The UF-ECT

Applications to different models and realistic bug scenarios.

Teo Price-Broncucia - CU Boulder - June 10, 2024

Collaborative Work With:

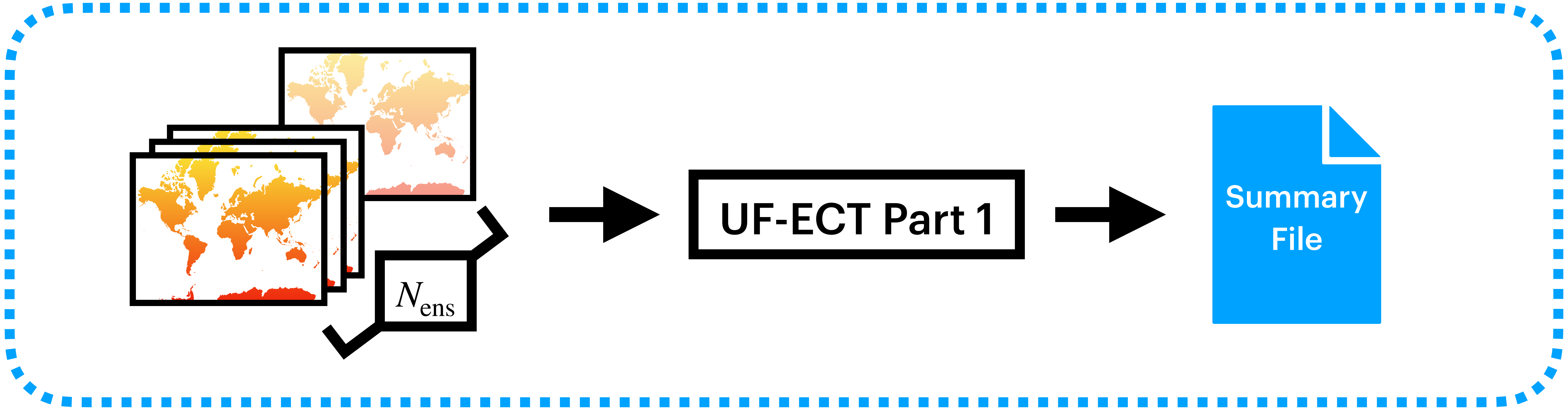
- Allison Baker: NSF-NCAR, CISL-ASP
- Dorit Hammerling: Colorado School of Mines
- Michael Duda: NSF-NCAR, Mesoscale & Microscale Meteorology Lab
- Rebecca Morrison: CU-Boulder

What is the UF-ECT?

- Ultra-Fast Ensemble Consistency Test
- Designed to identify changes in model outputs exceeding internal model variability.
  - When bit-for-bit equivalence (BFB) is not feasible.
- Require minimal computational expense to test new model configurations.

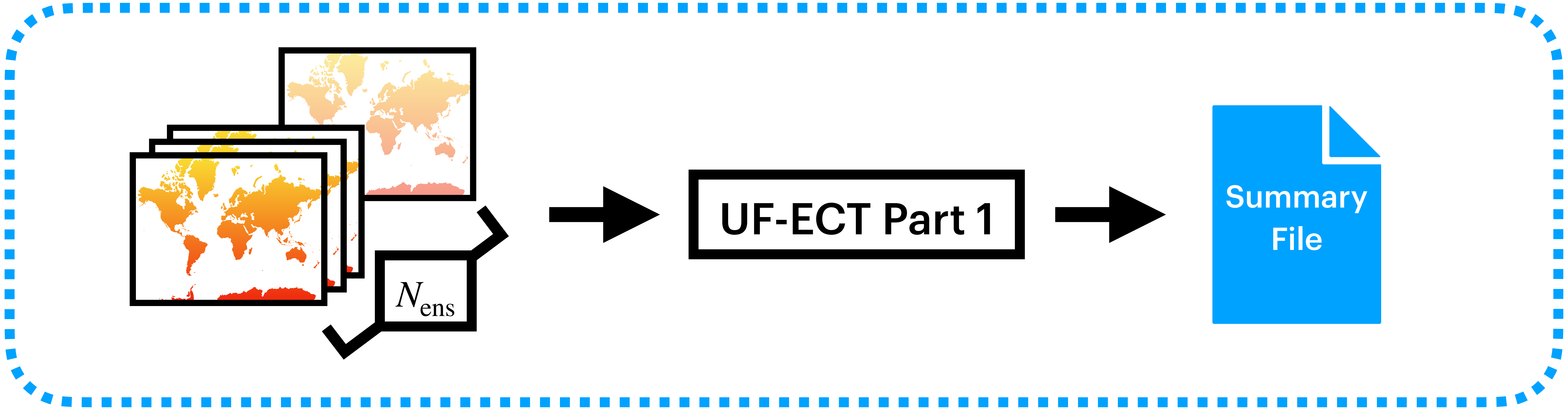
What is the UF-ECT?

