Towards accurate extreme event likelihoods from diffusion model climate emulators

Peter Manshausen, Noah Brenowitz, Julius Berner, Karthik Kashinath, Mike Pritchard

ML for Earth Observation and Prediction, Bologna

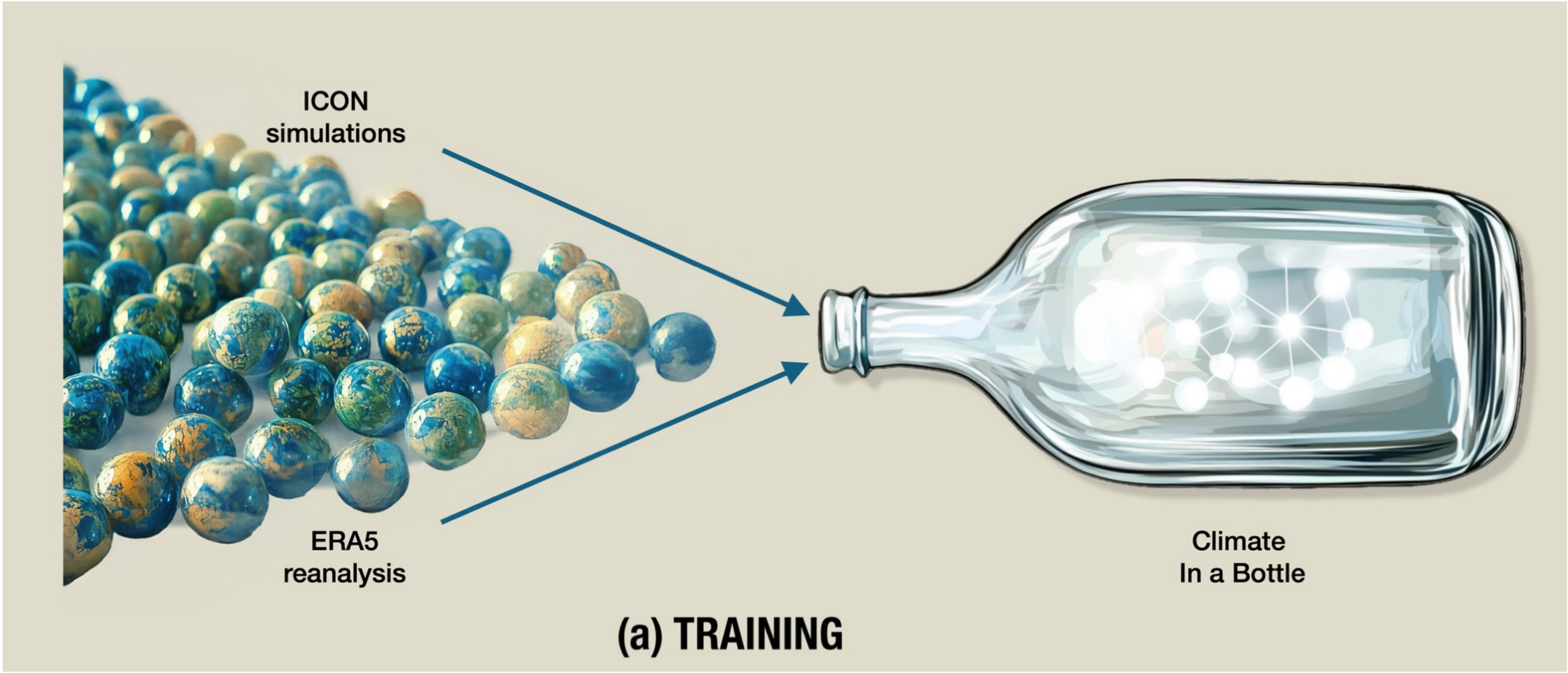
16 April 2026

Why we need (interactive) climate model emulators

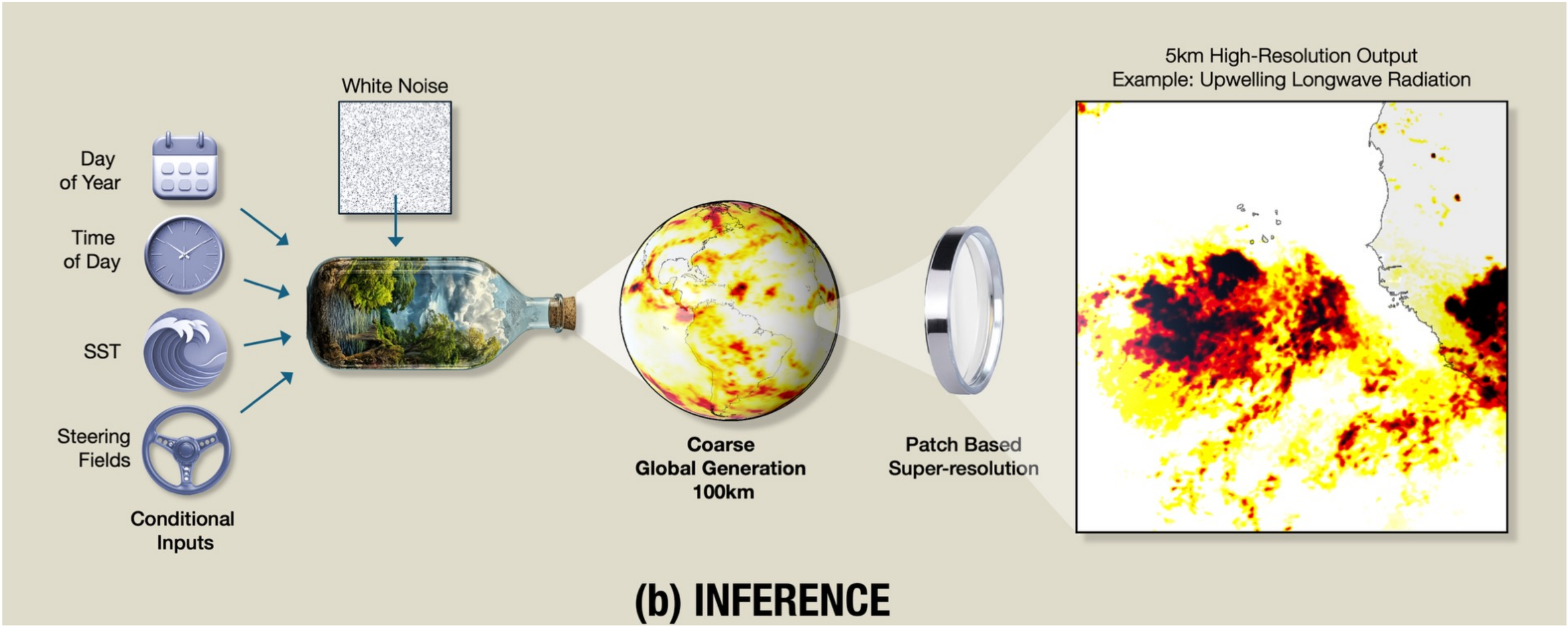
- Traditional climate models (GCMs) are expensive and slow.
- Petabyte-scale high-resolution simulations (ICON, SCREAM) are difficult to store, move, and analyze.
- Climate extremes (e.g., heatwaves, hurricanes) require targeted sampling and ensemble generation.
- Need for interactive, scalable, generative models.

