twopc Documentation

J. Michael Burgess

Aug 24, 2021

CONTENTS

1 Indices and tables 25

A simple library to generate posterior predictive checks from 3ML bayesian fits to spectra. Ypu should do this because if feels good and is the best way to know if your data match your modeling.

{

"cells": [

{ "cell_type": "markdown", "id": "0703e08d", "metadata": {}, "source": [

"# Creating PPC from GBM SpectrumLike fitsn", "n", "A very useful way to check the quality of Bayesian fits is through posterior predicitve checks.n", "We can create these with the machinery inside of 3ML for certain plugins (perhaps more in the future).n", "n"

]

}, {

"cell_type": "code", "execution_count": 1, "id": "ca29fe49", "metadata": {

"execution": { "iopub.execute_input": "2021-08-24T15:09:33.262786Z", "iopub.status.busy": "2021-08-24T15:09:33.259376Z", "iopub.status.idle": "2021-08-24T15:09:37.781301Z", "shell.execute_reply": "2021-08-24T15:09:37.780481Z"

}, "lines_to_next_cell": 2

}, "outputs": [

{ "name": "stdout", "output_type": "stream", "text": [

"[u001b[35mWARNING u001b[0m]u001b[35m The naima package is not available. Models that depend on it will not be availableu001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[35mWARNING u001b[0m]u001b[35m The GSL library or the pygsl wrapper cannot be loaded. Models that depend on it will not be available.u001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[35mWARNING u001b[0m]u001b[35m The ebltable package is not available. Models that depend on it will not be availableu001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m Starting 3ML!u001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[35mWARNING u001b[0m]u001b[35m no display variable set. using backend for graphics without display (agg)u001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[35mWARNING u001b[0m]u001b[35m ROOT minimizer not availableu001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[35mWARNING u001b[0m]u001b[35m Multinest minimizer not availableu001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[35mWARNING u001b[0m]u001b[35m PyGMO is not availableu001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[35mWARNING u001b[0m]u001b[35m The cthreeML package is not installed. You will not be able to use plugins which require the C/C++ interface (currently HAWC)u001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[35mWARNING u001b[0m]u001b[35m Could not import plugin FermiLATLike.py. Do you have the relative instrument software installed and configured?u001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[35mWARNING u001b[0m]u001b[35m Could not import plugin HAWCLike.py. Do you have the relative instrument software installed and configured?u001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[35mWARNING u001b[0m]u001b[35m Env. variable OMP_NUM_THREADS is not set. Please set it to 1 for optimal performances in 3MLu001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[35mWARNING u001b[0m]u001b[35m Env. variable MKL_NUM_THREADS is not set. Please set it to 1 for optimal performances in 3MLu001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[35mWARNING u001b[0m]u001b[35m Env. variable NUMEXPR_NUM_THREADS is not set. Please set it to 1 for optimal performances in 3MLu001b[0mn"

]

}

], "source": [

"import numpy as npn", "from threeML.io.package_data import get_path_of_data_filen", "from threeML.utils.OGIP.response import OGIPResponsen", "from threeML import (n", " silence_warnings,n", " DispersionSpectrumLike,n", " display_spectrum_model_counts,n", ")n", "from threeML import set_threeML_style, BayesianAnalysis, DataList, threeML_confign", "from astromodels import Blackbody, Powerlaw, PointSource, Model, Log_normaln", "n", "threeML_config.plugins.ogip.data_plot.counts_color = "#FCE902"n", "threeML_config.plugins.ogip.data_plot.background_color = "#CC0000""

]

}, {

"cell_type": "code", "execution_count": 2, "id": "a8cff239", "metadata": {

"execution": { "iopub.execute_input": "2021-08-24T15:09:37.790102Z", "iopub.status.busy": "2021-08-24T15:09:37.789456Z", "iopub.status.idle": "2021-08-24T15:09:37.801153Z", "shell.execute_reply": "2021-08-24T15:09:37.801820Z"

}, "lines_to_next_cell": 0, "nbsphinx": "hidden"

}, "outputs": [], "source": [

"from jupyterthemes import jtplotn", "n", "%matplotlib inlinen", "jtplot.style(context="talk", fscale=1, ticks=True, grid=False)n", "n", "silence_warnings()n", "n", "set_threeML_style()n", "n", "threeML_config.plugins.ogip.data_plot.counts_color = "#FCE902"n", "threeML_config.plugins.ogip.data_plot.background_color = "#CC0000"n", "n", "n", "threeML_config.plugins.ogip.fit_plot.background_mpl_kwargs = dict(ls="-", lw=0.7)n", "threeML_config.interface.multi_progress_cmap = "Reds""

]

}, {

"cell_type": "markdown", "id": "19a1e337", "metadata": {}, "source": [

"## Generate some synthetic datan", "n", "First, lets use 3ML to generate some fake data. Here we will have a background spectrum that is a power law and a source that is a black body.n", "n", "### Grab a demo response from the 3ML library"

]

}, {

"cell_type": "code", "execution_count": 3, "id": "9394d05a", "metadata": {

"execution": { "iopub.execute_input": "2021-08-24T15:09:37.805957Z", "iopub.status.busy": "2021-08-24T15:09:37.804589Z", "iopub.status.idle": "2021-08-24T15:09:37.830106Z", "shell.execute_reply": "2021-08-24T15:09:37.828956Z"

}

}, "outputs": [], "source": [

"# we will use a demo responsen", "response = OGIPResponse(get_path_of_data_file("datasets/ogip_powerlaw.rsp"))"

]

}, {

"cell_type": "markdown", "id": "5d6a3b84", "metadata": {}, "source": [

"### Simulate the datan", "n", "We will mimic the some normal c-stat style data."