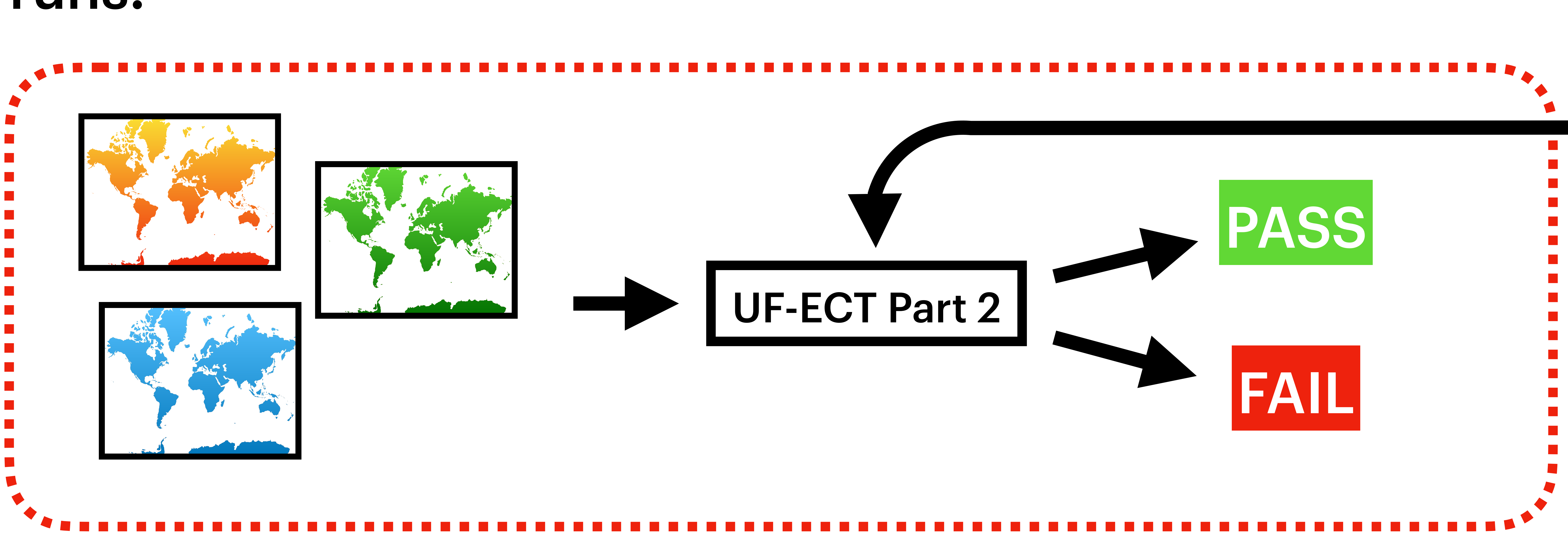
1) Characterize model variability using large ensemble of accepted model runs.



What is the UF-ECT?