Climate-in-a-Bottle: Petabytes to Gigabytes to Interactive Sampling

Generative Data Compression and Reconstruction

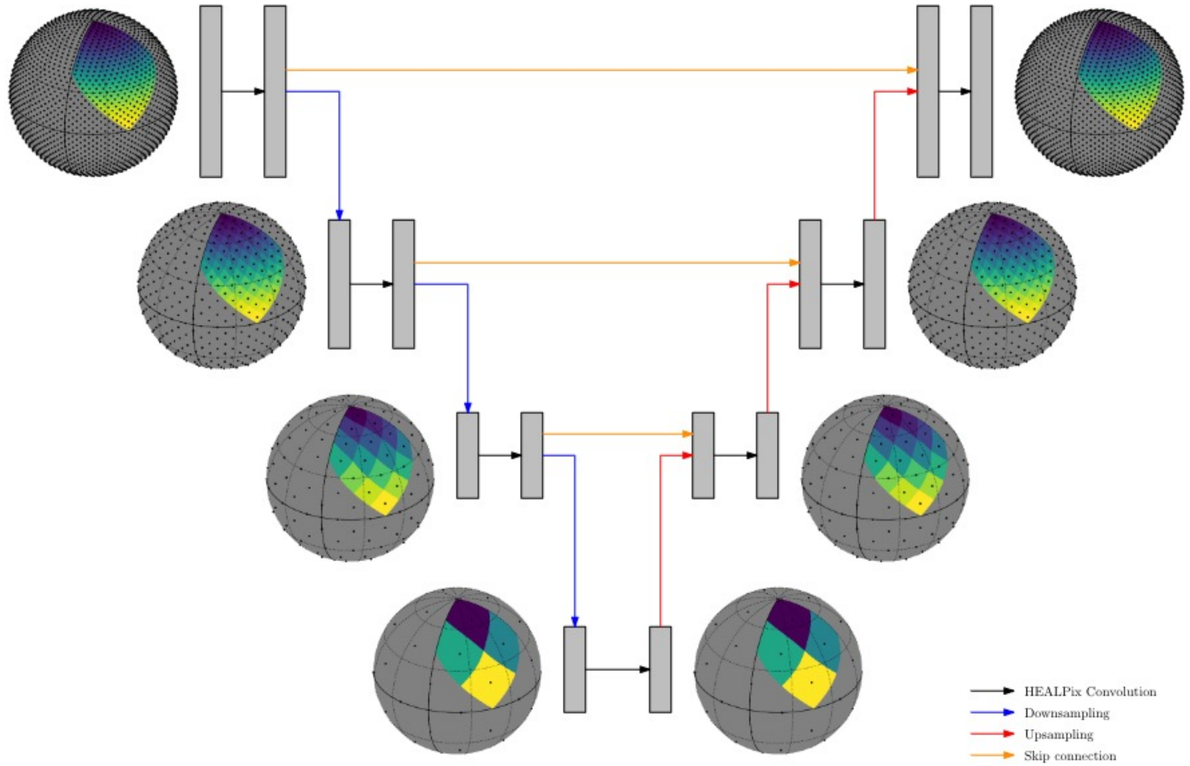


Conditional Weather Generation



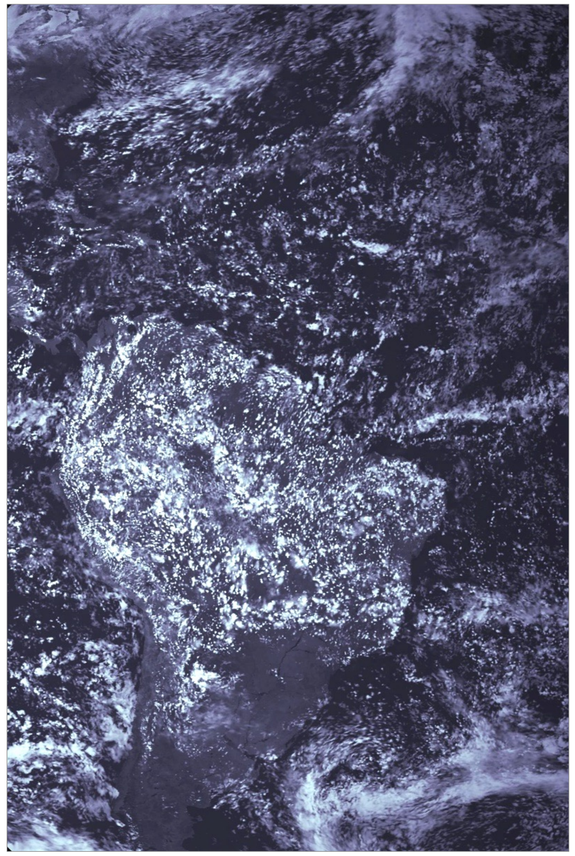
Architecture: Song UNet with Elucidated Diffusion on a HEALPix Sphere

Including Padding Adapted for the HEALPix Grid

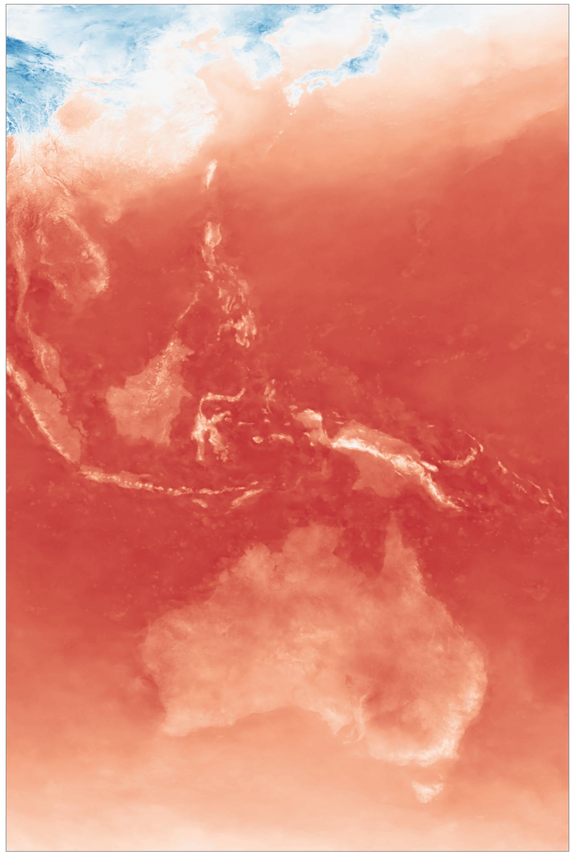


Climate-in-a-Bottle: Multi-Modal 5km Global Generation

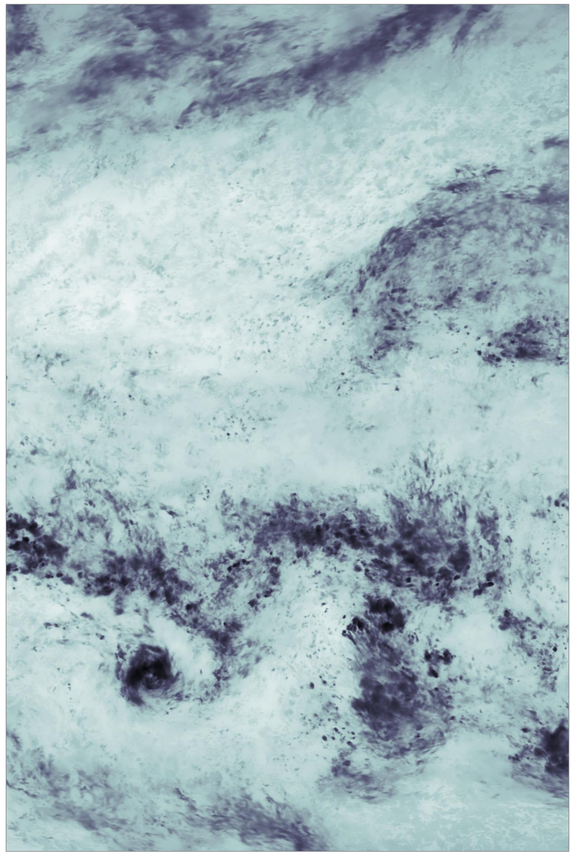
From Petabyte Archives to Gigabyte Models