]

}, {

"cell_type": "code", "execution_count": 4, "id": "56d4e417", "metadata": {

"execution": { "iopub.execute_input": "2021-08-24T15:09:37.837371Z", "iopub.status.busy": "2021-08-24T15:09:37.836579Z", "iopub.status.idle": "2021-08-24T15:09:42.337891Z", "shell.execute_reply": "2021-08-24T15:09:42.338380Z"

}

}, "outputs": [

{ "name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m Auto-probed noise models:u001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m - observation: poissonu001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m - background: Noneu001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m Auto-probed noise models:u001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m - observation: poissonu001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m - background: Noneu001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m Auto-probed noise models:u001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m - observation: poissonu001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m - background: poissonu001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m Auto-probed noise models:u001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m - observation: poissonu001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m - background: poissonu001b[0mn"

]

}, {

"data": { "image/png": "iVBORw0KGgoAAAANSUhEUgAAApYAAAIuCAYAAAAFa0gSAAAAOXRFWH

"<Figure size 748.8x655.2 with 1 Axes>"
]
}, "metadata": {}, "output_type": "display_data"
}
], "source": [
"np.random.seed(1234)n", "n", "# rescale the functions for the responsen", "source_function = Blackbody(K=1e-7, kT=500.0)n", "background_function = Powerlaw(K=1, index=-1.5, piv=1.0e3)n", "spectrum_generator = DispersionSpectrumLike.from_function(n", " "fake",n", " source_function=source_function,n", " background_function=background_function,n", " response=response,n", ")n", "n", "fig = spectrum_generator.view_count_spectrum()"
]
}, {
"cell_type": "markdown", "id": "1cc09071", "metadata": {}, "source": [
"## Fit the datan", "n", "We will quickly fit the data via Bayesian posterior sampling… After all you need a posterior to do posterior predictive checks!n"
]
}, {
"cell_type": "code", "execution_count": 5, "id": "456a47df", "metadata": {
"execution": { "iopub.execute_input": "2021-08-24T15:09:42.347075Z", "iopub.status.busy": "2021-08-24T15:09:42.346432Z", "iopub.status.idle": "2021-08-24T15:09:42.349790Z", "shell.execute_reply": "2021-08-24T15:09:42.349202Z"
}
}, "outputs": [], "source": [
"source_function.K.prior = Log_normal(mu=np.log(1e-7), sigma=1)n", "source_function.kT.prior = Log_normal(mu=np.log(300), sigma=2)n", "n", "ps = PointSource("demo", 0, 0, spectral_shape=source_function)n", "n", "model = Model(ps)"
]
}, {
"cell_type": "code", "execution_count": 6, "id": "9ba5280d", "metadata": {
"execution": { "iopub.execute_input": "2021-08-24T15:09:42.356400Z", "iopub.status.busy": "2021-08-24T15:09:42.354819Z", "iopub.status.idle": "2021-08-24T15:09:42.357066Z", "shell.execute_reply": "2021-08-24T15:09:42.357574Z"
}
}, "outputs": [], "source": [
"ba = BayesianAnalysis(model, DataList(spectrum_generator))"
]
}, {

"cell_type": "code", "execution_count": 7, "id": "7d0a8182", "metadata": {

"execution": { "iopub.execute_input": "2021-08-24T15:09:42.362900Z", "iopub.status.busy": "2021-08-24T15:09:42.362244Z", "iopub.status.idle": "2021-08-24T15:10:09.148475Z", "shell.execute_reply": "2021-08-24T15:10:09.149092Z"

}

}, "outputs": [

{ "name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m sampler set to emceeu001b[0mn"

]

}, {

"data": {

"application/vnd.jupyter.widget-view+json": { "model_id": "321b8709103540cc9100bc59f347c8ad", "version_major": 2, "version_minor": 0

}, "text/plain": [

" 0%| | 0/250 [00:00<?, ?it/s]"

]

}, "metadata": {}, "output_type": "display_data"

}, {

"data": {

"application/vnd.jupyter.widget-view+json": { "model_id": "8eae23c6d15f4d55b5c11ffd325cea1c", "version_major": 2, "version_minor": 0

}, "text/plain": [

" 0%| | 0/1000 [00:00<?, ?it/s]"

]

}, "metadata": {}, "output_type": "display_data"

}, {

"name": "stdout", "output_type": "stream", "text": [

"[u001b[32mINFO u001b[0m]u001b[32m Mean acceptance fraction: 0.71642u001b[0mn"

]

}, {

"name": "stdout", "output_type": "stream", "text": [

"Maximum a posteriori probability (MAP) point:n", "n"

]

}, {

"data": {

“text/html”: [“ n”, “<style scoped>n”, ” .dataframe tbody tr th:only-of-type {n”, ” vertical-align: middle;n”, ” }n”, “n”, ” .dataframe tbody tr th {n”, ” vertical-align: top;n”, ” }n”, “n”, ” .dataframe thead th {n”, ” text-align: right;n”, ” }n”, “</style>n”, “<table border="1" class="dataframe">n”, ” <thead>n”, ” <tr style="text-align: right;">n”, ” <th></th>n”, ” <th>result</th>n”, ” <th>unit</th>n”, ” </tr>n”, ” <tr>n”, ” <th>parameter</th>n”, ” <th></th>n”, ” <th></th>n”, ” </tr>n”, ” </thead>n”, ” <tbody>n”, ” <tr>n”, ” <th>demo.spectrum.main.Blackbody.K</th>n”, ” <td>(9.7 +/- 0.5) x 10^-8</td>n”, ” <td>1 / (cm2 keV3 s)</td>n”, ” </tr>n”, ” <tr>n”, ” <th>demo.spectrum.main.Blackbody.kT</th>n”, ” <td>(5.06 +/- 0.09) x 10^2</td>n”, ” <td>keV</td>n”, ” </tr>n”, ” </tbody>n”, “</table>n”, “ ”

