        "process_type": {
          "title": "Process Type",
          "env_names": "'process_type'",
          "type": "string"
        },
        "threads_per_process": {
          "title": "Threads Per Process",
          "default": 1,
          "env_names": "'threads_per_process'",
          "type": "integer"
        },
        "thread_type": {
          "title": "Thread Type",
          "env_names": "'thread_type'",
          "type": "string"
        }
      }
    },
    "additionalProperties": false
  }
},

```

(continues on next page)

(continued from previous page)

```

"GPUReqs": {
  "title": "GPUReqs",
  "description": "radical.entk task.gpu_reqs parameters.",
  "type": "object",
  "properties": {
    "processes": {
      "title": "Processes",
      "default": 0,
      "env_names": "'processes'",
      "type": "integer"
    },
    "process_type": {
      "title": "Process Type",
      "env_names": "'process_type'",
      "type": "string"
    },
    "threads_per_process": {
      "title": "Threads Per Process",
      "default": 0,
      "env_names": "'threads_per_process'",
      "type": "integer"
    },
    "thread_type": {
      "title": "Thread Type",
      "env_names": "'thread_type'",
      "type": "string"
    }
  },
  "additionalProperties": false
},
"MachineLearningTaskConfig": {
  "title": "MachineLearningTaskConfig",
  "description": "Base class for specific model configs to inherit.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'experiment_directory'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    "task_idx": {
      "title": "Task Idx",
      "default": 0,
      "env_names": "'task_idx'",

```

(continues on next page)

(continued from previous page)

```

        "type": "integer"
    },
    "output_path": {
        "title": "Output Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'output_path'",
        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'node_local_path'",
        "type": "string",
        "format": "path"
    },
    "model_tag": {
        "title": "Model Tag",
        "default": "set_by_deepdrivemd",
        "env_names": "'model_tag'",
        "type": "string"
    },
    "init_weights_path": {
        "title": "Init Weights Path",
        "env_names": "'init_weights_path'",
        "type": "string",
        "format": "path"
    }
}
}
}
}

```

Fields

- *retrain_freq* (int)
- *task_config* (*deepdrivemd.config.MachineLearningTaskConfig*)

field *retrain_freq*: int = 1

field *task_config*: *deepdrivemd.config.MachineLearningTaskConfig* [Required]

pydantic settings *deepdrivemd.config.MachineLearningTaskConfig*

Base class for specific model configs to inherit.

```

{
    "title": "MachineLearningTaskConfig",
    "description": "Base class for specific model configs to inherit.",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",

```

(continues on next page)

(continued from previous page)

```

        "default": "set_by_deepdrivemd",
        "env_names": "{ 'experiment_directory' }",
        "type": "string",
        "format": "path"
    },
    "stage_idx": {
        "title": "Stage Idx",
        "default": 0,
        "env_names": "{ 'stage_idx' }",
        "type": "integer"
    },
    "task_idx": {
        "title": "Task Idx",
        "default": 0,
        "env_names": "{ 'task_idx' }",
        "type": "integer"
    },
    "output_path": {
        "title": "Output Path",
        "default": "set_by_deepdrivemd",
        "env_names": "{ 'output_path' }",
        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "{ 'node_local_path' }",
        "type": "string",
        "format": "path"
    },
    "model_tag": {
        "title": "Model Tag",
        "default": "set_by_deepdrivemd",
        "env_names": "{ 'model_tag' }",
        "type": "string"
    },
    "init_weights_path": {
        "title": "Init Weights Path",
        "env_names": "{ 'init_weights_path' }",
        "type": "string",
        "format": "path"
    }
}

```

Config

- **extra:** *str = allow*

Fields

- *init_weights_path* (*Optional[pathlib.Path]*)

- `model_tag` (`str`)

`field init_weights_path: Optional[pathlib.Path] = None`

`field model_tag: str = 'set_by_deepdrivemd'`

`pydantic settings deepdrivemd.config.ModelSelectionStageConfig`

Global ML configuration (written one per experiment).

```
{
  "title": "ModelSelectionStageConfig",
  "description": "Global ML configuration (written one per experiment).",
  "type": "object",
  "properties": {
    "pre_exec": {
      "title": "Pre Exec",
      "default": [],
      "env_names": "'pre_exec'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "executable": {
      "title": "Executable",
      "default": "",
      "env_names": "'executable'",
      "type": "string"
    },
    "arguments": {
      "title": "Arguments",
      "default": [],
      "env_names": "'arguments'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "cpu_reqs": {
      "title": "Cpu Reqs",
      "default": {
        "processes": 1,
        "process_type": null,
        "threads_per_process": 1,
        "thread_type": null
      },
      "env_names": "'cpu_reqs'",
      "allOf": [
        {
          "$ref": "#/definitions/CPUReqs"
        }
      ]
    },
    "gpu_reqs": {
```

(continues on next page)

(continued from previous page)

```

    "title": "Gpu Reqs",
    "default": {
      "processes": 0,
      "process_type": null,
      "threads_per_process": 0,
      "thread_type": null
    },
    "env_names": "'gpu_reqs'",
    "allOf": [
      {
        "$ref": "#/definitions/GPUReqs"
      }
    ]
  },
  "task_config": {
    "title": "Task Config",
    "env_names": "'task_config'",
    "allOf": [
      {
        "$ref": "#/definitions/ModelSelectionTaskConfig"
      }
    ]
  }
},
"required": [
  "task_config"
],
"additionalProperties": false,
"definitions": {
  "CPUReqs": {
    "title": "CPUReqs",
    "description": "radical.entk task.cpu_reqs parameters.",
    "type": "object",
    "properties": {
      "processes": {
        "title": "Processes",
        "default": 1,
        "env_names": "'processes'",
        "type": "integer"
      },
      "process_type": {
        "title": "Process Type",
        "env_names": "'process_type'",
        "type": "string"
      },
      "threads_per_process": {
        "title": "Threads Per Process",
        "default": 1,
        "env_names": "'threads_per_process'",
        "type": "integer"
      },
      "thread_type": {

```

(continues on next page)

(continued from previous page)

```

        "title": "Thread Type",
        "env_names": "{ 'thread_type' }",
        "type": "string"
    }
},
"additionalProperties": false
},
"GPUReqs": {
    "title": "GPUReqs",
    "description": "radical.entk task.gpu_reqs parameters.",
    "type": "object",
    "properties": {
        "processes": {
            "title": "Processes",
            "default": 0,
            "env_names": "{ 'processes' }",
            "type": "integer"
        },
        "process_type": {
            "title": "Process Type",
            "env_names": "{ 'process_type' }",
            "type": "string"
        },
        "threads_per_process": {
            "title": "Threads Per Process",
            "default": 0,
            "env_names": "{ 'threads_per_process' }",
            "type": "integer"
        },
        "thread_type": {
            "title": "Thread Type",
            "env_names": "{ 'thread_type' }",
            "type": "string"
        }
    },
    "additionalProperties": false
},
"ModelSelectionTaskConfig": {
    "title": "ModelSelectionTaskConfig",
    "description": "Base class for specific model selection configs to inherit.
↪",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",
            "default": "set_by_deepdrivemd",
            "env_names": "{ 'experiment_directory' }",
            "type": "string",
            "format": "path"
        },
        "stage_idx": {
            "title": "Stage Idx",

```

(continues on next page)

(continued from previous page)

```

        "default": 0,
        "env_names": "{ 'stage_idx' }",
        "type": "integer"
    },
    "task_idx": {
        "title": "Task Idx",
        "default": 0,
        "env_names": "{ 'task_idx' }",
        "type": "integer"
    },
    "output_path": {
        "title": "Output Path",
        "default": "set_by_deepdrivemd",
        "env_names": "{ 'output_path' }",
        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "{ 'node_local_path' }",
        "type": "string",
        "format": "path"
    }
}
}
}
}

```

Fields

- `task_config` (`deepdrivemd.config.ModelSelectionTaskConfig`)

field `task_config`: `deepdrivemd.config.ModelSelectionTaskConfig` [Required]

pydantic settings `deepdrivemd.config.ModelSelectionTaskConfig`

Base class for specific model selection configs to inherit.

```

{
    "title": "ModelSelectionTaskConfig",
    "description": "Base class for specific model selection configs to inherit.",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",
            "default": "set_by_deepdrivemd",
            "env_names": "{ 'experiment_directory' }",
            "type": "string",
            "format": "path"
        },
        "stage_idx": {
            "title": "Stage Idx",

```

(continues on next page)

(continued from previous page)

```

        "default": 0,
        "env_names": {"stage_idx"},
        "type": "integer"
    },
    "task_idx": {
        "title": "Task Idx",
        "default": 0,
        "env_names": {"task_idx"},
        "type": "integer"
    },
    "output_path": {
        "title": "Output Path",
        "default": "set_by_deepdrivemd",
        "env_names": {"output_path"},
        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": {"node_local_path"},
        "type": "string",
        "format": "path"
    }
}

```

Config

- **extra:** *str* = *allow*

Fields

- *experiment_directory* (*pathlib.Path*)
- *node_local_path* (*Optional[pathlib.Path]*)
- *output_path* (*pathlib.Path*)
- *stage_idx* (*int*)
- *task_idx* (*int*)

```

field experiment_directory: pathlib.Path = PosixPath('set_by_deepdrivemd')
field node_local_path: Optional[pathlib.Path] = PosixPath('set_by_deepdrivemd')
field output_path: pathlib.Path = PosixPath('set_by_deepdrivemd')
field stage_idx: int = 0
field task_idx: int = 0

```

pydantic settings `deepdrivemd.config.MolecularDynamicsStageConfig`

Global MD configuration (written one per experiment).

```

{
  "title": "MolecularDynamicsStageConfig",
  "description": "Global MD configuration (written one per experiment).",
  "type": "object",
  "properties": {
    "pre_exec": {
      "title": "Pre Exec",
      "default": [],
      "env_names": "'pre_exec'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "executable": {
      "title": "Executable",
      "default": "",
      "env_names": "'executable'",
      "type": "string"
    },
    "arguments": {
      "title": "Arguments",
      "default": [],
      "env_names": "'arguments'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "cpu_reqs": {
      "title": "Cpu Reqs",
      "default": {
        "processes": 1,
        "process_type": null,
        "threads_per_process": 1,
        "thread_type": null
      },
      "env_names": "'cpu_reqs'",
      "allOf": [
        {
          "$ref": "#/definitions/CPUReqs"
        }
      ]
    },
    "gpu_reqs": {
      "title": "Gpu Reqs",
      "default": {
        "processes": 0,
        "process_type": null,
        "threads_per_process": 0,
        "thread_type": null
      },
      "env_names": "'gpu_reqs'",

```

(continues on next page)

(continued from previous page)

```

    "allOf": [
      {
        "$ref": "#/definitions/GPUReqs"
      }
    ]
  },
  "num_tasks": {
    "title": "Num Tasks",
    "default": 1,
    "env_names": "'num_tasks'",
    "type": "integer"
  },
  "task_config": {
    "title": "Task Config",
    "env_names": "'task_config'",
    "allOf": [
      {
        "$ref": "#/definitions/MolecularDynamicsTaskConfig"
      }
    ]
  }
},
"required": [
  "task_config"
],
"additionalProperties": false,
"definitions": {
  "CPUReqs": {
    "title": "CPUReqs",
    "description": "radical.entk task.cpu_reqs parameters.",
    "type": "object",
    "properties": {
      "processes": {
        "title": "Processes",
        "default": 1,
        "env_names": "'processes'",
        "type": "integer"
      },
      "process_type": {
        "title": "Process Type",
        "env_names": "'process_type'",
        "type": "string"
      },
      "threads_per_process": {
        "title": "Threads Per Process",
        "default": 1,
        "env_names": "'threads_per_process'",
        "type": "integer"
      },
      "thread_type": {
        "title": "Thread Type",
        "env_names": "'thread_type'",

```

(continues on next page)

(continued from previous page)

```

        "type": "string"
    }
},
"additionalProperties": false
},
"GPUReqs": {
    "title": "GPUReqs",
    "description": "radical.entk task.gpu_reqs parameters.",
    "type": "object",
    "properties": {
        "processes": {
            "title": "Processes",
            "default": 0,
            "env_names": "'processes'",
            "type": "integer"
        },
        "process_type": {
            "title": "Process Type",
            "env_names": "'process_type'",
            "type": "string"
        },
        "threads_per_process": {
            "title": "Threads Per Process",
            "default": 0,
            "env_names": "'threads_per_process'",
            "type": "integer"
        },
        "thread_type": {
            "title": "Thread Type",
            "env_names": "'thread_type'",
            "type": "string"
        }
    },
    "additionalProperties": false
},
"MolecularDynamicsTaskConfig": {
    "title": "MolecularDynamicsTaskConfig",
    "description": "Auto-generates configuration file for MD tasks.",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",
            "default": "set_by_deepdrivemd",
            "env_names": "'experiment_directory'",
            "type": "string",
            "format": "path"
        },
        "stage_idx": {
            "title": "Stage Idx",
            "default": 0,
            "env_names": "'stage_idx'",
            "type": "integer"
        }
    }
}

```

(continues on next page)

(continued from previous page)

```

    },
    "task_idx": {
        "title": "Task Idx",
        "default": 0,
        "env_names": "'task_idx'",
        "type": "integer"
    },
    "output_path": {
        "title": "Output Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'output_path'",
        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'node_local_path'",
        "type": "string",
        "format": "path"
    },
    "pdb_file": {
        "title": "Pdb File",
        "default": "set_by_deepdrivemd",
        "env_names": "'pdb_file'",
        "type": "string",
        "format": "path"
    },
    "initial_pdb_dir": {
        "title": "Initial Pdb Dir",
        "env_names": "'initial_pdb_dir'",
        "type": "string",
        "format": "path"
    }
},
"required": [
    "initial_pdb_dir"
]
}
}
}

```

Fields

- *num_tasks* (*int*)
- *task_config* (*deepdrivemd.config.MolecularDynamicsTaskConfig*)

field num_tasks: *int* = 1

field task_config: *deepdrivemd.config.MolecularDynamicsTaskConfig* [Required]

pydantic settings *deepdrivemd.config.MolecularDynamicsTaskConfig*

Auto-generates configuration file for MD tasks.

```

{
  "title": "MolecularDynamicsTaskConfig",
  "description": "Auto-generates configuration file for MD tasks.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'experiment_directory'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    "task_idx": {
      "title": "Task Idx",
      "default": 0,
      "env_names": "'task_idx'",
      "type": "integer"
    },
    "output_path": {
      "title": "Output Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'output_path'",
      "type": "string",
      "format": "path"
    },
    "node_local_path": {
      "title": "Node Local Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'node_local_path'",
      "type": "string",
      "format": "path"
    },
    "pdb_file": {
      "title": "Pdb File",
      "default": "set_by_deepdrivemd",
      "env_names": "'pdb_file'",
      "type": "string",
      "format": "path"
    },
    "initial_pdb_dir": {
      "title": "Initial Pdb Dir",
      "env_names": "'initial_pdb_dir'",
      "type": "string",
      "format": "path"
    }
  },
  "required": [

```

(continues on next page)

(continued from previous page)

```

    "initial_pdb_dir"
  ]
}

```

Config

- **extra:** *str = allow*

Fields

- *initial_pdb_dir* (*pathlib.Path*)
- *pdb_file* (*Optional[pathlib.Path]*)

```
field initial_pdb_dir:  pathlib.Path [Required]
```

```
field pdb_file:  Optional[pathlib.Path] = PosixPath('set_by_deepdrivemd')
```

pydantic settings deepdrivemd.config.StreamingAgentStageConfig

```

{
  "title": "StreamingAgentStageConfig",
  "description": "Global agent configuration (written one per experiment).",
  "type": "object",
  "properties": {
    "pre_exec": {
      "title": "Pre Exec",
      "default": [],
      "env_names": "'pre_exec'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "executable": {
      "title": "Executable",
      "default": "",
      "env_names": "'executable'",
      "type": "string"
    },
    "arguments": {
      "title": "Arguments",
      "default": [],
      "env_names": "'arguments'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "cpu_reqs": {
      "title": "Cpu Reqs",
      "default": {
        "processes": 1,
        "process_type": null,

```

(continues on next page)

(continued from previous page)

```

        "threads_per_process": 1,
        "thread_type": null
    },
    "env_names": "'cpu_reqs'",
    "allOf": [
        {
            "$ref": "#/definitions/CPUReqs"
        }
    ]
},
"gpu_reqs": {
    "title": "Gpu Reqs",
    "default": {
        "processes": 0,
        "process_type": null,
        "threads_per_process": 0,
        "thread_type": null
    },
    "env_names": "'gpu_reqs'",
    "allOf": [
        {
            "$ref": "#/definitions/GPUReqs"
        }
    ]
},
"task_config": {
    "title": "Task Config",
    "env_names": "'task_config'",
    "allOf": [
        {
            "$ref": "#/definitions/AgentTaskConfig"
        }
    ]
},
"num_tasks": {
    "title": "Num Tasks",
    "default": 1,
    "env_names": "'num_tasks'",
    "type": "integer"
}
},
"required": [
    "task_config"
],
"additionalProperties": false,
"definitions": {
    "CPUReqs": {
        "title": "CPUReqs",
        "description": "radical.entk task.cpu_reqs parameters.",
        "type": "object",
        "properties": {
            "processes": {

```

(continues on next page)

(continued from previous page)

```

        "title": "Processes",
        "default": 1,
        "env_names": "{ 'processes' }",
        "type": "integer"
    },
    "process_type": {
        "title": "Process Type",
        "env_names": "{ 'process_type' }",
        "type": "string"
    },
    "threads_per_process": {
        "title": "Threads Per Process",
        "default": 1,
        "env_names": "{ 'threads_per_process' }",
        "type": "integer"
    },
    "thread_type": {
        "title": "Thread Type",
        "env_names": "{ 'thread_type' }",
        "type": "string"
    }
},
"additionalProperties": false
},
"GPUReqs": {
    "title": "GPUReqs",
    "description": "radical.entk task.gpu_reqs parameters.",
    "type": "object",
    "properties": {
        "processes": {
            "title": "Processes",
            "default": 0,
            "env_names": "{ 'processes' }",
            "type": "integer"
        },
        "process_type": {
            "title": "Process Type",
            "env_names": "{ 'process_type' }",
            "type": "string"
        },
        "threads_per_process": {
            "title": "Threads Per Process",
            "default": 0,
            "env_names": "{ 'threads_per_process' }",
            "type": "integer"
        },
        "thread_type": {
            "title": "Thread Type",
            "env_names": "{ 'thread_type' }",
            "type": "string"
        }
    }
},

```

(continues on next page)

(continued from previous page)

```

    "additionalProperties": false
  },
  "AgentTaskConfig": {
    "title": "AgentTaskConfig",
    "description": "Base class for specific agent configs to inherit.",
    "type": "object",
    "properties": {
      "experiment_directory": {
        "title": "Experiment Directory",
        "default": "set_by_deepdrivemd",
        "env_names": "'experiment_directory'",
        "type": "string",
        "format": "path"
      },
      "stage_idx": {
        "title": "Stage Idx",
        "default": 0,
        "env_names": "'stage_idx'",
        "type": "integer"
      },
      "task_idx": {
        "title": "Task Idx",
        "default": 0,
        "env_names": "'task_idx'",
        "type": "integer"
      },
      "output_path": {
        "title": "Output Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'output_path'",
        "type": "string",
        "format": "path"
      },
      "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'node_local_path'",
        "type": "string",
        "format": "path"
      }
    }
  }
}

```

Fields

- *num_tasks* (*int*)

```
field num_tasks: int = 1
```

pydantic settings deepdrivemd.config.StreamingAggregationStageConfig

```

{
  "title": "StreamingAggregationStageConfig",
  "description": "Global aggregation configuration (written one per experiment).",
  "type": "object",
  "properties": {
    "pre_exec": {
      "title": "Pre Exec",
      "default": [],
      "env_names": "'pre_exec'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "executable": {
      "title": "Executable",
      "default": "",
      "env_names": "'executable'",
      "type": "string"
    },
    "arguments": {
      "title": "Arguments",
      "default": [],
      "env_names": "'arguments'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "cpu_reqs": {
      "title": "Cpu Reqs",
      "default": {
        "processes": 1,
        "process_type": null,
        "threads_per_process": 1,
        "thread_type": null
      },
      "env_names": "'cpu_reqs'",
      "allOf": [
        {
          "$ref": "#/definitions/CPUReqs"
        }
      ]
    },
    "gpu_reqs": {
      "title": "Gpu Reqs",
      "default": {
        "processes": 0,
        "process_type": null,
        "threads_per_process": 0,
        "thread_type": null
      },
      "env_names": "'gpu_reqs'",

```

(continues on next page)

(continued from previous page)

```

    "allOf": [
      {
        "$ref": "#/definitions/GPUReqs"
      }
    ]
  },
  "skip_aggregation": {
    "title": "Skip Aggregation",
    "default": false,
    "env_names": "'skip_aggregation'",
    "type": "boolean"
  },
  "task_config": {
    "title": "Task Config",
    "env_names": "'task_config'",
    "allOf": [
      {
        "$ref": "#/definitions/AggregationTaskConfig"
      }
    ]
  },
  "num_tasks": {
    "title": "Num Tasks",
    "default": 1,
    "env_names": "'num_tasks'",
    "type": "integer"
  }
},
"required": [
  "task_config"
],
"additionalProperties": false,
"definitions": {
  "CPUReqs": {
    "title": "CPUReqs",
    "description": "radical.entk task.cpu_reqs parameters.",
    "type": "object",
    "properties": {
      "processes": {
        "title": "Processes",
        "default": 1,
        "env_names": "'processes'",
        "type": "integer"
      },
      "process_type": {
        "title": "Process Type",
        "env_names": "'process_type'",
        "type": "string"
      },
      "threads_per_process": {
        "title": "Threads Per Process",
        "default": 1,

```

(continues on next page)

(continued from previous page)

```

        "env_names": "{ 'threads_per_process' }",
        "type": "integer"
    },
    "thread_type": {
        "title": "Thread Type",
        "env_names": "{ 'thread_type' }",
        "type": "string"
    }
},
"additionalProperties": false
},
"GPUReqs": {
    "title": "GPUReqs",
    "description": "radical.entk task.gpu_reqs parameters.",
    "type": "object",
    "properties": {
        "processes": {
            "title": "Processes",
            "default": 0,
            "env_names": "{ 'processes' }",
            "type": "integer"
        },
        "process_type": {
            "title": "Process Type",
            "env_names": "{ 'process_type' }",
            "type": "string"
        },
        "threads_per_process": {
            "title": "Threads Per Process",
            "default": 0,
            "env_names": "{ 'threads_per_process' }",
            "type": "integer"
        },
        "thread_type": {
            "title": "Thread Type",
            "env_names": "{ 'thread_type' }",
            "type": "string"
        }
    },
    "additionalProperties": false
},
"AggregationTaskConfig": {
    "title": "AggregationTaskConfig",
    "description": "Base class for specific aggregation configs to inherit.",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",
            "default": "set_by_deepdrivemd",
            "env_names": "{ 'experiment_directory' }",
            "type": "string",
            "format": "path"
        }
    }
}

```

(continues on next page)

(continued from previous page)