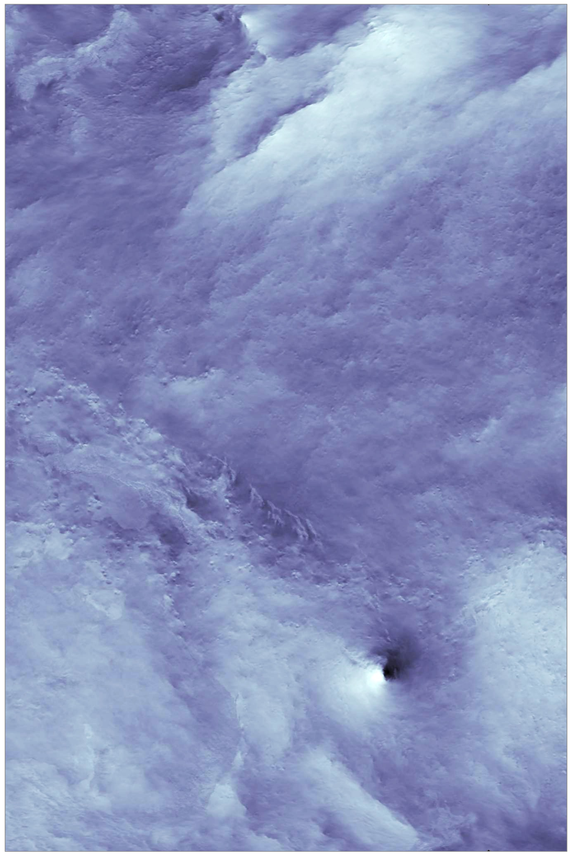
Visible



Temperature



Outgoing Longwave Infrared



East-west surface winds

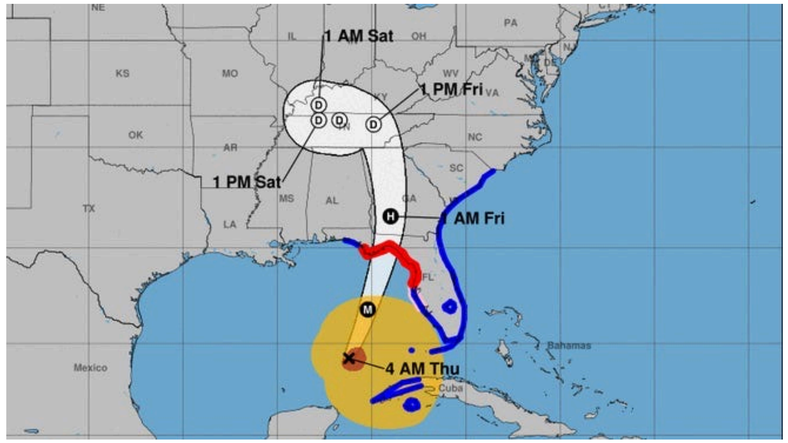
NVIDIA.

What-If Scenarios & Steerable Cyclone Sampling

Steerable Climate Sampling Would Be Useful

“AI-on-Top” Should Enable Interactive Experiences

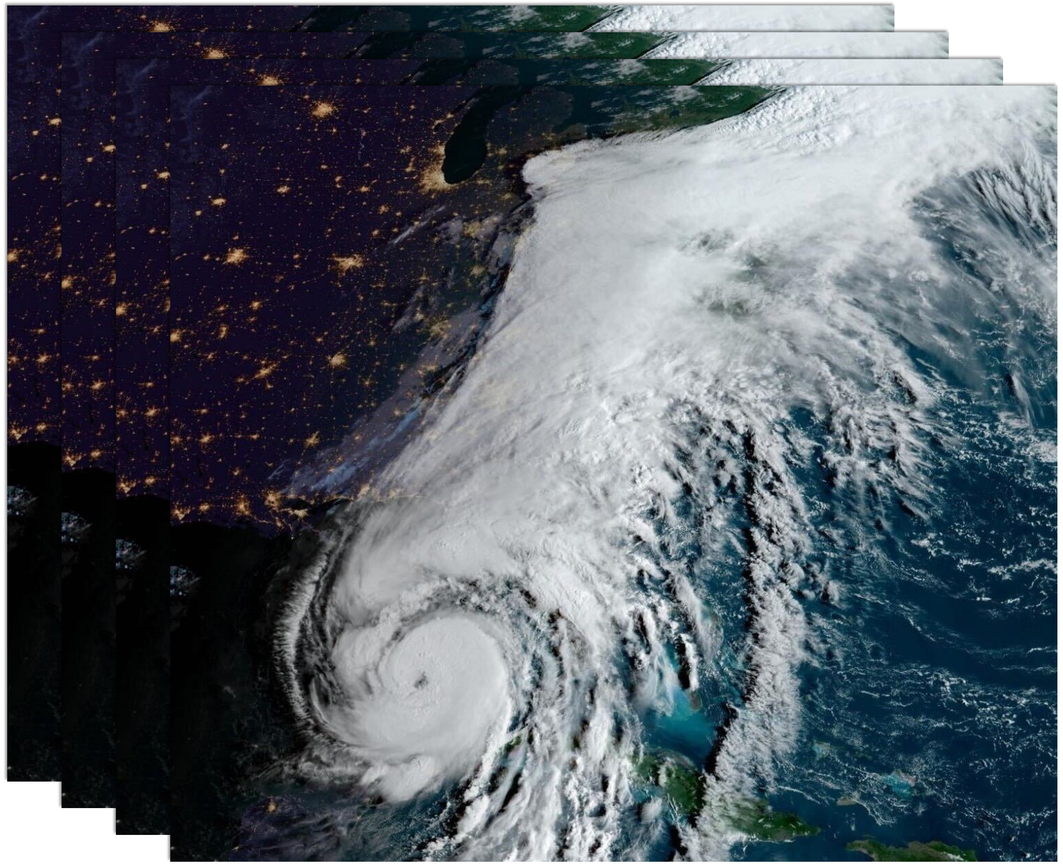
User guided scenario



or

Prompt: Show me several realizations of a potential hurricane impacting the gulf coast given the current average surface conditions. Output precipitation and wind damage maps.