], “text/plain”: [

” result unitn”, “parameter n”, “demo.spectrum.main.Blackbody.K (9.7 +/- 0.5) x 10^-8 1 / (cm2 keV3 s)n”, “demo.spectrum.main.Blackbody.kT (5.06 +/- 0.09) x 10^2 keV”

]

}, “metadata”: {}, “output_type”: “display_data”

}, {

“name”: “stdout”, “output_type”: “stream”, “text”: [

“n”, “Values of -log(posterior) at the minimum:n”, “n”

]

}, {

“data”: {

“text/html”: [“ n”, “<style scoped>n”, ” .dataframe tbody tr th:only-of-type {n”, ” vertical-align: middle;n”, ” }n”, “n”, ” .dataframe tbody tr th {n”, ” vertical-align: top;n”, ” }n”, “n”, ” .dataframe thead th {n”, ” text-align: right;n”, ” }n”, “</style>n”, “<table border="1" class="dataframe">n”, ” <thead>n”, ” <tr style="text-align: right;">n”, ” <th></th>n”, ” <th>-log(posterior)</th>n”, ” </tr>n”, ” </thead>n”, ” <tbody>n”, ” <tr>n”, ” <th>fake</th>n”, ” <td>-631.515661</td>n”, ” </tr>n”, ” <tr>n”, ” <th>total</th>n”, ” <td>-631.515661</td>n”, ” </tr>n”, ” </tbody>n”, “</table>n”, “ ”

], “text/plain”: [

” -log(posterior)n”, “fake -631.515661n”, “total -631.515661”

]

}, “metadata”: {}, “output_type”: “display_data”

}, {

“name”: “stdout”, “output_type”: “stream”, “text”: [

“n”, “Values of statistical measures:n”, “n”

]

}, {

"data": {

"text/html": [" n", "<style scoped>n", " .dataframe tbody tr th:only-of-type {n", " vertical-align: middle;n", " }n", "n", " .dataframe tbody tr th {n", " vertical-align: top;n", " }n", "n", " .dataframe thead th {n", " text-align: right;n", " }n", "</style>n", "<table border="1" class="dataframe">n", " <thead>n", " <tr style="text-align: right;">n", " <th></th>n", " <th>statistical measures</th>n", " </tr>n", " </thead>n", " <tbody>n", " <tr>n", " <th>AIC</th>n", " <td>1267.127322</td>n", " </tr>n", " <tr>n", " <th>BIC</th>n", " <td>1272.735383</td>n", " </tr>n", " <tr>n", " <th>DIC</th>n", " <td>1267.112042</td>n", " </tr>n", " <tr>n", " <th>PDIC</th>n", " <td>2.029160</td>n", " </tr>n", " </tbody>n", "</table>n", " "

], "text/plain": [

" statistical measuresn", "AIC 1267.127322n", "BIC 1272.735383n", "DIC 1267.112042n", "PDIC 2.029160"

]

}, "metadata": {}, "output_type": "display_data"

}

], "source": [

"ba.set_sampler("emcee")n", "ba.sampler.setup(n_iterations=1000, n_walkers="50")n", "ba.sample(quiet=False)"

]

}, {

"cell_type": "markdown", "id": "4811f8d8", "metadata": {}, "source": [

"Ok, we have a pretty decent fit. But is it?"

]

}, {

"cell_type": "code", "execution_count": 8, "id": "46cb19bc", "metadata": {

"execution": { "iopub.execute_input": "2021-08-24T15:10:09.170356Z", "iopub.status.busy": "2021-08-24T15:10:09.153829Z", "iopub.status.idle": "2021-08-24T15:10:14.972438Z", "shell.execute_reply": "2021-08-24T15:10:14.971782Z"

}, "lines_to_next_cell": 2

}, "outputs": [

{ "name": "stderr", "output_type": "stream", "text": [

"n", "WARNING MatplotlibDeprecationWarning: The 'nonposy' parameter of __init__() has been renamed 'nonpositive' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.n", "n"

]

}, {

"data": { "image/png": "iVBORw0KGgoAAAANSUhEUgAAAtsAAAKBCAYAAACGZWc5AAAAOXR

"<Figure size 748.8x655.2 with 2 Axes>"

```
                ]
            }, "metadata": {}, "output_type": "display_data"
        }
    ], "source": [
        "ba.restore_median_fit()n", "fig = display_spectrum_model_counts(n", " ba,n", " data_color="#CC0000",n", " model_color="#FCE902",n", " background_color="k",n", " show_background=True,n", " source_only=False,n", " min_rate=10,n", ")n", "ax = fig.get_axes()[0]n", "n", "_ = ax.set_ylim(1e-3)"
    ]
}, {
    "cell_type": "markdown", "id": "c11d68cc", "metadata": {}, "source": [
        "## Compute the PPCsn", "n", "We want to check the validiity of the fit via posterior predicitive checks (PPCs).n", "Essentially:n", "n", "$$\pi\left(y^{\mathrm{rep}} \mid y\right)=\int \mathrm{d} \theta \pi(\theta \mid y) \pi\left(y^{\mathrm{rep}} \mid \theta\right)$$n", "n", "n", "So, we will randomly sample the posterior, fold those point of the posterior model throught the likelihood and sample new counts. As this is an expensive process, we will want to save this to disk (in an HDF5 file).n"
    ]
}, {
    "cell_type": "code", "execution_count": 9, "id": "a5d5f626", "metadata": {
        "execution": { "iopub.execute_input": "2021-08-24T15:10:14.977254Z", "iopub.status.busy": "2021-08-24T15:10:14.976548Z", "iopub.status.idle": "2021-08-24T15:10:14.988626Z", "shell.execute_reply": "2021-08-24T15:10:14.989137Z"
        }
    }, "outputs": [], "source": [
        "from twopc import compute_ppc"
    ]
}, {
    "cell_type": "code", "execution_count": 10, "id": "b88d6e2a", "metadata": {
        "execution": { "iopub.execute_input": "2021-08-24T15:10:14.994430Z", "iopub.status.busy": "2021-08-24T15:10:14.993817Z", "iopub.status.idle": "2021-08-24T15:10:31.477080Z", "shell.execute_reply": "2021-08-24T15:10:31.476432Z"
        }
    }, "outputs": [
        {
            "data": {
                "application/vnd.jupyter.widget-view+json": { "model_id": "ce8c91244c6c4d0983371a04d96487ce", "version_major": 2, "version_minor": 0
```

