NHC Ensemble-based Sensitivity Code Readme

Ryan Torn, University at Albany, SUNY, rtorn@albany.edu
Updated: January 15, 2026

This document provides a description of the enclosed code that utilizes the ensemble-based sensitivity technique to evaluate the sensitivity of various TC-related forecast outcomes, including track, intensity, 2D maximum wind speed and precipitation to forecast fields at an earlier lead time. The enclosed code is written in conda python and is designed to work with any grib files that contain forecast data on a lat/lon grid. The specific python packages are described in the section below. The execution of the program is controlled by the numerous settings within a configuration/parameter file, while the date, storm, and parameter file itself are input as arguments on the command line since they change based on storm and initialization time. Furthermore, the code is designed to work for a variety of models and computing locations. Those differences related to various locations and models is also isolated to the individual i/o module, which has common routine names, but does all of the model/location specific differences inside of it and is transparent to the rest of the code.

The code itself consists of four distinct stages, all of which are controlled by run_NHC_sens.py

1. Data staging and preparation (model and platform specific)
2. Computing forecast metrics (fcst_metrics_tc.py)
3. Compute forecast fields to compute sensitivity to (compute_tc_fields.py)
4. Compute sensitivity and generate maps (nhc_sens.py)

In order to generate sensitivity output, the user should run the following code from the unix command line, which is the command for the Hurricane Laura forecast initialized 0000 UTC 22 August 2020:

python run_NHC_sens.py –init 2020082200 –storm laura13l –param nhc_ecmwf.py,

where the –init argument is the forecast initialization date in yyyymmddhh format, –storm is the TC name, including both the name, TC number, and the basin. The TC number and basin are necessary as the code parses the storm text string to figure out the basin and TC number. Finally, –param is the path to the paramter/configuration file, that contains a number of configuration options that are meant to be static from one initialization time/storm to another, but still gives the user the option to change how the code executes, or how the plots look. Most of the configuration options have default values, though some MUST be set within the file for the code to work. The tables below list the individual parameter/configuration options available and the default values, where appropriate.

The outcome of running this code is a set of graphics and gridded sensitivity output, which are organized using the following directory format:

{figure_dir}/{storm}_{yyyymmddhh}/{metric}/sens/{field},

where storm is the name of the TC (same as –storm line above), yyyymmddhh is the initialization date (same as –init line above), metric is the name of each forecast metric, where each forecast metric has its own directory. For example, the integrated track metric (the default metric of the code) is named f120_intmajtrack, with f120 meaning that the integrated track forecast window ends at 120 h, while intmajtrack is the name of the integrated track metric. The user can specify additional metrics to compute sensitivity for using the metrics configuration option. More information on the definition of each metric is located in the Metric section below.

Within each forecast metric directory are two sub-directories, one is called sens, which are the sensitivity plots/grids on a fixed domain, while the sens_sc provides storm-centered graphics. Within each of these directories is a set of subdirectories that represent individual forecast fields that you are computing the sensitivity to. The forecast hour in each file's name is the forecast lead time that you are computing the sensitivity to (i.e., a file starting with 202008200_f036 is the sensitivity of the metric to the 36 h forecast fields.). The fields section contains information on what each of the forecast fields represents, how it is calculated, and what metric(s) it is used for.

Gridded sensitivity output, in netCDF format, is stored in tar files in {figure_dir}/awips/{yyyymmddhh}.tar. Each tar file contains all of the storms for a particular initialization time using the following convention: {yyyymmddhh}/{storm}/{metshort}, where {metshort} is the short name for each of the metric (track, inten, wind, pcp). These files can be ingested into the AWIPS system for discrimination by forecasters or for other purposes.

Forecast Metrics

By default, the code will compute the sensitivity of TC track, intensity, wind speed, and precipitation independently. The description below is the default behavior of the code that has been tested over many cases, though the user has the option of modifying the timing and/or the domain of each.