Multiple realizations



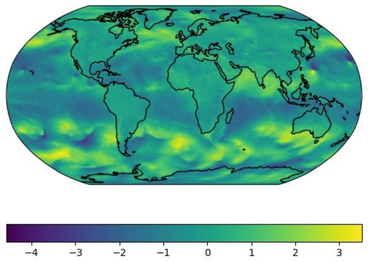
Hurricane Helene, Sept 26, *NOAA*

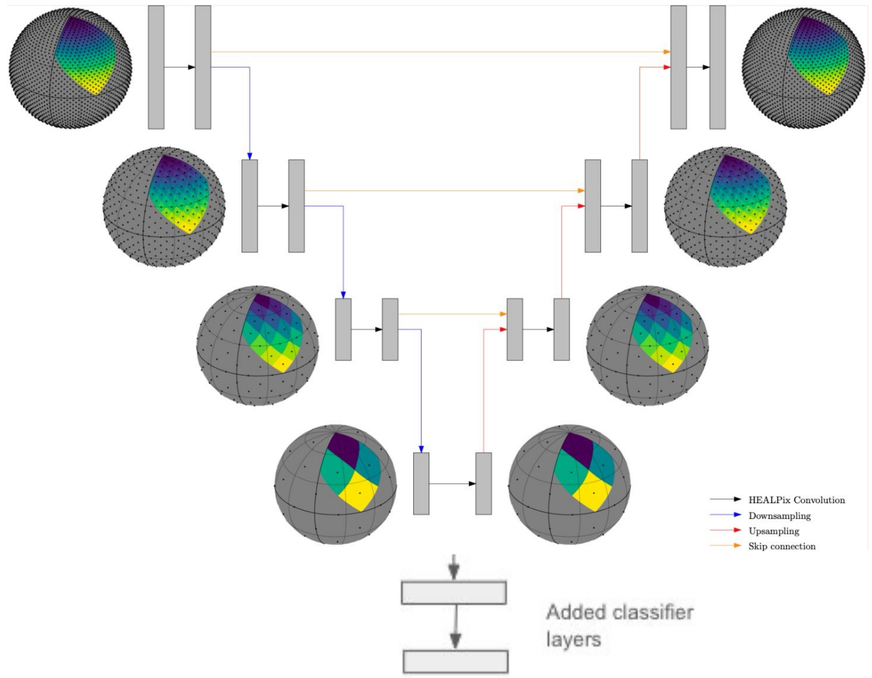
NVIDIA.

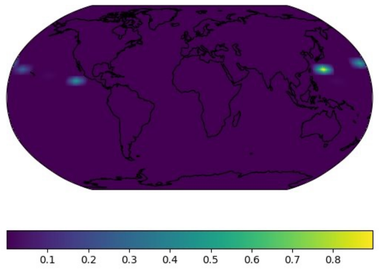
Can we add Classifier Guidance to Query cBottle?

Desired Feature: Ability to Ask for a Sample with a Tropical Cyclone in a Location of Interest