}, “text/plain”: [

“sampling posterior: 0%| | 0/1000 [00:00<?, ?it/s]”

]

}, “metadata”: {}, “output_type”: “display_data”

}

], “source”: [

“ppc = compute_ppc(n”, ” ba, ba.results, n_sims=1000, file_name="my_ppc.h5", overwrite=True, return_ppc=Truen”, “)”

]

}, {

“cell_type”: “markdown”, “id”: “2da199b0”, “metadata”: {}, “source”: [

“Lets take a look at the results of the background substracted rate”

]

}, {

“cell_type”: “code”, “execution_count”: 11, “id”: “f0f29a60”, “metadata”: {

“execution”: { “iopub.execute_input”: “2021-08-24T15:10:31.504484Z”, “iopub.status.busy”: “2021-08-24T15:10:31.503632Z”, “iopub.status.idle”: “2021-08-24T15:10:34.161219Z”, “shell.execute_reply”: “2021-08-24T15:10:34.161837Z”

}

}, “outputs”: [

{

“data”: { “image/png”: “iVBORw0KGgoAAAANSUhEUgAAApYAAAIuCAYAAAAFa0gSAAAAOXRF
“text/plain”: [

“<Figure size 748.8x655.2 with 1 Axes>”

]

}, “metadata”: {}, “output_type”: “display_data”

}

], “source”: [

“fig = ppc.fake.plot(bkg_subtract=True, colors=["#FF1919", "#CC0000", "#7F0000"])”

]

}, {

“cell_type”: “markdown”, “id”: “d7982521”, “metadata”: {}, “source”: [

“And the PPCs with source + background”

]

}, {

“cell_type”: “code”, “execution_count”: 12, “id”: “23e714b6”, “metadata”: {

“execution”: { “iopub.execute_input”: “2021-08-24T15:10:34.188100Z”, “iopub.status.busy”: “2021-08-24T15:10:34.186766Z”, “iopub.status.idle”: “2021-08-24T15:10:37.196659Z”, “shell.execute_reply”: “2021-08-24T15:10:37.196118Z”

}

}, “outputs”: [

{

“data”: { “image/png”: “iVBORw0KGgoAAAANSUhEUgAAApYAAAIuCAYAAAAFa0gSAAAAOXRF

“<Figure size 748.8x655.2 with 1 Axes>”

]

}, “metadata”: {}, “output_type”: “display_data”

}

], “source”: [

“fig = ppc.fake.plot(bkg_subtract=False, colors=["#FF1919", "#CC0000", "#7F0000"])”

]

}, {

“cell_type”: “markdown”, “id”: “20bb5644”, “metadata”: {}, “source”: [

“** AWESOME! ** It looks like our fit and accurately produce future data!”

]

}, {

“cell_type”: “code”, “execution_count”: null, “id”: “8011a706”, “metadata”: {}, “outputs”: [], “source”: []

}

], “metadata”: {

“jupytext”: { “formats”: “ipynb,md”

}, “kernelspec”: {

“display_name”: “Python 3”, “language”: “python”, “name”: “python3”

}, “language_info”: {

“codemirror_mode”: { “name”: “ipython”, “version”: 3

}, “file_extension”: “.py”, “mimetype”: “text/x-python”, “name”: “python”, “nbconvert_exporter”: “python”, “pygments_lexer”: “ipython3”, “version”: “3.8.11”

}, “widgets”: {

“application/vnd.jupyter.widget-state+json”: {

“state”: {

“1b6b06a2312742de82eca7b5559ed708”: { “model_module”: “@jupyter-widgets/controls”, “model_module_version”: “1.5.0”, “model_name”: “FloatProgressModel”, “state”: {

"_dom_classes": [], "_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "FloatProgressModel", "_view_count": null, "_view_module": "@jupyter-widgets/controls", "_view_module_version": "1.5.0", "_view_name": "ProgressView", "bar_style": "success", "description": "", "description_tooltip": null, "layout": "IPY_MODEL_1f10834375494d639c7c4b400201b306", "max": 1000.0, "min": 0.0, "orientation": "horizontal", "style": "IPY_MODEL_4e136e4935a7434b95672ae461a3f501", "value": 1000.0

}

}, "1f10834375494d639c7c4b400201b306": {

"model_module": "@jupyter-widgets/base", "model_module_version": "1.2.0", "model_name": "LayoutModel", "state": {

"_model_module": "@jupyter-widgets/base", "_model_module_version": "1.2.0", "_model_name": "LayoutModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "LayoutView", "align_content": null, "align_items": null, "align_self": null, "border": null, "bottom": null, "display": null, "flex": "2", "flex_flow": null, "grid_area": null, "grid_auto_columns": null, "grid_auto_flow": null, "grid_auto_rows": null, "grid_column": null, "grid_gap": null, "grid_row": null, "grid_template_areas": null, "grid_template_columns": null, "grid_template_rows": null, "height": null, "justify_content": null, "justify_items": null, "left": null, "margin": null, "max_height": null, "max_width": null, "min_height": null, "min_width": null, "object_fit": null, "object_position": null, "order": null, "overflow": null, "overflow_x": null, "overflow_y": null, "padding": null, "right": null, "top": null, "visibility": null, "width": null