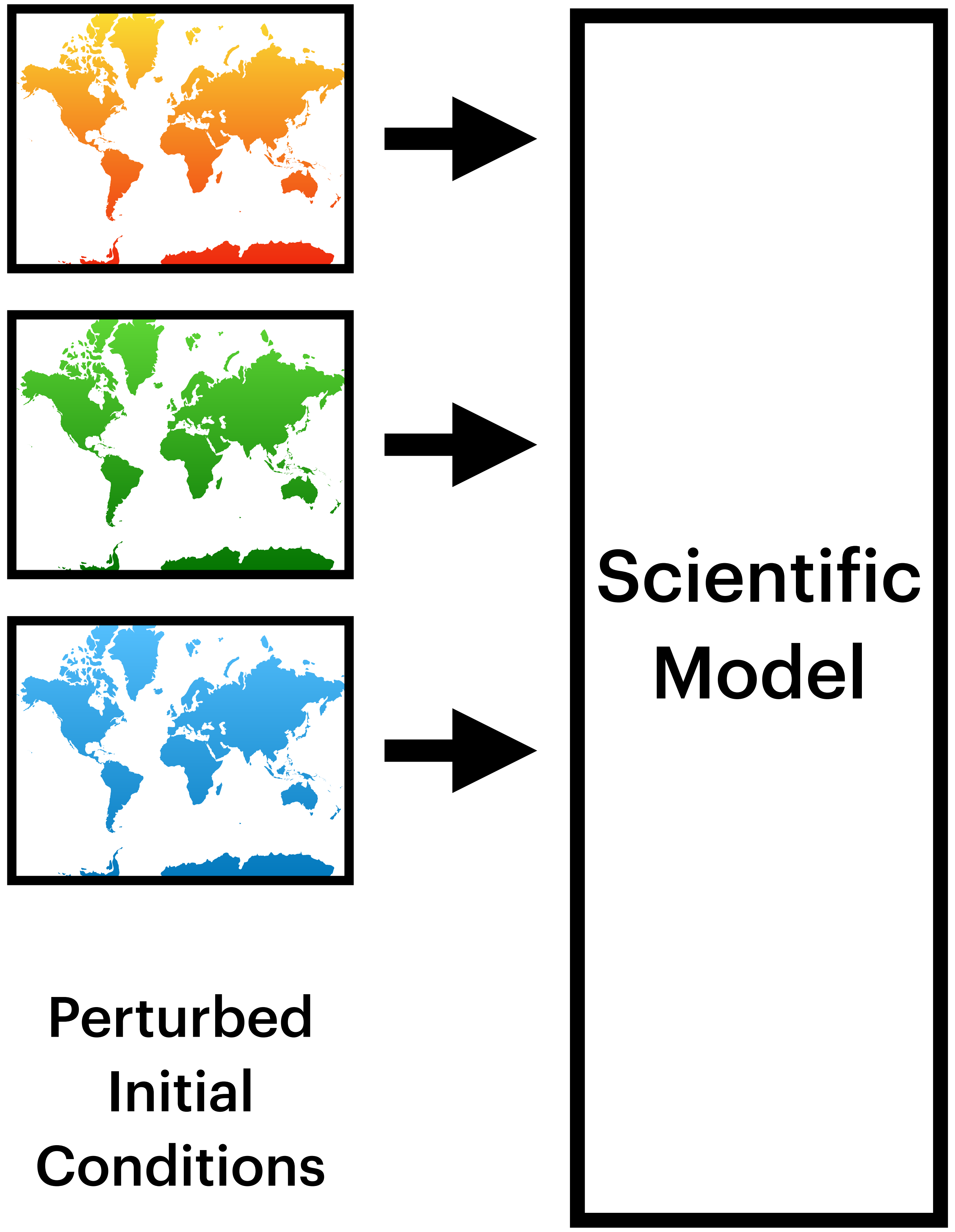
1) Characterize model variability using large ensemble of accepted model runs.



