

twinlab.CalibrateParams

```
class twinlab.CalibrateParams(y_std_model=False, return_summary=True,
iterations=10000, n_chains=2, force_sequential=False, seed=None)
```

Parameter configuration for inverting a trained emulator to estimate the input parameters that generated a given output.

Variables:

- **y_std_model** (*Union[[bool](#), [pd.DataFrame](#)], optional*) –
Whether to include model noise covariance in the likelihood.
 - If `True` TODO ...
 - If `False` TODO ...
 - If a *pandas.DataFrame* is supplied, it must contain the same columns as the set of emulator outputs.

The default value is `False`.

- **return_summary** (*[bool](#), optional*) –
Should the result of the inversion be presented as a summary or as the full solution?
 - If `True` then return a summary of the inverse solution.
 - If `False` return the entire solution in the form of the points sampled.

The default value is `True`.

- **iterations** (*[int](#), optional*) – The number of points to sample in each inversion chain. More points is better. The default value is `10,000`.
- **n_chains** (*[int](#), optional*) – The number of independent chains to use for the inversion process. More is better, so that the solution derived between indepent chains can be compared and convergence can be checked. The default value is `2`.
- **force_sequential** (*[bool](#), optional*) – TODO The default value is False.
- **seed** (*Union[[int](#), [None](#)], optional*) – Specifies the seed used by the random number generator to start the inversion process. Setting the seed to an integer is good for reproducibility. The default value is `None`, which means the seed is randomly generated each time.

```
__init__(y_std_model=False, return_summary=True, iterations=10000,
```

Methods

`__init__`([y_std_model, return_summary, ...])

`unpack_parameters`()

< Previous
[twinlab.RecommendParams](#)

Next >
[Distributions](#)

© Copyright 2024, twinLab Dev Team.

Created using [Sphinx](#) 7.3.7.

Built with the [PyData Sphinx Theme](#) 0.15.2.