# **CESM** Unified Postprocessing and Diagnostics (CUPiD)

Teagan King and Mike Levy Mar 4th, 2024 - SEWG Meeting





This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

# **GENERAL THOUGHTS FOR FLOW OF PRESENTATION - DELETE SLIDE LATER**

- Mike [5 min]
  - ESDS Background
  - Vision of project / anticipated milestones
- Teagan [10 min]
  - Where we're at
  - ➤ A few things more relevant to SEWG:
    - i. Specific framework-related things we're addressing
    - ii. GeoCAT collaboration thoughts
    - iii. Collaborative work time thoughts?
    - iv. Any ways we may be able to improve CUPiD or strengthen collaborations with help from folks in this meeting?



Collaboration and grassroots effort started in Fall 2020 among NCAR/UCAR labs, centers, programs, and offices (LCPOs) focused on improving Earth system data science

- Led by a committee with representation from CGD, CISL, EOL, MMM, RAL, and Unidata
- Has been integral in the "push to python" (helping users transition from NCL to open source python packages)
- Community-building and creating a culture of open science and development





# **CESM Diagnostics and ESDS**



PANGE

- CESM users were hitting the limits of what NCL-based analysis could handle (and NCL is no longer being developed)
  - Advantage to making sure everyone replaced NCL with the same product
  - ESDS helped community find a solution
    - Pangeo stack: numpy, xarray, matplotlib, dask, etc
- Still had several independent efforts for analyzing CESM output
  - ADF, NBscuid, individuals writing one-off notebooks
- Next step: build a common framework
  - CESM Unified Postprocessing and Diagnostics (CUPiD)









### **CUPiD Collaborators**

- **AMP:** Dani Coleman, Cecile Hannay, Brian Medeiros, Christina McCluskey, Jesse Nusbaumer, Justin Richling
- CAS: John Fasullo, Adam Phillips, Isla Simpson
- CCR: Gary Strand
- **CSEG:** Brian Dobbins
- **CESM:** Dave Lawrence
- ESDS: Katie Dagon, Teagan King, Mike Levy
- ESMF: Bill Sacks
- GeoCAT (CISL): Orhan Eroglu, Katelyn FitzGerald, Anissa Zacharias
- **OS:** Anna Deppenmeier, Gustavo Marques, Lev Romashkov
- **PPC:** Dave Bailey, Kate Thayer-Calder
- TSS: Sam Levis, Will Wieder



# **CUPiD Project Vision**

A collaborative effort that unifies all CESM component diagnostics and provides

- Python code that
  - i. runs in an easy-to-generate conda environment, and
  - ii. can be launched via CIME workflow or independently
- Diagnostics for single/multiple runs and single/multiple components
- Ability to call post-processing tools that other groups are working on
- An API that makes it easy to include outside code
- Ongoing support and software maintenance





#### **Anticipated Timeline**





### • Open development on <u>github</u> (under NCAR organization)

- Documentation on github.io
- Fork / branch / PR strategy
- Issues and Project Board

📄 🎧 NCAR / Projects / Bare-bones Deployment 🛆		Q Type 🛛 to search		
A Bare-bones Deployment				
🔟 kanban board 📼 🕂 New view				
〒 Filter by keyword or by field				
O Todo 13	O In Progress 7 ····	O Done (2)		
This item hasn't been started	This is actively being worked on	This has been completed		
Clean up notebooks	Work through engineering concerns	Keep conda environment files in same directory		
	⊙ CUPiD #6			
Write python script that allows users to run diagnostics from all components	Determine which diagnostics to include for Atmospheric component	<ul> <li>CUPID #22</li> <li>Bring mom6-tools in as an external via manage-externals</li> </ul>		
	• CUPID #7			
Run diagnostics from CIME workflow	Determine which diagnostics to include for Land component			
	🔿 CUPID #8			
Add flags to run particular component diagnostics	Determine which diagnostics to include for Sea Ice component			
	O CUPID #9			
Common location for observational datasets?	COPID #9     Determine which diagnostics to include for     Ocean component			
• CUPID #2	Ocean component			
+ Add item	+ Add item	+ Add item		



- Single command line call to produce basic diagnostics for atmosphere, ocean, land, and sea-ice
  - User-editable YAML file controls what is run
  - Uses ploomber (which wraps papermill) to run notebooks and pass parameters

(cupid-dev) \$ (cupid-run) config.yml) (1910)	
(cupid-dev) \$ (cupid-run) config.yml)	
index	
adf_quick_run	
ocean_surface	
land_comparison	
seaice	
/glade/work/mlevy/conda-envs/cupid-dev/lib/python3.11/site-packages/ploomber/dag/dag.py:455: U	UserWarning:
NotebookRunner: index -> File('computed_notebookck-run/index.ipynb')	
/glade/work/mlevy/codes/CUPiD/examples/nblibrary/index.ipynb	
These parameters are not used in the task's source code: 'CESM_output_dir', 'lc_kwargs', and '	
NotebookRunner: adf_quick_run -> File('computed_notebookdf_quick_run.ipynb')	
/glade/work/mlevy/codes/CUPiD/examples/nblibrary/adf_quick_run.ipynb	
These parameters are not used in the task's source code: 'CESM_output_dir', 'lc_kwargs', and '	
NotebookRunner: ocean_surface -> File('computed_notebookcean_surface.ipynb')	
/glade/work/mlevy/codes/CUPiD/examples/nblibrary/ocean_surface.ipynb	
These parameters are not used in the task's source code: 'subset_kwargs'	
NotebookRunner: land_comparison -> File('computed_notebookd_comparison.ipynb')	
/glade/work/mlevy/codes/CUPiD/examples/nblibrary/land_comparison.ipynb	
These parameters are not used in the task's source code: 'lc_kwargs', and 'subset_kwargs'	
NotebookRunner: seaice -> File('computed_notebookk-run/seaice.ipynb')	
/glade/work/mlevy/codes/CUPiD/examples/nblibrary/seaice.ipynb	
These parameters are not used in the task's source code: 'subset_kwargs'	
Summary (5 tasks)	
NotebookRunner: index -> File('computed_notebookck-run/index.ipynb')	
NotebookRunner: adf_quick_run -> File('computed_notebookdf_quick_run.ipynb') NotebookRunner: ocean_surface -> File('computed_notebookcean_surface.ipynb')	
NotebookRunner: lord_comparison -> File('computed_notebookd_comparison.ipynb')	
NotebookRunner: seaice -> File('computed_notebookk-run/seaice.ipynb')	
DAG render with warnings	