```

    },
    "stage_idx": {
        "title": "Stage Idx",
        "default": 0,
        "env_names": "'stage_idx'",
        "type": "integer"
    },
    "task_idx": {
        "title": "Task Idx",
        "default": 0,
        "env_names": "'task_idx'",
        "type": "integer"
    },
    "output_path": {
        "title": "Output Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'output_path'",
        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'node_local_path'",
        "type": "string",
        "format": "path"
    }
}
}
}
}
}

```

Fields

- `num_tasks` (*int*)

```
field num_tasks: int = 1
```

pydantic settings `deepdrivemd.config.StreamingExperimentConfig`

```

{
    "title": "StreamingExperimentConfig",
    "description": "Main configuration.",
    "type": "object",
    "properties": {
        "title": {
            "title": "Title",
            "env_names": "'title'",
            "type": "string"
        },
        "resource": {
            "title": "Resource",

```

(continues on next page)

(continued from previous page)

```

    "env_names": "{ 'resource' }",
    "type": "string"
  },
  "queue": {
    "title": "Queue",
    "env_names": "{ 'queue' }",
    "type": "string"
  },
  "schema_": {
    "title": "Schema ",
    "env_names": "{ 'schema_' }",
    "type": "string"
  },
  "project": {
    "title": "Project",
    "env_names": "{ 'project' }",
    "type": "string"
  },
  "walltime_min": {
    "title": "Walltime Min",
    "env_names": "{ 'walltime_min' }",
    "type": "integer"
  },
  "max_iteration": {
    "title": "Max Iteration",
    "env_names": "{ 'max_iteration' }",
    "type": "integer"
  },
  "cpus_per_node": {
    "title": "Cpus Per Node",
    "env_names": "{ 'cpus_per_node' }",
    "type": "integer"
  },
  "gpus_per_node": {
    "title": "Gpus Per Node",
    "env_names": "{ 'gpus_per_node' }",
    "type": "integer"
  },
  "hardware_threads_per_cpu": {
    "title": "Hardware Threads Per Cpu",
    "env_names": "{ 'hardware_threads_per_cpu' }",
    "type": "integer"
  },
  "experiment_directory": {
    "title": "Experiment Directory",
    "env_names": "{ 'experiment_directory' }",
    "type": "string",
    "format": "path"
  },
  "node_local_path": {
    "title": "Node Local Path",
    "env_names": "{ 'node_local_path' }",

```

(continues on next page)

(continued from previous page)

```

    "type": "string",
    "format": "path"
  },
  "molecular_dynamics_stage": {
    "title": "Molecular Dynamics Stage",
    "env_names": "'molecular_dynamics_stage'",
    "allOf": [
      {
        "$ref": "#/definitions/MolecularDynamicsStageConfig"
      }
    ]
  },
  "aggregation_stage": {
    "title": "Aggregation Stage",
    "env_names": "'aggregation_stage'",
    "allOf": [
      {
        "$ref": "#/definitions/StreamingAggregationStageConfig"
      }
    ]
  },
  "machine_learning_stage": {
    "title": "Machine Learning Stage",
    "env_names": "'machine_learning_stage'",
    "allOf": [
      {
        "$ref": "#/definitions/StreamingMachineLearningStageConfig"
      }
    ]
  },
  "model_selection_stage": {
    "title": "Model Selection Stage",
    "env_names": "'model_selection_stage'",
    "allOf": [
      {
        "$ref": "#/definitions/ModelSelectionStageConfig"
      }
    ]
  },
  "agent_stage": {
    "title": "Agent Stage",
    "env_names": "'agent_stage'",
    "allOf": [
      {
        "$ref": "#/definitions/StreamingAgentStageConfig"
      }
    ]
  },
  "adios_xml_sim": {
    "title": "Adios Xml Sim",
    "env_names": "'adios_xml_sim'",
    "type": "string",

```

(continues on next page)

(continued from previous page)

```

    "format": "path"
  },
  "adios_xml_agg": {
    "title": "Adios Xml Agg",
    "env_names": "'adios_xml_agg'",
    "type": "string",
    "format": "path"
  },
  "adios_xml_agg_4ml": {
    "title": "Adios Xml Agg 4Ml",
    "env_names": "'adios_xml_agg_4ml'",
    "type": "string",
    "format": "path"
  },
  "adios_xml_file": {
    "title": "Adios Xml File",
    "env_names": "'adios_xml_file'",
    "type": "string",
    "format": "path"
  },
  "config_directory": {
    "title": "Config Directory",
    "env_names": "'config_directory'",
    "type": "string",
    "format": "path"
  },
  "software_directory": {
    "title": "Software Directory",
    "env_names": "'software_directory'",
    "type": "string",
    "format": "path"
  },
  "init_pdb_file": {
    "title": "Init Pdb File",
    "env_names": "'init_pdb_file'",
    "type": "string",
    "format": "path"
  },
  "top_file1": {
    "title": "Top File1",
    "env_names": "'top_file1'",
    "type": "string",
    "format": "path"
  },
  "ref_pdb_file": {
    "title": "Ref Pdb File",
    "env_names": "'ref_pdb_file'",
    "type": "string",
    "format": "path"
  },
  "multi_ligand_table": {
    "title": "Multi Ligand Table",

```

(continues on next page)

(continued from previous page)

```

    "env_names": "{ 'multi_ligand_table' }",
    "type": "string",
    "format": "path"
  },
  "model": {
    "title": "Model",
    "env_names": "{ 'model' }",
    "type": "string"
  }
},
"required": [
  "title",
  "resource",
  "queue",
  "schema_",
  "project",
  "walltime_min",
  "max_iteration",
  "cpus_per_node",
  "gpus_per_node",
  "hardware_threads_per_cpu",
  "experiment_directory",
  "molecular_dynamics_stage",
  "aggregation_stage",
  "machine_learning_stage",
  "agent_stage",
  "adios_xml_sim",
  "adios_xml_agg",
  "adios_xml_agg_4ml",
  "adios_xml_file",
  "config_directory",
  "software_directory",
  "init_pdb_file",
  "model"
],
"additionalProperties": false,
"definitions": {
  "CPUReqs": {
    "title": "CPUReqs",
    "description": "radical.entk task.cpu_reqs parameters.",
    "type": "object",
    "properties": {
      "processes": {
        "title": "Processes",
        "default": 1,
        "env_names": "{ 'processes' }",
        "type": "integer"
      },
      "process_type": {
        "title": "Process Type",
        "env_names": "{ 'process_type' }",
        "type": "string"
      }
    }
  }
}

```

(continues on next page)

(continued from previous page)

```

    },
    "threads_per_process": {
      "title": "Threads Per Process",
      "default": 1,
      "env_names": "{ 'threads_per_process' }",
      "type": "integer"
    },
    "thread_type": {
      "title": "Thread Type",
      "env_names": "{ 'thread_type' }",
      "type": "string"
    }
  },
  "additionalProperties": false
},
"GPUReqs": {
  "title": "GPUReqs",
  "description": "radical.entk task.gpu_reqs parameters.",
  "type": "object",
  "properties": {
    "processes": {
      "title": "Processes",
      "default": 0,
      "env_names": "{ 'processes' }",
      "type": "integer"
    },
    "process_type": {
      "title": "Process Type",
      "env_names": "{ 'process_type' }",
      "type": "string"
    },
    "threads_per_process": {
      "title": "Threads Per Process",
      "default": 0,
      "env_names": "{ 'threads_per_process' }",
      "type": "integer"
    },
    "thread_type": {
      "title": "Thread Type",
      "env_names": "{ 'thread_type' }",
      "type": "string"
    }
  },
  "additionalProperties": false
},
"MolecularDynamicsTaskConfig": {
  "title": "MolecularDynamicsTaskConfig",
  "description": "Auto-generates configuration file for MD tasks.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",

```

(continues on next page)

(continued from previous page)

```

        "default": "set_by_deepdrivemd",
        "env_names": "{ 'experiment_directory' }",
        "type": "string",
        "format": "path"
    },
    "stage_idx": {
        "title": "Stage Idx",
        "default": 0,
        "env_names": "{ 'stage_idx' }",
        "type": "integer"
    },
    "task_idx": {
        "title": "Task Idx",
        "default": 0,
        "env_names": "{ 'task_idx' }",
        "type": "integer"
    },
    "output_path": {
        "title": "Output Path",
        "default": "set_by_deepdrivemd",
        "env_names": "{ 'output_path' }",
        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "{ 'node_local_path' }",
        "type": "string",
        "format": "path"
    },
    "pdb_file": {
        "title": "Pdb File",
        "default": "set_by_deepdrivemd",
        "env_names": "{ 'pdb_file' }",
        "type": "string",
        "format": "path"
    },
    "initial_pdb_dir": {
        "title": "Initial Pdb Dir",
        "env_names": "{ 'initial_pdb_dir' }",
        "type": "string",
        "format": "path"
    }
},
"required": [
    "initial_pdb_dir"
],
"MolecularDynamicsStageConfig": {
    "title": "MolecularDynamicsStageConfig",
    "description": "Global MD configuration (written one per experiment).",

```

(continues on next page)

(continued from previous page)

```

"type": "object",
"properties": {
  "pre_exec": {
    "title": "Pre Exec",
    "default": [],
    "env_names": "'pre_exec'",
    "type": "array",
    "items": {
      "type": "string"
    }
  },
  "executable": {
    "title": "Executable",
    "default": "",
    "env_names": "'executable'",
    "type": "string"
  },
  "arguments": {
    "title": "Arguments",
    "default": [],
    "env_names": "'arguments'",
    "type": "array",
    "items": {
      "type": "string"
    }
  },
  "cpu_reqs": {
    "title": "Cpu Reqs",
    "default": {
      "processes": 1,
      "process_type": null,
      "threads_per_process": 1,
      "thread_type": null
    },
    "env_names": "'cpu_reqs'",
    "allOf": [
      {
        "$ref": "#/definitions/CPUReqs"
      }
    ]
  },
  "gpu_reqs": {
    "title": "Gpu Reqs",
    "default": {
      "processes": 0,
      "process_type": null,
      "threads_per_process": 0,
      "thread_type": null
    },
    "env_names": "'gpu_reqs'",
    "allOf": [
      {

```

(continues on next page)

(continued from previous page)

```

        "$ref": "#/definitions/GPUReqs"
    }
]
},
"num_tasks": {
    "title": "Num Tasks",
    "default": 1,
    "env_names": "'num_tasks'",
    "type": "integer"
},
"task_config": {
    "title": "Task Config",
    "env_names": "'task_config'",
    "allOf": [
        {
            "$ref": "#/definitions/MolecularDynamicsTaskConfig"
        }
    ]
}
},
"required": [
    "task_config"
],
"additionalProperties": false
},
"AggregationTaskConfig": {
    "title": "AggregationTaskConfig",
    "description": "Base class for specific aggregation configs to inherit.",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",
            "default": "set_by_deepdrivemd",
            "env_names": "'experiment_directory'",
            "type": "string",
            "format": "path"
        },
        "stage_idx": {
            "title": "Stage Idx",
            "default": 0,
            "env_names": "'stage_idx'",
            "type": "integer"
        },
        "task_idx": {
            "title": "Task Idx",
            "default": 0,
            "env_names": "'task_idx'",
            "type": "integer"
        },
        "output_path": {
            "title": "Output Path",
            "default": "set_by_deepdrivemd",

```

(continues on next page)

(continued from previous page)

```

        "env_names": "{ 'output_path' }",
        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "{ 'node_local_path' }",
        "type": "string",
        "format": "path"
    }
},
"StreamingAggregationStageConfig": {
    "title": "StreamingAggregationStageConfig",
    "description": "Global aggregation configuration (written one per_
↪ experiment).",
    "type": "object",
    "properties": {
        "pre_exec": {
            "title": "Pre Exec",
            "default": [],
            "env_names": "{ 'pre_exec' }",
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "executable": {
            "title": "Executable",
            "default": "",
            "env_names": "{ 'executable' }",
            "type": "string"
        },
        "arguments": {
            "title": "Arguments",
            "default": [],
            "env_names": "{ 'arguments' }",
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "cpu_reqs": {
            "title": "Cpu Reqs",
            "default": {
                "processes": 1,
                "process_type": null,
                "threads_per_process": 1,
                "thread_type": null
            },
            "env_names": "{ 'cpu_reqs' }",

```

(continues on next page)

(continued from previous page)

```

        "allOf": [
            {
                "$ref": "#/definitions/CPUReqs"
            }
        ],
    },
    "gpu_reqs": {
        "title": "Gpu Reqs",
        "default": {
            "processes": 0,
            "process_type": null,
            "threads_per_process": 0,
            "thread_type": null
        },
        "env_names": "'gpu_reqs'",
        "allOf": [
            {
                "$ref": "#/definitions/GPUReqs"
            }
        ],
    },
    "skip_aggregation": {
        "title": "Skip Aggregation",
        "default": false,
        "env_names": "'skip_aggregation'",
        "type": "boolean"
    },
    "task_config": {
        "title": "Task Config",
        "env_names": "'task_config'",
        "allOf": [
            {
                "$ref": "#/definitions/AggregationTaskConfig"
            }
        ],
    },
    "num_tasks": {
        "title": "Num Tasks",
        "default": 1,
        "env_names": "'num_tasks'",
        "type": "integer"
    },
    },
    "required": [
        "task_config"
    ],
    "additionalProperties": false
},
"MachineLearningTaskConfig": {
    "title": "MachineLearningTaskConfig",
    "description": "Base class for specific model configs to inherit.",
    "type": "object",

```

(continues on next page)

(continued from previous page)

```

"properties": {
  "experiment_directory": {
    "title": "Experiment Directory",
    "default": "set_by_deepdrivemd",
    "env_names": "{ 'experiment_directory' }",
    "type": "string",
    "format": "path"
  },
  "stage_idx": {
    "title": "Stage Idx",
    "default": 0,
    "env_names": "{ 'stage_idx' }",
    "type": "integer"
  },
  "task_idx": {
    "title": "Task Idx",
    "default": 0,
    "env_names": "{ 'task_idx' }",
    "type": "integer"
  },
  "output_path": {
    "title": "Output Path",
    "default": "set_by_deepdrivemd",
    "env_names": "{ 'output_path' }",
    "type": "string",
    "format": "path"
  },
  "node_local_path": {
    "title": "Node Local Path",
    "default": "set_by_deepdrivemd",
    "env_names": "{ 'node_local_path' }",
    "type": "string",
    "format": "path"
  },
  "model_tag": {
    "title": "Model Tag",
    "default": "set_by_deepdrivemd",
    "env_names": "{ 'model_tag' }",
    "type": "string"
  },
  "init_weights_path": {
    "title": "Init Weights Path",
    "env_names": "{ 'init_weights_path' }",
    "type": "string",
    "format": "path"
  }
},
"StreamingMachineLearningStageConfig": {
  "title": "StreamingMachineLearningStageConfig",
  "description": "Global ML configuration (written one per experiment).",
  "type": "object",

```

(continues on next page)

(continued from previous page)

```

"properties": {
  "pre_exec": {
    "title": "Pre Exec",
    "default": [],
    "env_names": "'pre_exec'",
    "type": "array",
    "items": {
      "type": "string"
    }
  },
  "executable": {
    "title": "Executable",
    "default": "",
    "env_names": "'executable'",
    "type": "string"
  },
  "arguments": {
    "title": "Arguments",
    "default": [],
    "env_names": "'arguments'",
    "type": "array",
    "items": {
      "type": "string"
    }
  },
  "cpu_reqs": {
    "title": "Cpu Reqs",
    "default": {
      "processes": 1,
      "process_type": null,
      "threads_per_process": 1,
      "thread_type": null
    },
    "env_names": "'cpu_reqs'",
    "allOf": [
      {
        "$ref": "#/definitions/CPUReqs"
      }
    ]
  },
  "gpu_reqs": {
    "title": "Gpu Reqs",
    "default": {
      "processes": 0,
      "process_type": null,
      "threads_per_process": 0,
      "thread_type": null
    },
    "env_names": "'gpu_reqs'",
    "allOf": [
      {
        "$ref": "#/definitions/GPUReqs"
      }
    ]
  }
}

```

(continues on next page)

(continued from previous page)

```

    }
  ]
},
"retrain_freq": {
  "title": "Retrain Freq",
  "default": 1,
  "env_names": "'retrain_freq'",
  "type": "integer"
},
"task_config": {
  "title": "Task Config",
  "env_names": "'task_config'",
  "allOf": [
    {
      "$ref": "#/definitions/MachineLearningTaskConfig"
    }
  ]
},
"num_tasks": {
  "title": "Num Tasks",
  "default": 1,
  "env_names": "'num_tasks'",
  "type": "integer"
}
},
"required": [
  "task_config"
],
"additionalProperties": false
},
"ModelSelectionTaskConfig": {
  "title": "ModelSelectionTaskConfig",
  "description": "Base class for specific model selection configs to inherit.
↪",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'experiment_directory'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    "task_idx": {
      "title": "Task Idx",
      "default": 0,

```

(continues on next page)

(continued from previous page)

```

        "env_names": "'task_idx'",
        "type": "integer"
    },
    "output_path": {
        "title": "Output Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'output_path'",
        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'node_local_path'",
        "type": "string",
        "format": "path"
    }
},
"ModelSelectionStageConfig": {
    "title": "ModelSelectionStageConfig",
    "description": "Global ML configuration (written one per experiment).",
    "type": "object",
    "properties": {
        "pre_exec": {
            "title": "Pre Exec",
            "default": [],
            "env_names": "'pre_exec'",
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "executable": {
            "title": "Executable",
            "default": "",
            "env_names": "'executable'",
            "type": "string"
        },
        "arguments": {
            "title": "Arguments",
            "default": [],
            "env_names": "'arguments'",
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "cpu_reqs": {
            "title": "Cpu Reqs",
            "default": {
                "processes": 1,

```

(continues on next page)

(continued from previous page)

```

        "process_type": null,
        "threads_per_process": 1,
        "thread_type": null
    },
    "env_names": "'cpu_reqs'",
    "allOf": [
        {
            "$ref": "#/definitions/CPUReqs"
        }
    ]
},
"gpu_reqs": {
    "title": "Gpu Reqs",
    "default": {
        "processes": 0,
        "process_type": null,
        "threads_per_process": 0,
        "thread_type": null
    },
    "env_names": "'gpu_reqs'",
    "allOf": [
        {
            "$ref": "#/definitions/GPUReqs"
        }
    ]
},
"task_config": {
    "title": "Task Config",
    "env_names": "'task_config'",
    "allOf": [
        {
            "$ref": "#/definitions/ModelSelectionTaskConfig"
        }
    ]
},
},
"required": [
    "task_config"
],
"additionalProperties": false
},
"AgentTaskConfig": {
    "title": "AgentTaskConfig",
    "description": "Base class for specific agent configs to inherit.",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",
            "default": "set_by_deepdrivemd",
            "env_names": "'experiment_directory'",
            "type": "string",
            "format": "path"
        }
    }
}

```

(continues on next page)

(continued from previous page)

```

    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    },
    "task_idx": {
      "title": "Task Idx",
      "default": 0,
      "env_names": "'task_idx'",
      "type": "integer"
    },
    },
    "output_path": {
      "title": "Output Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'output_path'",
      "type": "string",
      "format": "path"
    },
    },
    "node_local_path": {
      "title": "Node Local Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'node_local_path'",
      "type": "string",
      "format": "path"
    },
    }
  },
  "StreamingAgentStageConfig": {
    "title": "StreamingAgentStageConfig",
    "description": "Global agent configuration (written one per experiment).",
    "type": "object",
    "properties": {
      "pre_exec": {
        "title": "Pre Exec",
        "default": [],
        "env_names": "'pre_exec'",
        "type": "array",
        "items": {
          "type": "string"
        }
      },
    },
    "executable": {
      "title": "Executable",
      "default": "",
      "env_names": "'executable'",
      "type": "string"
    },
    },
    "arguments": {
      "title": "Arguments",
      "default": [],

```

(continues on next page)

(continued from previous page)

```

        "env_names": "'arguments'",
        "type": "array",
        "items": {
            "type": "string"
        }
    },
    "cpu_reqs": {
        "title": "Cpu Reqs",
        "default": {
            "processes": 1,
            "process_type": null,
            "threads_per_process": 1,
            "thread_type": null
        },
        "env_names": "'cpu_reqs'",
        "allOf": [
            {
                "$ref": "#/definitions/CPUReqs"
            }
        ]
    },
    "gpu_reqs": {
        "title": "Gpu Reqs",
        "default": {
            "processes": 0,
            "process_type": null,
            "threads_per_process": 0,
            "thread_type": null
        },
        "env_names": "'gpu_reqs'",
        "allOf": [
            {
                "$ref": "#/definitions/GPUReqs"
            }
        ]
    },
    "task_config": {
        "title": "Task Config",
        "env_names": "'task_config'",
        "allOf": [
            {
                "$ref": "#/definitions/AgentTaskConfig"
            }
        ]
    },
    "num_tasks": {
        "title": "Num Tasks",
        "default": 1,
        "env_names": "'num_tasks'",
        "type": "integer"
    }
},

```

(continues on next page)

(continued from previous page)

```

    "required": [
        "task_config"
    ],
    "additionalProperties": false
  }
}

```

Fields

- *adios_xml_agg* (*pathlib.Path*)
- *adios_xml_agg_4ml* (*pathlib.Path*)
- *adios_xml_file* (*pathlib.Path*)
- *adios_xml_sim* (*pathlib.Path*)
- *agent_stage* (*deepdrivemd.config.StreamingAgentStageConfig*)
- *aggregation_stage* (*deepdrivemd.config.StreamingAggregationStageConfig*)
- *config_directory* (*pathlib.Path*)
- *init_pdb_file* (*pathlib.Path*)
- *machine_learning_stage* (*deepdrivemd.config.StreamingMachineLearningStageConfig*)
- *model* (*str*)
- *model_selection_stage* (*Optional[deepdrivemd.config.ModelSelectionStageConfig]*)
- *multi_ligand_table* (*Optional[pathlib.Path]*)
- *ref_pdb_file* (*Optional[pathlib.Path]*)
- *software_directory* (*pathlib.Path*)
- *top_file1* (*Optional[pathlib.Path]*)

Validators

