

---

# **DeepSphere**

***Release 0.1.0***

**Arcanite Solutions**

**Feb 14, 2020**



**CONTENTS:**

- 1 deepsphere package** **1**
  - 1.1 Subpackages . . . . . 1
  - 1.2 Module contents . . . . . 4
- 2 deepsphere.tests package** **5**
  - 2.1 Submodules . . . . . 5
  - 2.2 deepsphere.tests.test\_foo module . . . . . 5
  - 2.3 Module contents . . . . . 5
- 3 More info** **7**
- 4 Indices and tables** **9**
- Python Module Index** **11**
- Index** **13**



## DEEPSPHERE PACKAGE

### 1.1 Subpackages

#### 1.1.1 deepsphere.data package

Subpackages

`deepsphere.data.datasets` package

Submodules

`deepsphere.data.datasets.dataset` module

Module contents

`deepsphere.data.transforms` package

Submodules

`deepsphere.data.transforms.transforms` module

Module contents

Module contents

#### 1.1.2 deepsphere.layers package

Subpackages

`deepsphere.layers.samplings` package

Submodules

`deepsphere.layers.samplings.equiangular_pool_unpool` module

`deepsphere.layers.samplings.healpix_pool_unpool` module

`deepsphere.layers.samplings.icosahedron_pool_unpool` module

#### Module contents

DeepSphere Base Documentation doc

#### Submodules

`deepsphere.layers.chebyshev` module

#### Module contents

### 1.1.3 `deepsphere.models` package

#### Subpackages

`deepsphere.models.spherical_unet` package

#### Submodules

`deepsphere.models.spherical_unet.decoder` module

`deepsphere.models.spherical_unet.encoder` module

`deepsphere.models.spherical_unet.unet_model` module

`deepsphere.models.spherical_unet.utils` module

#### Module contents

#### Module contents

### 1.1.4 `deepsphere.utils` package

#### Submodules

`deepsphere.utils.initialization` module

`deepsphere.utils.laplacian_funcs` module

`deepsphere.utils.parser` module

Command Line Parser related functions. One function creates the parser. Another function allows hybrid usage of: - a yaml file with predefined parameters and - user inputted parameters through the command line.

`deepsphere.utils.parser.create_parser()`

Creates a parser with all the variables that can be edited by the user.

**Returns** a parser for the command line

**Return type** parser

`deepsphere.utils.parser.parse_config(parser)`

Takes the yaml file given through the command line Adds all the yaml file parameters, unless they have already been defined in the command line. Checks all values have been set else raises a Value error. :param parser: parser to be updated by the yaml file parameters :type parser: argparse.parser

**Raises** **ValueError** – All fields must be set in the yaml config file or in the command line. Raises error if value is None (was not set).

**Returns** parsed args of the parser

**Return type** dict

## deepsphere.utils.samplings module

Different samplings require various calculations. The calculations present here are for equiangular, healpix, icosahedron samplings.

`deepsphere.utils.samplings.equiangular_bandwidth(nodes)`

Calculate the equiangular bandwidth based on input nodes

**Parameters** **nodes** (*int*) – the number of nodes should be a power of 4

**Returns** the corresponding bandwidth

**Return type** int

`deepsphere.utils.samplings.equiangular_calculator(tensor, ratio)`

From a 3D input tensor and a known ratio between the latitude dimension and longitude dimension of the data, reformat the 3D input into a 4D output while also obtaining the bandwidth.

**Parameters**

- **tensor** (*torch.tensor*) – 3D input tensor
- **ratio** (*float*) – the ratio between the latitude and longitude dimension of the data

**Returns** 4D tensor, the bandwidths for lat. and long.

**Return type** torch.tensor, int, int

`deepsphere.utils.samplings.equiangular_dimension_unpack(nodes, ratio)`

Calculate the two underlying dimensions from the total number of nodes

**Parameters**

- **nodes** (*int*) – combined dimensions
- **ratio** (*float*) – ratio between the two dimensions

**Returns** separated dimensions

**Return type** int, int

`deepsphere.utils.samplings.healpix_resolution_calculator(nodes)`

Calculate the resolution of a healpix graph for a given number of nodes.