}

}, "2dab145f330345f3a5b38fe44ef5da25": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "HTMLModel", "state": {

"_dom_classes": [], "_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "HTMLModel", "_view_count": null, "_view_module": "@jupyter-widgets/controls", "_view_module_version": "1.5.0", "_view_name": "HTMLView", "description": "", "description_tooltip": null, "layout": "IPY_MODEL_8ec7844c0726434da4b023f494ed86eb", "placeholder": "", "style": "IPY_MODEL_fcd72c266765484ca1d3466683960019", "value": "100%"

}

}, "2edd5e1fc8a14a299469b9120fdfd390": {

"model_module": "@jupyter-widgets/controls",

"model_module_version": "1.5.0", "model_name": "ProgressStyleModel", "state": {

"_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "ProgressStyleModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "StyleView", "bar_color": null, "description_width": ""

}

}, "2eed223f01cb43729142ea11c611c71b": {

"model_module": "@jupyter-widgets/base", "model_module_version": "1.2.0", "model_name": "LayoutModel", "state": {

"_model_module": "@jupyter-widgets/base", "_model_module_version": "1.2.0", "_model_name": "LayoutModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "LayoutView", "align_content": null, "align_items": null, "align_self": null, "border": null, "bottom": null, "display": null, "flex": null, "flex_flow": null, "grid_area": null, "grid_auto_columns": null, "grid_auto_flow": null, "grid_auto_rows": null, "grid_column": null, "grid_gap": null, "grid_row": null, "grid_template_areas": null, "grid_template_columns": null, "grid_template_rows": null, "height": null, "justify_content": null, "justify_items": null, "left": null, "margin": null, "max_height": null, "max_width": null, "min_height": null, "min_width": null, "object_fit": null, "object_position": null, "order": null, "overflow": null, "overflow_x": null, "overflow_y": null, "padding": null, "right": null, "top": null, "visibility": null, "width": null

}

}, "321b8709103540cc9100bc59f347c8ad": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "HBoxModel", "state": {

"_dom_classes": [], "_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "HBoxModel", "_view_count": null, "_view_module": "@jupyter-widgets/controls", "_view_module_version": "1.5.0", "_view_name": "HBoxView", "box_style": "", "children": [

"IPY_MODEL_41ea0bc11be84415bb288f3735d459c0",
"IPY_MODEL_ad507479d6974a08bba099e7f5ebf681",
"IPY_MODEL_f398009217034a1a9fea93bf23c9beea"

], "layout": "IPY_MODEL_a23f439a41fe4d29a577f5fd559af4cf"

}

}, "3eb54496faf04b33b9b5df36a7a1544b": {

"model_module": "@jupyter-widgets/base", "model_module_version": "1.2.0", "model_name": "LayoutModel", "state": {

"_model_module": "@jupyter-widgets/base", "_model_module_version": "1.2.0", "_model_name": "LayoutModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "LayoutView", "align_content": null, "align_items": null, "align_self": null, "border": null, "bottom": null, "display": null, "flex": null, "flex_flow": null, "grid_area": null, "grid_auto_columns": null, "grid_auto_flow": null, "grid_auto_rows": null, "grid_column": null, "grid_gap": null, "grid_row": null, "grid_template_areas": null, "grid_template_columns": null, "grid_template_rows": null, "height": null, "justify_content": null, "justify_items": null, "left": null, "margin": null, "max_height": null, "max_width": null, "min_height": null, "min_width": null, "object_fit": null, "object_position": null, "order": null, "overflow": null, "overflow_x": null, "overflow_y": null, "padding": null, "right": null, "top": null, "visibility": null, "width": null

}

}, "4149e77e943c4d16a0f6edf3f618f4a5": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "DescriptionStyleModel", "state": {

"_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "DescriptionStyleModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "StyleView", "description_width": ""

}

}, "41ea0bc11be84415bb288f3735d459c0": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "HTMLModel", "state": {

"_dom_classes": [], "_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "HTMLModel", "_view_count": null, "_view_module": "@jupyter-widgets/controls", "_view_module_version": "1.5.0", "_view_name": "HTMLView", "description": "", "description_tooltip": null, "layout": "IPY_MODEL_f1091902982d43fd8d1c5da523bec127", "placeholder": "", "style": "IPY_MODEL_90a517716ed145918c5634ae3c70118c", "value": "100%"

}

}, "435e27dd8f714ef2a0ee3b2005c18ce1": {

"model_module": "@jupyter-widgets/base", "model_module_version": "1.2.0", "model_name": "LayoutModel",

"state": {

"_model_module": "@jupyter-widgets/base", "_model_module_version": "1.2.0", "_model_name": "LayoutModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "LayoutView", "align_content": null, "align_items": null, "align_self": null, "border": null, "bottom": null, "display": null, "flex": null, "flex_flow": null, "grid_area": null, "grid_auto_columns": null, "grid_auto_flow": null, "grid_auto_rows": null, "grid_column": null, "grid_gap": null, "grid_row": null, "grid_template_areas": null, "grid_template_columns": null, "grid_template_rows": null, "height": null, "justify_content": null, "justify_items": null, "left": null, "margin": null, "max_height": null, "max_width": null, "min_height": null, "min_width": null, "object_fit": null, "object_position": null, "order": null, "overflow": null, "overflow_x": null, "overflow_y": null, "padding": null, "right": null, "top": null, "visibility": null, "width": null