```

field adios_xml_agg:  pathlib.Path [Required]
field adios_xml_agg_4ml:  pathlib.Path [Required]
field adios_xml_file:  pathlib.Path [Required]
field adios_xml_sim:  pathlib.Path [Required]
field agent_stage:  deepdrivemd.config.StreamingAgentStageConfig [Required]
field aggregation_stage:  deepdrivemd.config.StreamingAggregationStageConfig [Required]
field config_directory:  pathlib.Path [Required]
field init_pdb_file:  pathlib.Path [Required]

```

```
field machine_learning_stage:
    deepdrivemd.config.StreamingMachineLearningStageConfig [Required]

field model: str [Required]

field model_selection_stage: Optional[deepdrivemd.config.ModelSelectionStageConfig]
    = PydanticUndefined

field multi_ligand_table: Optional[pathlib.Path] = PydanticUndefined

field ref_pdb_file: Optional[pathlib.Path] = PydanticUndefined

field software_directory: pathlib.Path [Required]

field top_file1: Optional[pathlib.Path] = PydanticUndefined
```

pydantic settings deepdrivemd.config.StreamingMachineLearningStageConfig

```
{
  "title": "StreamingMachineLearningStageConfig",
  "description": "Global ML configuration (written one per experiment).",
  "type": "object",
  "properties": {
    "pre_exec": {
      "title": "Pre Exec",
      "default": [],
      "env_names": "'pre_exec'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "executable": {
      "title": "Executable",
      "default": "",
      "env_names": "'executable'",
      "type": "string"
    },
    "arguments": {
      "title": "Arguments",
      "default": [],
      "env_names": "'arguments'",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "cpu_reqs": {
      "title": "Cpu Reqs",
      "default": {
        "processes": 1,
        "process_type": null,
        "threads_per_process": 1,
        "thread_type": null
      }
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

    "env_names": "'cpu_reqs'",
    "allOf": [
      {
        "$ref": "#/definitions/CPUReqs"
      }
    ]
  },
  "gpu_reqs": {
    "title": "Gpu Reqs",
    "default": {
      "processes": 0,
      "process_type": null,
      "threads_per_process": 0,
      "thread_type": null
    },
    "env_names": "'gpu_reqs'",
    "allOf": [
      {
        "$ref": "#/definitions/GPUReqs"
      }
    ]
  },
  "retrain_freq": {
    "title": "Retrain Freq",
    "default": 1,
    "env_names": "'retrain_freq'",
    "type": "integer"
  },
  "task_config": {
    "title": "Task Config",
    "env_names": "'task_config'",
    "allOf": [
      {
        "$ref": "#/definitions/MachineLearningTaskConfig"
      }
    ]
  },
  "num_tasks": {
    "title": "Num Tasks",
    "default": 1,
    "env_names": "'num_tasks'",
    "type": "integer"
  }
},
"required": [
  "task_config"
],
"additionalProperties": false,
"definitions": {
  "CPUReqs": {
    "title": "CPUReqs",
    "description": "radical.entk task.cpu_reqs parameters.",

```

(continues on next page)

(continued from previous page)

```

    "type": "object",
    "properties": {
      "processes": {
        "title": "Processes",
        "default": 1,
        "env_names": "{ 'processes' }",
        "type": "integer"
      },
      "process_type": {
        "title": "Process Type",
        "env_names": "{ 'process_type' }",
        "type": "string"
      },
      "threads_per_process": {
        "title": "Threads Per Process",
        "default": 1,
        "env_names": "{ 'threads_per_process' }",
        "type": "integer"
      },
      "thread_type": {
        "title": "Thread Type",
        "env_names": "{ 'thread_type' }",
        "type": "string"
      }
    },
    "additionalProperties": false
  },
  "GPUReqs": {
    "title": "GPUReqs",
    "description": "radical.entk task.gpu_reqs parameters.",
    "type": "object",
    "properties": {
      "processes": {
        "title": "Processes",
        "default": 0,
        "env_names": "{ 'processes' }",
        "type": "integer"
      },
      "process_type": {
        "title": "Process Type",
        "env_names": "{ 'process_type' }",
        "type": "string"
      },
      "threads_per_process": {
        "title": "Threads Per Process",
        "default": 0,
        "env_names": "{ 'threads_per_process' }",
        "type": "integer"
      },
      "thread_type": {
        "title": "Thread Type",
        "env_names": "{ 'thread_type' }",

```

(continues on next page)

(continued from previous page)

```

        "type": "string"
    },
    },
    "additionalProperties": false
},
"MachineLearningTaskConfig": {
    "title": "MachineLearningTaskConfig",
    "description": "Base class for specific model configs to inherit.",
    "type": "object",
    "properties": {
        "experiment_directory": {
            "title": "Experiment Directory",
            "default": "set_by_deepdrivemd",
            "env_names": "'experiment_directory'",
            "type": "string",
            "format": "path"
        },
        "stage_idx": {
            "title": "Stage Idx",
            "default": 0,
            "env_names": "'stage_idx'",
            "type": "integer"
        },
        "task_idx": {
            "title": "Task Idx",
            "default": 0,
            "env_names": "'task_idx'",
            "type": "integer"
        },
        "output_path": {
            "title": "Output Path",
            "default": "set_by_deepdrivemd",
            "env_names": "'output_path'",
            "type": "string",
            "format": "path"
        },
        "node_local_path": {
            "title": "Node Local Path",
            "default": "set_by_deepdrivemd",
            "env_names": "'node_local_path'",
            "type": "string",
            "format": "path"
        },
        "model_tag": {
            "title": "Model Tag",
            "default": "set_by_deepdrivemd",
            "env_names": "'model_tag'",
            "type": "string"
        },
        "init_weights_path": {
            "title": "Init Weights Path",
            "env_names": "'init_weights_path'",

```

(continues on next page)

(continued from previous page)

```

        "type": "string",
        "format": "path"
    }
}
}
}
}

```

Fields

- `num_tasks` (`int`)

field `num_tasks`: `int` = 1

`deepdrivemd.config.generate_sample_config()` → `deepdrivemd.config.ExperimentConfig`

deepdrivemd.data**Modules**

`deepdrivemd.data.analysis`

`deepdrivemd.data.api`

`deepdrivemd.data.stream`

`deepdrivemd.data.utils` Data utility functions for handling HDF5 files.

deepdrivemd.data.analysis**Classes**

`DeepDriveMDAnalysis(experiment_directory)`

class `deepdrivemd.data.analysis.DeepDriveMDAnalysis(experiment_directory: Union[str, pathlib.Path])`

apply_analysis_fn(`fn: Callable[[Iterable[Dict[str, List[str]]], Any]`, `num_workers: Optional[int] = None`, `n: Optional[int] = None`, `data_file_suffix: str = '.h5'`, `traj_file_suffix: str = '.dcd'`, `structure_file_suffix: str = '.pdb'`) → `List[Any]`

get_agent_h5(`iterations: int = -1`, `fields: List[str] = []`) → `List[Dict[str, npt.ArrayLike]]`

get_agent_json(`iterations: int = -1`) → `List[Optional[List[Dict[str, Any]]]]`

deepdrivemd.data.api

Functions

<i>glob_file_from_dirs</i> (dirs, pattern)	Return a list of all items matching <i>pattern</i> in multiple <i>dirs</i> .
--	--

Classes

DeepDriveMD_API(experiment_directory)

Stage_API(experiment_dir, stage_dir_name)

class deepdrivemd.data.api.**DeepDriveMD_API**(*experiment_directory*: Union[str, pathlib.Path])

AGENT_DIR = 'agent_runs'

AGGREGATE_DIR = 'aggregation_runs'

MACHINE_LEARNING_DIR = 'machine_learning_runs'

MODEL_SELECTION_DIR = 'model_selection_runs'

MOLECULAR_DYNAMICS_DIR = 'molecular_dynamics_runs'

static *get_initial_pdb*s(*initial_pdb_dir*: Union[str, pathlib.Path]) → List[pathlib.Path]

Return a list of PDB paths from the *initial_pdb_dir*.

Parameters *initial_pdb_dir* (Union[str, Path]) – Initial data directory passed containing PDBs and optional topologies.

Returns List[Path] – List of paths to initial PDB files.

Raises **ValueError** – If any of the PDB file names contain a double underscore __.

get_last_n_md_runs(*n*: Optional[int] = None, *data_file_suffix*: str = '.h5', *traj_file_suffix*: str = '.dcd', *structure_file_suffix*: str = '.pdb') → Dict[str, List[str]]

Get the last *n* MD run directories data file paths.

Return a dictionary of data file paths for the last *n* MD runs including the training data files, the trajectory files, and the coordinate files.

Parameters

- **n** (int, optional) – Number of latest MD run directories to glob data files from. Defaults to all MD run directories.
- **data_file_suffix** (int, optional) – The suffix of the training data file. Defaults to “.h5”.
- **traj_file_suffix** (str, optional) – The suffix of the traj file. Defaults to “.dcd”.
- **structure_file_suffix** (str, optional) – The suffix of the structure file. Defaults to “.pdb”.

Returns Dict[str, List[str]] – A dictionary with keys “data_files”, “traj_files” and “structure_files” each containing a list of *n* paths globed from the the latest *n* MD run directories.

get_restart_pdb(*index: int, stage_idx: int = - 1, task_idx: int = 0*) → Dict[str, Any]

Gets a single datum for the restart points JSON file.

Parameters *index* (*int*) – Index into the agent_{}.json file of the latest DeepDriveMD iteration.

Returns *Dict[Any]* – Dictionary entry written by the outlier detector.

static get_system_name(*pdb_file: Union[str, pathlib.Path]*) → str

Parse the system name from a PDB file.

Parameters *pdb_file* (*Union[str, Path]*) – The PDB file to parse. Can be absolute path, relative path, or filename.

Returns *str* – The system name used to identify system topology.

Examples

```
>>> pdb_file = "/path/to/system_name__anything.pdb"
>>> DeepDriveMD_API.get_system_name(pdb_file)
'system_name'
```

```
>>> pdb_file = "/path/to/system_name/anything.pdb"
>>> DeepDriveMD_API.get_system_name(pdb_file)
'system_name'
```

static get_system_pdb_name(*pdb_file: Union[str, pathlib.Path]*) → str

Generate PDB file name with correct system name.

Parse *pdb_file* for the system name and generate a PDB file name that is parseable by DeepDriveMD. If *pdb_file* name is already compatible with DeepDriveMD, the returned name will be the same.

Parameters *pdb_file* (*Union[str, Path]*) – The PDB file to parse. Can be absolute path, relative path, or filename.

Returns *str* – The new PDB file name. File is not created.

Raises **ValueError** – If *pdb_file* contains more than one __.

Examples

```
>>> pdb_file = "/path/to/system_name__anything.pdb"
>>> DeepDriveMD_API.get_system_pdb_name(pdb_file)
'system_name__anything.pdb'
```

```
>>> pdb_file = "/path/to/system_name/anything.pdb"
>>> DeepDriveMD_API.get_system_pdb_name(pdb_file)
'system_name__anything.pdb'
```

static get_topology(*initial_pdb_dir: Union[str, pathlib.Path], pdb_file: Union[str, pathlib.Path], suffix: str = '.top'*) → Optional[pathlib.Path]

Get the topology file for the system.

Parse *pdb_file* for the system name and then retrieve the topology file from the correct subdirectory, given by the system name, in the *initial_pdb_dir* directory or return None if the system doesn't have a topology.

Parameters

- **initial_pdb_dir** (*Union[str, Path]*) – Initial data directory passed containing system sub-directories with PDBs and optional topologies.
- **pdb_file** (*Union[str, Path]*) – The PDB file to parse. Can be absolute path, relative path, or filename.
- **suffix** (*str*) – Suffix of the topology file (.top, .prmtop, etc).

Returns *Optional[Path]* – The path to the topology file, or None if system has no topology.

get_total_iterations() → int

static write_pdb(*output_pdb_file: Union[str, pathlib.Path], input_pdb_file: Union[str, pathlib.Path], traj_file: Union[str, pathlib.Path], frame: int, in_memory: bool = False*) → None

Write a PDB file.

Writes *output_pdb_file* to disk containing coordinates of a single *frame* from a given input PDB *input_pdb_file* and trajectory file *traj_file*.

Parameters

- **output_pdb_file** (*Union[str, Path]*) – The path of the output PDB file to be written to.
- **input_pdb_file** (*Union[str, Path]*) – The path of the input PDB file used to open *traj_file* in MDAnalysis.Universe().
- **traj_file** (*Union[str, Path]*) – The path of the trajectory file to be read from.
- **frame** (*int*) – The frame index into *traj_file* used to write *output_pdb_file*.
- **in_memory** (*bool, optional*) – If true, will load the MDAnalysis.Universe() trajectory into memory.

Examples

```
>>> output_pdb_file = "/path/to/output.pdb"
>>> input_pdb_file = "/path/to/input.pdb"
>>> traj_file = "/path/to/traj.dcd"
>>> frame = 10
>>> DeepDriveMD_API.write_pdb(output_pdb_file, input_pdb_file, traj_file, frame)
```

class deepdrivemd.data.api.**Stage_API**(*experiment_dir: pathlib.Path, stage_dir_name: str*)

config_path(*stage_idx: int = -1, task_idx: int = 0*) → *Optional[pathlib.Path]*

static get_count(*path: pathlib.Path, pattern: str, is_dir: bool = False*) → int

static get_latest(*path: pathlib.Path, pattern: str, is_dir: bool = False, key: typing.Callable[[pathlib.Path], pathlib.Path] = <function Stage_API.<lambda>>>*) → *Optional[pathlib.Path]*

json_path(*stage_idx: int = -1, task_idx: int = 0*) → *Optional[pathlib.Path]*

read_task_json(*stage_idx: int = -1, task_idx: int = 0*) → *Optional[List[Dict[str, Any]]]*

property runs_dir: *pathlib.Path*

stage_dir(*stage_idx: int = - 1*) → Optional[pathlib.Path]

Return the stage directory containing task subdirectories.

Each stage type has a directory containing subdirectories stageXXXX. In each stageXXXX there are several task directories labeled taskXXXX. This function returns a particular stageXXXX directory selected with *stage_idx*. Each iteration of DeepDriveMD corresponds to a stageXXXX directory, they are labeled in increasing order.

stage_dir_count() → int

Return the number of stage directories.

static stage_name(*stage_idx: int*) → str

task_dir(*stage_idx: int = - 1, task_idx: int = 0, mkdir: bool = False*) → Optional[pathlib.Path]

static task_name(*task_idx: int*) → str

static unique_name(*task_path: pathlib.Path*) → str

write_task_json(*data: List[Dict[str, Any]], stage_idx: int = - 1, task_idx: int = 0*) → None

Dump *data* to a new JSON file for the agent.

Dump *data* to a JSON file written to the directory specified by *stage_idx* and *task_idx*.

Parameters *data* (List[Dict[str, Any]]) – List of dictionarys to pass to *json.dump()*. Values in the dictionarys must be JSON serializable.

deepdrivemd.data.api.glob_file_from_dirs(*dirs: List[str], pattern: str*) → List[str]

Return a list of all items matching *pattern* in multiple *dirs*.

deepdrivemd.data.stream

Modules

deepdrivemd.data.stream.OutlierDB

deepdrivemd.data.stream.adios_utils Utility functions for ADIOS2.

deepdrivemd.data.stream.aggregator_reader

deepdrivemd.data.stream.enumerations

deepdrivemd.data.stream.OutlierDB

Classes

OutlierDB(*dir, restarts*) Stores the metadata for outliers to be used by simulations.

class **deepdrivemd.data.stream.OutlierDB.OutlierDB**(*dir: str, restarts: List[Tuple[float, str]]*)

Stores the metadata for outliers to be used by simulations.

dir

directory with published outliers

Type str

sorted_index

list of md5sums of outlier positions (used as a name of an outlier pdb or numpy file) sorted by the corresponding rmsd

Type List[str]

dictionary

maps md5sum to rmsd

Type Dict

__init__(*dir*: str, *restarts*: List[Tuple[float, str]])

Constructor

Parameters

- **dir** (str) – directory with published outliers
- **restarts** (List[Tuple[float, str]]) – list of outliers given as tuples of rmsd and md5sum of positions (used as a file name)

next_random(*m*: Optional[int] = None, *alpha*: int = 1, *beta*: int = 25) → str

Return next outlier using beta distribution that prefers smaller rmsds

Parameters

- **m** (int, default = None) – if *m* is not None, restrict the random selection to the first *m* elements of *sorted_index*, otherwise - any element can be chosen.
- **alpha** (int, default = 1)
- **beta** (int, default = 25) – *alpha* and *beta* are parameters of beta distribution.

print(*n*: int = 5)

deepdrivemd.data.stream.adios_utils

Utility functions for ADIOS2.

Classes

<i>AdiosStreamStepRW</i> (connections, variables)	Read/Write step by step adios stream using Full API.
---	--

```
class deepdrivemd.data.stream.adios_utils.AdiosStreamStepRW(connections: Dict[int,
    Tuple[adios2.adios2.ADIOS,
    adios2.adios2.IO,
    adios2.adios2.Engine]], variables:
    Dict[str, Tuple[type, deep-
    drivemd.data.stream.enumerations.DataStructure]])
```

Read/Write step by step adios stream using Full API.

connections

dictionary of adios connections; key - integer, in aggregator it is simulation task id; value - a tuple of adios objects

Type Dict[int, Tuple[adios2.adios2.ADIOS, adios2.adios2.IO, adios2.adios2.Engine]]

variables

dictionary describing variables; key - adios column name, value - a tuple of variable type and enumeration describing the structure type: scalar, numpy array, string; other class attributes are created on the fly using *setattr*: for each key two attributes are created: *var_<key>* - adios variable, *d_<key>* - data which stores the result of reading a particular variable *key* from a step of adios stream.

Type Dict[str, Tuple[type, [DataStructure](#)]],

__init__(*connections*: Dict[int, Tuple[adios2.adios2.ADIOS, adios2.adios2.IO, adios2.adios2.Engine]],
variables: Dict[str, Tuple[type, [deepdrivemd.data.stream.enumerations.DataStructure](#)]])

Initialize AdiosStreamStepRW object.

Parameters

- **connections** (Dict[int, Tuple[adios2.adios2.ADIOS, adios2.adios2.IO, adios2.adios2.Engine]]) – dictionary of adios connections; key - integer, in aggregator it is simulation task id, value - a tuple of adios objects
- **variables** (Dict[str, Tuple[type, [DataStructure](#)]]) – dictionary describing variables; key - adios column name, value - a tuple of variable type and enumeration describing the structure type: scalar, numpy array, string.

read_step(*sim_task_id*: int) → bool

Read the next step from adios stream given by *connections[sim_task_id]*.

Parameters *sim_task_id* (int) – is used as a key to get the corresponding adios objects from *connections*

Returns *bool* – *True* if reading a step succeeded, *False* - otherwise.

write_step(*wstream*: adios2.adios2.Engine, *variables*: Dict[str, Tuple[type, [deepdrivemd.data.stream.enumerations.DataStructure](#)]], *end_step*: bool = False)

Write the next step from class *d_...* variables into *wstream* adios stream.

Parameters

- **wstream** (*adios2.adios2.Engine*) – adios stream to which the data is written
- **variables** (Dict[str, Tuple[type, [DataStructure](#)]]) – a dictionary indexed by adios column names, value is a tuple - data type, structure type; structure type can be scalar, array, string
- **end_step** (*bool*, *default* = False) – if this is *True*, the write of the last variable would be marked by *end_step* = *True* meaning that the step writing is done; otherwise, terminating the step should be done outside of the method

deepdrivemd.data.stream.aggregator_reader

Classes

<code>AdiosReader</code> (fn, config, stream_name, variables)	This class is used to read the next N steps from an adios stream.
<code>StreamContactMapVariable</code> (name, dtype, structure)	Implementation of <code>StreamVariable</code> that handles contact maps: unpack bits to 1D array, convert 1D array to 2D array.
<code>StreamScalarVariable</code> (name, dtype, structure)	Implementation of <code>StreamVariable</code> that handles scalar variables.
<code>StreamVariable</code> (name, dtype, structure)	This class is used to read a variable from BP file.
<code>Streams</code> (files, variables[, config, ...])	The class keeps <i>lastN</i> steps from each aggregator.