**Parameters** **nodes** (*int*) – number of nodes in healpix sampling

**Returns** resolution for the matching healpix graph

**Return type** int

`deepsphere.utils.samplings.icosahedron_nodes_calculator` (*order*)

Calculate the number of nodes corresponding to the order of an icosahedron graph

**Parameters** `order` (*int*) – order of an icosahedron graph

**Returns** number of nodes in icosahedron sampling for that order

**Return type** `int`

`deepsphere.utils.samplings.icosahedron_order_calculator` (*nodes*)

Calculate the order of a icosahedron graph for a given number of nodes.

**Parameters** `nodes` (*int*) – number of nodes in icosahedron sampling

**Returns** order for the matching icosahedron graph

**Return type** `int`

**deepsphere.utils.stats\_extractor module**

**Module contents**

## 1.2 Module contents

DeepSphere Base Documentation doc



## DEEPSPHERE.TESTS PACKAGE

### 2.1 Submodules

### 2.2 deepsphere.tests.test\_foo module

Fake file to test the doc

```
class deepsphere.tests.test_foo.TestFoo (methodName='runTest')  
    Bases: unittest.case.TestCase  
  
    Fake test class in order to setup the tests module  
  
    test_foo()  
        Fake test method in order to setup the test module
```

### 2.3 Module contents

The `tests` module contains different directory and files that have the goal to test different parts of the code

#### 2.3.1 Class

You can see in this module the `TestFoo` that contain the different method:

<code>TestFoo.test_foo()</code>	Fake test method in order to setup the test module
---------------------------------	--

#### 2.3.2 More Doc / Example

You can add then more doc and even examples



**MORE INFO**

So many info



## INDICES AND TABLES

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



## PYTHON MODULE INDEX

### d

- `deepsphere`, 4
- `deepsphere.data`, 1
- `deepsphere.data.datasets`, 1
- `deepsphere.data.transforms`, 1
- `deepsphere.layers`, 2
- `deepsphere.layers.samplings`, 2
- `deepsphere.models`, 2
- `deepsphere.models.spherical_unet`, 2
- `deepsphere.tests`, 5
- `deepsphere.tests.test_foo`, 5
- `deepsphere.utils`, 4
- `deepsphere.utils.parser`, 2
- `deepsphere.utils.samplings`, 3





## INDEX

### C

`create_parser()` (in module *deepsphere.utils.parser*), 2

### D

*deepsphere* (module), 4  
*deepsphere.data* (module), 1  
*deepsphere.data.datasets* (module), 1  
*deepsphere.data.transforms* (module), 1  
*deepsphere.layers* (module), 2  
*deepsphere.layers.samplings* (module), 2  
*deepsphere.models* (module), 2  
*deepsphere.models.spherical\_unet* (module), 2  
*deepsphere.tests* (module), 5  
*deepsphere.tests.test\_foo* (module), 5  
*deepsphere.utils* (module), 4  
*deepsphere.utils.parser* (module), 2  
*deepsphere.utils.samplings* (module), 3

### E

`equiangular_bandwidth()` (in module *deepsphere.utils.samplings*), 3  
`equiangular_calculator()` (in module *deepsphere.utils.samplings*), 3  
`equiangular_dimension_unpack()` (in module *deepsphere.utils.samplings*), 3

### H

`healpix_resolution_calculator()` (in module *deepsphere.utils.samplings*), 3

### I

`icosahedron_nodes_calculator()` (in module *deepsphere.utils.samplings*), 3  
`icosahedron_order_calculator()` (in module *deepsphere.utils.samplings*), 4

### P

`parse_config()` (in module *deepsphere.utils.parser*), 3

### T

`test_foo()` (*deepsphere.tests.test\_foo.TestFoo* method), 5  
*TestFoo* (class in *deepsphere.tests.test\_foo*), 5