}

}, "444df54a412b4ab6af6acf074c24068f": {

"model_module": "@jupyter-widgets/base", "model_module_version": "1.2.0", "model_name": "LayoutModel", "state": {

"_model_module": "@jupyter-widgets/base", "_model_module_version": "1.2.0", "_model_name": "LayoutModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "LayoutView", "align_content": null, "align_items": null, "align_self": null, "border": null, "bottom": null, "display": "inline-flex", "flex": null, "flex_flow": "row wrap", "grid_area": null, "grid_auto_columns": null, "grid_auto_flow": null, "grid_auto_rows": null, "grid_column": null, "grid_gap": null, "grid_row": null, "grid_template_areas": null, "grid_template_columns": null, "grid_template_rows": null, "height": null, "justify_content": null, "justify_items": null, "left": null, "margin": null, "max_height": null, "max_width": null, "min_height": null, "min_width": null, "object_fit": null, "object_position": null, "order": null, "overflow": null, "overflow_x": null, "overflow_y": null, "padding": null, "right": null, "top": null, "visibility": null, "width": "100%"

}

}, "48b09d6b10ea4896b35f184388059d7d": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "ProgressStyleModel", "state": {

"_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "ProgressStyleModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0",

"_view_name": "StyleView", "bar_color": null, "description_width": ""

}

}, "4bf0acfcb9d746ebb87d3de33ddaea47": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "FloatProgressModel", "state": {

"_dom_classes": [], "_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "FloatProgressModel", "_view_count": null, "_view_module": "@jupyter-widgets/controls", "_view_module_version": "1.5.0", "_view_name": "ProgressView", "bar_style": "success", "description": "", "description_tooltip": null, "layout": "IPY_MODEL_4ea44658a843408ab02e63546060c3cb", "max": 1000.0, "min": 0.0, "orientation": "horizontal", "style": "IPY_MODEL_48b09d6b10ea4896b35f184388059d7d", "value": 1000.0

}

}, "4e136e4935a7434b95672ae461a3f501": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "ProgressStyleModel", "state": {

"_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "ProgressStyleModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "StyleView", "bar_color": "#440154", "description_width": ""

}

}, "4ea44658a843408ab02e63546060c3cb": {

"model_module": "@jupyter-widgets/base", "model_module_version": "1.2.0", "model_name": "LayoutModel", "state": {

"_model_module": "@jupyter-widgets/base", "_model_module_version": "1.2.0", "_model_name": "LayoutModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "LayoutView", "align_content": null, "align_items": null, "align_self": null, "border": null, "bottom": null, "display": null, "flex": null, "flex_flow": null, "grid_area": null, "grid_auto_columns": null, "grid_auto_flow": null, "grid_auto_rows": null, "grid_column": null, "grid_gap": null, "grid_row": null, "grid_template_areas": null, "grid_template_columns": null, "grid_template_rows": null, "height": null, "justify_content": null, "justify_items": null, "left": null, "margin": null, "max_height": null, "max_width": null, "min_height": null, "min_width": null, "object_fit": null,

"object_position": null, "order": null, "overflow": null, "overflow_x": null, "overflow_y": null, "padding": null, "right": null, "top": null, "visibility": null, "width": null

}

}, "5a649ae7052944b59e52da6ee278902b": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "HTMLModel", "state": {

"_dom_classes": [], "_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "HTMLModel", "_view_count": null, "_view_module": "@jupyter-widgets/controls", "_view_module_version": "1.5.0", "_view_name": "HTMLView", "description": "", "description_tooltip": null, "layout": "IPY_MODEL_5ae5893c1e2c47c8897717efe8eaeaf0", "placeholder": "", "style": "IPY_MODEL_7bdd3da1758c449fa450aefcd3597f0f", "value": " 1000/1000 [00:15<00:00, 64.31it/s]"

}

}, "5ae5893c1e2c47c8897717efe8eaeaf0": {

"model_module": "@jupyter-widgets/base", "model_module_version": "1.2.0", "model_name": "LayoutModel", "state": {

"_model_module": "@jupyter-widgets/base", "_model_module_version": "1.2.0", "_model_name": "LayoutModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "LayoutView", "align_content": null, "align_items": null, "align_self": null, "border": null, "bottom": null, "display": null, "flex": null, "flex_flow": null, "grid_area": null, "grid_auto_columns": null, "grid_auto_flow": null, "grid_auto_rows": null, "grid_column": null, "grid_gap": null, "grid_row": null, "grid_template_areas": null, "grid_template_columns": null, "grid_template_rows": null, "height": null, "justify_content": null, "justify_items": null, "left": null, "margin": null, "max_height": null, "max_width": null, "min_height": null, "min_width": null, "object_fit": null, "object_position": null, "order": null, "overflow": null, "overflow_x": null, "overflow_y": null, "padding": null, "right": null, "top": null, "visibility": null, "width": null

}

}, "64a4559bb9b74c30b7daec3c2a1a10f9": {

"model_module": "@jupyter-widgets/base", "model_module_version": "1.2.0", "model_name": "LayoutModel", "state": {

"_model_module": "@jupyter-widgets/base", "_model_module_version": "1.2.0", "_model_name": "LayoutModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0",

"_view_name": "LayoutView", "align_content": null, "align_items": null, "align_self": null, "border": null, "bottom": null, "display": null, "flex": null, "flex_flow": null, "grid_area": null, "grid_auto_columns": null, "grid_auto_flow": null, "grid_auto_rows": null, "grid_column": null, "grid_gap": null, "grid_row": null, "grid_template_areas": null, "grid_template_columns": null, "grid_template_rows": null, "height": null, "justify_content": null, "justify_items": null, "left": null, "margin": null, "max_height": null, "max_width": null, "min_height": null, "min_width": null, "object_fit": null, "object_position": null, "order": null, "overflow": null, "overflow_x": null, "overflow_y": null, "padding": null, "right": null, "top": null, "visibility": null, "width": null