```
class deepdrivemd.data.stream.aggregator_reader.AdiosReader(fn: str, config: pathlib.Path,
stream_name: str, variables:
List[deepdrivemd.data.stream.aggregator_reader.StreamVariable])
```

This class is used to read the next N steps from an adios stream.

adios

Type adios2.adios2.ADIOS

io

Type adios2.adios2.IO

stream

Type adios2.adios2.Engine

__del__()

Destructor: clean the adios resources.

__init__(fn: str, config: pathlib.Path, stream_name: str, variables:
List[deepdrivemd.data.stream.aggregator_reader.StreamVariable])

Initialize AdiosReader object.

Parameters

- **fn** (*str*) – file name of bp file or sst socket (without sst extension)
- **config** (*Path*) – path to *adios.xml* file
- **stream_name** (*str*) – name of a stream in *adios.xml* file

next(N : int) → Dict[str, Union[numpy.array, str, int, float]]

Read the next N steps of all variables.

Parameters N (*int*) – read that many steps

Returns Dict[str, Union[np.array, str, int, float]] – values for different variables whose names are used as keys

```
class deepdrivemd.data.stream.aggregator_reader.StreamContactMapVariable(name: str, dtype:
type, structure: deep-
drivemd.data.stream.enumerations.DataType)
```

Implementation of `StreamVariable` that handles contact maps: unpack bits to 1D array, convert 1D array to 2D array.

next(*ARW*)

Get the variable value for the next time step and append it to *total*.

Parameters *ARW* (*AdiosStreamStepRW*) – low level object for reading data from ADIOS stream (BP file or SST stream)

```
class deepdrivemd.data.stream.aggregator_reader.StreamScalarVariable(name: str, dtype: type,
                                                                    structure: deep-
                                                                    drivemd.data.stream.enumerations.DataStruc
```

Implementation of *StreamVariable* that handles scalar variables.

next(*ARW*)

Get the variable value for the next time step and append it to *total*.

Parameters *ARW* (*AdiosStreamStepRW*) – low level object for reading data from ADIOS stream (BP file or SST stream)

```
class deepdrivemd.data.stream.aggregator_reader.StreamVariable(name: str, dtype: type, structure:
                                                                deep-
                                                                drivemd.data.stream.enumerations.DataStructure)
```

This class is used to read a variable from BP file.

name

variable name in adios file

Type str

dtype

variable type, for example, np.uint8

Type type

structure

enumeration: array, scalar, string

Type *DataStructure*

total

list of variable values for different steps

Type List

```
__init__(name: str, dtype: type, structure: deepdrivemd.data.stream.enumerations.DataStructure)
```

Parameters

- **name** (*str*) – variable name in adios file
- **dtype** (*type*) – variable type, for example, np.uint8
- **structure** (*DataStructure*) – structure type: array, scalar, string

next(*ARW*: *deepdrivemd.data.stream.adios_utils.AdiosStreamStepRW*)

Get the variable value for the next time step and append it to *total*.

Parameters *ARW* (*AdiosStreamStepRW*) – low level object for reading data from ADIOS stream (BP file or SST stream)

```

class deepdrivemd.data.stream.aggregator_reader.Streams(files: List[str], variables:
    List[deepdrivemd.data.stream.aggregator_reader.StreamVariable],
    config: pathlib.Path =
    PosixPath('./aggregate/adios.xml'),
    stream_name: str = 'AdiosOutput', lastN:
    int = 2000, batch: int = 10000)

```

The class keeps *lastN* steps from each aggregator.

readers

a dictionary of *AdiosReader* indexed by the corresponding adios file name

Type Dict[str, *AdiosReader*]

positions

Type Dict[str, np.ndarray]

md5

Type Dict[str, str]

steps

Type Dict[str, np.ndarray]

rmsds

Type Dict[str, np.ndarray]

cm

Type Dict[str, np.ndarray]

velocities

Type Dict[str, np.ndarray]

lastN

keep that many last steps from each aggregator

Type int

batch

up to how many steps to read from each adios file at a time

Type int

```

__init__(files: List[str], variables: List[deepdrivemd.data.stream.aggregator_reader.StreamVariable],
    config: pathlib.Path = PosixPath('./aggregate/adios.xml'), stream_name: str = 'AdiosOutput',
    lastN: int = 2000, batch: int = 10000)

```

Initialize Streams object.

Parameters

- **files** (*List[str]*) – adios files from each aggregator,
- **variables** (*List[StreamVariable]*) – list of variables to read from the aggregator file
- **config** (*Path*) – adios xml file for the files,
- **stream_name** (*str*) – corresponding stream name in adios.xml
- **lastN** (*int*) – number of last steps to keep from each adios file

- **batch** (*int*) – up to how many steps to read from each adios file at a time (call of *next()*)

next() → Dict[str, Union[numpy.array, str, int, float]]

Provide *lastN* steps from each aggregator.

Returns Dict[str, Union[np.array, int, float, str]] – values for the the variables whose names are used as keys

deepdrivemd.data.stream.enumerations

Classes

<i>DataStructure</i> (value)	An enumeration.
------------------------------	-----------------

class deepdrivemd.data.stream.enumerations.**DataStructure**(value)

An enumeration.

array = 1

scalar = 2

string = 3

deepdrivemd.data.utils

Data utility functions for handling HDF5 files.

Functions

<i>concatenate_virtual_h5</i> (input_file_names, ...)	Concatenate HDF5 files into a virtual HDF5 file.
<i>get_virtual_h5_file</i> (output_path, all_h5_files)	Create and return a virtual HDF5 file.
<i>parse_h5</i> (path, fields)	Helper function for accessing data fields in a HDF5 file.

deepdrivemd.data.utils.**concatenate_virtual_h5**(input_file_names: List[str], output_name: str, fields: Optional[List[str]] = None) → None

Concatenate HDF5 files into a virtual HDF5 file.

Concatenates a list input_file_names of HDF5 files containing the same format into a single virtual dataset.

Parameters

- **input_file_names** (List[str]) – List of HDF5 file names to concatenate.
- **output_name** (str) – Name of output virtual HDF5 file.
- **fields** (Optional[List[str]], default=None) – Which dataset fields to concatenate. Will concatenate all fields by default.

deepdrivemd.data.utils.**get_virtual_h5_file**(output_path: pathlib.Path, all_h5_files: List[str], last_n: int = 0, k_random_old: int = 0, virtual_name: str = 'virtual', node_local_path: Optional[pathlib.Path] = None) → Tuple[pathlib.Path, List[str]]

Create and return a virtual HDF5 file.

Create a virtual HDF5 file from the *last_n* files in *all_h5_files* and a random selection of *k_random_old*.

Parameters

- **output_path** (*Path*) – Directory to write virtual HDF5 file to.
- **all_h5_files** (*List[str]*) – List of HDF5 files to select from.
- **last_n** (*int, optional*) – Chooses the last *n* files in *all_h5_files* to concatenate into a virtual HDF5 file. Defaults to all the files.
- **k_random_old** (*int, default=0*) – Chooses *k* random files not in the *last_n* files to concatenate into the virtual HDF5 file. Defaults to choosing no random old files.
- **virtual_name** (*str, default="virtual"*) – The name of the virtual HDF5 file to be written e.g. `virtual_name == virtual` implies the file will be written to `output_path/virtual.h5`.
- **node_local_path** (*Optional[Path], default=None*) – An optional path to write the virtual file to that could be a node local storage. Will also copy all selected HDF5 files in *all_h5_files* to the same directory.

Returns

- *Path* – The path to the created virtual HDF5 file.
- *List[str]* – The selected HDF5 files from *last_n* and *k_random_old* used to make the virtual HDF5 file.

Raises `ValueError` – If *all_h5_files* is empty. If *:obj:last_n* is greater than `len(all_h5_files)`.

`deepdrivemd.data.utils.parse_h5(path: Union[str, pathlib.Path], fields: List[str]) → Dict[str, npt.ArrayLike]`

Helper function for accessing data fields in a HDF5 file.

Parameters

- **path** (*Union[Path, str]*) – Path to HDF5 file.
- **fields** (*List[str]*) – List of dataset field names inside of the HDF5 file.

Returns *Dict[str, npt.ArrayLike]* – A dictionary mapping each field name in *fields* to a numpy array containing the data from the associated HDF5 dataset.

deepdrivemd.deepdrivemd

Functions

`generate_task(cfg)`

Classes

PipelineManager(cfg)

```
class deepdrivemd.deepdrivemd.PipelineManager(cfg: deepdrivemd.config.ExperimentConfig)

    AGENT_STAGE_NAME = 'Agent'
    AGGREGATION_STAGE_NAME = 'Aggregating'
    MACHINE_LEARNING_STAGE_NAME = 'MachineLearning'
    MODEL_SELECTION_STAGE_NAME = 'ModelSelection'
    MOLECULAR_DYNAMICS_STAGE_NAME = 'MolecularDynamics'
    PIPELINE_NAME = 'DeepDriveMD'

    func_condition() → None
    func_on_false() → None
    func_on_true() → None

    generate_agent_stage() → radical.entk.stage.Stage
    generate_aggregating_stage() → radical.entk.stage.Stage
    generate_machine_learning_stage() → radical.entk.stage.Stage
    generate_model_selection_stage() → radical.entk.stage.Stage
    generate_molecular_dynamics_stage() → radical.entk.stage.Stage
    generate_pipelines() → List[radical.entk.pipeline.Pipeline]

deepdrivemd.deepdrivemd.generate_task(cfg: deepdrivemd.config.BaseStageConfig) →
    radical.entk.task.Task
```

deepdrivemd.deepdrivemd_stream

Functions

compute_number_of_nodes(cfg)

generate_task(cfg)

Classes

PipelineManager(cfg)

class deepdrivemd.deepdrivemd_stream.**PipelineManager**(*cfg*: deepdrivemd.config.StreamingExperimentConfig)

AGENT_PIPELINE_NAME = 'AgentPipeline'

AGENT_STAGE_NAME = 'Agent'

AGGREGATION_PIPELINE_NAME = 'AggregatingPipeline'

AGGREGATION_STAGE_NAME = 'Aggregating'

MACHINE_LEARNING_PIPELINE_NAME = 'MachineLearningPipeline'

MACHINE_LEARNING_STAGE_NAME = 'MachineLearning'

MOLECULAR_DYNAMICS_PIPELINE_NAME = 'MolecularDynamicsPipeline'

MOLECULAR_DYNAMICS_STAGE_NAME = 'MolecularDynamics'

generate_agent_stage() → radical.entk.stage.Stage

generate_aggregating_stage() → radical.entk.stage.Stage

generate_machine_learning_stage() → radical.entk.stage.Stage

generate_molecular_dynamics_stage() → radical.entk.stage.Stage

generate_pipelines() → List[radical.entk.pipeline.Pipeline]

deepdrivemd.deepdrivemd_stream.**compute_number_of_nodes**(*cfg*: deepdrivemd.config.StreamingExperimentConfig) → int

deepdrivemd.deepdrivemd_stream.**generate_task**(*cfg*: deepdrivemd.config.BaseStageConfig) → radical.entk.task.Task

deepdrivemd.models

Modules

deepdrivemd.models.aae

deepdrivemd.models.aae_stream

deepdrivemd.models.keras_cvae

deepdrivemd.models.keras_cvae_stream

deepdrivemd.models.aae

Modules

deepdrivemd.models.aae.config

deepdrivemd.models.aae.config

pydantic settings deepdrivemd.models.aae.config.**AAEModelConfig**

```
{
  "title": "AAEModelConfig",
  "description": "Base class for specific model configs to inherit.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'experiment_directory'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    "task_idx": {
      "title": "Task Idx",
      "default": 0,
      "env_names": "'task_idx'",
      "type": "integer"
    },
    "output_path": {
      "title": "Output Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'output_path'",
      "type": "string",
      "format": "path"
    },
    "node_local_path": {
      "title": "Node Local Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'node_local_path'",
      "type": "string",
      "format": "path"
    },
    "model_tag": {
      "title": "Model Tag",
```

(continues on next page)

(continued from previous page)

```

    "default": "set_by_deepdrivemd",
    "env_names": "{ 'model_tag' }",
    "type": "string"
  },
  "init_weights_path": {
    "title": "Init Weights Path",
    "env_names": "{ 'init_weights_path' }",
    "type": "string",
    "format": "path"
  },
  "last_n_h5_files": {
    "title": "Last N H5 Files",
    "default": 10,
    "env_names": "{ 'last_n_h5_files' }",
    "type": "integer"
  },
  "k_random_old_h5_files": {
    "title": "K Random Old H5 Files",
    "default": 0,
    "env_names": "{ 'k_random_old_h5_files' }",
    "type": "integer"
  },
  "dataset_name": {
    "title": "Dataset Name",
    "default": "point_cloud",
    "env_names": "{ 'dataset_name' }",
    "type": "string"
  },
  "rmsd_name": {
    "title": "Rmsd Name",
    "default": "rmsd",
    "env_names": "{ 'rmsd_name' }",
    "type": "string"
  },
  "fnc_name": {
    "title": "Fnc Name",
    "default": "fnc",
    "env_names": "{ 'fnc_name' }",
    "type": "string"
  },
  "num_points": {
    "title": "Num Points",
    "default": 3375,
    "env_names": "{ 'num_points' }",
    "type": "integer"
  },
  "num_features": {
    "title": "Num Features",
    "default": 0,
    "env_names": "{ 'num_features' }",
    "type": "integer"
  },

```

(continues on next page)

(continued from previous page)

```

"initial_epochs": {
  "title": "Initial Epochs",
  "default": 10,
  "env_names": "'initial_epochs'",
  "type": "integer"
},
"epochs": {
  "title": "Epochs",
  "default": 10,
  "env_names": "'epochs'",
  "type": "integer"
},
"batch_size": {
  "title": "Batch Size",
  "default": 32,
  "env_names": "'batch_size'",
  "type": "integer"
},
"optimizer_name": {
  "title": "Optimizer Name",
  "default": "Adam",
  "env_names": "'optimizer_name'",
  "type": "string"
},
"optimizer_lr": {
  "title": "Optimizer Lr",
  "default": 0.0001,
  "env_names": "'optimizer_lr'",
  "type": "number"
},
"latent_dim": {
  "title": "Latent Dim",
  "default": 64,
  "env_names": "'latent_dim'",
  "type": "integer"
},
"encoder_filters": {
  "title": "Encoder Filters",
  "default": [
    64,
    128,
    256,
    256,
    512
  ],
  "env_names": "'encoder_filters'",
  "type": "array",
  "items": {
    "type": "integer"
  }
},
"encoder_kernel_sizes": {

```

(continues on next page)

(continued from previous page)

```

    "title": "Encoder Kernel Sizes",
    "default": [
        5,
        5,
        3,
        1,
        1
    ],
    "env_names": "'encoder_kernel_sizes'",
    "type": "array",
    "items": {
        "type": "integer"
    }
},
"generator_filters": {
    "title": "Generator Filters",
    "default": [
        64,
        128,
        512,
        1024
    ],
    "env_names": "'generator_filters'",
    "type": "array",
    "items": {
        "type": "integer"
    }
},
"discriminator_filters": {
    "title": "Discriminator Filters",
    "default": [
        512,
        512,
        128,
        64
    ],
    "env_names": "'discriminator_filters'",
    "type": "array",
    "items": {
        "type": "integer"
    }
},
"encoder_relu_slope": {
    "title": "Encoder Relu Slope",
    "default": 0.0,
    "env_names": "'encoder_relu_slope'",
    "type": "number"
},
"generator_relu_slope": {
    "title": "Generator Relu Slope",
    "default": 0.0,
    "env_names": "'generator_relu_slope'",

```

(continues on next page)

(continued from previous page)

```

    "type": "number"
  },
  "discriminator_relu_slope": {
    "title": "Discriminator Relu Slope",
    "default": 0.0,
    "env_names": "'discriminator_relu_slope'",
    "type": "number"
  },
  "use_encoder_bias": {
    "title": "Use Encoder Bias",
    "default": true,
    "env_names": "'use_encoder_bias'",
    "type": "boolean"
  },
  "use_generator_bias": {
    "title": "Use Generator Bias",
    "default": true,
    "env_names": "'use_generator_bias'",
    "type": "boolean"
  },
  "use_discriminator_bias": {
    "title": "Use Discriminator Bias",
    "default": true,
    "env_names": "'use_discriminator_bias'",
    "type": "boolean"
  },
  "noise_mu": {
    "title": "Noise Mu",
    "default": 0.0,
    "env_names": "'noise_mu'",
    "type": "number"
  },
  "noise_std": {
    "title": "Noise Std",
    "default": 1.0,
    "env_names": "'noise_std'",
    "type": "number"
  },
  "lambda_rec": {
    "title": "Lambda Rec",
    "default": 0.5,
    "env_names": "'lambda_rec'",
    "type": "number"
  },
  "lambda_gp": {
    "title": "Lambda Gp",
    "default": 10,
    "env_names": "'lambda_gp'",
    "type": "number"
  },
  "embed_interval": {
    "title": "Embed Interval",

```

(continues on next page)

(continued from previous page)

```

        "default": 1,
        "env_names": "'embed_interval'",
        "type": "integer"
    },
    "tsne_interval": {
        "title": "Tsne Interval",
        "default": 5,
        "env_names": "'tsne_interval'",
        "type": "integer"
    },
    "sample_interval": {
        "title": "Sample Interval",
        "default": 20,
        "env_names": "'sample_interval'",
        "type": "integer"
    },
    "num_data_workers": {
        "title": "Num Data Workers",
        "default": 0,
        "env_names": "'num_data_workers'",
        "type": "integer"
    },
    "dataset_location": {
        "title": "Dataset Location",
        "default": "storage",
        "env_names": "'dataset_location'",
        "type": "string"
    },
    "wandb_project_name": {
        "title": "Wandb Project Name",
        "env_names": "'wandb_project_name'",
        "type": "string"
    }
}

```

Config

- **extra:** *str = allow*

Fields

- *batch_size* (*int*)
- *dataset_location* (*str*)
- *dataset_name* (*str*)
- *discriminator_filters* (*List[int]*)
- *discriminator_relu_slope* (*float*)
- *embed_interval* (*int*)
- *encoder_filters* (*List[int]*)
- *encoder_kernel_sizes* (*List[int]*)

- *encoder_relu_slope* (float)
- *epochs* (int)
- *fnc_name* (str)
- *generator_filters* (List[int])
- *generator_relu_slope* (float)
- *initial_epochs* (int)
- *k_random_old_h5_files* (int)
- *lambda_gp* (float)
- *lambda_rec* (float)
- *last_n_h5_files* (int)
- *latent_dim* (int)
- *noise_mu* (float)
- *noise_std* (float)
- *num_data_workers* (int)
- *num_features* (int)
- *num_points* (int)
- *optimizer_lr* (float)
- *optimizer_name* (str)
- *rmsd_name* (str)
- *sample_interval* (int)
- *tsne_interval* (int)
- *use_discriminator_bias* (bool)
- *use_encoder_bias* (bool)
- *use_generator_bias* (bool)
- *wandb_project_name* (Optional[str])

```
field batch_size:  int = 32
field dataset_location:  str = 'storage'
field dataset_name:  str = 'point_cloud'
field discriminator_filters:  List[int] = [512, 512, 128, 64]
field discriminator_relu_slope:  float = 0.0
field embed_interval:  int = 1
field encoder_filters:  List[int] = [64, 128, 256, 256, 512]
field encoder_kernel_sizes:  List[int] = [5, 5, 3, 1, 1]
field encoder_relu_slope:  float = 0.0
```

```

field epochs: int = 10
field fnc_name: str = 'fnc'
field generator_filters: List[int] = [64, 128, 512, 1024]
field generator_relu_slope: float = 0.0
field initial_epochs: int = 10
field k_random_old_h5_files: int = 0
field lambda_gp: float = 10
field lambda_rec: float = 0.5
field last_n_h5_files: int = 10
field latent_dim: int = 64
field noise_mu: float = 0.0
field noise_std: float = 1.0
field num_data_workers: int = 0
field num_features: int = 0
field num_points: int = 3375
field optimizer_lr: float = 0.0001
field optimizer_name: str = 'Adam'
field rmsd_name: str = 'rmsd'
field sample_interval: int = 20
field tsne_interval: int = 5
field use_discriminator_bias: bool = True
field use_encoder_bias: bool = True
field use_generator_bias: bool = True
field wandb_project_name: Optional[str] = PydanticUndefined

```

deepdrivemd.models.aae_stream

Modules

[*deepdrivemd.models.aae_stream.config*](#)

[*deepdrivemd.models.aae_stream.train*](#)

[*deepdrivemd.models.aae_stream.utils*](#)

deepdrivemd.models.aae_stream.config

Classes

Point3dAAEConfig(*args, **kwargs)

```
class deepdrivemd.models.aae_stream.config.Point3dAAEConfig(*args: Any, **kwargs: Any)
    adios_xml_agg
    adios_xml_agg_4ml
    agg_dir = PosixPath('.')
    batch_size = 32
    checkpoint_dir
    cms_transform = True
    decoder_affine_widths = [64, 128, 512, 1024]
    decoder_bias = True
    decoder_relu_slope = 0.0
    discriminator_affine_widths = [512, 128, 64]
    discriminator_bias = True
    discriminator_relu_slope = 0.0
    encoder_bias = True
    encoder_filters = [64, 128, 256, 256, 512]
    encoder_kernels = [5, 5, 3, 1, 1]
    encoder_relu_slope = 0.0
    epochs = 30
    experiment_directory
    init_weights = ''
    init_weights_path
    input_path = PosixPath('/p/gpfs1/yakushin/Outputs/305t/molecular_dynamics_runs/
stage0000/task0000/0/trajectory.bp')
    lambda_gp = 10.0
    lambda_rec = 0.5
    latent_dim = 16
```

```
max_loss = 10000
max_steps = 8000
min_step_increment = 5000
model = 'aae'
model_tag = 'aae'
node_local_path = PosixPath('/tmp')
noise_mu = 0.0
noise_std = 1.0
num_agg = 12
num_data_workers = 16
num_features = 0
num_points = 200
output_path = PosixPath('TODO')
prefetch_factor = 2
published_model_dir
read_batch = 10000
reinit = False
resume_checkpoint = None
scalar_dset_names = []
scalar_requires_grad = False
seed = 333
shuffle = True
split_pct = 0.8
stage_idx = 0
task_idx = 0
timeout1 = 30
timeout2 = 10
use_model_checkpoint = True
```

deepdrivemd.models.aae_stream.train**Functions**

build_model(cfg)

main(cfg)

next_input(cfg, streams) Read the next batch of contact maps from aggregated files.

train(train_loader, model, disc_optimizer, ...)

train_model(model, ae_optimizer, ...)

validate(valid_loader, model, device, cfg)

wait_for_input(cfg) Wait for the expected number of sufficiently large agg.bp files to be produced.

```
deepdrivemd.models.aae_stream.train.build_model(cfg: deep-  
drivemd.models.aae_stream.config.Point3dAAEConfig)
```

```
deepdrivemd.models.aae_stream.train.main(cfg:  
deepdrivemd.models.aae_stream.config.Point3dAAEConfig)
```

```
deepdrivemd.models.aae_stream.train.next_input(cfg: deep-  
drivemd.models.aae_stream.config.Point3dAAEConfig,  
streams:  
deepdrivemd.data.stream.aggregator_reader.Streams)  
→ Tuple[numpy.ndarray, numpy.ndarray]
```

Read the next batch of contact maps from aggregated files.

Returns *Tuple[np.ndarray, np.ndarray]* – Training and validation sets.