Table 1: Forecast Metric Descriptions

Metric Name	Description

intmajtrack	Forecast metric that represents the track perturbation that explains the greatest amount of variability in the ensemble forecast tracks for a time window in the forecast. By default, the time window is 24-120 h, hence the graphics are in the f120_intmajtrack directory and the grids are in the track directory in the tar file. The calculation is done by taking the EOF decomposition of the ensemble latitude/longitude values as a function of time, where the metric is the principle component. By convention, positive values of the metric are indicative of a TC that will end up further along and/or to the right of the ensemble-mean track. The advantage of using this position metric is that it does not require specifying a particular lead time and takes into account the temporal correlation of forecast tracks (i.e., members that are further west early in the forecast will end up further west later on). The user can modify how this metric is calculated via the track_eof namelist options in the metric block of the parameter file.
intmslp	Forecast metric that represents the minimum SLP time series perturbation that explains the greatest amount of variability in the ensemble minimum SLP for a time window in the forecast. By default, the time window is 48-96 h, hence the graphics are in the f096_intmslp directory and the grids are in the inter directory in the tar file. The calculation is done by taking the EOF decomposition of the ensemble minimum SLP values as a function of time, where the metric is the principle component. By convention, positive values of the metric are indicative of a TC that will have lower minimum SLP relative to the ensemble mean. The user can modify how this metric is calculated via the intensity_eof namelist options in the metric block of the parameter file.
wndeof	Forecast metric that represents the 2D forecast wind pattern that explains the greatest amount of variability in the 10 m wind speed within the domain within a window of time. By default, the time window is 48-96 h, hence the graphics are in the f096_wndeof directory and the grids are in the wind directory in the tar file. The domain is automatically determined for each storm and initialization time by considering all of the grid points that are within 300 km of any ensemble member's TC center. The maximum wind speed is computed for each grid point and member by looping over all times within the window and finding the highest wind speed. The greater time frequency, the smoother this field will appear. The metric is then calculated by taking the EOF decomposition of the 2D maximum wind speed field, where the metric is the principle component. By convention, positive values of the metric are indicative of higher winds over the largest amount of the domain, though the metric is often representative of across-track position shifts. The user can modify how this metric is calculated via the ' wind_speed_eof namelist options in the metric block of the parameter file. Note the user can create a customized version of the metric by adding a text file (see namelist options).

pcpeof	Forecast metric that represents the 2D forecast precipitation pattern that explains the greatest amount of variability in the precipitation within the domain within a window of time. By default, the time window is 48-120 h, hence the graphics are in the f120_pcpeof directory and the grids are in the pop directory in the tar file. The domain is automatically determined for each storm and initialization time by considering all contiguous grid points where the precipitation exceeds 0.5 in $(24\ h)^{-1}$ during the window, is close to an ensemble member's track, and is over land. The metric is then calculated by taking the EOF decomposition of the 2D precipitation field for the contiguous set of points, where the metric is the principle component. By convention, positive values of the metric are indicative of higher precipitation over the largest amount of the domain, though the metric is often representative of across-track position shifts. The user can modify how this metric is calculated via the ' precip_eof namelist options in the metric block of the parameter file. Note the user can create a customized version of the metric by adding a text file (see namelist options).

Forecast Fields

The table below gives the list of fields for which the sensitivity of TC track/position forecasts are computed and what they represent:

Parameter File Information

Table 2: Configuration options for the "model" subset.

Parameter Name	Type	Description
model_src	string	Name of the model being used in the sensitivity calculation. This is mainly used in plot titles, so the user can set to whatever they want. No default value.
io_module	string	Name of the module to use for obtaining and reading the grib and ATCF file. Each platform and model will have its own module. This value MUST be set by the user.
projection	string	Map projection for plots. This is a placeholder for models that are not on lat/lon grid. Default: PlateCarree
flip_lon	boolean	Swap longitude from -180 to 180 to 0 to 360. By default, the code uses the -180 to 180 longitude convention. Default False

num_ens	integer	Number of perturbation ensemble members (i.e., ECMWF has 50 perturbed members, GEFS has 30). No default value, so it must be set.
fcst_hour_int	integer	Forecast hour interval for computing forecast fields for sensitivity calculations in hours. Default: 12 h
input_hour_int	integer	Forecast hour internal of the input forecast fields. Default: 6 h
fcst_hour_max	integer	Last forecast hour to compute forecast fields for sensitvity calculations in hours. Default 120 h
tigge_forecast_time	string	List of forecast hours to read from ECMWF TIGGE archive.
tigge_forecast_grid_space	string	Requested grid spacing for data pulled from the TIGGE respository. Default: 1.0/1.0 (units: degrees)
tigge_surface_area	string	Latitude and Longitude grid boundaries to download surface fields from TIGGE. Default: 90/-180/-90/179
tigge_surface_time	string	List of forecast hours to read surface fields from the TIGGE archive.
tigge_surface_grid_space	string	Requested grid spacing for surface data pulled from the TIGGE respository. Default: 0.25/0.25 (units: degrees)

Table 3: Configuration options for the "locations" subset.