- During inference, **guide with classifier towards desired states**

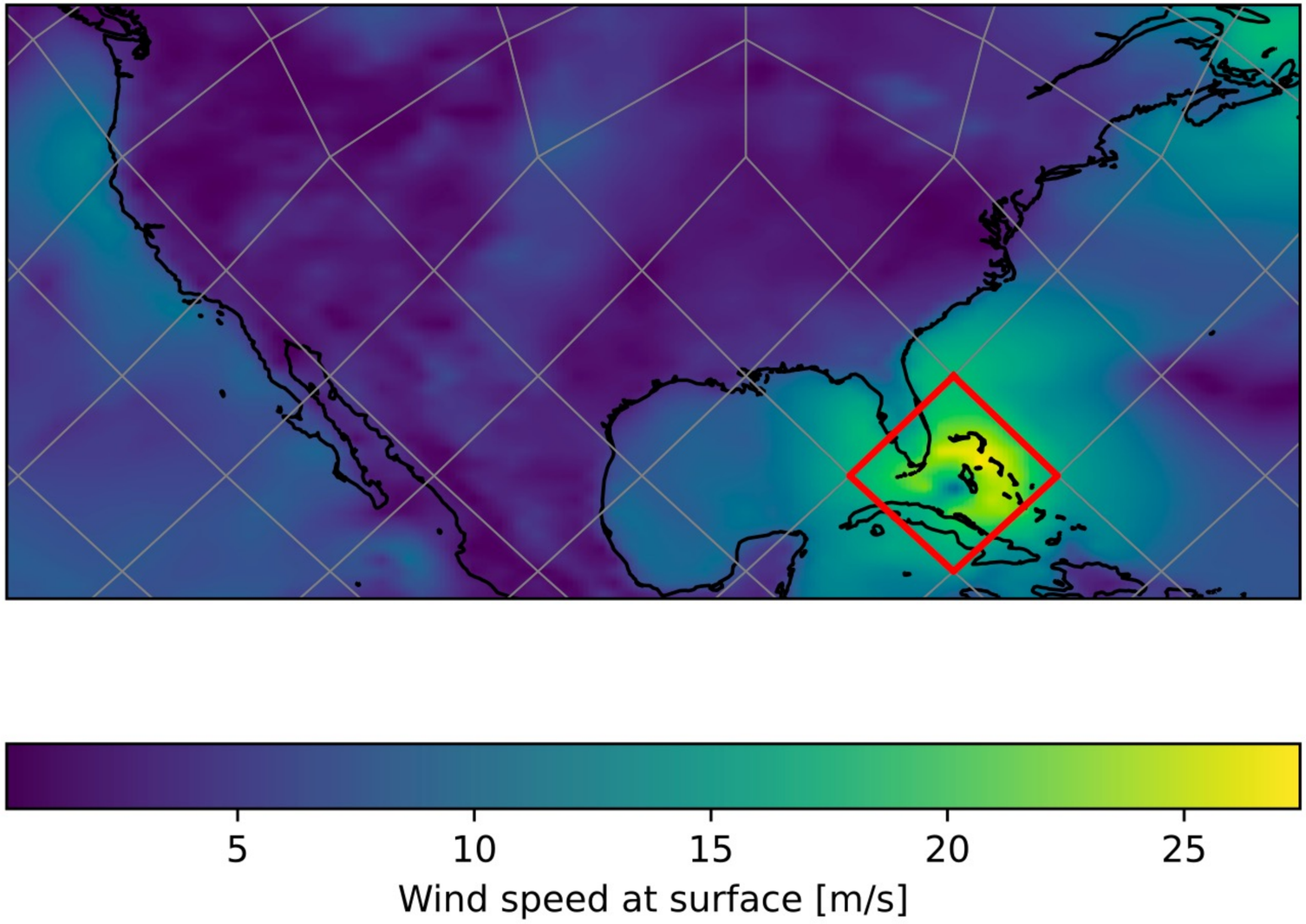


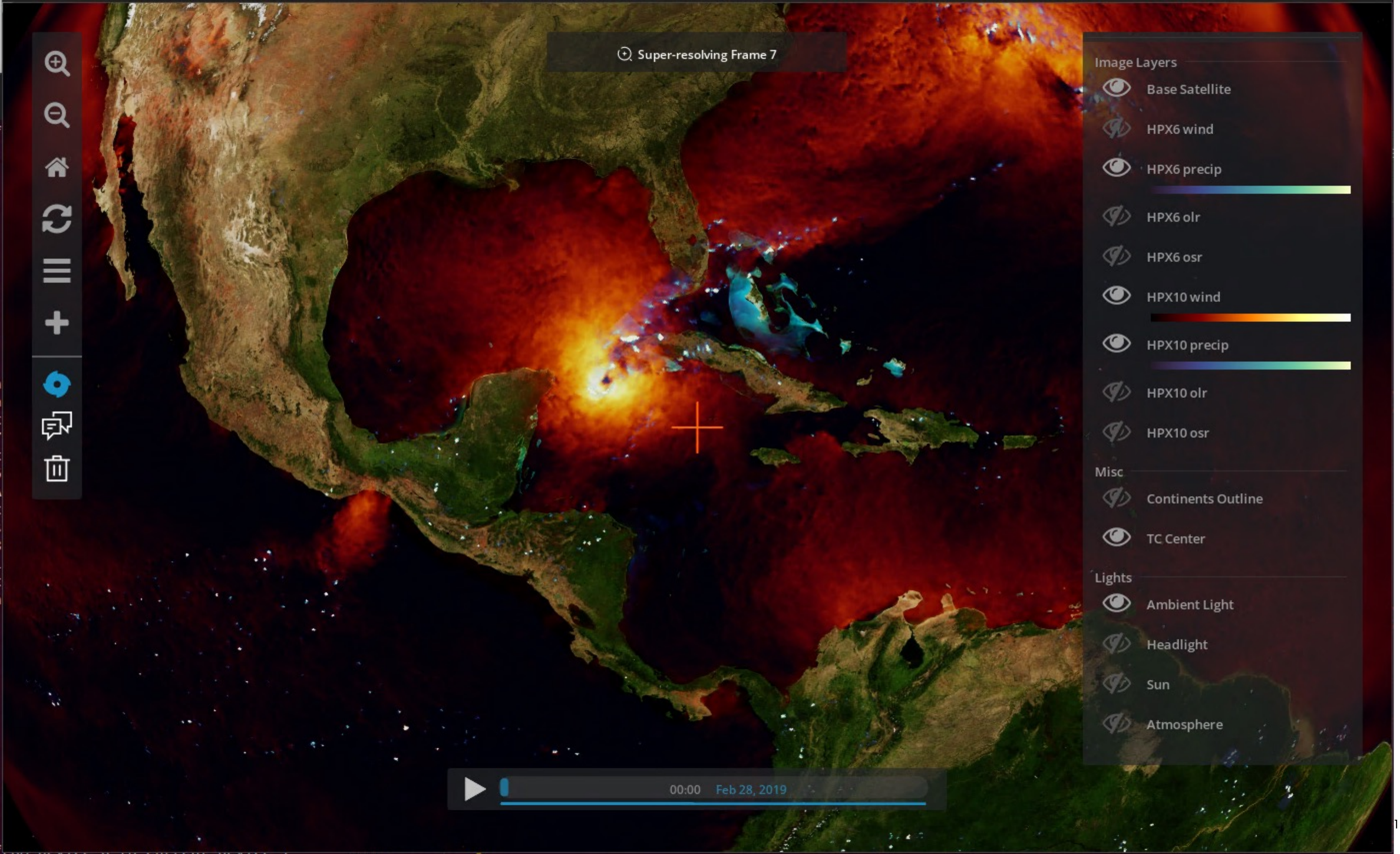




Zoom

Red square shows low-res healpix pixel where TC guidance applied




Super-resolving Frame 7
Image Layers
Base Satellite
HPX6 wind
HPX6 precip
HPX6 olr
HPX6 osr
HPX10 wind
HPX10 precip
HPX10 olr
HPX10 osr
Misc
Continents Outline
TC Center
Lights
Ambient Light
Headlight
Sun
Atmosphere
00:00
Feb 28, 2019

NVIDIA.

But now, every sample has a TC!

How can we get anything like an event frequency back from these?

But now, every sample has a TC!

How can we get anything like an event frequency back from these?

- Importance sampling:
  - Idea is to oversample the part of the distribution one is interested in
  - Then downweight the samples with the odds ratio of the oversampling

- **Before** (Monte Carlo estimate): $$p(TC) \approx \frac{1}{K}\sum_{i=1}^{K}\mathbb{I}_{TC}(\mathbf{x}'_i), \qquad where \;\; \mathbf{x}'_i \sim p_{\text{cBottle}}$$

- **After** (Importance Sampling): $$p_{IS}(TC) \approx \frac{1}{K}\sum_{i=1}^{K}\mathbb{I}_{TC}(\mathbf{x}_i)o(\mathbf{x}_i), \qquad where \;\; \mathbf{x}_i \sim p_{\text{guided}}$$

$$o(\mathbf{x}) = \frac{p_{\text{unguided}}(\mathbf{x})}{p_{\text{guided}}(\mathbf{x})}$$

How do we find odds ratios?

Using the log probability calculated from the Probability Flow ODE

- Song et al. (2021), Appendix D, give the expression for evaluating the probability density of a sample from the denoising ODE:
- Trained denoiser approximates score: $\nabla_{\mathbf{x}} \log p(\mathbf{x};\sigma) \approx \frac{\mathcal{D}_\theta(\mathbf{x};\sigma) - \mathbf{x}}{\sigma^2}$
- Denoising ODE: $\frac{d\mathbf{x}}{d\sigma} = \frac{\mathbf{x} - \mathcal{D}_\theta(\mathbf{x};\sigma)}{\sigma}$
- Probability is composed of final probability of noise plus change of probability density
- Continuity equation: change in probability density is equal to divergence of flow (RHS of ODE)

$$\log p(\mathbf{x}_0) = \log p(\mathbf{x}_T) + \int_0^T \nabla \cdot \mathbf{v}_\theta(\mathbf{x}(t), t)\, dt$$

- Goal: evaluate for guided and unguided model

$$o(\mathbf{x}) = \frac{p_{\text{unguided}}(\mathbf{x})}{p_{\text{guided}}(\mathbf{x})}$$

How do we find odds ratios?

Using the log probability calculated from the Probability Flow ODE

- Song et al. (2021), Appendix D, give the expression for evaluating the probability density of a sample from the denoising ODE:

- Trained denoiser approximates score: $$\nabla_{\mathbf{x}} \log p(\mathbf{x};\sigma) \approx \frac{\mathcal{D}_\theta(\mathbf{x};\sigma) - \mathbf{x}}{\sigma^2}$$

- Denoising ODE: $$\frac{d\mathbf{x}}{d\sigma} = \frac{\mathbf{x} - \mathcal{D}_\theta(\mathbf{x};\sigma)}{\sigma} + \gamma \nabla_{\mathbf{x}} \mathcal{L}_{BCE}(\mathcal{C}(\mathbf{x}), y)$$

- Probability is composed of final probability of noise plus change of probability density
- Continuity equation: change in probability density is equal to divergence of flow (RHS of ODE)

$$\log p(\mathbf{x}_0) = \log p(\mathbf{x}_T) + \int_0^T \nabla \cdot \mathbf{v}_\theta(\mathbf{x}(t), t)\, dt$$