```
deepdrivemd.models.aae_stream.train.train(train_loader, model:  
mdlearn.nn.models.aae.point_3d_aae.AAE3d, disc_optimizer,  
ae_optimizer, device, cfg:  
deepdrivemd.models.aae_stream.config.Point3dAAEConfig)
```

```
deepdrivemd.models.aae_stream.train.train_model(model, ae_optimizer, disc_optimizer, train_loader,  
valid_loader, device, cfg: deep-  
drivemd.models.aae_stream.config.Point3dAAEConfig)
```

```
deepdrivemd.models.aae_stream.train.validate(valid_loader, model:  
mdlearn.nn.models.aae.point_3d_aae.AAE3d, device, cfg:  
deep-  
drivemd.models.aae_stream.config.Point3dAAEConfig)
```

```
deepdrivemd.models.aae_stream.train.wait_for_input(cfg: deep-  
drivemd.models.aae_stream.config.Point3dAAEConfig)  
→ List[str]
```

Wait for the expected number of sufficiently large agg.bp files to be produced.

Returns *List[str]* – List of paths to aggregated files.

deepdrivemd.models.aae_stream.utils**Functions**

`read_adios_file(input_path)`

Classes

`CenterOfMassTransform(data)`

`PointCloudDatasetInMemory(*args, **kwargs)` PyTorch Dataset class to load point cloud data.

class deepdrivemd.models.aae_stream.utils.**CenterOfMassTransform**(data: *numpy.ndarray*)

__init__(data: *numpy.ndarray*) → None

Computes center of mass transformation :Parameters: **data** (*np.ndarray*) – Dataset of positions with shape (num_examples, 3, num_points).

transform(x: *numpy.ndarray*) → *numpy.ndarray*

Normalize example by bias and scale factors :Parameters: **x** (*np.ndarray*) – Data to transform shape (3, num_points). Modifies **x**.

Returns *np.ndarray* – The transformed data

Raises **ValueError** – If NaN encountered in input

class deepdrivemd.models.aae_stream.utils.**PointCloudDatasetInMemory**(*args: Any, **kwargs: Any)

PyTorch Dataset class to load point cloud data. Optionally, uses HDF5 files to only read into memory what is necessary for one batch.

__init__(data: *numpy.ndarray*, scalars: *Dict[str, numpy.ndarray]* = {}, cms_transform: *bool* = False, scalar_requires_grad: *bool* = False)
Parameters

- **data** (*np.ndarray*) – Dataset of positions with shape (num_examples, 3, num_points)
- **scalars** (*Dict[str, np.ndarray]*, *default={}*) – Dictionary of scalar arrays. For instance, the root mean squared deviation (RMSD) for each feature vector can be passed via {"rmsd": *np.array(...)*}. The dimension of each scalar array should match the number of input feature vectors N.
- **cms_transform** (*bool*) – If True, subtract center of mass from batch and shift and scale batch by the full dataset statistics.
- **scalar_requires_grad** (*bool*) – Sets requires_grad torch.Tensor parameter for scalars specified by scalar_dset_names. Set to True, to use scalars for learning. If scalars are only required for plotting, then set it as False.

deepdrivemd.models.aae_stream.utils.**read_adios_file**(input_path: *pathlib.Path*)

deepdrivemd.models.keras_cvae**Modules**

deepdrivemd.models.keras_cvae.config

deepdrivemd.models.keras_cvae.utils

deepdrivemd.models.keras_cvae.config**pydantic settings** `deepdrivemd.models.keras_cvae.config.KerasCVAEModelConfig`

```
{
  "title": "KerasCVAEModelConfig",
  "description": "Base class for specific model configs to inherit.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'experiment_directory'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    "task_idx": {
      "title": "Task Idx",
      "default": 0,
      "env_names": "'task_idx'",
      "type": "integer"
    },
    "output_path": {
      "title": "Output Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'output_path'",
      "type": "string",
      "format": "path"
    },
    "node_local_path": {
      "title": "Node Local Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'node_local_path'",
      "type": "string",
      "format": "path"
    }
  },
}
```

(continues on next page)

(continued from previous page)

```

"model_tag": {
  "title": "Model Tag",
  "default": "set_by_deepdrivemd",
  "env_names": "{}model_tag{}",
  "type": "string"
},
"init_weights_path": {
  "title": "Init Weights Path",
  "env_names": "{}init_weights_path{}",
  "type": "string",
  "format": "path"
},
"last_n_h5_files": {
  "title": "Last N H5 Files",
  "default": 10,
  "env_names": "{}last_n_h5_files{}",
  "type": "integer"
},
"k_random_old_h5_files": {
  "title": "K Random Old H5 Files",
  "default": 0,
  "env_names": "{}k_random_old_h5_files{}",
  "type": "integer"
},
"dataset_name": {
  "title": "Dataset Name",
  "default": "contact_map",
  "env_names": "{}dataset_name{}",
  "type": "string"
},
"initial_shape": {
  "title": "Initial Shape",
  "default": [
    28,
    28
  ],
  "env_names": "{}initial_shape{}",
  "type": "array",
  "minItems": 2,
  "maxItems": 2,
  "items": [
    {
      "type": "integer"
    },
    {
      "type": "integer"
    }
  ]
},
"final_shape": {
  "title": "Final Shape",
  "default": [

```

(continues on next page)

(continued from previous page)

```

        28,
        28,
        1
    ],
    "env_names": "{final_shape}",
    "type": "array",
    "minItems": 3,
    "maxItems": 3,
    "items": [
        {
            "type": "integer"
        },
        {
            "type": "integer"
        },
        {
            "type": "integer"
        }
    ]
},
"initial_epochs": {
    "title": "Initial Epochs",
    "default": 10,
    "env_names": "{initial_epochs}",
    "type": "integer"
},
"epochs": {
    "title": "Epochs",
    "default": 10,
    "env_names": "{epochs}",
    "type": "integer"
},
"batch_size": {
    "title": "Batch Size",
    "default": 32,
    "env_names": "{batch_size}",
    "type": "integer"
},
"split_pct": {
    "title": "Split Pct",
    "default": 0.8,
    "env_names": "{split_pct}",
    "type": "number"
},
"shuffle": {
    "title": "Shuffle",
    "default": true,
    "env_names": "{shuffle}",
    "type": "boolean"
},
"latent_dim": {
    "title": "Latent Dim",

```

(continues on next page)

(continued from previous page)

```

    "default": 10,
    "env_names": "'latent_dim'",
    "type": "integer"
  },
  "conv_layers": {
    "title": "Conv Layers",
    "default": 4,
    "env_names": "'conv_layers'",
    "type": "integer"
  },
  "conv_filters": {
    "title": "Conv Filters",
    "default": [
      64,
      64,
      64,
      64
    ],
    "env_names": "'conv_filters'",
    "type": "array",
    "items": {
      "type": "integer"
    }
  },
  "conv_filter_shapes": {
    "title": "Conv Filter Shapes",
    "default": [
      [
        3,
        3
      ],
      [
        3,
        3
      ],
      [
        3,
        3
      ],
      [
        3,
        3
      ]
    ],
    "env_names": "'conv_filter_shapes'",
    "type": "array",
    "items": {
      "type": "array",
      "minItems": 2,
      "maxItems": 2,
      "items": [
        {

```

(continues on next page)

(continued from previous page)

```

        "type": "integer"
    },
    {
        "type": "integer"
    }
]
},
"conv_strides": {
    "title": "Conv Strides",
    "default": [
        [
            1,
            1
        ],
        [
            2,
            2
        ],
        [
            1,
            1
        ],
        [
            1,
            1
        ]
    ],
    "env_names": "'conv_strides'",
    "type": "array",
    "items": {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
            {
                "type": "integer"
            },
            {
                "type": "integer"
            }
        ]
    }
},
"dense_layers": {
    "title": "Dense Layers",
    "default": 1,
    "env_names": "'dense_layers'",
    "type": "integer"
},
"dense_neurons": {
    "title": "Dense Neurons",

```

(continues on next page)

(continued from previous page)

```

    "default": [
        128
    ],
    "env_names": "'dense_neurons'",
    "type": "array",
    "items": {
        "type": "integer"
    }
},
"dense_dropouts": {
    "title": "Dense Dropouts",
    "default": [
        0.25
    ],
    "env_names": "'dense_dropouts'",
    "type": "array",
    "items": {
        "type": "number"
    }
},
"use_model_checkpoint": {
    "title": "Use Model Checkpoint",
    "default": false,
    "env_names": "'use_model_checkpoint'",
    "type": "boolean"
}
}
}

```

Config

- **extra:** *str = allow*

Fields

- *batch_size* (*int*)
- *conv_filter_shapes* (*List[Tuple[int, int]]*)
- *conv_filters* (*List[int]*)
- *conv_layers* (*int*)
- *conv_strides* (*List[Tuple[int, int]]*)
- *dataset_name* (*str*)
- *dense_dropouts* (*List[float]*)
- *dense_layers* (*int*)
- *dense_neurons* (*List[int]*)
- *epochs* (*int*)
- *final_shape* (*Tuple[int, int, int]*)
- *initial_epochs* (*int*)
- *initial_shape* (*Tuple[int, int]*)

- *k_random_old_h5_files* (*int*)
- *last_n_h5_files* (*int*)
- *latent_dim* (*int*)
- *shuffle* (*bool*)
- *split_pct* (*float*)

```
field batch_size: int = 32
field conv_filter_shapes: List[Tuple[int, int]] = [(3, 3), (3, 3), (3, 3), (3, 3)]
field conv_filters: List[int] = [64, 64, 64, 64]
field conv_layers: int = 4
field conv_strides: List[Tuple[int, int]] = [(1, 1), (2, 2), (1, 1), (1, 1)]
field dataset_name: str = 'contact_map'
field dense_dropouts: List[float] = [0.25]
field dense_layers: int = 1
field dense_neurons: List[int] = [128]
field epochs: int = 10
field final_shape: Tuple[int, int, int] = (28, 28, 1)
field initial_epochs: int = 10
field initial_shape: Tuple[int, int] = (28, 28)
field k_random_old_h5_files: int = 0
field last_n_h5_files: int = 10
field latent_dim: int = 10
field shuffle: bool = True
field split_pct: float = 0.8
```

deepdrivemd.models.keras_cvae.utils

Functions

<i>sparse_to_dense</i> (h5_file, dataset_name, ...)	Convert sparse COO formatted contact maps to dense.
---	---

deepdrivemd.models.keras_cvae.utils.**sparse_to_dense**(h5_file: Union[str, pathlib.Path], dataset_name: str, initial_shape: Tuple[int, int], final_shape: Union[Tuple[int, int, int], Tuple[int, int]]) → npt.ArrayLike

Convert sparse COO formatted contact maps to dense.

Parameters

- **h5_file** (*PathLike*) – The HDF5 file containing contact maps.
- **dataset_name** (*str*) – The dataset name containing the contact map indices.
- **initial_shape** (*Tuple[int, int]*) – The shape of the contact map saved in the HDF5 file.
- **final_shape** (*Union[Tuple[int, int, int], Tuple[int, int]]*) – The final shape of the contact map incase adding an extra dimension is necessary e.g. (D, D, 1) where D is the number of residues or the cropping shape.

Returns *npt.ArrayLike* – The output array of contact maps of shape (N, D, D) or (N, D, D, 1) depending on **final_shape** where N is the number of contact maps in the HDF5 file.

deepdrivemd.models.keras_cvae_stream**Modules**

`deepdrivemd.models.keras_cvae_stream.config`

deepdrivemd.models.keras_cvae_stream.config

pydantic settings `deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig`

```
{
  "title": "KerasCVAEModelConfig",
  "description": "Base class for specific model configs to inherit.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'experiment_directory'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    "task_idx": {
      "title": "Task Idx",
      "default": 0,
      "env_names": "'task_idx'",
      "type": "integer"
    },
    "output_path": {
      "title": "Output Path",
      "default": "set_by_deepdrivemd",

```

(continues on next page)

(continued from previous page)

```

    "env_names": "{ 'output_path' }",
    "type": "string",
    "format": "path"
  },
  "node_local_path": {
    "title": "Node Local Path",
    "default": "set_by_deepdrivemd",
    "env_names": "{ 'node_local_path' }",
    "type": "string",
    "format": "path"
  },
  "model_tag": {
    "title": "Model Tag",
    "default": "set_by_deepdrivemd",
    "env_names": "{ 'model_tag' }",
    "type": "string"
  },
  "init_weights_path": {
    "title": "Init Weights Path",
    "env_names": "{ 'init_weights_path' }",
    "type": "string",
    "format": "path"
  },
  "final_shape": {
    "title": "Final Shape",
    "default": [
      28,
      28,
      1
    ],
    "env_names": "{ 'final_shape' }",
    "type": "array",
    "items": {
      "type": "integer"
    }
  },
  "epochs": {
    "title": "Epochs",
    "default": 10,
    "env_names": "{ 'epochs' }",
    "type": "integer"
  },
  "batch_size": {
    "title": "Batch Size",
    "default": 32,
    "env_names": "{ 'batch_size' }",
    "type": "integer"
  },
  "split_pct": {
    "title": "Split Pct",
    "default": 0.8,
    "env_names": "{ 'split_pct' }",

```

(continues on next page)

(continued from previous page)

```

    "type": "number"
  },
  "shuffle": {
    "title": "Shuffle",
    "default": true,
    "env_names": "'shuffle'",
    "type": "boolean"
  },
  "latent_dim": {
    "title": "Latent Dim",
    "default": 10,
    "env_names": "'latent_dim'",
    "type": "integer"
  },
  "conv_layers": {
    "title": "Conv Layers",
    "default": 4,
    "env_names": "'conv_layers'",
    "type": "integer"
  },
  "conv_filters": {
    "title": "Conv Filters",
    "default": [
      64,
      64,
      64,
      64
    ],
    "env_names": "'conv_filters'",
    "type": "array",
    "items": {
      "type": "integer"
    }
  },
  "conv_filter_shapes": {
    "title": "Conv Filter Shapes",
    "default": [
      [
        3,
        3
      ],
      [
        3,
        3
      ],
      [
        3,
        3
      ],
      [
        3,
        3
      ]
    ]
  }
}

```

(continues on next page)

(continued from previous page)

```

    ]
  ],
  "env_names": "{ 'conv_filter_shapes' }",
  "type": "array",
  "items": {
    "type": "array",
    "minItems": 2,
    "maxItems": 2,
    "items": [
      {
        "type": "integer"
      },
      {
        "type": "integer"
      }
    ]
  }
},
"conv_strides": {
  "title": "Conv Strides",
  "default": [
    [
      1,
      1
    ],
    [
      2,
      2
    ],
    [
      1,
      1
    ],
    [
      1,
      1
    ]
  ],
  "env_names": "{ 'conv_strides' }",
  "type": "array",
  "items": {
    "type": "array",
    "minItems": 2,
    "maxItems": 2,
    "items": [
      {
        "type": "integer"
      },
      {
        "type": "integer"
      }
    ]
  }
}

```

(continues on next page)

(continued from previous page)

```

    }
  },
  "dense_layers": {
    "title": "Dense Layers",
    "default": 1,
    "env_names": "'dense_layers'",
    "type": "integer"
  },
  "dense_neurons": {
    "title": "Dense Neurons",
    "default": [
      128
    ],
    "env_names": "'dense_neurons'",
    "type": "array",
    "items": {
      "type": "integer"
    }
  },
  "dense_dropouts": {
    "title": "Dense Dropouts",
    "default": [
      0.25
    ],
    "env_names": "'dense_dropouts'",
    "type": "array",
    "items": {
      "type": "number"
    }
  },
  "min_step_increment": {
    "title": "Min Step Increment",
    "default": 5000,
    "env_names": "'min_step_increment'",
    "type": "integer"
  },
  "max_steps": {
    "title": "Max Steps",
    "default": 8000,
    "env_names": "'max_steps'",
    "type": "integer"
  },
  "max_loss": {
    "title": "Max Loss",
    "default": 10000,
    "env_names": "'max_loss'",
    "type": "integer"
  },
  "num_agg": {
    "title": "Num Agg",
    "default": 12,
    "env_names": "'num_agg'",

```

(continues on next page)

(continued from previous page)

```

    "type": "integer"
  },
  "timeout1": {
    "title": "Timeout1",
    "default": 30,
    "env_names": "'timeout1'",
    "type": "integer"
  },
  "timeout2": {
    "title": "Timeout2",
    "default": 10,
    "env_names": "'timeout2'",
    "type": "integer"
  },
  "agg_dir": {
    "title": "Agg Dir",
    "default": ".",
    "env_names": "'agg_dir'",
    "type": "string",
    "format": "path"
  },
  "published_model_dir": {
    "title": "Published Model Dir",
    "env_names": "'published_model_dir'",
    "type": "string",
    "format": "path"
  },
  "checkpoint_dir": {
    "title": "Checkpoint Dir",
    "env_names": "'checkpoint_dir'",
    "type": "string",
    "format": "path"
  },
  "adios_xml_agg": {
    "title": "Adios Xml Agg",
    "env_names": "'adios_xml_agg'",
    "type": "string",
    "format": "path"
  },
  "reinit": {
    "title": "Reinit",
    "default": true,
    "env_names": "'reinit'",
    "type": "boolean"
  },
  "read_batch": {
    "title": "Read Batch",
    "default": 10000,
    "env_names": "'read_batch'",
    "type": "integer"
  },
  "model": {

```

(continues on next page)

(continued from previous page)

```

        "title": "Model",
        "default": "cvae",
        "env_names": "'model'",
        "type": "string"
    },
    "use_model_checkpoint": {
        "title": "Use Model Checkpoint",
        "default": true,
        "env_names": "'use_model_checkpoint'",
        "type": "boolean"
    }
},
"required": [
    "published_model_dir",
    "checkpoint_dir",
    "adios_xml_agg"
]
}

```

Config

- **extra:** *str = allow*

Fields

- *adios_xml_agg* (*pathlib.Path*)
- *agg_dir* (*pathlib.Path*)
- *batch_size* (*int*)
- *checkpoint_dir* (*pathlib.Path*)
- *conv_filter_shapes* (*List[Tuple[int, int]]*)
- *conv_filters* (*List[int]*)
- *conv_layers* (*int*)
- *conv_strides* (*List[Tuple[int, int]]*)
- *dense_dropouts* (*List[float]*)
- *dense_layers* (*int*)
- *dense_neurons* (*List[int]*)
- *epochs* (*int*)
- *final_shape* (*Tuple[int, ...]*)
- *latent_dim* (*int*)
- *max_loss* (*int*)
- *max_steps* (*int*)
- *min_step_increment* (*int*)
- *model* (*str*)
- *num_agg* (*int*)

- *published_model_dir* (*pathlib.Path*)
- *read_batch* (*int*)
- *reinit* (*bool*)
- *shuffle* (*bool*)
- *split_pct* (*float*)
- *timeout1* (*int*)
- *timeout2* (*int*)

```
field adios_xml_agg:  pathlib.Path [Required]
field agg_dir:  pathlib.Path = PosixPath('.')
field batch_size:  int = 32
field checkpoint_dir:  pathlib.Path [Required]
field conv_filter_shapes:  List[Tuple[int, int]] = [(3, 3), (3, 3), (3, 3), (3, 3)]
field conv_filters:  List[int] = [64, 64, 64, 64]
field conv_layers:  int = 4
field conv_strides:  List[Tuple[int, int]] = [(1, 1), (2, 2), (1, 1), (1, 1)]
field dense_dropouts:  List[float] = [0.25]
field dense_layers:  int = 1
field dense_neurons:  List[int] = [128]
field epochs:  int = 10
field final_shape:  Tuple[int, ...] = (28, 28, 1)
field latent_dim:  int = 10
field max_loss:  int = 10000
field max_steps:  int = 8000
field min_step_increment:  int = 5000
field model:  str = 'cvae'
field num_agg:  int = 12
field published_model_dir:  pathlib.Path [Required]
field read_batch:  int = 10000
field reinit:  bool = True
field shuffle:  bool = True
field split_pct:  float = 0.8
field timeout1:  int = 30
field timeout2:  int = 10
```

deepdrivemd.selection**Modules***deepdrivemd.selection.latest***deepdrivemd.selection.latest****Modules***deepdrivemd.selection.latest.config**deepdrivemd.selection.latest.select_model***deepdrivemd.selection.latest.config****pydantic settings** `deepdrivemd.selection.latest.config.LatestCheckpointConfig`

Config for selecting the latest model checkpoint.

```
{
  "title": "LatestCheckpointConfig",
  "description": "Config for selecting the latest model checkpoint.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'experiment_directory'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    "task_idx": {
      "title": "Task Idx",
      "default": 0,
      "env_names": "'task_idx'",
      "type": "integer"
    },
    "output_path": {
      "title": "Output Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'output_path'"
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

        "type": "string",
        "format": "path"
    },
    "node_local_path": {
        "title": "Node Local Path",
        "default": "set_by_deepdrivemd",
        "env_names": "'node_local_path'",
        "type": "string",
        "format": "path"
    },
    "retrain_freq": {
        "title": "Retrain Freq",
        "default": 1,
        "env_names": "'retrain_freq'",
        "type": "integer"
    },
    "checkpoint_dir": {
        "title": "Checkpoint Dir",
        "default": "checkpoint",
        "env_names": "'checkpoint_dir'",
        "type": "string"
    },
    "checkpoint_suffix": {
        "title": "Checkpoint Suffix",
        "default": ".pt",
        "env_names": "'checkpoint_suffix'",
        "type": "string"
    }
}

```

Config

- **extra:** *str = allow*

Fields

- *checkpoint_dir (str)*
- *checkpoint_suffix (str)*
- *retrain_freq (int)*

```
field checkpoint_dir: str = 'checkpoint'
```

```
field checkpoint_suffix: str = '.pt'
```

```
field retrain_freq: int = 1
```

deepdrivemd.selection.latest.select_model**Functions**

<code>get_model_path([stage_idx, task_idx, api, ...])</code>	Get the current best model.
<code>latest_checkpoint(api[, checkpoint_dir, ...])</code>	Select latest PyTorch model checkpoint.
<code>latest_model_checkpoint(cfg)</code>	Select the latest model checkpoint and write path to JSON.

`deepdrivemd.selection.latest.select_model.get_model_path(stage_idx: int = -1, task_idx: int = 0, api: Optional[deepdrivemd.data.api.DeepDriveMD_API] = None, experiment_dir: Optional[Union[str, pathlib.Path]] = None) → Optional[Tuple[pathlib.Path, pathlib.Path]]`

Get the current best model.

Should be imported by other stages to retrieve the best model path.

Parameters

- **api** (*DeepDriveMD_API*, optional) – API to DeepDriveMD to access the machine learning model path.
- **experiment_dir** (*Union[str, Path]*, optional) – Experiment directory to initialize DeepDriveMD_API.

Returns

- *None* – If model selection has not run before.
- **model_config** (*Path*, optional) – Path to the most recent model YAML configuration file selected by the model selection stage. Contains hyperparameters.
- **model_checkpoint** (*Path*, optional) – Path to the most recent model weights selected by the model selection stage.

Raises **ValueError** – If both `api` and `experiment_dir` are *None*.

`deepdrivemd.selection.latest.select_model.latest_checkpoint(api: deepdrivemd.data.api.DeepDriveMD_API, checkpoint_dir: str = 'checkpoint', checkpoint_suffix: str = '.pt') → pathlib.Path`

Select latest PyTorch model checkpoint.

Assuming the model outputs a `checkpoint_dir` directory with `checkpoint_suffix` checkpoint files with the form `XXX_<epoch-index>_YYY_ZZZ...<checkpoint_suffix>`, return the path to the latest training epoch model checkpoint.

Parameters

- **api** (*DeepDriveMD_API*) – API to DeepDriveMD to access the machine learning model path.
- **checkpoint_dir** (*str*, *default="checkpoint"*) – Name of the checkpoint directory inside the model path. Note, if checkpoint files are stored in the top level directory, set `checkpoint_dir=""`.

- **checkpoint_suffix** (*str*, *default*=".pt") – The file extension for checkpoint files (.pt, .h5, etc).

Returns *Path* – Path to the latest model checkpoint file.

deepdrivemd.selection.latest.select_model.latest_model_checkpoint(*cfg*: deepdrivemd.selection.latest.config.LatestCheckpointConfig) → None

Select the latest model checkpoint and write path to JSON.

Find the latest model checkpoint written by the machine learning stage and write the path into a JSON file to be consumed by the agent stage.

Parameters *cfg* (*LatestCheckpointConfig*) – pydantic YAML configuration for model selection task.

deepdrivemd.sim

Modules

deepdrivemd.sim.openmm

deepdrivemd.sim.openmm_stream

deepdrivemd.sim.openmm

Modules

deepdrivemd.sim.openmm.config

deepdrivemd.sim.openmm.config

pydantic settings deepdrivemd.sim.openmm.config.OpenMMConfig