Parameter Name	Type	Description
atcf_dir	string	Path to raw ATCF forecast data on local server. Usage will depend on which io_module that you use. No default value
model_dir	string	Path to raw model data on local server. Usage will depend on which io_module that you use. No default value
best_dir	string	Path to best track data on local server. Usage will depend on which io_module that you use. No default value
work_dir	string	Path to work directory where sensitivity calculations are carried out No default value.
output_dir	string	Path to directory to save certain output of the sensitivity calculations, if desired. No default value.
script_dir	string	Path to python scripts and modules (i.e., where this code is located.) No default value.
figure_dir	string	Path to directory where output figures will be placed. No default value.
outgrid_dir	string	Path to directory where gridded sensitivity output will be placed. No default value.
log_dir	string	Path to directory where log file output from the python logging function will be placed. No default value.

log_level	string	Logging level to output into the log file. Default: INFO
archive_metric	boolean	True to save metric netcdf files into the appropriate output_dir location. Default: False
output_dir	string	Path to directory where archived metric and field files will be placed. Default: False
archive_fields	boolean	True to save forecast field netcdf files into the appropriate output_dir. Default: False
save_work_dir	boolean	True to save work directory at the end of the execution. Otherwise, this directory will be deleted to save space. Default: False

Table 4: Configuration options for the “vitals_plot” subset. This controls diagnostic figures, such as the generic track, intensity and precipitation figures.

Parameter Name	Type	Description
trackfile	string	Name of figure that shows the TC track forecast. Default: track.png
track_output_dir	string	Name of directory to place the track plot. Default: figure_dir from above
intfile	string	Name of figure that shows the TC intensity forecast. Default: intensity.png
int_output_dir	string	Name of directory to place the intensity plot. Default: figure_dir from above
forecast_hour_int	float	Frequency of forecasts from model (hours). Default: 6
forecast_hour_max	string	Maximum forecast hour for track and intensity plots. Default: 120 hours
plot_ellipse	boolean	True to plot TC position ellipses on plots. Default: True
ellipse_frequency	float	Frequency to plot the position ellipses (hours). Default: 24
plot_best	boolean	True to plot best track information, if available. Default: True
projection	string	Map projection to use for vitals plot. Default: PlateCarree
grid_interval	float	Latitude and Longitude line grid interval (degrees). Default: 5
title_string	string	customized string for TC track plots. Overwrites the default string
precip_hour_1	integer list	list of start forecast hours for precipitation plots. No Default
precip_hour_2	integer list	list of end forecast hours for precipitation plots. No Default
min_lat_precip	float	Minimum latitude for precipitation forecast plots. Default: 22.0

max_lat_precip	float	Maximum latitude for precipitation forecast plots. Default: 50.0
min_lon_precip	float	Minimum longitude for precipitation forecast plots. Default: -100.0
max_lon_precip	float	Maximum longitude for precipitation forecast plots. Default: -65.0

Table 5: Configuration options for the “metric” subset. This set of namelist values control the forecast metric definition and how it is calculated.