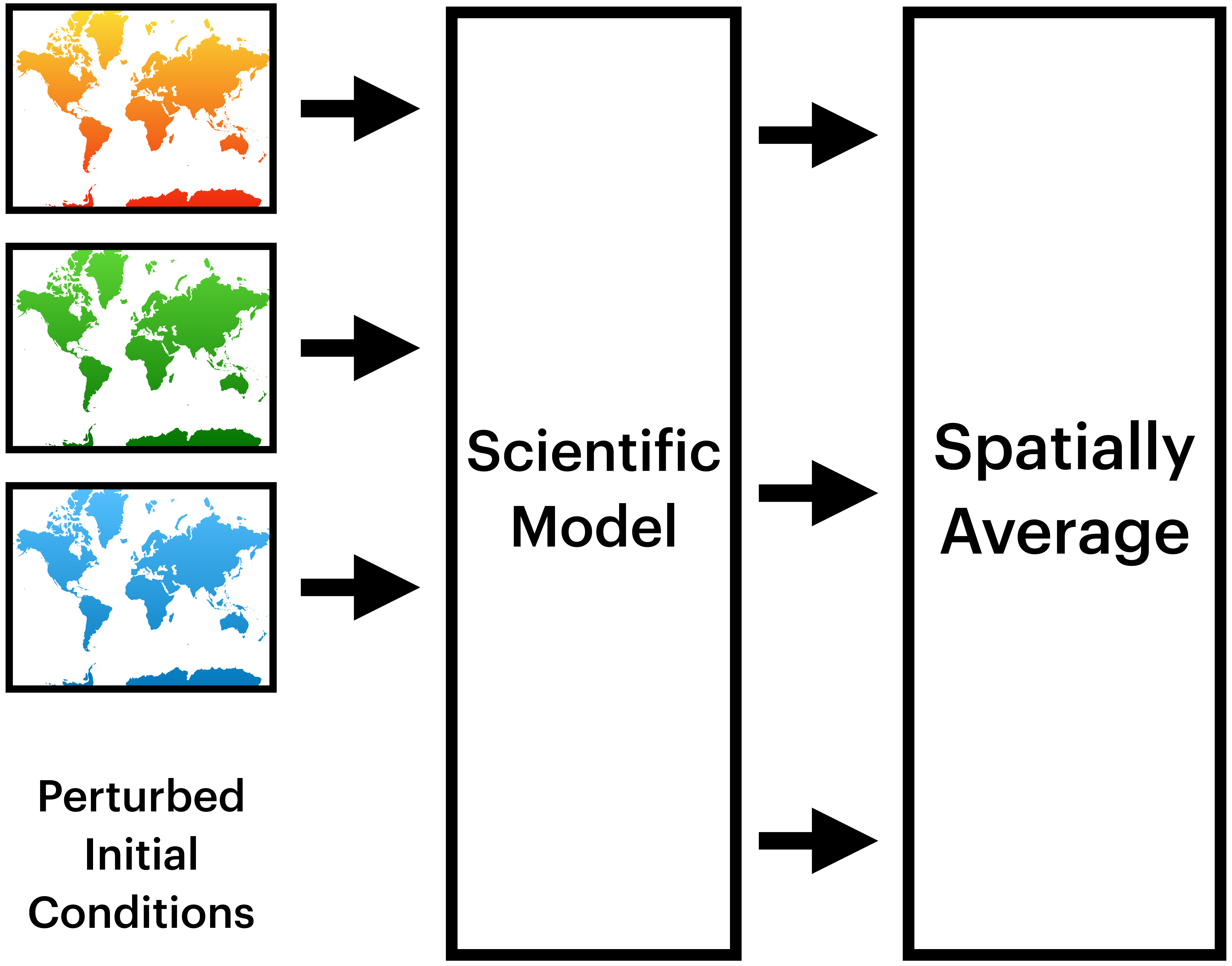


2) Test new configurations using a small set of new model runs.

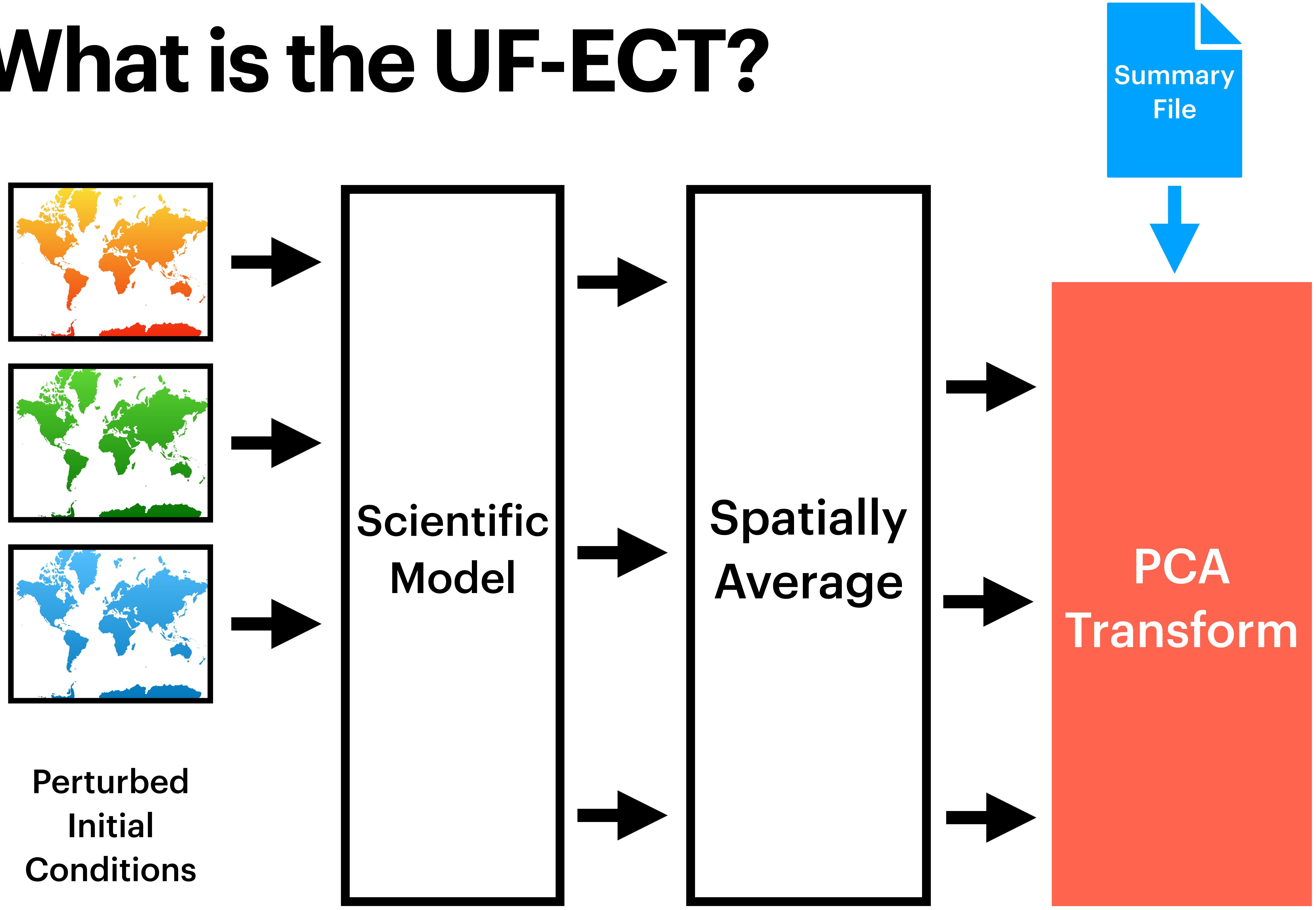
What is the UF-ECT?