```
{
  "title": "OpenMMConfig",
  "description": "Auto-generates configuration file for MD tasks.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'{ 'experiment_directory' }'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'{ 'stage_idx' }'",
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

    "type": "integer"
  },
  "task_idx": {
    "title": "Task Idx",
    "default": 0,
    "env_names": "'task_idx'",
    "type": "integer"
  },
  "output_path": {
    "title": "Output Path",
    "default": "set_by_deepdrivemd",
    "env_names": "'output_path'",
    "type": "string",
    "format": "path"
  },
  "node_local_path": {
    "title": "Node Local Path",
    "default": "set_by_deepdrivemd",
    "env_names": "'node_local_path'",
    "type": "string",
    "format": "path"
  },
  "pdb_file": {
    "title": "Pdb File",
    "default": "set_by_deepdrivemd",
    "env_names": "'pdb_file'",
    "type": "string",
    "format": "path"
  },
  "initial_pdb_dir": {
    "title": "Initial Pdb Dir",
    "env_names": "'initial_pdb_dir'",
    "type": "string",
    "format": "path"
  },
  "solvent_type": {
    "default": "implicit",
    "env_names": "'solvent_type'",
    "allOf": [
      {
        "$ref": "#/definitions/MDSolvent"
      }
    ]
  },
  "top_suffix": {
    "title": "Top Suffix",
    "default": ".top",
    "env_names": "'top_suffix'",
    "type": "string"
  },
  "simulation_length_ns": {
    "title": "Simulation Length Ns",

```

(continues on next page)

(continued from previous page)

```

    "default": 10,
    "env_names": "{simulation_length_ns}",
    "type": "number"
  },
  "report_interval_ps": {
    "title": "Report Interval Ps",
    "default": 50,
    "env_names": "{report_interval_ps}",
    "type": "number"
  },
  "dt_ps": {
    "title": "Dt Ps",
    "default": 0.002,
    "env_names": "{dt_ps}",
    "type": "number"
  },
  "temperature_kelvin": {
    "title": "Temperature Kelvin",
    "default": 310.0,
    "env_names": "{temperature_kelvin}",
    "type": "number"
  },
  "heat_bath_friction_coef": {
    "title": "Heat Bath Friction Coef",
    "default": 1.0,
    "env_names": "{heat_bath_friction_coef}",
    "type": "number"
  },
  "wrap": {
    "title": "Wrap",
    "default": false,
    "env_names": "{wrap}",
    "type": "boolean"
  },
  "reference_pdb_file": {
    "title": "Reference Pdb File",
    "env_names": "{reference_pdb_file}",
    "type": "string",
    "format": "path"
  },
  "openmm_selection": {
    "title": "Openmm Selection",
    "default": [
      "CA"
    ],
    "env_names": "{openmm_selection}",
    "type": "array",
    "items": {
      "type": "string"
    }
  },
  "mda_selection": {

```

(continues on next page)

(continued from previous page)

```

        "title": "Mda Selection",
        "default": "protein and name CA",
        "env_names": "{ 'mda_selection' }",
        "type": "string"
    },
    "threshold": {
        "title": "Threshold",
        "default": 8.0,
        "env_names": "{ 'threshold' }",
        "type": "number"
    },
    "contact_map": {
        "title": "Contact Map",
        "default": true,
        "env_names": "{ 'contact_map' }",
        "type": "boolean"
    },
    "point_cloud": {
        "title": "Point Cloud",
        "default": true,
        "env_names": "{ 'point_cloud' }",
        "type": "boolean"
    },
    "fraction_of_contacts": {
        "title": "Fraction Of Contacts",
        "default": true,
        "env_names": "{ 'fraction_of_contacts' }",
        "type": "boolean"
    },
    "in_memory": {
        "title": "In Memory",
        "default": true,
        "env_names": "{ 'in_memory' }",
        "type": "boolean"
    }
},
"required": [
    "initial_pdb_dir"
],
"definitions": {
    "MDSolvent": {
        "title": "MDSolvent",
        "description": "An enumeration.",
        "enum": [
            "implicit",
            "explicit"
        ],
        "type": "string"
    }
}
}

```

Config

- **extra:** *str = allow*

Fields

- *contact_map* (*bool*)
- *dt_ps* (*float*)
- *fraction_of_contacts* (*bool*)
- *heat_bath_friction_coef* (*float*)
- *in_memory* (*bool*)
- *mda_selection* (*str*)
- *openmm_selection* (*List[str]*)
- *point_cloud* (*bool*)
- *reference_pdb_file* (*Optional[pathlib.Path]*)
- *report_interval_ps* (*float*)
- *simulation_length_ns* (*float*)
- *solvent_type* (*deepdrivemd.sim.openmm.config.OpenMMConfig.MDSolvent*)
- *temperature_kelvin* (*float*)
- *threshold* (*float*)
- *top_suffix* (*Optional[str]*)
- *wrap* (*bool*)

field contact_map: *bool = True*

Validated by

- *explicit_solvent_requires_top_suffix*

field dt_ps: *float = 0.002*

Validated by

- *explicit_solvent_requires_top_suffix*

field fraction_of_contacts: *bool = True*

Validated by

- *explicit_solvent_requires_top_suffix*

field heat_bath_friction_coef: *float = 1.0*

Validated by

- *explicit_solvent_requires_top_suffix*

field in_memory: *bool = True*

Validated by

- *explicit_solvent_requires_top_suffix*

field mda_selection: str = 'protein and name CA'

Validated by

- *explicit_solvent_requires_top_suffix*

field openmm_selection: List[str] = ['CA']

Validated by

- *explicit_solvent_requires_top_suffix*

field point_cloud: bool = True

Validated by

- *explicit_solvent_requires_top_suffix*

field reference_pdb_file: Optional[pathlib.Path] = PydanticUndefined

Validated by

- *explicit_solvent_requires_top_suffix*

field report_interval_ps: float = 50

Validated by

- *explicit_solvent_requires_top_suffix*

field simulation_length_ns: float = 10

Validated by

- *explicit_solvent_requires_top_suffix*

field solvent_type: *deepdrivemd.sim.openmm.config.OpenMMConfig.MDSolvent* = MDSolvent.implicit

Validated by

- *explicit_solvent_requires_top_suffix*

field temperature_kelvin: float = 310.0

Validated by

- *explicit_solvent_requires_top_suffix*

field threshold: float = 8.0

Validated by

- *explicit_solvent_requires_top_suffix*

field top_suffix: Optional[str] = '.top'

Validated by

- *explicit_solvent_requires_top_suffix*

field wrap: bool = False

Validated by

- *explicit_solvent_requires_top_suffix*

```
class MDSolvent(value)
    An enumeration.

    explicit = 'explicit'

    implicit = 'implicit'

    validator explicit_solvent_requires_top_suffix » all fields
```

deepdrivemd.sim.openmm_stream

Modules

deepdrivemd.sim.openmm_stream.config

deepdrivemd.sim.openmm_stream.config

pydantic settings deepdrivemd.sim.openmm_stream.config.OpenMMConfig

```
{
  "title": "OpenMMConfig",
  "description": "Auto-generates configuration file for MD tasks.",
  "type": "object",
  "properties": {
    "experiment_directory": {
      "title": "Experiment Directory",
      "default": "set_by_deepdrivemd",
      "env_names": "'experiment_directory'",
      "type": "string",
      "format": "path"
    },
    "stage_idx": {
      "title": "Stage Idx",
      "default": 0,
      "env_names": "'stage_idx'",
      "type": "integer"
    },
    "task_idx": {
      "title": "Task Idx",
      "default": 0,
      "env_names": "'task_idx'",
      "type": "integer"
    },
    "output_path": {
      "title": "Output Path",
      "default": "set_by_deepdrivemd",
      "env_names": "'output_path'",
      "type": "string",
      "format": "path"
    }
  },
}
```

(continues on next page)

(continued from previous page)

```

"node_local_path": {
  "title": "Node Local Path",
  "default": "set_by_deepdrivemd",
  "env_names": "${'node_local_path'}",
  "type": "string",
  "format": "path"
},
"pdb_file": {
  "title": "Pdb File",
  "default": "set_by_deepdrivemd",
  "env_names": "${'pdb_file'}",
  "type": "string",
  "format": "path"
},
"initial_pdb_dir": {
  "title": "Initial Pdb Dir",
  "default": ".",
  "env_names": "${'initial_pdb_dir'}",
  "type": "string",
  "format": "path"
},
"solvent_type": {
  "default": "implicit",
  "env_names": "${'solvent_type'}",
  "allOf": [
    {
      "$ref": "#/definitions/MDSolvent"
    }
  ]
},
"top_suffix": {
  "title": "Top Suffix",
  "default": ".top",
  "env_names": "${'top_suffix'}",
  "type": "string"
},
"simulation_length_ns": {
  "title": "Simulation Length Ns",
  "default": 10,
  "env_names": "${'simulation_length_ns'}",
  "type": "number"
},
"report_interval_ps": {
  "title": "Report Interval Ps",
  "default": 50,
  "env_names": "${'report_interval_ps'}",
  "type": "number"
},
"dt_ps": {
  "title": "Dt Ps",
  "default": 0.002,
  "env_names": "${'dt_ps'}",

```

(continues on next page)

(continued from previous page)

```

    "type": "number"
  },
  "temperature_kelvin": {
    "title": "Temperature Kelvin",
    "default": 310.0,
    "env_names": "'temperature_kelvin'",
    "type": "number"
  },
  "heat_bath_friction_coef": {
    "title": "Heat Bath Friction Coef",
    "default": 1.0,
    "env_names": "'heat_bath_friction_coef'",
    "type": "number"
  },
  "reference_pdb_file": {
    "title": "Reference Pdb File",
    "env_names": "'reference_pdb_file'",
    "type": "string",
    "format": "path"
  },
  "openmm_selection": {
    "title": "Openmm Selection",
    "default": [
      "CA"
    ],
    "env_names": "'openmm_selection'",
    "type": "array",
    "items": {
      "type": "string"
    }
  },
  "mda_selection": {
    "title": "Mda Selection",
    "default": "protein and name CA",
    "env_names": "'mda_selection'",
    "type": "string"
  },
  "threshold": {
    "title": "Threshold",
    "default": 8.0,
    "env_names": "'threshold'",
    "type": "number"
  },
  "in_memory": {
    "title": "In Memory",
    "default": true,
    "env_names": "'in_memory'",
    "type": "boolean"
  },
  "bp_file": {
    "title": "Bp File",
    "default": "md.bp",

```

(continues on next page)

(continued from previous page)

```

    "env_names": "'bp_file'",
    "type": "string",
    "format": "path"
  },
  "adios_cfg": {
    "title": "Adios Cfg",
    "default": "adios.xml",
    "env_names": "'adios_cfg'",
    "type": "string",
    "format": "path"
  },
  "adios_xml_sim": {
    "title": "Adios Xml Sim",
    "default": "adios.xml",
    "env_names": "'adios_xml_sim'",
    "type": "string",
    "format": "path"
  },
  "compute_rmsd": {
    "title": "Compute Rmsd",
    "default": true,
    "env_names": "'compute_rmsd'",
    "type": "boolean"
  },
  "divisibleby": {
    "title": "Divisibleby",
    "default": 2,
    "env_names": "'divisibleby'",
    "type": "integer"
  },
  "outliers_dir": {
    "title": "Outliers Dir",
    "default": ".",
    "env_names": "'outliers_dir'",
    "type": "string",
    "format": "path"
  },
  "pickle_db": {
    "title": "Pickle Db",
    "default": ".",
    "env_names": "'pickle_db'",
    "type": "string",
    "format": "path"
  },
  "copy_velocities_p": {
    "title": "Copy Velocities P",
    "default": 0.5,
    "env_names": "'copy_velocities_p'",
    "type": "number"
  },
  "current_dir": {
    "title": "Current Dir",

```

(continues on next page)

(continued from previous page)

```

        "default": ".",
        "env_names": "'current_dir'",
        "type": "string",
        "format": "path"
    },
    "zcentroid_atoms": {
        "title": "Zcentroid Atoms",
        "default": "",
        "env_names": "'zcentroid_atoms'",
        "type": "string"
    },
    "compute_zcentroid": {
        "title": "Compute Zcentroid",
        "default": false,
        "env_names": "'compute_zcentroid'",
        "type": "boolean"
    },
    "ligand": {
        "title": "Ligand",
        "default": -1,
        "env_names": "'ligand'",
        "type": "integer"
    },
    "multi_ligand_table": {
        "title": "Multi Ligand Table",
        "default": ".",
        "env_names": "'multi_ligand_table'",
        "type": "string",
        "format": "path"
    },
    "adios_xml_file": {
        "title": "Adios Xml File",
        "default": ".",
        "env_names": "'adios_xml_file'",
        "type": "string",
        "format": "path"
    },
    "top_file1": {
        "title": "Top File1",
        "default": ".",
        "env_names": "'top_file1'",
        "type": "string",
        "format": "path"
    },
    "model": {
        "title": "Model",
        "default": "cvae",
        "env_names": "'model'",
        "type": "string"
    }
},
"definitions": {

```

(continues on next page)

(continued from previous page)