Parameter Name	Type	Description
metric_hours	float	Vector list of forecast hours to compute individual time forecast metrics, such as position, intensity, etc. This capability is old and not maintained at this point. No default value
track_eof_metric	boolean	True to calculate the integrated track metric. Default True
track_eof_hour_init	float	Initial forecast hour to use for integrated track metric (hours). Default: 24
track_eof_hour_int	float	Forecast hour interval to use for integrated track metric (hours). Default: 6
track_eof_hour_final	float	Final forecast hour to use for integrated track metric (hours). Default: 120
track_eof_member_frac	float	Fraction of members that need to be present during a forecast hour for track EOF. Default: 0.5
track_eof_esign	float	Factor to multiply EOF PC by. Default 1.0
title_string	string	Title string that is placed at the top of the track EOF plot. This overwrites the default plot title if it exists. Default: None
intensity_eof_metric	boolean	True to calculate the time-integrated intensity metric based on min. SLP. Default True
intensity_eof_hour_init	float	Initial forecast hour to use for integrated intensity metric (hours). Default: 24
intensity_eof_hour_int	float	Forecast hour interval to use for integrated intensity metric (hours). Default: 6
intensity_eof_hour_final	float	Final forecast hour to use for integrated intensity metric (hours). Default: 96
intensity_eof_member_frac	float	Fraction of members that need to be present during a forecast hour for intensity EOF. Default: 0.3
track_inten_eof_metric	boolean	True to calculate the time-integrated combined track/intensity metric. Default: False

track_inten_eof_hour_init	float	Initial forecast hour to use for integrated track/intensity metric (hours). Default: 24
track_inten_eof_hour_int	float	Forecast hour interval to use for integrated track/intensity metric (hours). Default: 6
track_inten_eof_hour_final	float	Final forecast hour to use for integrated track/intensity metric (hours). Default: 120
track_inten_eof_member_frac	float	Fraction of members that need to be present during a forecast hour for integrated track/intensity EOF. Default: 0.5
track_inten_eof_esign	float	Factor to multiply EOF PC by. Default 1.0
kinetic_energy_metric	boolean	True to calculate area-average kinetic energy metric. Default: False
kinetic_energy_radius	float	Radius over which to calculate the average kinetic energy (km). Default: 200
kinetic_energy_level	float	Pressure level to use for kinetic energy metric (hPa). Default: 1000
wind_speed_eof_metric	boolean	True to calculate forecast metric that is EOF/PC of the wind speed over the specified time window and area. Default: False
wind_metric_file	string	Path to file that contains precipitation EOF metric settings. Assumes the file has the format {yyyymmddhh}_{storm}_wind. No default value.
wind_speed_eof_forecast_hour1	integer	initial forecast hour for calculating the maximum wind speed metric (hour). This value can be overwritten with a value in the wind_metric_file. Default: 48
wind_speed_eof_forecast_hour2	integer	final forecast hour for calculating the maximum wind speed metric (hour). This value can be overwritten with a value in the wind_metric_file. Default: 96
wind_speed_eof_adapt	boolean	True to use adaptive algorithm for identifying the wind speed EOF metric domain based on the forecast times provided. Default: True
wind_speed_eof_dom_buffer	float	Distance from forecasted TC center over which the maximum wind speed metric EOF is calculated. Default: 300
precipitation_metric	boolean	True to calculate forecast metric that is the average precipitation over the specified time window and area. Currently experimental. Default: False
precipitation_eof_metric	boolean	True to calculate forecast metric that is EOF/PC of the precipitation over the specified time window and area. Default: False
precip_metric_file	string	Path to file that contains precipitation EOF metric settings. Assumes the file has the format {yyyymmddhh}_{storm}_precip. No default value.