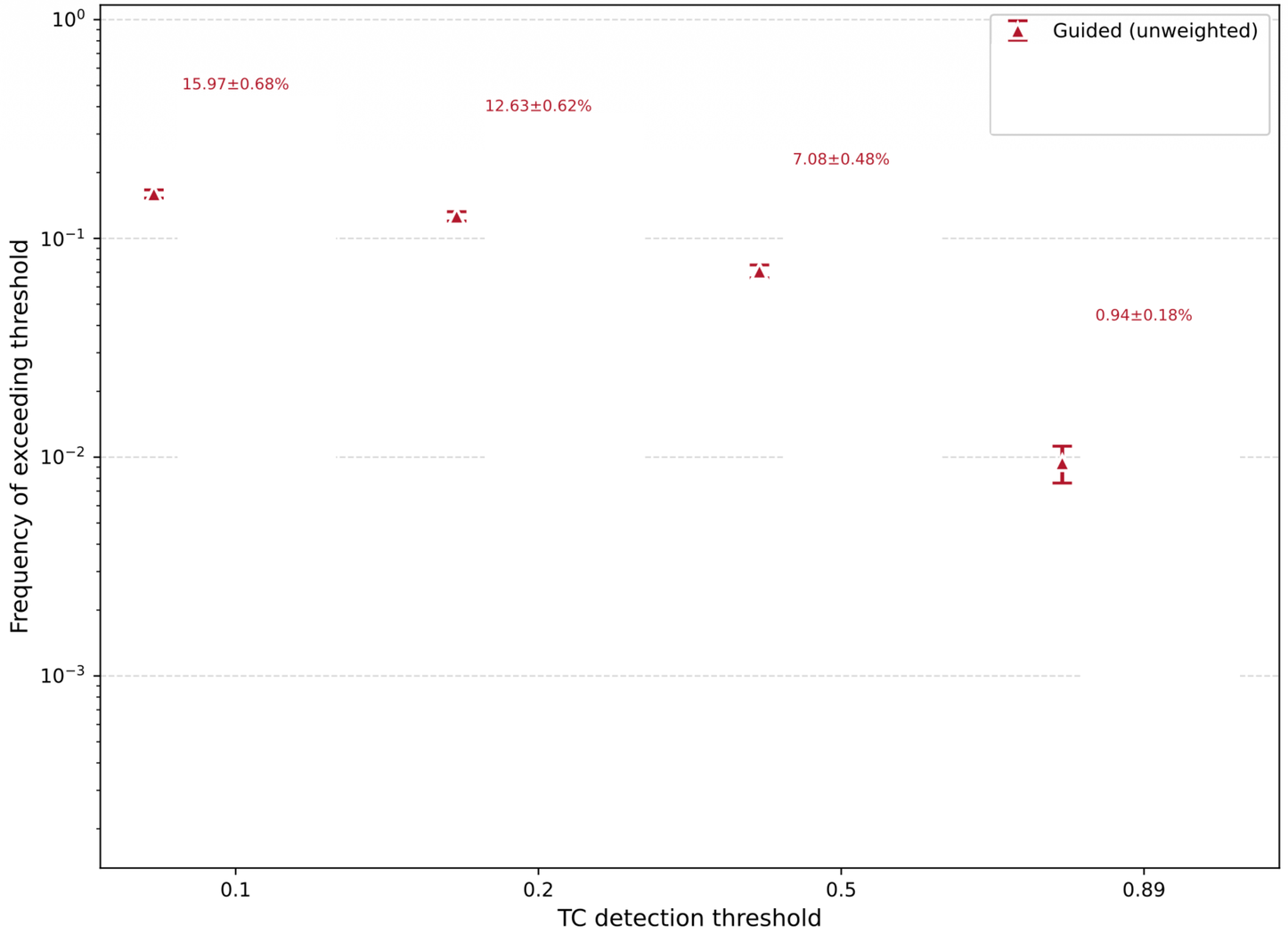
- Goal: evaluate for guided and unguided model

$$o(\mathbf{x}) = \frac{p_{\text{unguided}}(\mathbf{x})}{p_{\text{guided}}(\mathbf{x})}$$

NVIDIA.

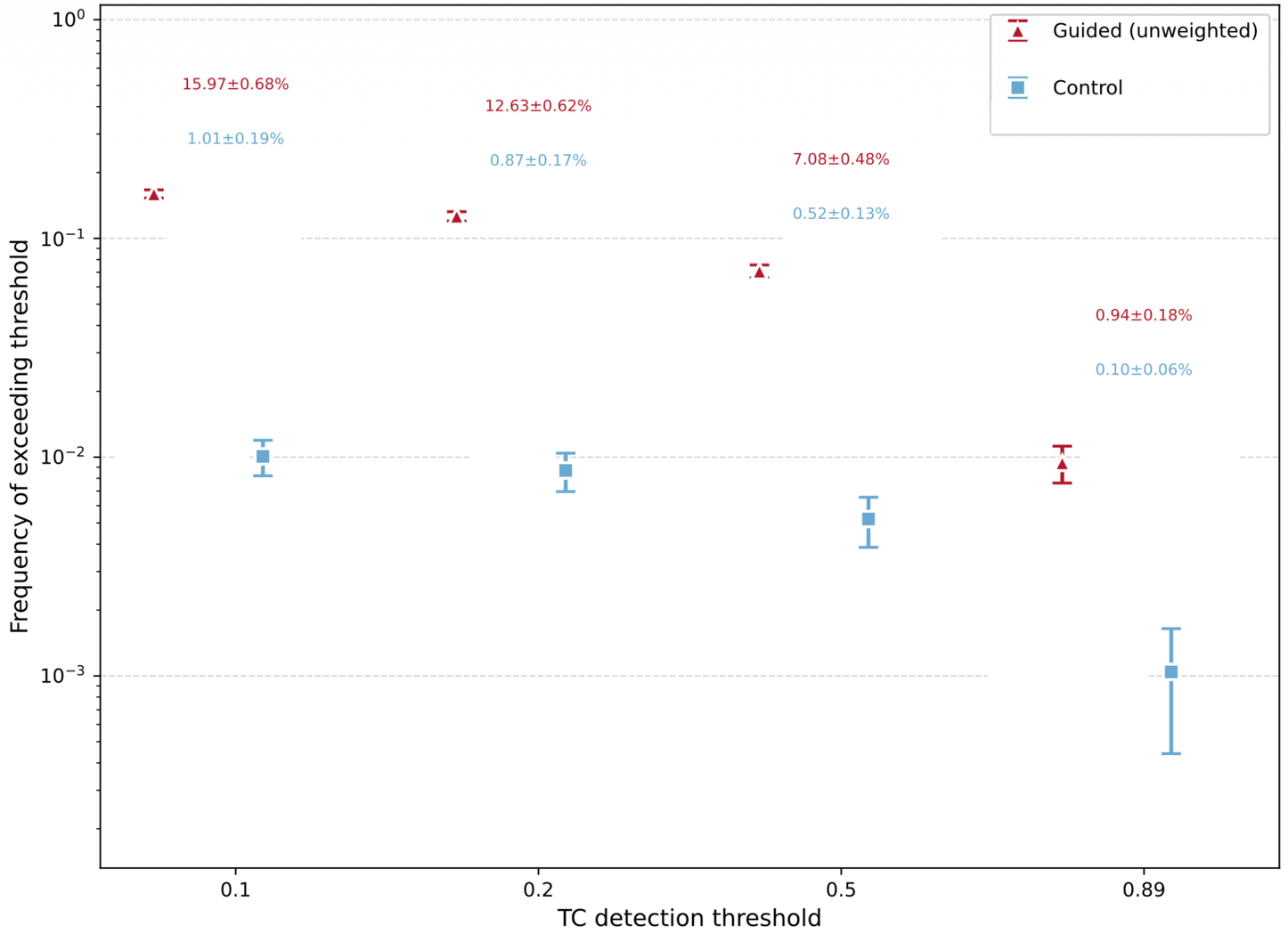
How well does the Importance sampling work?