```

    "MDSolvent": {
        "title": "MDSolvent",
        "description": "An enumeration.",
        "enum": [
            "implicit",
            "explicit"
        ],
        "type": "string"
    }
}

```

Config

- **extra:** *str = allow*

Fields

- *adios_cfg* (*pathlib.Path*)
- *adios_xml_file* (*pathlib.Path*)
- *adios_xml_sim* (*pathlib.Path*)
- *bp_file* (*pathlib.Path*)
- *compute_rmsd* (*bool*)
- *compute_zcentroid* (*bool*)
- *copy_velocities_p* (*float*)
- *current_dir* (*pathlib.Path*)
- *divisibleby* (*int*)
- *dt_ps* (*float*)
- *heat_bath_friction_coef* (*float*)
- *in_memory* (*bool*)
- *initial_pdb_dir* (*pathlib.Path*)
- *ligand* (*int*)
- *mda_selection* (*str*)
- *multi_ligand_table* (*pathlib.Path*)
- *openmm_selection* (*List[str]*)
- *outliers_dir* (*pathlib.Path*)
- *pickle_db* (*pathlib.Path*)
- *reference_pdb_file* (*Optional[pathlib.Path]*)
- *report_interval_ps* (*float*)
- *simulation_length_ns* (*float*)
- *solvent_type* (*deepdrivemd.sim.openmm_stream.config.OpenMMConfig.MDSolvent*)

- *temperature_kelvin* (float)
- *threshold* (float)
- *top_file1* (pathlib.Path)
- *top_suffix* (Optional[str])
- *zcentroid_atoms* (Optional[str])

field adios_cfg: pathlib.Path = 'adios.xml'

Validated by

- *explicit_solvent_requires_top_suffix*

field adios_xml_file: pathlib.Path = PosixPath('.')

Validated by

- *explicit_solvent_requires_top_suffix*

field adios_xml_sim: pathlib.Path = 'adios.xml'

Validated by

- *explicit_solvent_requires_top_suffix*

field bp_file: pathlib.Path = 'md.bp'

Validated by

- *explicit_solvent_requires_top_suffix*

field compute_rmsd: bool = True

Validated by

- *explicit_solvent_requires_top_suffix*

field compute_zcentroid: bool = False

Validated by

- *explicit_solvent_requires_top_suffix*

field copy_velocities_p: float = 0.5

Validated by

- *explicit_solvent_requires_top_suffix*

field current_dir: pathlib.Path = PosixPath('.')

Validated by

- *explicit_solvent_requires_top_suffix*

field divisibleby: int = 2

Validated by

- *explicit_solvent_requires_top_suffix*

field dt_ps: float = 0.002

Validated by

- *explicit_solvent_requires_top_suffix*

field heat_bath_friction_coef: float = 1.0

Validated by

- *explicit_solvent_requires_top_suffix*

field in_memory: bool = True

Validated by

- *explicit_solvent_requires_top_suffix*

field initial_pdb_dir: pathlib.Path = PosixPath('.')

Validated by

- *explicit_solvent_requires_top_suffix*

field ligand: int = -1

Validated by

- *explicit_solvent_requires_top_suffix*

field mda_selection: str = 'protein and name CA'

Validated by

- *explicit_solvent_requires_top_suffix*

field multi_ligand_table: pathlib.Path = PosixPath('.')

Validated by

- *explicit_solvent_requires_top_suffix*

field openmm_selection: List[str] = ['CA']

Validated by

- *explicit_solvent_requires_top_suffix*

field outliers_dir: pathlib.Path = PosixPath('.')

Validated by

- *explicit_solvent_requires_top_suffix*

field pickle_db: pathlib.Path = PosixPath('.')

Validated by

- *explicit_solvent_requires_top_suffix*

field reference_pdb_file: Optional[pathlib.Path] = PydanticUndefined

Validated by

- *explicit_solvent_requires_top_suffix*

field report_interval_ps: float = 50

Validated by

- `explicit_solvent_requires_top_suffix`

field simulation_length_ns: float = 10

Validated by

- `explicit_solvent_requires_top_suffix`

field solvent_type: `deepdrivemd.sim.openmm_stream.config.OpenMMConfig.MDSolvent` = `MDSolvent.implicit`

Validated by

- `explicit_solvent_requires_top_suffix`

field temperature_kelvin: float = 310.0

Validated by

- `explicit_solvent_requires_top_suffix`

field threshold: float = 8.0

Validated by

- `explicit_solvent_requires_top_suffix`

field top_file1: `pathlib.Path` = `PosixPath('.')`

Validated by

- `explicit_solvent_requires_top_suffix`

field top_suffix: `Optional[str]` = `'.top'`

Validated by

- `explicit_solvent_requires_top_suffix`

field zcentroid_atoms: `Optional[str]` = `''`

Validated by

- `explicit_solvent_requires_top_suffix`

class `MDSolvent`(*value*)

An enumeration.

`explicit` = `'explicit'`

`implicit` = `'implicit'`

validator `explicit_solvent_requires_top_suffix` » *all fields*

deepdrivemd.utils

Functions

<code>bestk(a, k[, smallest])</code>	Return the best k values and correspondding indices.
<code>get_frameinfo()</code>	
<code>hash2intarray(h)</code>	
<code>intarray2hash(ia)</code>	
<code>parse_args()</code>	
<code>setup_mpi([comm])</code>	
<code>setup_mpi_comm(distributed)</code>	
<code>t1Dto2D(B)</code>	
<code>t2Dto1D(A)</code>	
<code>timer(label[, start, frameinfo])</code>	

Classes

<code>Timer(label)</code>

class deepdrivemd.utils.**Timer**(*label: str*)

deepdrivemd.utils.**bestk**(*a: npt.ArrayLike, k: int, smallest: bool = True*) → Tuple[npt.ArrayLike, npt.ArrayLike]

Return the best k values and correspondding indices.

Parameters

- **a** (*npt.ArrayLike*) – Array of dim (N,)
- **k** (*int*) – Specifies which element to partition upon.
- **smallest** (*bool*) – True if the best values are small (or most negative). False if the best values are most positive.

Returns

- *npt.ArrayLike* – Of length k containing the k smallest values in a .
- *npt.ArrayLike* – Of length k containing indices of input array a corresponding to the k smallest values in a .

deepdrivemd.utils.**get_frameinfo**() → inspect.Traceback

deepdrivemd.utils.**hash2intarray**(*h*)

deepdrivemd.utils.**intarray2hash**(ia)

deepdrivemd.utils.**parse_args**() → argparse.Namespace

deepdrivemd.utils.**setup_mpi**(comm: *Optional[Any] = None*) → Tuple[int, int]

deepdrivemd.utils.**setup_mpi_comm**(distributed: *bool*) → Optional[Any]

deepdrivemd.utils.**t1Dto2D**(B)

deepdrivemd.utils.**t2Dto1D**(A)

deepdrivemd.utils.**timer**(label: *str*, start: *int = 1*, frameinfo: *Optional[inspect.Traceback] = None*) → None

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

d

- deepdrivemd, 8
- deepdrivemd.agents, 8
- deepdrivemd.agents.lof, 8
- deepdrivemd.agents.lof.config, 9
- deepdrivemd.agents.stream, 13
- deepdrivemd.agents.stream.config, 13
- deepdrivemd.aggregation, 22
- deepdrivemd.aggregation.basic, 22
- deepdrivemd.aggregation.basic.aggregate, 23
- deepdrivemd.aggregation.basic.config, 23
- deepdrivemd.aggregation.stream, 25
- deepdrivemd.aggregation.stream.aggregator, 25
- deepdrivemd.aggregation.stream.config, 26
- deepdrivemd.config, 29
- deepdrivemd.data, 108
- deepdrivemd.data.analysis, 108
- deepdrivemd.data.api, 109
- deepdrivemd.data.stream, 112
- deepdrivemd.data.stream.adios_utils, 113
- deepdrivemd.data.stream.aggregator_reader, 115
- deepdrivemd.data.stream.enumerations, 118
- deepdrivemd.data.stream.OutlierDB, 112
- deepdrivemd.data.utils, 118
- deepdrivemd.deepdrivemd, 119
- deepdrivemd.deepdrivemd_stream, 120
- deepdrivemd.models, 121
- deepdrivemd.models.aae, 122
- deepdrivemd.models.aae.config, 122
- deepdrivemd.models.aae_stream, 129
- deepdrivemd.models.aae_stream.config, 130
- deepdrivemd.models.aae_stream.train, 132
- deepdrivemd.models.aae_stream.utils, 133
- deepdrivemd.models.keras_cvae, 134
- deepdrivemd.models.keras_cvae.config, 134
- deepdrivemd.models.keras_cvae.utils, 140
- deepdrivemd.models.keras_cvae_stream, 141
- deepdrivemd.models.keras_cvae_stream.config, 141
- deepdrivemd.selection, 149
- deepdrivemd.selection.latest, 149
- deepdrivemd.selection.latest.config, 149
- deepdrivemd.selection.latest.select_model, 151
- deepdrivemd.sim, 152
- deepdrivemd.sim.openmm, 152
- deepdrivemd.sim.openmm.config, 152
- deepdrivemd.sim.openmm_stream, 158
- deepdrivemd.sim.openmm_stream.config, 158
- deepdrivemd.utils, 167

Symbols

<code>__del__()</code> (<code>deepdrivemd.data.stream.aggregator_reader.AdiosReader</code> method), 115	<code>drivemd.models.aae_stream.config.Point3dAAEConfig</code> attribute), 130
<code>__init__()</code> (<code>deepdrivemd.data.stream.OutlierDB.OutlierDB</code> method), 113	<code>ad战略ios_xml_file</code> (<code>deepdrivemd.config.StreamingExperimentConfig</code> attribute), 103
<code>__init__()</code> (<code>deepdrivemd.data.stream.adios_utils.AdiosStreamStepRW</code> method), 114	<code>ad战略ios_xml_file</code> (<code>deepdrivemd.sim.openmm_stream.config.OpenMMConfig</code> attribute), 164
<code>__init__()</code> (<code>deepdrivemd.data.stream.aggregator_reader.AdiosReader</code> method), 115	<code>ad战略ios_xml_sim</code> (<code>deepdrivemd.config.StreamingExperimentConfig</code> attribute), 103
<code>__init__()</code> (<code>deepdrivemd.data.stream.aggregator_reader.StreamVariable</code> method), 116	<code>ad战略ios_xml_sim</code> (<code>deepdrivemd.sim.openmm_stream.config.OpenMMConfig</code> attribute), 164
<code>__init__()</code> (<code>deepdrivemd.data.stream.aggregator_reader.Streams</code> method), 117	<code>AdiosReader</code> (class in <code>deepdrivemd.data.stream.aggregator_reader</code>), 115
<code>__init__()</code> (<code>deepdrivemd.models.aae_stream.utils.CenterOfMassTransform</code> method), 133	<code>AdiosStreamStepRW</code> (class in <code>deepdrivemd.data.stream.adios_utils</code>), 113
<code>__init__()</code> (<code>deepdrivemd.models.aae_stream.utils.PointCloudDatasetInMemory</code> method), 133	<code>AGENT_DIR</code> (<code>deepdrivemd.data.api.DeepDriveMD_API</code> attribute), 109
<code>ad战略ios</code> (<code>deepdrivemd.data.stream.aggregator_reader.AdiosReader</code> attribute), 115	<code>AGENT_PIPELINE_NAME</code> (<code>deepdrivemd.deepdrivemd_stream.PipelineManager</code> attribute), 121
<code>ad战略ios_cfg</code> (<code>deepdrivemd.sim.openmm_stream.config.OpenMMConfig</code> attribute), 164	<code>agent_stage</code> (<code>deepdrivemd.config.ExperimentConfig</code> attribute), 60
<code>ad战略ios_xml_agg</code> (<code>deepdrivemd.agents.stream.config.OutlierDetectionConfig</code> attribute), 21	<code>agent_stage</code> (<code>deepdrivemd.config.StreamingExperimentConfig</code> attribute), 103
<code>ad战略ios_xml_agg</code> (<code>deepdrivemd.aggregation.stream.config.StreamAggregation</code> attribute), 28	<code>AGENT_STAGE_NAME</code> (<code>deepdrivemd.deepdrivemd.PipelineManager</code> attribute), 120
<code>ad战略ios_xml_agg</code> (<code>deepdrivemd.config.StreamingExperimentConfig</code> attribute), 103	<code>AGENT_STAGE_NAME</code> (<code>deepdrivemd.deepdrivemd_stream.PipelineManager</code> attribute), 121
<code>ad战略ios_xml_agg</code> (<code>deepdrivemd.models.aae_stream.config.Point3dAAEConfig</code> attribute), 130	<code>agg_dir</code> (<code>deepdrivemd.agents.stream.config.OutlierDetectionConfig</code> attribute), 21
<code>ad战略ios_xml_agg</code> (<code>deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig</code> attribute), 148	<code>agg_dir</code> (<code>deepdrivemd.models.aae_stream.config.Point3dAAEConfig</code> attribute), 130
<code>ad战略ios_xml_agg_4ml</code> (<code>deepdrivemd.config.StreamingExperimentConfig</code> attribute), 103	<code>agg_dir</code> (<code>deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig</code> attribute), 148
<code>ad战略ios_xml_agg_4ml</code> (<code>deepdrivemd.aggregation.stream.aggregator</code>),	<code>aggregate()</code> (in module <code>deepdrivemd.aggregation.stream.aggregator</code>),

25			
AGGREGATE_DIR	(deep- drivemd.data.api.DeepDriveMD_API attribute), 109	checkpoint_dir	(deep- drivemd.selection.latest.config.LatestCheckpointConfig attribute), 150
AGGREGATION_PIPELINE_NAME	(deep- drivemd.deepdrivemd_stream.PipelineManager attribute), 121	checkpoint_suffix	(deep- drivemd.selection.latest.config.LatestCheckpointConfig attribute), 150
aggregation_stage	(deep- drivemd.config.ExperimentConfig attribute), 60	cm (deepdrivemd.data.stream.aggregator_reader.Streams attribute), 117	
aggregation_stage	(deep- drivemd.config.StreamingExperimentConfig attribute), 103	cms_transform	(deep- drivemd.models.aae_stream.config.Point3dAAEConfig attribute), 130
AGGREGATION_STAGE_NAME	(deep- drivemd.deepdrivemd.PipelineManager attribute), 120	compute_number_of_nodes() (in module deep- drivemd.deepdrivemd_stream), 121	
AGGREGATION_STAGE_NAME	(deep- drivemd.deepdrivemd_stream.PipelineManager attribute), 121	compute_rmsd (deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute), 21	
apply_analysis_fn()	(deep- drivemd.data.analysis.DeepDriveMD_Analysis method), 108	compute_rmsd (deepdrivemd.aggregation.stream.config.StreamAggregation attribute), 28	
arguments	(deepdrivemd.config.BaseStageConfig attribute), 41	compute_rmsd (deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute), 164	
array (deepdrivemd.data.stream.enumerations.DataStructure attribute), 118		compute_zcentroid	(deep- drivemd.agents.stream.config.OutlierDetectionConfig attribute), 21
B		compute_zcentroid	(deep- drivemd.aggregation.stream.config.StreamAggregation attribute), 28
batch (deepdrivemd.data.stream.aggregator_reader.Streams attribute), 117		compute_zcentroid	(deep- drivemd.sim.openmm_stream.config.OpenMMConfig attribute), 164
batch_size (deepdrivemd.models.aae.config.AAEModelConfig attribute), 128		concatenate_last_n_h5() (in module deep- drivemd.aggregation.basic.aggregate), 23	
batch_size (deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 130		concatenate_virtual_h5() (in module deep- drivemd.data.utils), 118	
batch_size (deepdrivemd.models.keras_cvae.config.KerasCVAEModelConfig attribute), 140		config_directory	(deep- drivemd.config.StreamingExperimentConfig attribute), 103
batch_size (deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig attribute), 148		config_path() (deepdrivemd.data.api.Stage_API method), 111	
best_model (deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute), 21		connect_to_input() (in module deep- drivemd.aggregation.stream.aggregator), 26	
bestk() (in module deepdrivemd.utils), 167		connections (deepdrivemd.data.stream.adios_utils.AdiosStreamStepRW attribute), 113	
bp_file (deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute), 164		contact_map (deepdrivemd.aggregation.basic.config.BasicAggregation attribute), 25	
build_model() (in module deep- drivemd.models.aae_stream.train), 132		contact_map (deepdrivemd.sim.openmm.config.OpenMMConfig attribute), 156	
C		conv_filter_shapes	(deep- drivemd.agents.stream.config.OutlierDetectionConfig attribute), 21
CenterOfMassTransform (class in deep- drivemd.models.aae_stream.utils), 133		conv_filter_shapes	(deep- drivemd.models.keras_cvae.config.KerasCVAEModelConfig attribute), 140
checkpoint_dir	(deep- drivemd.models.aae_stream.config.Point3dAAEConfig attribute), 130	conv_filter_shapes	(deep- drivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig attribute), 140
checkpoint_dir	(deep- drivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig attribute), 140		

`drivemd.models.keras_cvae_stream.config.KerasCVAEStreamConfig` (deepdrivemd.config module), 148
`conv_filters` (deepdrivemd.agents.stream.config.OutlierDetectorConfig), 21
`conv_filters` (deepdrivemd.models.keras_cvae.config.KerasCVAEModelConfig), 140
`conv_filters` (deepdrivemd.models.keras_cvae_stream.config.KerasCVAEStreamConfig), 148
`conv_layers` (deepdrivemd.agents.stream.config.OutlierDetectorConfig), 21
`conv_layers` (deepdrivemd.models.keras_cvae.config.KerasCVAEModelConfig), 140
`conv_layers` (deepdrivemd.models.keras_cvae_stream.config.KerasCVAEStreamConfig), 148
`conv_strides` (deepdrivemd.agents.stream.config.OutlierDetectorConfig), 21
`conv_strides` (deepdrivemd.models.keras_cvae.config.KerasCVAEModelConfig), 140
`conv_strides` (deepdrivemd.models.keras_cvae_stream.config.KerasCVAEStreamConfig), 148
`copy_velocities_p` (deepdrivemd.sim.openmm_stream.config.OpenMMConfig), 164
`cpu_reqs` (deepdrivemd.config.BaseStageConfig), 41
`cpus_per_node` (deepdrivemd.config.ExperimentConfig), 60
`current_dir` (deepdrivemd.sim.openmm_stream.config.OpenMMConfig), 164
D
`dataset_location` (deepdrivemd.models.aae.config.AAEModelConfig), 128
`dataset_name` (deepdrivemd.models.aae.config.AAEModelConfig), 128
`dataset_name` (deepdrivemd.models.keras_cvae.config.KerasCVAEModelConfig), 140
`DataStructure` (class in deepdrivemd.data.stream.enumerations), 118
`decoder_affine_widths` (deepdrivemd.models.aae_stream.config.Point3dAAEConfig), 130
`decoder_bias` (deepdrivemd.models.aae_stream.config.Point3dAAEConfig), 130
`decoder_relu_slope` (deepdrivemd.models.aae_stream.config.Point3dAAEConfig), 130
`deepdrivemd` module, 8
`deepdrivemd.agents` module, 8
`deepdrivemd.agents.lof` module, 8
`deepdrivemd.config` module, 9
`deepdrivemd.config.agents` module, 13
`deepdrivemd.config.agents.stream` module, 13
`deepdrivemd.config.agents.stream.config` module, 13
`deepdrivemd.config.agents.stream.config.outlier_detector` module, 22
`deepdrivemd.config.aggregation.basic` module, 22
`deepdrivemd.config.aggregation.basic.aggregate` module, 23
`deepdrivemd.config.aggregation.basic.config` module, 23
`deepdrivemd.config.aggregation.stream` module, 25
`deepdrivemd.config.aggregation.stream.aggregator` module, 25
`deepdrivemd.config.aggregation.stream.config` module, 26
`deepdrivemd.config` module, 29
`deepdrivemd.data` module, 108
`deepdrivemd.data.analysis` module, 108
`deepdrivemd.data.api` module, 109
`deepdrivemd.data.stream` module, 112
`deepdrivemd.data.stream.adios_utils` module, 113
`deepdrivemd.data.stream.aggregator_reader` module, 115
`deepdrivemd.data.stream.enumerations` module, 118
`deepdrivemd.data.stream.OutlierDB` module, 112
`deepdrivemd.data.utils` module, 118
`deepdrivemd.deepdrivemd` module, 119
`deepdrivemd.deepdrivemd_stream` module, 120
`deepdrivemd.models` module, 121
`deepdrivemd.models.aae` module, 122
`deepdrivemd.models.aae.config` module, 122
`deepdrivemd.models.aae_stream` module, 129
`deepdrivemd.models.aae_stream.config` module, 130

deepdrivemd.models.aae_stream.train
 module, 132
 deepdrivemd.models.aae_stream.utils
 module, 133
 deepdrivemd.models.keras_cvae
 module, 134
 deepdrivemd.models.keras_cvae.config
 module, 134
 deepdrivemd.models.keras_cvae.utils
 module, 140
 deepdrivemd.models.keras_cvae_stream
 module, 141
 deepdrivemd.models.keras_cvae_stream.config
 module, 141
 deepdrivemd.selection
 module, 149
 deepdrivemd.selection.latest
 module, 149
 deepdrivemd.selection.latest.config
 module, 149
 deepdrivemd.selection.latest.select_model
 module, 151
 deepdrivemd.sim
 module, 152
 deepdrivemd.sim.openmm
 module, 152
 deepdrivemd.sim.openmm.config
 module, 152
 deepdrivemd.sim.openmm_stream
 module, 158
 deepdrivemd.sim.openmm_stream.config
 module, 158
 deepdrivemd.utils
 module, 167
 DeepDriveMD_Analysis (class in deep-
 drivemd.data.analysis), 108
 DeepDriveMD_API (class in deepdrivemd.data.api), 109
 dense_dropouts (deep-
 drivemd.agents.stream.config.OutlierDetectionConfig
 attribute), 21
 dense_dropouts (deep-
 drivemd.models.keras_cvae.config.KerasCVAEModelConfig
 attribute), 140
 dense_dropouts (deep-
 drivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig
 attribute), 148
 dense_layers (deepdrivemd.agents.stream.config.OutlierDetectionConfig
 attribute), 21
 dense_layers (deepdrivemd.models.keras_cvae.config.KerasCVAEModelConfig
 attribute), 140
 dense_layers (deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig
 attribute), 148
 dense_neurons (deep-
 drivemd.agents.stream.config.OutlierDetectionConfig
 attribute), 21
 dense_neurons (deep-
 drivemd.models.keras_cvae.config.KerasCVAEModelConfig
 attribute), 140
 dense_neurons (deep-
 drivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig
 attribute), 148
 dictionary (deepdrivemd.data.stream.OutlierDB.OutlierDB
 attribute), 113
 dir (deepdrivemd.data.stream.OutlierDB.OutlierDB at-
 tribute), 112
 discriminator_affine_widths (deep-
 drivemd.models.aae_stream.config.Point3dAAEConfig
 attribute), 130
 discriminator_bias (deep-
 drivemd.models.aae_stream.config.Point3dAAEConfig
 attribute), 130
 discriminator_filters (deep-
 drivemd.models.aae.config.AAEModelConfig
 attribute), 128
 discriminator_relu_slope (deep-
 drivemd.models.aae.config.AAEModelConfig
 attribute), 128
 discriminator_relu_slope (deep-
 drivemd.models.aae_stream.config.Point3dAAEConfig
 attribute), 130
 divisibleby (deepdrivemd.sim.openmm_stream.config.OpenMMConfig
 attribute), 164
 dt_ps (deepdrivemd.sim.openmm.config.OpenMMConfig
 attribute), 156
 dt_ps (deepdrivemd.sim.openmm_stream.config.OpenMMConfig
 attribute), 164
 dtype (deepdrivemd.data.stream.aggregator_reader.StreamVariable
 attribute), 116
 dump_yaml() (deepdrivemd.config.BaseSettings
 method), 39
E
 embed_interval (deep-
 drivemd.models.aae.config.AAEModelConfig
 attribute), 128
 encoder_bias (deepdrivemd.models.aae_stream.config.Point3dAAEConfig
 attribute), 130
 encoder_filters (deep-
 drivemd.models.aae.config.AAEModelConfig
 attribute), 128
 encoder_filters (deep-
 drivemd.models.aae_stream.config.Point3dAAEConfig
 attribute), 130
 encoder_kernel_sizes (deep-
 drivemd.models.aae.config.AAEModelConfig
 attribute), 128
 encoder_kernels (deep-
 drivemd.models.aae_stream.config.Point3dAAEConfig
 attribute), 130

attribute), 130
 encoder_relu_slope (deepdrivemd.models.aae.config.AAEModelConfig attribute), 128
 encoder_relu_slope (deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 130
 epochs (deepdrivemd.models.aae.config.AAEModelConfig attribute), 128
 epochs (deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 130
 epochs (deepdrivemd.models.keras_cvae.config.KerasCVAEModelConfig attribute), 140
 epochs (deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig attribute), 148
 executable (deepdrivemd.config.BaseStageConfig attribute), 42
 experiment_directory (deepdrivemd.config.AgentTaskConfig attribute), 33
 experiment_directory (deepdrivemd.config.AggregationTaskConfig attribute), 38
 experiment_directory (deepdrivemd.config.BaseTaskConfig attribute), 43
 experiment_directory (deepdrivemd.config.ExperimentConfig attribute), 60
 experiment_directory (deepdrivemd.config.ModelSelectionTaskConfig attribute), 72
 experiment_directory (deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 130
 experiment_directory_cannot_exist() (deepdrivemd.config.ExperimentConfig class method), 61
 explicit (deepdrivemd.sim.openmm.config.OpenMMConfig attribute), 158
 explicit (deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute), 166
 explicit_solvent_requires_top_suffix() (deepdrivemd.sim.openmm.config.OpenMMConfig class method), 158
 explicit_solvent_requires_top_suffix() (deepdrivemd.sim.openmm_stream.config.OpenMMConfig class method), 166
 extrinsic_score (deepdrivemd.agents.lof.config.OutlierDetectionConfig attribute), 11

attribute), 21
 final_shape (deepdrivemd.models.keras_cvae.config.KerasCVAEModelConfig attribute), 140
 final_shape (deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig attribute), 148
 find_input() (in module deepdrivemd.aggregation.stream.aggregator), 26
 fnc (deepdrivemd.aggregation.basic.config.BasicAggregation attribute), 25
 fnc_name (deepdrivemd.models.aae.config.AAEModelConfig attribute), 129
 fraction_of_contacts (deepdrivemd.models.openmm.config.OpenMMConfig attribute), 156
 from_yaml() (deepdrivemd.config.BaseSettings class method), 39
 func_condition() (deepdrivemd.deepdrivemd.PipelineManager method), 120
 func_on_false() (deepdrivemd.deepdrivemd.PipelineManager method), 120
 func_on_true() (deepdrivemd.deepdrivemd.PipelineManager method), 120

G

generate_agent_stage() (deepdrivemd.deepdrivemd.PipelineManager method), 120
 generate_agent_stage() (deepdrivemd.deepdrivemd_stream.PipelineManager method), 121
 generate_aggregating_stage() (deepdrivemd.deepdrivemd.PipelineManager method), 120
 generate_aggregating_stage() (deepdrivemd.deepdrivemd_stream.PipelineManager method), 121
 generate_machine_learning_stage() (deepdrivemd.deepdrivemd.PipelineManager method), 120
 generate_machine_learning_stage() (deepdrivemd.deepdrivemd_stream.PipelineManager method), 121
 generate_model_selection_stage() (deepdrivemd.deepdrivemd.PipelineManager method), 120
 generate_molecular_dynamics_stage() (deepdrivemd.deepdrivemd.PipelineManager method), 120
 generate_molecular_dynamics_stage() (deepdrivemd.deepdrivemd_stream.PipelineManager method), 121

F

final_shape (deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute), 11

method), 121

generate_pipelines() (deepdrivemd.deepdrivemd.PipelineManager method), 120

generate_pipelines() (deepdrivemd.deepdrivemd_stream.PipelineManager method), 121

generate_sample_config() (in module deepdrivemd.config), 108

generate_task() (in module deepdrivemd.deepdrivemd), 120

generate_task() (in module deepdrivemd.deepdrivemd_stream), 121

generator_filters (deepdrivemd.models.aae.config.AAEModelConfig attribute), 129

generator_relu_slope (deepdrivemd.models.aae.config.AAEModelConfig attribute), 129

get_agent_h5() (deepdrivemd.data.analysis.DeepDriveMD_Analysis method), 108