precip_eof_forecast_hour1	integer	initial forecast hour for calculating the precipitation metric (hour). This value can be overwritten with a value in the precip_metric_file. Default: 48
precip_eof_forecast_hour2	integer	final forecast hour for calculating the precipitation metric (hour). This value can be overwritten with a value in the precip_metric_file. Default: 120
precip_eof_dom_buffer	float	Distance from forecasted TC center over which the precipitation metric EOF is calculated. Can be overwritten by precip_metric_file. Default: 300
land_mask_minimum	float	Minimum value of the landmask value that is considered land. 1 = all land. Can be overwritten by precip_metric_file. Default: 0.2
precip_eof_land_mask	boolean	True to only consider grid points with a value above land_mask_minimum. Can be overwritten by precip_metric_file. Default: True
precip_eof_adapt	boolean	True to use adaptive algorithm for identifying the precipitation EOF metric domain based on the forecast times provided. Can be overwritten by precip_metric_file. Default: True
precip_eof_time_adapt	boolean	True to use adaptive algorithm for identifying the time period for the precipitation EOF metric. Can be overwritten by precip_metric_file. Default: True
precip_eof_time_adapt_domain	float	Distance in degrees from the min/max latitude and longitude to consider within the adaptive precipitation EOF metric metric. Can be overwritten by precip_metric_file. Default: 2.0
precip_eof_time_adapt_freq	integer	Forecast frequency to use for calculating time period for the adaptive precipitation EOF metric. Can be overwritten by precip_metric_file. Default: 6 hours
precip_eof_adapt_pcp_min	float	Minimum precipitation threshold to consider for the adaptive precipitation EOF metric. Can be overwritten by precip_metric_file. Default: 12.7 (mm)
wind_precip_eof_metric	boolean	True to calculate forecast metric that is EOF/PC of the maximum wind speed and precipitation over the specified time window and area. Uses parameters from the wind and precipitation EOF metrics. Default False
fcst_int	integer	Forecast frequency to use for calculating forecast metrics over time. Default: 6 hours
static_fields_file	string	Path to a grib file that contains static fields, such as the landmask for use in calculating metrics. Default: None
projection	string	Map projection to use for metric plot. Default: PlateCarree
grid_interval	float	Latitude and Longitude line grid interval in metric plots (degrees). Default: 5

Parameter Name	Description
usteer	Zonal component of the steering flow. By default, this is designated as the average wind between 300-850 hPa (vortex removed), but this can be changed in the configuration file.
vsteer	Meridional component of the steering flow. By default, this is designated as the average wind between 300-850 hPa (vortex removed), but this can be changed in the configuration file.
masteer	Major axis winds are the wind component that is in the direction of greatest track variability for that particular case (positive values are either along and/or right of track). In most situations, the sensitivity to the major axis wind is the most useful for sensitivity calculations because it most closely relates to variability in subsequent TC position, which is not often in the Cartesian directions.
csteer	Vorticity of the steering flow that is computed using the method described in unsteer, steer
pv250hPa	250 hPa potential vorticity
h500	500 hPa height

Table 6: Configuration options for the field subset. This controls the forecast fields for which the sensitivity is computed to.

Parameter Name	Type	Description
multiprocessor	boolean	True to calculate fields in serial using multiple processors. Default: False
processes	integer	Number of processes to use for computing fields. Default: None
calc_uvsteer	boolean	True to compute the u/v component of the steering wind. Default: True
calc_steer_circ	boolean	True to compute the circulation/vorticity from the steering wind. Default: False
steer_level1	float	Lowest pressure level to use to compute the layer-average steering wind (hPa). Default: 300
steer_level2	float	Highest pressure level to use to compute the layer-average steering wind (hPa). Default: 850
steer_radius	float	Radius to use for removing the TC from the model fields (km). Default: 333
calc_height	boolean	True to compute gepotential height on specified pressure levels. Default: True
height_levels	float	List of pressure levels to compute the geopotential height (hPa). Default: 500

calc_pv_pres	boolean	True to compute potential vorticity on specified pressure levels. Default: True
pv_levels	float	List of pressure levels to compute the potential vorticity (hPa). Default: 250, 850 hPa
calc_theta-e	boolean	True to compute equivalent potential temperature at specified pressure levels. Default: False
theta-e_levels	float	List of pressure levels to compute equivalent potential temperature (hPa). Default: 700, 850
calc_q500-850hPa	boolean	True to compute the integrated water vapor between 500 and 850 hPa. Default: False
calc_ivt	boolean	True to compute the integrated vapor transport from the available pressure levels. Default: True
calc_winds	boolean	True to compute zonal and meridional wind at specified pressure levels. Default: False
wind_levels	float	List of pressure levels to compute zonal and meridional wind (hPa). Default: 850
calc_vorticity	boolean	True to compute relative vorticity at specified pressure levels. Default: False
vorticity_levels	float	List of pressure levels to compute relative vorticity (hPa). Default: 850
vorticity_radius	float	Radius to use for calculating the area-average of vorticity, which is the effective scale of rotation (km). Default: 115 km
min_lat	float	Minimum latitude to compute forecast fields over. Default: 0.0
max_lat	float	Maximum latitude to compute forecast fields over. Default: 65.0
min_lon	float	Minimum longitude to compute forecast fields over. Default: -180.0
max_lon	float	Maximum longitude to compute forecast fields over. Default: -10.0
global	boolean	True to calculate fields based on a global forecast grid. Default: False

Table 7: Configuration options for the sens subset. This set of parameters set the display options for the sensitivity plots.