}

}, "6927b84a7a214439b18c722a59b15e8a": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "HTMLModel", "state": {

"_dom_classes": [], "_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "HTMLModel", "_view_count": null, "_view_module": "@jupyter-widgets/controls", "_view_module_version": "1.5.0", "_view_name": "HTMLView", "description": "", "description_tooltip": null, "layout": "IPY_MODEL_435e27dd8f714ef2a0ee3b2005c18ce1", "placeholder": "", "style": "IPY_MODEL_4149e77e943c4d16a0f6edf3f618f4a5", "value": " 1000/1000 [00:16<00:00, 62.68it/s]"

}

}, "7bdd3da1758c449fa450aefcd3597f0f": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "DescriptionStyleModel", "state": {

"_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "DescriptionStyleModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "StyleView", "description_width": ""

}

}, "8eae23c6d15f4d55b5c11ffd325cea1c": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "HBoxModel", "state": {

"_dom_classes": [], "_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "HBoxModel", "_view_count": null, "_view_module": "@jupyter-widgets/controls", "_view_module_version": "1.5.0", "_view_name": "HBoxView", "box_style": "", "children": [

"IPY_MODEL_2dab145f330345f3a5b38fe44ef5da25",
"IPY_MODEL_4bf0acfcb9d746ebb87d3de33ddaea47",
"IPY_MODEL_6927b84a7a214439b18c722a59b15e8a"

], "layout": "IPY_MODEL_2eed223f01cb43729142ea11c611c71b"

}

}, "8eb5e492e4cf4d92afcb75a985b5379f": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "DescriptionStyleModel", "state": {

"_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "DescriptionStyleModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "StyleView", "description_width": ""

}

}, "8ec7844c0726434da4b023f494ed86eb": {

"model_module": "@jupyter-widgets/base", "model_module_version": "1.2.0", "model_name": "LayoutModel", "state": {

"_model_module": "@jupyter-widgets/base", "_model_module_version": "1.2.0", "_model_name": "LayoutModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "LayoutView", "align_content": null, "align_items": null, "align_self": null, "border": null, "bottom": null, "display": null, "flex": null, "flex_flow": null, "grid_area": null, "grid_auto_columns": null, "grid_auto_flow": null, "grid_auto_rows": null, "grid_column": null, "grid_gap": null, "grid_row": null, "grid_template_areas": null, "grid_template_columns": null, "grid_template_rows": null, "height": null, "justify_content": null, "justify_items": null, "left": null, "margin": null, "max_height": null, "max_width": null, "min_height": null, "min_width": null, "object_fit": null, "object_position": null, "order": null, "overflow": null, "overflow_x": null, "overflow_y": null, "padding": null, "right": null, "top": null, "visibility": null, "width": null

}

}, "90a517716ed145918c5634ae3c70118c": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "DescriptionStyleModel", "state": {

"_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "DescriptionStyleModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "StyleView", "description_width": ""

}

}, “a23f439a41fe4d29a577f5fd559af4cf”: {

“model_module”: “@jupyter-widgets/base”, “model_module_version”: “1.2.0”, “model_name”: “LayoutModel”, “state”: {

“_model_module”: “@jupyter-widgets/base”, “_model_module_version”: “1.2.0”, “_model_name”: “LayoutModel”, “_view_count”: null, “_view_module”: “@jupyter-widgets/base”, “_view_module_version”: “1.2.0”, “_view_name”: “LayoutView”, “align_content”: null, “align_items”: null, “align_self”: null, “border”: null, “bottom”: null, “display”: null, “flex”: null, “flex_flow”: null, “grid_area”: null, “grid_auto_columns”: null, “grid_auto_flow”: null, “grid_auto_rows”: null, “grid_column”: null, “grid_gap”: null, “grid_row”: null, “grid_template_areas”: null, “grid_template_columns”: null, “grid_template_rows”: null, “height”: null, “justify_content”: null, “justify_items”: null, “left”: null, “margin”: null, “max_height”: null, “max_width”: null, “min_height”: null, “min_width”: null, “object_fit”: null, “object_position”: null, “order”: null, “overflow”: null, “overflow_x”: null, “overflow_y”: null, “padding”: null, “right”: null, “top”: null, “visibility”: null, “width”: null

}

}, “ad507479d6974a08bba099e7f5ebf681”: {

“model_module”: “@jupyter-widgets/controls”, “model_module_version”: “1.5.0”, “model_name”: “FloatProgressModel”, “state”: {

“_dom_classes”: [], “_model_module”: “@jupyter-widgets/controls”, “_model_module_version”: “1.5.0”, “_model_name”: “FloatProgressModel”, “_view_count”: null, “_view_module”: “@jupyter-widgets/controls”, “_view_module_version”: “1.5.0”, “_view_name”: “ProgressView”, “bar_style”: “success”, “description”: “”, “description_tooltip”: null, “layout”: “IPY_MODEL_cb81fd96890644ce8e1e2647e9c5ee6b”, “max”: 250.0, “min”: 0.0, “orientation”: “horizontal”, “style”: “IPY_MODEL_2edd5e1fc8a14a299469b9120fdfd390”, “value”: 250.0

}

}, “c8578b925601432b901a2ebf3c6ea531”: {

“model_module”: “@jupyter-widgets/controls”, “model_module_version”: “1.5.0”, “model_name”: “DescriptionStyleModel”, “state”: {