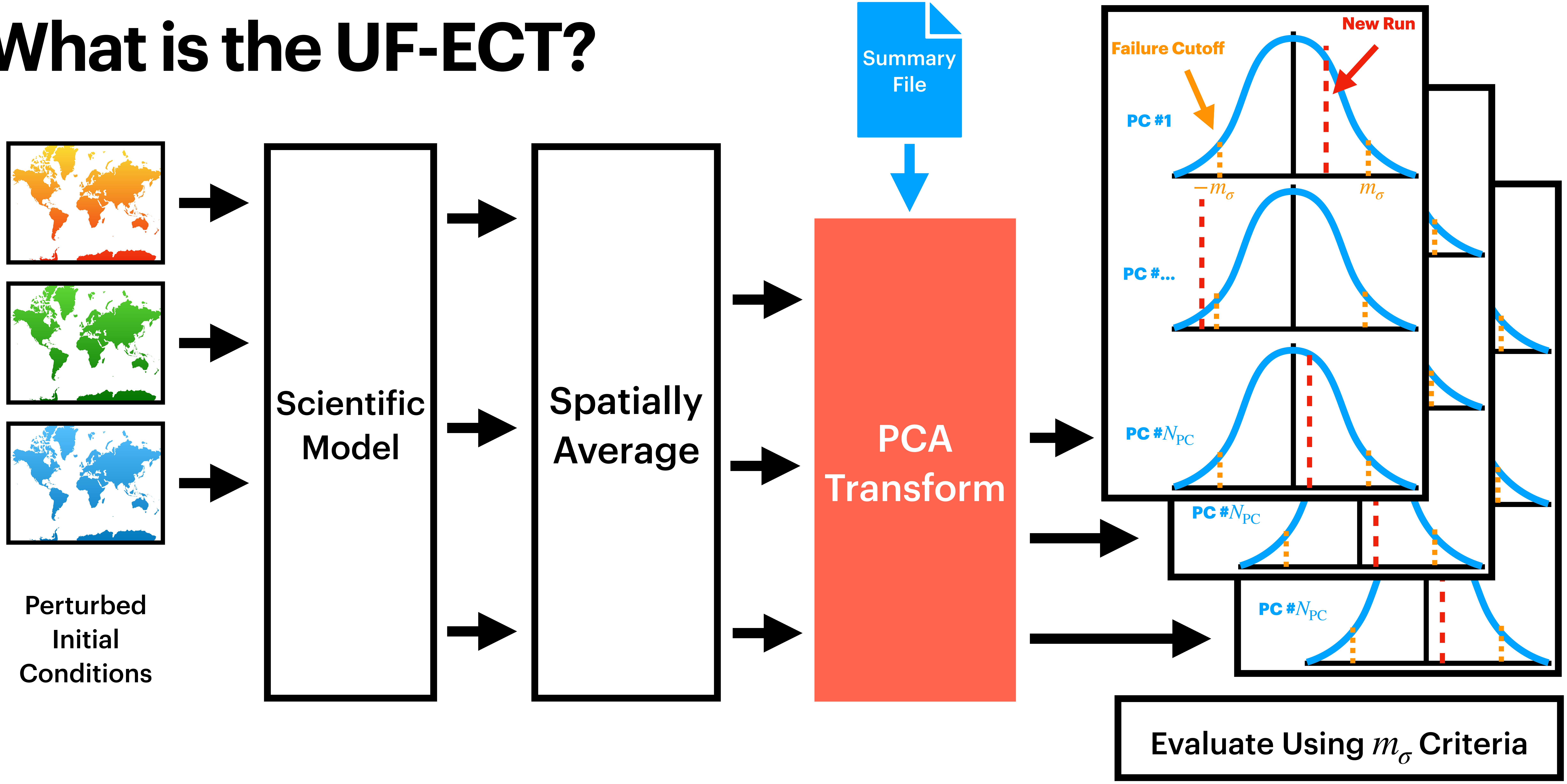
What is the UF-ECT?



