twinlab.Emulator.calibrate

Emulator.**calibrate**(*df_obs, df_std, params=<twinlab.params.CalibrateParams object>, wait=True, verbose=False*)

Solve an inverse problem using a trained emulator on the twinLab cloud.

A classic trained emulator can ingest X values and use these to predict corresponding y values. However, the emulator can also be used to solve an inverse problem, where the user has an observation of y and wants to find the corresponding X. Problems of this type are common in engineering and science, where the user has an observation of a system and wants to find the parameters that generated that observation. This operation can be numerically intensive, and the emulator can be used to solve this problem quickly and efficiently.

Parameters:

- **df_obs** (*pandas.DataFrame*) – A dataframe containing the single observation.
- **df_std** (*pandas.DataFrame*) – A dataframe containing the error on the single observation.
- **params** (*CalibrateParams, optional*) – A parameter configuration that contains all optional calibration parameters.
- **wait** (*bool, optional*) – If `True` wait for the job to complete, otherwise return the process ID and exit.
- **verbose** (*bool, optional*) – Display detailed information about the operation while running.

Returns:

By default, the solution to the inverse problem is either presented as a summary, or as the full set of points sampled from the posterior distribution. See the documentation for `CalibrateParams` for more information on the different options. Instead, if `wait=False`, the process ID is returned. The results can then be retrieved later using `Emulator.get_process(<process_id>)`. Process IDs associated with an emulator can be found using `Emulator.list_processes()`.

Return type:

pandas.DataFrame, str

Example

```
df_std = pd.DataFrame({'y': [0.01]})
emulator.calibrate(df_obs, df_std)
```

```
       mean      sd  hdi_3%  hdi_97%  mcse_mean  mcse_sd  ess_bulk  ess_tail  r_hat
x     0.496   0.013   0.471    0.521        0.0      0.0    2025.0    2538.0    1.0
```