get_agent_json() (deepdrivemd.data.analysis.DeepDriveMD_Analysis method), 108

get_count() (deepdrivemd.data.api.Stage_API static method), 111

get_frameinfo() (in module deepdrivemd.utils), 167

get_initial_pdb() (deepdrivemd.data.api.DeepDriveMD_API static method), 109

get_last_n_md_runs() (deepdrivemd.data.api.DeepDriveMD_API method), 109

get_latest() (deepdrivemd.data.api.Stage_API static method), 111

get_model_path() (in module deepdrivemd.selection.latest.select_model), 151

get_restart_pdb() (deepdrivemd.data.api.DeepDriveMD_API method), 109

get_system_name() (deepdrivemd.data.api.DeepDriveMD_API static method), 110

get_system_pdb_name() (deepdrivemd.data.api.DeepDriveMD_API static method), 110

get_topology() (deepdrivemd.data.api.DeepDriveMD_API static method), 110

get_total_iterations() (deepdrivemd.data.api.DeepDriveMD_API method), 111

get_virtual_h5_file() (in module deepdrivemd.data.utils), 118

glob_file_from_dirs() (in module deepdrivemd.data.api), 112

gpu_reqs (deepdrivemd.config.BaseStageConfig attribute), 42

gpus_per_node (deepdrivemd.config.ExperimentConfig attribute), 61

H

hardware_threads_per_cpu (deepdrivemd.config.ExperimentConfig attribute), 61

hash2intarray() (in module deepdrivemd.utils), 167

heat_bath_friction_coef (deepdrivemd.sim.openmm.config.OpenMMConfig attribute), 156

heat_bath_friction_coef (deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute), 165

I

implicit (deepdrivemd.sim.openmm.config.OpenMMConfig.MDSolvent attribute), 158

implicit (deepdrivemd.sim.openmm_stream.config.OpenMMConfig.MDSolvent attribute), 166

in_memory (deepdrivemd.sim.openmm.config.OpenMMConfig attribute), 156

in_memory (deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute), 165

inference_batch_size (deepdrivemd.agents.lof.config.OutlierDetectionConfig attribute), 11

init_eps (deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute), 21

init_min_samples (deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute), 21

init_pdb_file (deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute), 21

init_pdb_file (deepdrivemd.config.StreamingExperimentConfig attribute), 103

init_weights (deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 130

init_weights_path (deepdrivemd.config.MachineLearningTaskConfig attribute), 68

init_weights_path (deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 130

initial_epochs (deepdrivemd.models.aae.config.AAEModelConfig attribute), 129

initial_epochs (deepdrivemd.models.keras_cvae.config.KerasCVAEModelConfig attribute), 140
initial_pdb_dir (deepdrivemd.config.MolecularDynamicsTaskConfig attribute), 78
initial_pdb_dir (deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute), 165
initial_shape (deepdrivemd.models.keras_cvae.config.KerasCVAEModelConfig attribute), 140
input_path (deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 130
intarray2hash() (in module deepdrivemd.utils), 167
intrinsic_score (deepdrivemd.agents.lof.config.OutlierDetectionConfig attribute), 11
io (deepdrivemd.data.stream.aggregator_reader.AdiosReader attribute), 115

J

json_path() (deepdrivemd.data.api.Stage_API method), 111

K

k_random_old_h5_files (deepdrivemd.agents.lof.config.OutlierDetectionConfig attribute), 11
k_random_old_h5_files (deepdrivemd.models.aae.config.AAEModelConfig attribute), 129
k_random_old_h5_files (deepdrivemd.models.keras_cvae.config.KerasCVAEModelConfig attribute), 140

L

lambda_gp (deepdrivemd.models.aae.config.AAEModelConfig attribute), 129
lambda_gp (deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 130
lambda_rec (deepdrivemd.models.aae.config.AAEModelConfig attribute), 129
lambda_rec (deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 130
last_n_h5_files (deepdrivemd.aggregation.basic.config.BasicAggregation attribute), 25
last_n_h5_files (deepdrivemd.models.aae.config.AAEModelConfig attribute), 129
last_n_h5_files (deepdrivemd.models.keras_cvae.config.KerasCVAEModelConfig attribute), 140

M

MACHINE_LEARNING_DIR (deepdrivemd.data.api.DeepDriveMD_API attribute), 109
MACHINE_LEARNING_PIPELINE_NAME (deepdrivemd.deepdrivemd_stream.PipelineManager attribute), 121
machine_learning_stage (deepdrivemd.config.ExperimentConfig attribute), 61
machine_learning_stage (deepdrivemd.config.StreamingExperimentConfig attribute), 103
MACHINE_LEARNING_STAGE_NAME (deepdrivemd.deepdrivemd.PipelineManager attribute), 120
MACHINE_LEARNING_STAGE_NAME (deepdrivemd.deepdrivemd_stream.PipelineManager attribute), 121
main() (in module deepdrivemd.models.aae_stream.train), 132
max_iteration (deepdrivemd.config.ExperimentConfig attribute), 61
max_loss (deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 130
max_loss (deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig attribute), 148
max_steps (deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131
max_steps (deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig attribute), 148
md5() (deepdrivemd.data.stream.aggregator_reader.Streams attribute), 117

mda_selection	(deep- drivemd.sim.openmm.config.OpenMMConfig attribute), 156	deepdrivemd.aggregation.basic, 22 deepdrivemd.aggregation.basic.aggregate, 23
mda_selection	(deep- drivemd.sim.openmm_stream.config.OpenMMConfig attribute), 165	deepdrivemd.aggregation.basic.config, 23 deepdrivemd.aggregation.stream, 25 deepdrivemd.aggregation.stream.aggregator, 25
min_step_increment	(deep- drivemd.agents.stream.config.OutlierDetectionConfig attribute), 21	deepdrivemd.aggregation.stream.config, 26 deepdrivemd.config, 29
min_step_increment	(deep- drivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131	deepdrivemd.data, 108 deepdrivemd.data.analysis, 108 deepdrivemd.data.api, 109
min_step_increment	(deep- drivemd.models.keras_cvae_stream.config.KerasCVAEStreamConfig attribute), 148	deepdrivemd.data.stream, 112 deepdrivemd.data.stream.adios_utils, 113 deepdrivemd.data.stream.aggregator_reader, 115
model	(deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute), 21	deepdrivemd.data.stream.enumerations, 118 deepdrivemd.data.stream.OutlierDB, 112
model	(deepdrivemd.aggregation.stream.config.StreamAggregationConfig attribute), 28	deepdrivemd.data.utils, 118
model	(deepdrivemd.config.StreamingExperimentConfig attribute), 104	deepdrivemd.deepdrivemd, 119 deepdrivemd.deepdrivemd_stream, 120
model	(deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131	deepdrivemd.models, 121 deepdrivemd.models.aae, 122
model	(deepdrivemd.models.keras_cvae_stream.config.KerasCVAEStreamConfig attribute), 148	deepdrivemd.models.aae.config, 122 deepdrivemd.models.aae_stream, 129
MODEL_SELECTION_DIR	(deep- drivemd.data.api.DeepDriveMD_API attribute), 109	deepdrivemd.models.aae_stream.config, 130 deepdrivemd.models.aae_stream.train, 132 deepdrivemd.models.aae_stream.utils, 133
model_selection_stage	(deep- drivemd.config.ExperimentConfig attribute), 61	deepdrivemd.models.keras_cvae, 134 deepdrivemd.models.keras_cvae.config, 134 deepdrivemd.models.keras_cvae.utils, 140
model_selection_stage	(deep- drivemd.config.StreamingExperimentConfig attribute), 104	deepdrivemd.models.keras_cvae_stream, 141 deepdrivemd.models.keras_cvae_stream.config, 141
MODEL_SELECTION_STAGE_NAME	(deep- drivemd.deepdrivemd.PipelineManager attribute), 120	deepdrivemd.selection, 149 deepdrivemd.selection.latest, 149 deepdrivemd.selection.latest.config, 149
model_tag	(deepdrivemd.config.MachineLearningTaskConfig attribute), 68	deepdrivemd.selection.latest.select_model, 151
model_tag	(deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131	deepdrivemd.sim, 152 deepdrivemd.sim.openmm, 152
model_type	(deepdrivemd.agents.lof.config.OutlierDetectionConfig attribute), 11	deepdrivemd.sim.openmm.config, 152 deepdrivemd.sim.openmm_stream, 158
model_type_check()	(deep- drivemd.agents.lof.config.OutlierDetectionConfig class method), 12	deepdrivemd.sim.openmm_stream.config, 158 deepdrivemd.utils, 167
module		MOLECULAR_DYNAMICS_DIR (deep- drivemd.data.api.DeepDriveMD_API attribute), 109
deepdrivemd, 8		MOLECULAR_DYNAMICS_PIPELINE_NAME (deep- drivemd.deepdrivemd_stream.PipelineManager attribute), 121
deepdrivemd.agents, 8		molecular_dynamics_stage (deep- drivemd.config.ExperimentConfig attribute), 61
deepdrivemd.agents.lof, 8		
deepdrivemd.agents.lof.config, 9		
deepdrivemd.agents.stream, 13		
deepdrivemd.agents.stream.config, 13		
deepdrivemd.aggregation, 22		

MOLECULAR_DYNAMICS_STAGE_NAME (deep- node_local_path (deep-
drivemd.deepdrivemd.PipelineManager at- drivemd.config.ExperimentConfig attribute),
attribute), 120 61

MOLECULAR_DYNAMICS_STAGE_NAME (deep- node_local_path (deep-
drivemd.deepdrivemd_stream.PipelineManager drivemd.config.ModelSelectionTaskConfig
attribute), 121 attribute), 72

multi_ligand_table (deep- node_local_path (deep-
drivemd.agents.stream.config.OutlierDetectionConfig drivemd.models.aae_stream.config.Point3dAAEConfig
attribute), 21 attribute), 131

multi_ligand_table (deep- noise_mu (deepdrivemd.models.aae.config.AAEModelConfig
drivemd.aggregation.stream.config.StreamAggregation attribute), 129
attribute), 28 noise_mu (deepdrivemd.models.aae_stream.config.Point3dAAEConfig
attribute), 131

multi_ligand_table (deep- noise_std (deepdrivemd.models.aae.config.AAEModelConfig
drivemd.config.StreamingExperimentConfig attribute), 129
attribute), 104 noise_std (deepdrivemd.models.aae_stream.config.Point3dAAEConfig
attribute), 131

multi_ligand_table (deep- noise_std (deepdrivemd.models.aae_stream.config.Point3dAAEConfig
drivemd.sim.openmm_stream.config.OpenMMConfig attribute), 131
attribute), 165 num_agg (deepdrivemd.agents.stream.config.OutlierDetectionConfig
attribute), 21

N num_agg (deepdrivemd.models.aae_stream.config.Point3dAAEConfig
attribute), 131

n_most_recent_h5_files (deep- num_agg (deepdrivemd.models.keras_cvae_stream.config.KerasCVAEMode
drivemd.agents.lof.config.OutlierDetectionConfig attribute), 148
attribute), 12

n_sim (deepdrivemd.aggregation.stream.config.StreamAggregation num_data_workers (deep-
attribute), 28 drivemd.models.aae.config.AAEModelConfig
attribute), 129

n_traj_frames (deep- num_data_workers (deep-
drivemd.agents.lof.config.OutlierDetectionConfig attribute), 131
attribute), 12

name (deepdrivemd.data.stream.aggregator_reader.StreamVariable num_extrinsic_outliers (deep-
attribute), 116 drivemd.agents.lof.config.OutlierDetectionConfig
attribute), 12

next() (deepdrivemd.data.stream.aggregator_reader.AdiosReader num_features (deepdrivemd.agents.stream.config.OutlierDetectionConfig
method), 115 attribute), 21

next() (deepdrivemd.data.stream.aggregator_reader.StreamVariable num_features (deepdrivemd.models.aae.config.AAEModelConfig
method), 118 attribute), 129

next() (deepdrivemd.data.stream.aggregator_reader.StreamVariable num_features (deepdrivemd.models.aae_stream.config.Point3dAAEConfig
method), 116 attribute), 131

next() (deepdrivemd.data.stream.aggregator_reader.StreamVariable num_intrinsic_outliers (deep-
method), 116 drivemd.agents.lof.config.OutlierDetectionConfig
attribute), 12

next_input() (in module deep- num_outliers_check() (deep-
drivemd.models.aae_stream.train), 132 drivemd.agents.lof.config.OutlierDetectionConfig
class method), 12

next_random() (deep- num_points (deepdrivemd.agents.stream.config.OutlierDetectionConfig
drivemd.data.stream.OutlierDB.OutlierDB attribute), 21
method), 113 num_points (deepdrivemd.models.aae.config.AAEModelConfig
attribute), 129

node_local_path (deep- num_points (deepdrivemd.models.aae_stream.config.Point3dAAEConfig
drivemd.config.AgentTaskConfig attribute), 131
attribute), 38

node_local_path (deepdrivemd.config.BaseTaskConfig num_sim (deepdrivemd.agents.stream.config.OutlierDetectionConfig
attribute), 43 attribute), 22

num_tasks (deepdrivemd.aggregation.stream.config.StreamAggregation
attribute), 28

[num_tasks \(deepdrivemd.config.MolecularDynamicsStageConfig attribute\), 76](#)
[num_tasks \(deepdrivemd.config.StreamingAgentStageConfig attribute\), 81](#)
[num_tasks \(deepdrivemd.config.StreamingAggregationStageConfig attribute\), 85](#)
[num_tasks \(deepdrivemd.config.StreamingMachineLearningStageConfig attribute\), 108](#)
O
[openmm_selection \(deepdrivemd.sim.openmm.config.OpenMMConfig attribute\), 157](#)
[openmm_selection \(deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute\), 165](#)
[OpenMMConfig.MDSolvent \(class in deepdrivemd.sim.openmm.config\), 157](#)
[OpenMMConfig.MDSolvent \(class in deepdrivemd.sim.openmm_stream.config\), 166](#)
[optimizer_lr \(deepdrivemd.models.aae.config.AAEModelConfig attribute\), 129](#)
[optimizer_name \(deepdrivemd.models.aae.config.AAEModelConfig attribute\), 129](#)
[outlier_count \(deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute\), 22](#)
[outlier_max \(deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute\), 22](#)
[outlier_min \(deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute\), 22](#)
[outlier_selection \(deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute\), 22](#)
[OutlierDB \(class in deepdrivemd.data.stream.OutlierDB\), 112](#)
[outliers_dir \(deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute\), 165](#)
[output_path \(deepdrivemd.config.AgentTaskConfig attribute\), 33](#)
[output_path \(deepdrivemd.config.AggregationTaskConfig attribute\), 38](#)
[output_path \(deepdrivemd.config.BaseTaskConfig attribute\), 43](#)
[output_path \(deepdrivemd.config.ModelSelectionTaskConfig attribute\), 72](#)
[output_path \(deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute\), 131](#)
P
[parse_args\(\) \(in module deepdrivemd.utils\), 168](#)
[parse_h5\(\) \(in module deepdrivemd.data.utils\), 119](#)
[pickle_db \(deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute\), 165](#)
[PIPELINE_NAME \(deepdrivemd.deepdrivemd.PipelineManager attribute\), 120](#)
[PipelineManager \(class in deepdrivemd.deepdrivemd\), 120](#)
[PipelineManager \(class in deepdrivemd.deepdrivemd_stream\), 121](#)
[Point3dAAEConfig \(class in deepdrivemd.models.aae_stream.config\), 130](#)
[point_cloud \(deepdrivemd.aggregation.basic.config.BasicAggregation attribute\), 25](#)
[point_cloud \(deepdrivemd.sim.openmm.config.OpenMMConfig attribute\), 157](#)
[PointCloudDatasetInMemory \(class in deepdrivemd.models.aae_stream.utils\), 133](#)
[positions \(deepdrivemd.data.stream.aggregator_reader.Streams attribute\), 117](#)
[pre_exec \(deepdrivemd.config.BaseStageConfig attribute\), 42](#)
[prefetch_factor \(deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute\), 131](#)
[print\(\) \(deepdrivemd.data.stream.OutlierDB.OutlierDB method\), 113](#)
[process_type \(deepdrivemd.config.CPUReqs attribute\), 44](#)
[process_type \(deepdrivemd.config.GPUReqs attribute\), 62](#)
[process_type_check\(\) \(deepdrivemd.config.CPUReqs class method\), 44](#)
[process_type_check\(\) \(deepdrivemd.config.GPUReqs class method\), 62](#)
[processes \(deepdrivemd.config.CPUReqs attribute\), 44](#)
[processes \(deepdrivemd.config.GPUReqs attribute\), 62](#)
[project \(deepdrivemd.config.ExperimentConfig attribute\), 61](#)
[project_gpu \(deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute\), 22](#)
[project_lastN \(deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute\), 22](#)
[published_model_dir \(deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute\), 131](#)
[published_model_dir \(deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig attribute\), 148](#)

Q

queue (deepdrivemd.config.ExperimentConfig attribute), 61

R

read_adios_file() (in module deepdrivemd.models.aae_stream.utils), 133

read_batch(deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute), 22

read_batch(deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131

read_batch(deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig attribute), 148

read_step() (deepdrivemd.data.stream.adios_utils.AdiosStreamStep method), 114

read_task_json() (deepdrivemd.data.api.Stage_API method), 111

readers (deepdrivemd.data.stream.aggregator_reader.Streams attribute), 117

ref_pdb_file(deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute), 22

ref_pdb_file(deepdrivemd.config.StreamingExperimentConfig attribute), 104

reference_pdb_file (deepdrivemd.sim.openmm.config.OpenMMConfig attribute), 157

reference_pdb_file (deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute), 165

reinit(deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131

reinit(deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig attribute), 148

report_interval_ps (deepdrivemd.sim.openmm.config.OpenMMConfig attribute), 157

report_interval_ps (deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute), 165

resource (deepdrivemd.config.ExperimentConfig attribute), 61

resume_checkpoint (deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131

retrain_freq(deepdrivemd.config.MachineLearningStageConfig attribute), 66

retrain_freq(deepdrivemd.selection.latest.config.LatestCheckpointConfig attribute), 150

rmsd(deepdrivemd.aggregation.basic.config.BasicAggregation attribute), 25

rmsd_name(deepdrivemd.models.aae.config.AAEModelConfig attribute), 129

rmsds(deepdrivemd.data.stream.aggregator_reader.Streams attribute), 117

runs_dir (deepdrivemd.data.api.Stage_API property), 111

S

sample_interval (deepdrivemd.models.aae.config.AAEModelConfig attribute), 129

scalar(deepdrivemd.data.stream.enumerations.DataStructure attribute), 118

scalar_dataset_names (deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131

scalar_requires_grad (deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131

schema (deepdrivemd.config.ExperimentConfig attribute), 61

scoring_method_check() (deepdrivemd.agents.lof.config.OutlierDetectionConfig class method), 12

seed(deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131

setup_mpi() (in module deepdrivemd.utils), 168

setup_mpi_comm() (in module deepdrivemd.utils), 168

shuffle(deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131

shuffle(deepdrivemd.models.keras_cvae.config.KerasCVAEModelConfig attribute), 140

shuffle(deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModelConfig attribute), 148

simulation_length_ns (deepdrivemd.sim.openmm.config.OpenMMConfig attribute), 157

simulation_length_ns (deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute), 166

skip_aggregation (deepdrivemd.config.AggregationStageConfig attribute), 37

sklearn_num_jobs (deepdrivemd.agents.lof.config.OutlierDetectionConfig attribute), 12

sleep_time_bpfiles (deepdrivemd.aggregation.stream.config.StreamAggregation attribute), 28

software_directory (deepdrivemd.config.StreamingExperimentConfig attribute), 104

solvent_type(deepdrivemd.sim.openmm.config.OpenMMConfig attribute), 157

solvent_type(deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute), 166

sorted_index(deepdrivemd.data.stream.OutlierDB.OutlierDB attribute), 113

sparse_to_dense() (in module deepdrivemd.models.keras_cvae.utils), 140
 split_pct(deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131
 split_pct(deepdrivemd.models.keras_cvae.config.KerasCVAEModConfig attribute), 140
 split_pct(deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModConfig attribute), 148
 Stage_API(class in deepdrivemd.data.api), 111
 stage_dir()(deepdrivemd.data.api.Stage_API method), 111
 stage_dir_count()(deepdrivemd.data.api.Stage_API method), 112
 stage_idx(deepdrivemd.config.AgentTaskConfig attribute), 33
 stage_idx(deepdrivemd.config.AggregationTaskConfig attribute), 38
 stage_idx(deepdrivemd.config.BaseTaskConfig attribute), 43
 stage_idx(deepdrivemd.config.ModelSelectionTaskConfig attribute), 72
 stage_idx(deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131
 stage_name()(deepdrivemd.data.api.Stage_API static method), 112
 steps(deepdrivemd.data.stream.aggregator_reader.Streams attribute), 117
 stream(deepdrivemd.data.stream.aggregator_reader.AdiosReader attribute), 115
 StreamContactMapVariable(class in deepdrivemd.data.stream.aggregator_reader), 115
 Streams(class in deepdrivemd.data.stream.aggregator_reader), 116
 StreamScalarVariable(class in deepdrivemd.data.stream.aggregator_reader), 116
 StreamVariable(class in deepdrivemd.data.stream.aggregator_reader), 116
 string(deepdrivemd.data.stream.enumerations.DataStructure attribute), 118
 structure(deepdrivemd.data.stream.aggregator_reader.StreamVariable attribute), 116
 T
 t1Dto2D()(in module deepdrivemd.utils), 168
 t2Dto1D()(in module deepdrivemd.utils), 168
 task_config(deepdrivemd.config.AgentStageConfig attribute), 32
 task_config(deepdrivemd.config.AggregationStageConfig attribute), 37
 task_config(deepdrivemd.config.MachineLearningStageConfig attribute), 66
 task_config(deepdrivemd.config.ModelSelectionStageConfig attribute), 71
 task_config(deepdrivemd.config.MolecularDynamicsStageConfig attribute), 76
 task_config(deepdrivemd.config.TaskConfig attribute), 112
 task_idx(deepdrivemd.config.AgentTaskConfig attribute), 33
 task_idx(deepdrivemd.config.AggregationTaskConfig attribute), 38
 task_idx(deepdrivemd.config.BaseTaskConfig attribute), 43
 task_idx(deepdrivemd.config.ModelSelectionTaskConfig attribute), 72
 task_idx(deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131
 task_name()(deepdrivemd.data.api.Stage_API static method), 112
 temperature_kelvin(deepdrivemd.sim.openmm.config.OpenMMConfig attribute), 157
 temperature_kelvin(deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute), 166
 thread_type(deepdrivemd.config.CPUReqs attribute), 44
 thread_type(deepdrivemd.config.GPUReqs attribute), 62
 thread_type_check()(deepdrivemd.config.CPUReqs class method), 44
 thread_type_check()(deepdrivemd.config.GPUReqs class method), 62
 threads_per_process(deepdrivemd.config.CPUReqs attribute), 44
 threads_per_process(deepdrivemd.config.GPUReqs attribute), 62
 threshold(deepdrivemd.sim.openmm.config.OpenMMConfig attribute), 157
 threshold(deepdrivemd.sim.openmm_stream.config.OpenMMConfig attribute), 166
 timeout1(deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute), 22
 timeout1(deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131
 timeout1(deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModConfig attribute), 148
 timeout2(deepdrivemd.agents.stream.config.OutlierDetectionConfig attribute), 22
 timeout2(deepdrivemd.models.aae_stream.config.Point3dAAEConfig attribute), 131
 timeout2(deepdrivemd.models.keras_cvae_stream.config.KerasCVAEModConfig attribute), 148

Timer (class in *deepdrivemd.utils*), 167
 timer() (in module *deepdrivemd.utils*), 168
 title (*deepdrivemd.config.ExperimentConfig* attribute), 61
 top_file1 (*deepdrivemd.config.StreamingExperimentConfig* attribute), 104
 top_file1 (*deepdrivemd.sim.openmm_stream.config.OpenMMConfig* attribute), 166
 top_suffix (*deepdrivemd.sim.openmm.config.OpenMMConfig* attribute), 157
 top_suffix (*deepdrivemd.sim.openmm_stream.config.OpenMMConfig* attribute), 166
 total (*deepdrivemd.data.stream.aggregator_reader.StreamVariable* attribute), 116
 train() (in module *deepdrivemd.models.aae_stream.train*), 132
 train_model() (in module *deepdrivemd.models.aae_stream.train*), 132
 transform() (*deepdrivemd.models.aae_stream.utils.CenterOfMassTransform* method), 133
 tsne_interval (*deepdrivemd.models.aae.config.AAEModelConfig* attribute), 129

U

unique_name() (*deepdrivemd.data.api.Stage_API* static method), 112
 use_discriminator_bias (*deepdrivemd.models.aae.config.AAEModelConfig* attribute), 129
 use_encoder_bias (*deepdrivemd.models.aae.config.AAEModelConfig* attribute), 129
 use_generator_bias (*deepdrivemd.models.aae.config.AAEModelConfig* attribute), 129
 use_model_checkpoint (*deepdrivemd.models.aae_stream.config.Point3dAAEConfig* attribute), 131
 use_outliers (*deepdrivemd.agents.stream.config.OutlierDetectionConfig* attribute), 22
 use_random_outliers (*deepdrivemd.agents.stream.config.OutlierDetectionConfig* attribute), 22

V

validate() (in module *deepdrivemd.models.aae_stream.train*), 132
 variables (*deepdrivemd.data.stream.adios_utils.AdiosStreamStepRW* attribute), 114
 velocities (*deepdrivemd.data.stream.aggregator_reader.Streams* attribute), 117
 verbose (*deepdrivemd.aggregation.basic.config.BasicAggregation* attribute), 25

W

wait_for_input() (in module *deepdrivemd.models.aae_stream.train*), 132
 walltime_min (*deepdrivemd.config.ExperimentConfig* attribute), 61
 wandb_project_name (*deepdrivemd.models.aae.config.AAEModelConfig* attribute), 129
 wrap (*deepdrivemd.sim.openmm.config.OpenMMConfig* attribute), 157
 write_pub() (*deepdrivemd.data.api.DeepDriveMD_API* static method), 111
 write_step() (*deepdrivemd.data.stream.adios_utils.AdiosStreamStepRW* method), 114
 write_task_json() (*deepdrivemd.data.api.Stage_API* method), 112

Z

zcentroid_atoms (*deepdrivemd.sim.openmm_stream.config.OpenMMConfig* attribute), 166