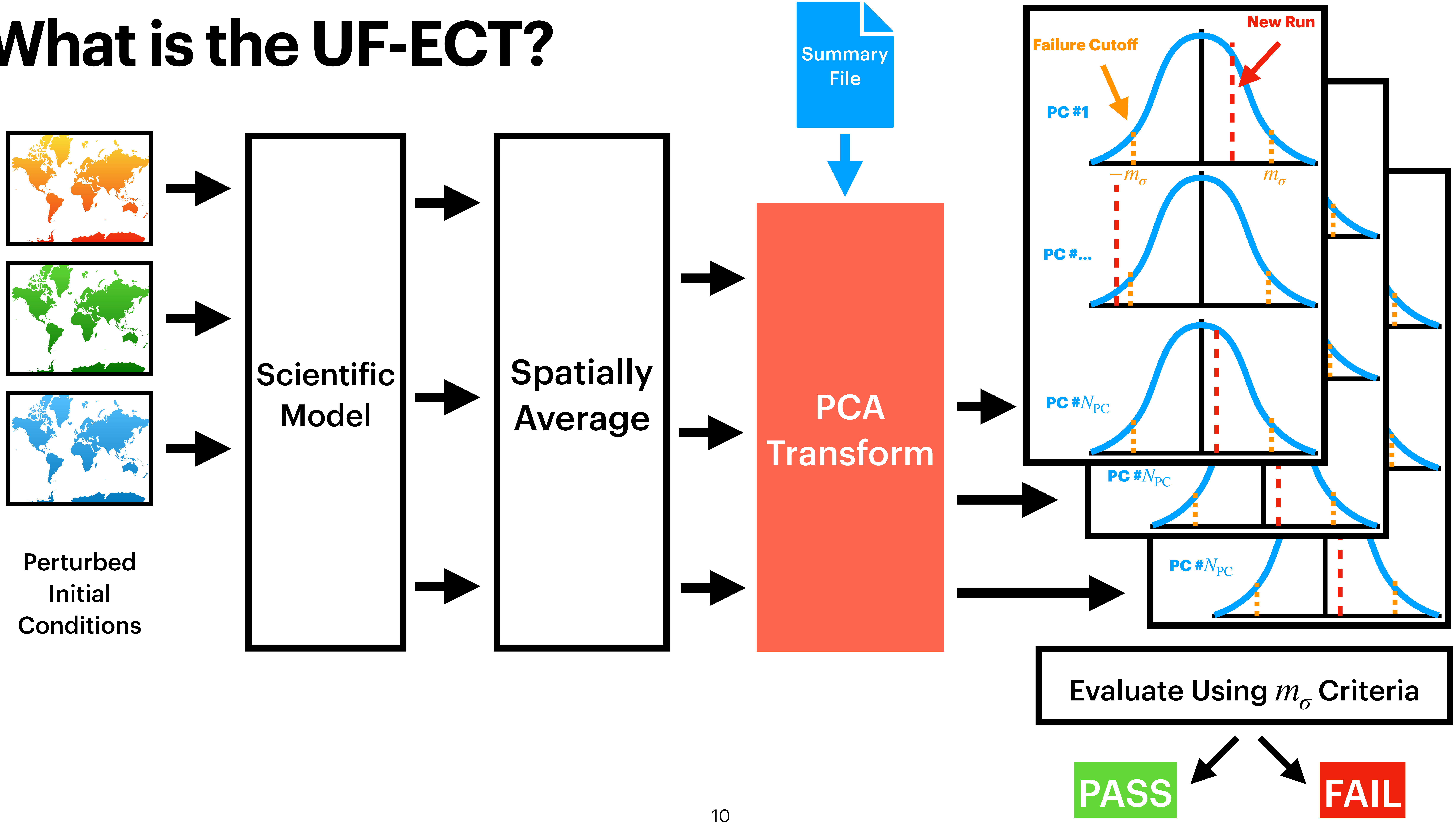
What is the UF-ECT?