Parameter Name	Type	Description
multiprocessor	boolean	True to calculate sensitivity in serial using multiple processors. Default: False
processes	integer	Number of processes to use for computing sensitivity. Default: None

metrics	string	List of names of forecast metrics to compute the sensitivity to. This parameter is not used anymore, but can be used. Default: none
min_lat	float	Minimum latitude for sensitivity plots. Default: 8.0, also set in run_NHC_sens.py based on basin.
max_lat	float	Maximum latitude for sensitivity plots. Default: 65.0, also set in run_NHC_sens.py based on basin.
min_lon	float	Minimum longitude for sensitivity plots. Default: -140.0, also set in run_NHC_sens.py based on basin.
max_lon	float	Maximum longitude for sensitivity plots. Default: -20.0, also set in run_NHC_sens.py based on basin.
zero_non_sig_sens	boolean	True to plot only the statistically significant sensitivity locations. Default: False
grid_interval	float	Latitude and Longitude line grid interval (degrees). Default: 10°.
barb_interval	integer	Number of grid points in between each wind barb in the plot. Default: 6 grid points
cbar_shrink	float	width of colorbar within the figure. Used to set shrink keyword in matplotlib. Default: 1.0
title_string	string	String that will overwrite the basic title string used in the plots.
storm_center_radius	float	Radius from the TC center used in storm-center plots (degrees). Default: 10
dropsonde_file	string	Full path to file of dropsonde locations. Default: none
drop_file_type	string	Type of dropsonde file to read. Default: nhc
drop_mark_size	integer	Marker size of dropsonde locations in plot. Default: 6
drop_mark_color	string	Dropsonde marker color in plot. Default: black
drop_mark_type	string	Dropsonde marker in plot. Default: +
rawinsonde_file	string	Full path to file of rawinsonde locations. Default: none
rawin_mark_size	integer	Marker size of rawinsonde locations in plot. Default: 6
rawin_mark_color	string	Rawinsonde marker color in plot. Default: gray
turns_file	string	Full path to file of aircraft turn locations. Default: none
turn_line_width	float	Line width of aircraft track. Default: 2
turn_line_color	string	Aircraft track line color in plot. Default: black
turn_file_type	string	Type of aircraft turn file to read. Default: nhc
range_rings	boolean	True to plot range rings from the predicted TC center. Default: True
ring_values	floats	List of range ring radii for plot in km. Default:
range_linewidth	float	Linewidth of the range rings. Default: 1
range_colors	float	Color of the range rings. Default: gray
plot_summary	boolean	True to plot summary style figures used for the NHC dashboard format. Default: False

output_sens	boolean	True to create netCDF file that contains gridded sensitivity fields that can be used in AWIPS or traveling salesman. Default: True
nhc_sens	boolean	True to create NHC version of the gridded netCDF file, which means that it includes a variable that is the absolute value of sensitivity (for traveling salesman software) and range rings. Default: False
wind_levels	integer	List of pressure levels to plot wind sensitivity. Default: None
vorticity_levels	integer	List of pressure levels to plot vorticity sensitivity. Default: None

1 Conda Information

The code requires the following conda packages to be installed at a minimum: ipython, numpy, cartopy, metpy, netcdf4, pandas, matplotlib, xarray, configparser, scipy, eofs, windspharm. In addition, grib files are read using cfgrib for most of the I/O modules, though other possibilities exist. All of these are fairly standard python packages, except windspharm, which is frozen. All of this packages can be downloaded from conda-forge stream. Once all of the packages are installed, you will need to compile the grid_calc fortran code into a python object. Computing spatial averages within python can be very expensive, so this speeds up the calculation greatly. In order to compile this code, go into the esens-util directory and type: “python -m numpy.f2py -c grid_calc.f90 -m grid_calc”. The resulting object file will need to be renamed grid_calc.so.