- Each notebook is run sequentially
- Runtime is fairly variable
- Notebooks can use dask LocalCluster for parallelization
  - Ocean notebook ~10 minutes without dask
    - With dask: 4:23 here, 3 minute best
  - Seaice notebook ~25 minutes without dask
    - With dask: 13:37 here, 9 minute best

warnings.warn(str(warnings_))	
Building task 'index': 0%    0/5 [00:00<2	?, ?it/s0
.02s - Debugger warning: It seems that frozen modules are being used, which may $10/3$ [00:00 ,</td <td>?cell/s]</td>	?cell/s]
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off	
0.00s - to python to disable frozen modules.	
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation	on.
Executing: 100%1   3/3 [00:09<00:00, 3.1	L7s/cell]
Building task 'adf_quick_run': 20%    1/5 [00:09<00:38, 9	).54s/it0
.00s - Debugger warning: It seems that frozen modules are being used, which may   0/63 [00:00 ,</td <td>?cell/s]</td>	?cell/s]
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off	
0.00s - to python to disable frozen modules.	
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation	on.
Executing: 100%    63/63 [01:06<00:00, 1.0	05s/cell]
Building task 'ocean_surface': 40%    2/5 [01:15<02:08, 42	2.83s/it0
.00s - Debugger warning: It seems that frozen modules are being used, which may   0/18 [00:00 ,</td <td>?cell/s]</td>	?cell/s]
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off	
0.00s - to python to disable frozen modules.	
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disp <del>ile this</del> validation	
Executing: 100%   18/16 [04:23<09:00, 14.6	
Building task 'land_comparison': 60%1	
.00s - Debugger warning: It seems that frozen modules are being used, which may $\mid$ 0/10 [00:00 ,</td <td>?cell/s]</td>	?cell/s]
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off	
0.00s - to python to disable frozen modules.	
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation	
Executing: 100%  [01:02<00:00, 6.2	1010
Building task 'seaice': 80%	
.00s - Debugger warning: It seems that frozen modules are being used, which may   0/22 [00:00 ,</td <td>?cell/s]</td>	?cell/s]
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off	
0.00s - to python to disable frozen modules.	
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation	
Executing: 100%	Advertised Advertised
Building task 'seaice': 100%	5.795/1t]
(cupid-dev) \$	



- CUPiD provides two environments
  - cupid-run executable in cupid-dev (plan: have CISL add to NPL)
  - All notebooks run in cupid-analysis
    - Build on pangeo stack
    - Uses geoCAT packages
      - Active migration of functions to GeoCAT



 Step-by-step instructions for adding new diagnostics in this framework are <u>available online</u>



CUPiD Documentation

Adding Notebooks

≡

#### 0700

#### How to add diagnostics notebooks

Generally, a good fit for a diagnostic notebook is one that reads in CESM output, does some processing, and outputs plots, values, and/or new files (images, data, etc.) that are useful for evaluating the run.

 Install the environments/cupid-analysis.yml environment (see installation instructions). Make sure that your notebook runs properly in this environment. If there are conflicts or missing dependencies, open an issue or talk to CUPiD developers so we can find a solution.

2. In your notebook, move all variables you might want to change (paths to data, dates to select, etc.) to a cell near the top. For example:

3. Tag this cell as parameters. This means that when the notebook is executed by <u>CUPiD</u>, a new cell will be inserted just below this one with all of the parameters specified in <u>config.yml</u> (see step 5). To tag it, in Jupyter Lab, click on the cell and click the button with two gears in the top right ("Property Inspector"). Open "Common Tools." There, you can see a section called "Cell Tags." Click "Add Tag," and add one called <u>parameters</u> (exactly as written). If you don't want to fully set up CUPiD, stop here and we can integrate the notebook into a CUPID workflow from here.



# **CUPiD** Output

- CUPiD can use JupyterBook to turn notebooks into web pages
  - cupid-build function name likely to change



(cupid-dev) \$ cupid-build config.yml
Running Jupyter-Book v0.15.1

Source Folder: /glade/work/mlevy/codes/CUPiD/examples/coupled\_model/computed\_notebooks/quick-run Config Path: /glade/work/mlevy/codes/CUPiD/examples/coupled\_model/computed\_notebooks/quick-run/\_config.yml Output Path: /glade/work/mlevy/codes/CUPiD/examples/coupled\_model/computed\_notebooks/quick-run/\_build/html Running Sphinx v5.0.2 making output directory... done

- Conda kernel available on Casper & Derecho
- Launch diagnostics from CIME workflow
- Port NCL-based code to python



# Summary

- ESDS is a cross-laboratory initiative at NCAR
- CUPiD is an ESDS project focused on CESM3 (and beyond) diagnostics
- Still in early stages of development, but showing lots of promise