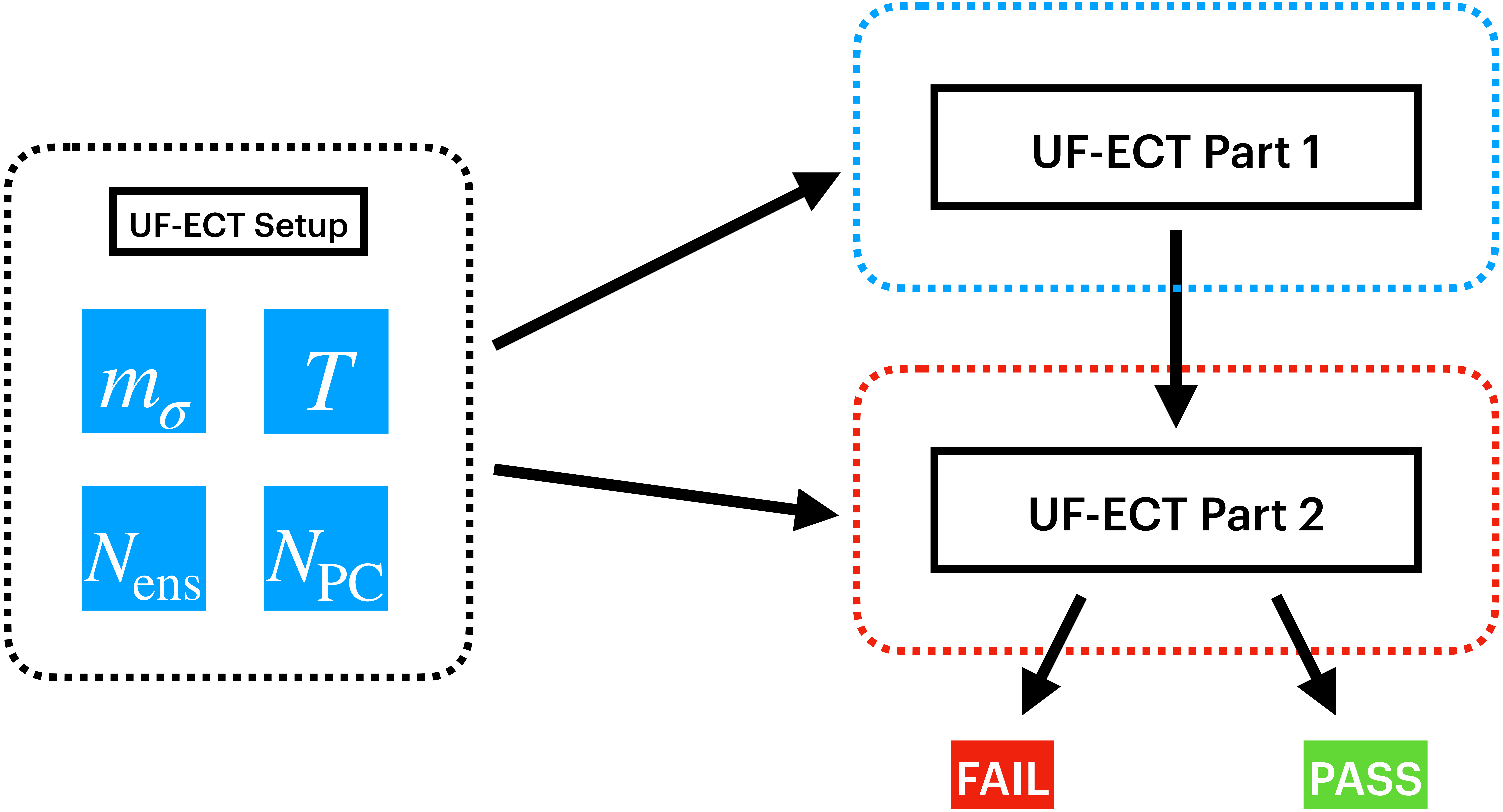
What is the UF-ECT?



What is the UF-ECT?



Recent work to make applying the UF-ECT to other models easier.



An Practical Example

How the UF-ECT Can Help Avoid Sneaky Bugs

An Practical Example

How the UF-ECT Can Help Avoid Sneaky Bugs

- One testing scenario we ran for MPAS-A involved detecting a fence post error.

An Practical Example

How the UF-ECT Can Help Avoid Sneaky Bugs

- One testing scenario we ran for MPAS-A involved detecting a fence post error.
  - Also known as an “off-by-one error”.

An Practical Example

How the UF-ECT Can Help Avoid Sneaky Bugs

- One testing scenario we ran for MPAS-A involved detecting a fence post error.
  - Also known as an “off-by-one error”.
- MPAS-A contains code to enforce a lower bound on eddy viscosities in the upper-most atmospheric layers (used in a horizontal Smagorinsky type filter).

An Practical Example

How the UF-ECT Can Help Avoid Sneaky Bugs

- One testing scenario we ran for MPAS-A involved detecting a fence post error.
  - Also known as an “off-by-one error”.
- MPAS-A contains code to enforce a lower bound on eddy viscosities in the upper-most atmospheric layers (used in a horizontal Smagorinsky type filter).
- Originally this filter was applied across a fixed number of layers (3) but was refactored to use a variable number of layers (N_{filter}).

An Practical Example

How the UF-ECT Can Help Avoid Sneaky Bugs

- One testing scenario we ran for MPAS-A involved detecting a fence post error.
  - Also known as an “off-by-one error”.
- MPAS-A contains code to enforce a lower bound on eddy viscosities in the upper-most atmospheric layers (used in a horizontal Smagorinsky type filter).
- Originally this filter was applied across a fixed number of layers (3) but was refactored to use a variable number of layers (N_{filter}).
- Even without compiler optimizations correct refactoring doesn’t yield BFB equivalent results.