“_model_module”: “@jupyter-widgets/controls”, “_model_module_version”: “1.5.0”, “_model_name”: “DescriptionStyleModel”, “_view_count”: null, “_view_module”: “@jupyter-widgets/base”, “_view_module_version”: “1.2.0”, “_view_name”: “StyleView”, “description_width”: “”

}

}, “cb81fd96890644ce8e1e2647e9c5ee6b”: {

“model_module”: “@jupyter-widgets/base”, “model_module_version”: “1.2.0”, “model_name”: “LayoutModel”, “state”: {

“_model_module”: “@jupyter-widgets/base”, “_model_module_version”: “1.2.0”, “_model_name”: “LayoutModel”, “_view_count”: null, “_view_module”: “@jupyter-widgets/base”, “_view_module_version”: “1.2.0”, “_view_name”: “LayoutView”, “align_content”: null, “align_items”: null, “align_self”: null, “border”: null, “bottom”: null, “display”: null, “flex”: null, “flex_flow”: null, “grid_area”: null, “grid_auto_columns”: null, “grid_auto_flow”: null, “grid_auto_rows”: null, “grid_column”: null, “grid_gap”: null, “grid_row”: null, “grid_template_areas”: null, “grid_template_columns”: null, “grid_template_rows”: null, “height”: null, “justify_content”: null, “justify_items”: null, “left”: null, “margin”: null, “max_height”: null, “max_width”: null, “min_height”: null, “min_width”: null, “object_fit”: null, “object_position”: null, “order”: null, “overflow”: null, “overflow_x”: null, “overflow_y”: null, “padding”: null, “right”: null, “top”: null, “visibility”: null, “width”: null

}

}, “ce8c91244c6c4d0983371a04d96487ce”: {

“model_module”: “@jupyter-widgets/controls”, “model_module_version”: “1.5.0”, “model_name”: “HBoxModel”, “state”: {

“_dom_classes”: [], “_model_module”: “@jupyter-widgets/controls”, “_model_module_version”: “1.5.0”, “_model_name”: “HBoxModel”, “_view_count”: null, “_view_module”: “@jupyter-widgets/controls”, “_view_module_version”: “1.5.0”, “_view_name”: “HBoxView”, “box_style”: “”, “children”: [

“IPY_MODEL_fe652b027f19462bb149c02ef27a269d”, “IPY_MODEL_1b6b06a2312742de82eca7b5559ed708”, “IPY_MODEL_5a649ae7052944b59e52da6ee278902b”

], “layout”: “IPY_MODEL_444df54a412b4ab6af6acf074c24068f”

}

}, “f1091902982d43fd8d1c5da523bec127”: {

“model_module”: “@jupyter-widgets/base”, “model_module_version”: “1.2.0”, “model_name”: “LayoutModel”, “state”: {

“_model_module”: “@jupyter-widgets/base”, “_model_module_version”: “1.2.0”, “_model_name”: “LayoutModel”, “_view_count”: null, “_view_module”: “@jupyter-widgets/base”, “_view_module_version”: “1.2.0”, “_view_name”: “LayoutView”, “align_content”: null, “align_items”: null, “align_self”: null, “border”: null, “bottom”: null, “display”: null, “flex”: null, “flex_flow”: null,

"grid_area": null, "grid_auto_columns": null, "grid_auto_flow": null, "grid_auto_rows": null, "grid_column": null, "grid_gap": null, "grid_row": null, "grid_template_areas": null, "grid_template_columns": null, "grid_template_rows": null, "height": null, "justify_content": null, "justify_items": null, "left": null, "margin": null, "max_height": null, "max_width": null, "min_height": null, "min_width": null, "object_fit": null, "object_position": null, "order": null, "overflow": null, "overflow_x": null, "overflow_y": null, "padding": null, "right": null, "top": null, "visibility": null, "width": null

}

}, "f398009217034a1a9fea93bf23c9beea": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "HTMLModel", "state": {

"_dom_classes": [], "_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "HTMLModel", "_view_count": null, "_view_module": "@jupyter-widgets/controls", "_view_module_version": "1.5.0", "_view_name": "HTMLView", "description": "", "description_tooltip": null, "layout": "IPY_MODEL_64a4559bb9b74c30b7daec3c2a1a10f9", "placeholder": "", "style": "IPY_MODEL_c8578b925601432b901a2ebf3c6ea531", "value": " 250/250 [00:04<00:00, 62.14it/s]"

}

}, "fcd72c266765484ca1d3466683960019": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "DescriptionStyleModel", "state": {

"_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "DescriptionStyleModel", "_view_count": null, "_view_module": "@jupyter-widgets/base", "_view_module_version": "1.2.0", "_view_name": "StyleView", "description_width": ""

}

}, "fe652b027f19462bb149c02ef27a269d": {

"model_module": "@jupyter-widgets/controls", "model_module_version": "1.5.0", "model_name": "HTMLModel", "state": {

"_dom_classes": [], "_model_module": "@jupyter-widgets/controls", "_model_module_version": "1.5.0", "_model_name": "HTMLModel", "_view_count": null, "_view_module": "@jupyter-widgets/controls", "_view_module_version": "1.5.0", "_view_name": "HTMLView", "description": "", "description_tooltip": null, "layout": "IPY_MODEL_3eb54496faf04b33b9b5df36a7a1544b", "placeholder": "", "style": "IPY_MODEL_8eb5e492e4cf4d92afcb75a985b5379f", "value": "sampling posterior: 100%"

```
                    }
                }
            }, "version_major": 2, "version_minor": 0
        }
    }
}, "nbformat": 4, "nbformat_minor": 5
}
```

CHAPTER

ONE

INDICES AND TABLES

- genindex
- modindex
- search