Showing guided TC frequencies as a function of detection threshold



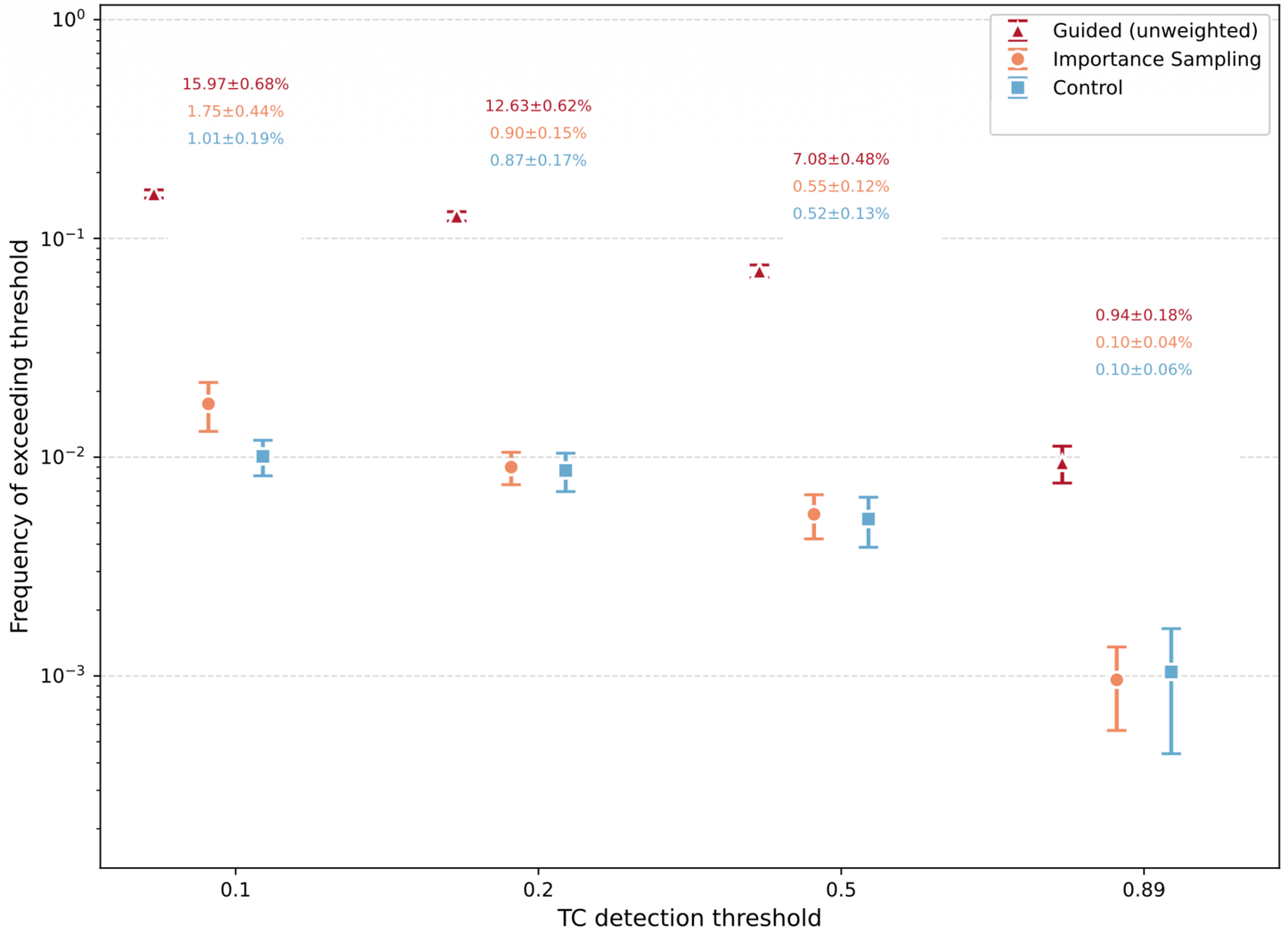
How well does the Importance sampling work?

Compare: unguided control simulation (Monte Carlo estimate)



How well does the Importance sampling work?

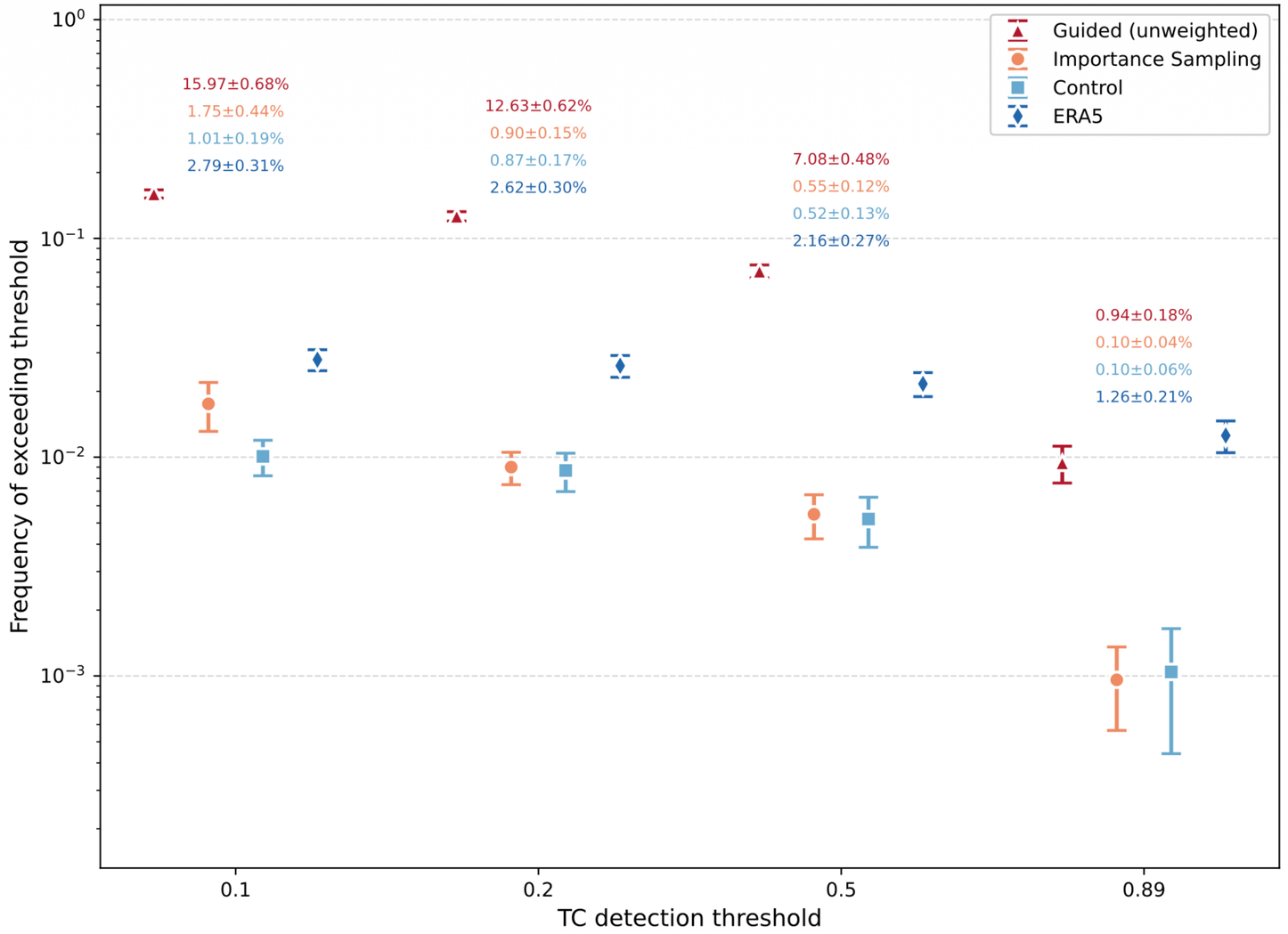
IS estimate agrees (mostly) with Control!