A Practical Example

Algorithm 1 Baseline

$$K(N_{\text{layer}}) = \max\{K(N_{\text{layer}}), \quad 3/3 * \mu\}$$

$$K(N_{\text{layer}} - 1) = \max\{K(N_{\text{layer}} - 1), \quad 2/3 * \mu\}$$

$$K(N_{\text{layer}} - 2) = \max\{K(N_{\text{layer}} - 2), \quad 1/3 * \mu\}$$

A Practical Example: Generalized

Algorithm 1 Baseline

$$K(N_{\text{layer}}) = \max\{K(N_{\text{layer}}), \quad 3/3 * \mu\}$$
$$K(N_{\text{layer}} - 1) = \max\{K(N_{\text{layer}} - 1), \quad 2/3 * \mu\}$$
$$K(N_{\text{layer}} - 2) = \max\{K(N_{\text{layer}} - 2), \quad 1/3 * \mu\}$$

Algorithm 2 Generalized

$i \leftarrow (N_{\text{layer}} - N_{\text{filter}} + 1)$

while $i \leq N_{\text{layer}}$ **do**

$\quad K(i) = \max\{K(i), \quad (1.0 - (N_{\text{layer}} - i)/N_{\text{filter}}) * \mu\}$

end while

A Practical Example: Generalized with Bug

Algorithm 1 Baseline

$$K(N_{\text{layer}}) = \max\{K(N_{\text{layer}}), \quad 3/3 * \mu\}$$
$$K(N_{\text{layer}} - 1) = \max\{K(N_{\text{layer}} - 1), \quad 2/3 * \mu\}$$
$$K(N_{\text{layer}} - 2) = \max\{K(N_{\text{layer}} - 2), \quad 1/3 * \mu\}$$

Algorithm 3 Generalized, with off-by-one error

$i \leftarrow (N_{\text{layer}} - N_{\text{filter}} + 1)$

while $i \leq N_{\text{layer}}$ **do**

$$K(i) = \max\{K(i), \quad (1.0 - (N_{\text{layer}} - i + 1)/N_{\text{filter}}) * \mu\}$$

end while

A Practical Example: Can we identify bug?

Algorithm 1 Baseline

$$K(N_{\text{layer}}) = \max\{K(N_{\text{layer}}), \quad 3/3 * \mu\}$$
$$K(N_{\text{layer}} - 1) = \max\{K(N_{\text{layer}} - 1), \quad 2/3 * \mu\}$$
$$K(N_{\text{layer}} - 2) = \max\{K(N_{\text{layer}} - 2), \quad 1/3 * \mu\}$$

Neither is BFB

Algorithm 2 Generalized

$i \leftarrow (N_{\text{layer}} - N_{\text{filter}} + 1)$
while $i \leq N_{\text{layer}}$ **do**
  $K(i) = \max\{K(i), \quad (1.0 - (N_{\text{layer}} - i)/N_{\text{filter}}) * \mu\}$
end while

Algorithm 3 Generalized, with off-by-one error

$i \leftarrow (N_{\text{layer}} - N_{\text{filter}} + 1)$
while $i \leq N_{\text{layer}}$ **do**
  $K(i) = \max\{K(i), \quad (1.0 - (N_{\text{layer}} - i + 1)/N_{\text{filter}}) * \mu\}$
end while

A Practical Example: Compare fields

(Original - Generalized)

Field	rms(Alg. 1 - Alg. 2)
u	0.12430
w	6.20378e-3

A Practical Example: Compare fields

(Original - Generalized) (Original - Bug)

Field	rms(Alg. 1 - Alg. 2)	rms(Alg. 1 - Alg. 3)
u	0.12430	0.12436
w	6.20378e-3	6.29749e-3

Very similar magnitudes!

A Practical Example: Compare fields

	(Original - Generalized)	(Original - Bug)	(Original - Generalized)	(Original - Bug)
Field	**rms(Alg. 1 - Alg. 2)**	**rms(Alg. 1 - Alg. 3)**	**max{|Alg. 1 - Alg. 2|}**	**max{|Alg. 1 - Alg. 3|}**
u	0.12430	0.12436	8.37051	10.84282
w	6.20378e-3	6.29749e-3	3.21694e-1	5.98148e-1

Again similar magnitudes.

Can the UF-ECT provide an automated way of identifying these bugs?

A Practical Example: UF-ECT

		Test Failure Rate
Generalized	Algorithm 2	0.22%
Bug	Algorithm 3	100%

Conclusions

Conclusions

- The UF-ECT is an ensemble-based testing framework to identify changes in expensive computer models.

Conclusions

- The UF-ECT is an ensemble-based testing framework to identify changes in expensive computer models.
- We have developed a setup framework to make it easier to apply the UF-ECT method to new computer models like MPAS-A and CESM 3.

Conclusions

- The UF-ECT is an ensemble-based testing framework to identify changes in expensive computer models.
- We have developed a setup framework to make it easier to apply the UF-ECT method to new computer models like MPAS-A and CESM 3.
- UF-ECT can help model developers identify unintentional bugs in a fairly automated way with high accuracy.

Thanks!