NVIDIA

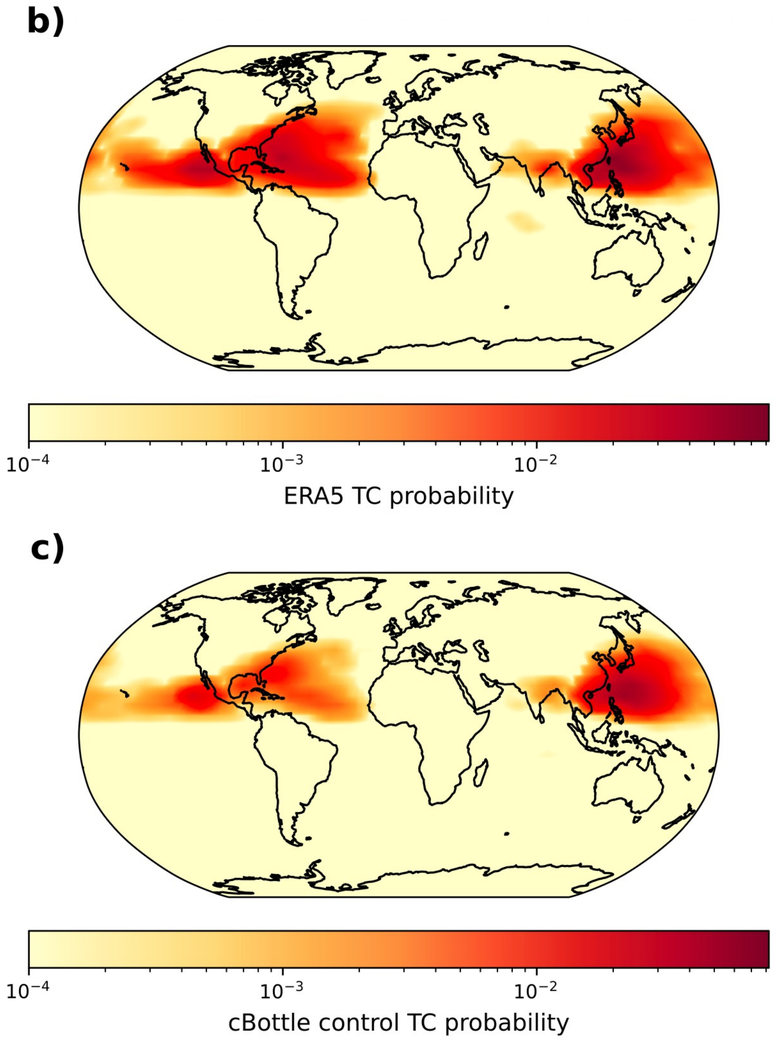
Problem: Underlying model (control) isn't perfect

cBottle systematically underestimates (stronger) TCs



Particularly in the Atlantic…

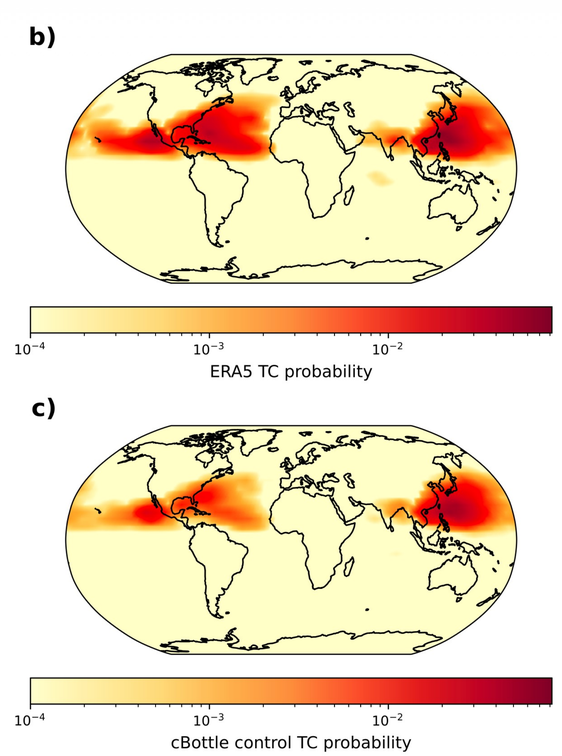
cBottle TCs are too rare and too weak



Improving the base model is WIP

cBottle may need architecture changes

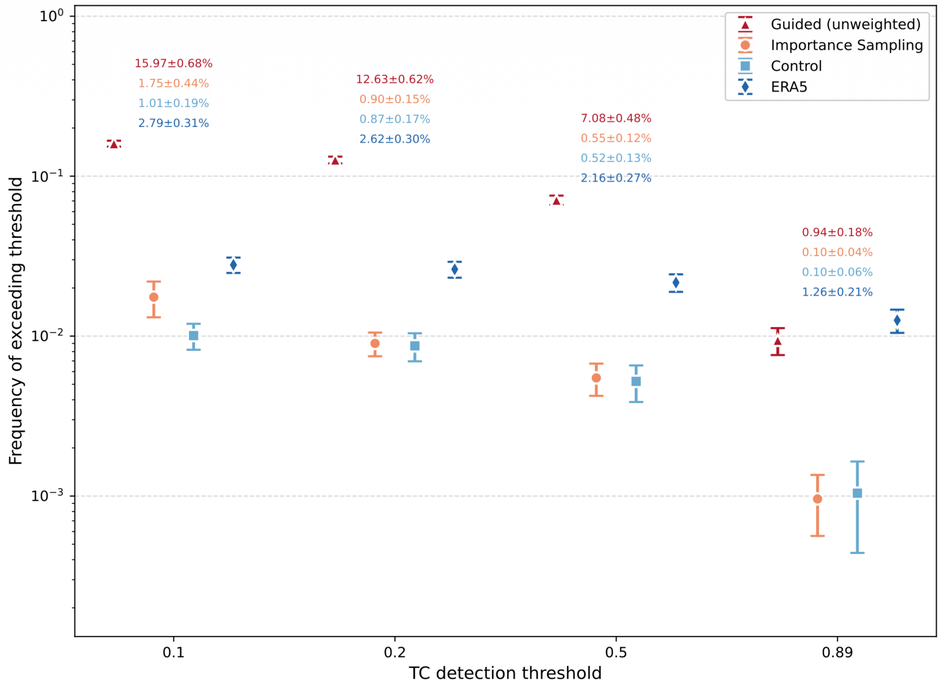
- Model is showing too few and too weak TCs, contrary to e.g. ACE2 (which is strong- and frequent biased)
- Improving the underlying model should improve the Monte Carlo estimate
- Importance sampling should follow suit



Note on compute

The evaluation of the PF-ODE is currently slow

- As shown, we obtain reduced uncertainties for the most extreme TCs, when MC estimates suffer from sampling uncertainty
- However, we can reduce sampling uncertainty by just sampling more often
- Currently, the IS estimate is 33x slower per sample, which could in theory be brought down to 2x at most through model distillation.

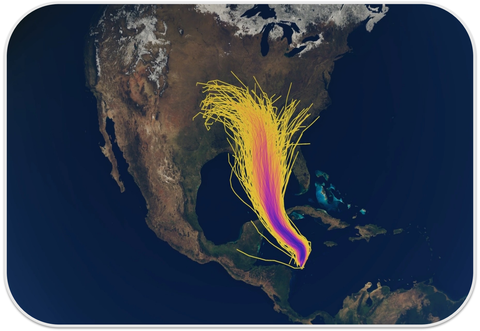


Summary

Towards interactive extreme events with likelihoods

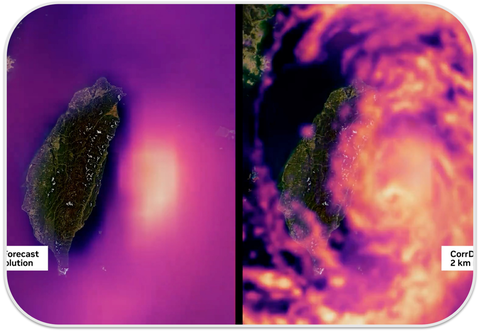
- cBottle is a diffusion model that emulates the present day climate
- cBottle can be interactively run to return samples containing Tropical Cyclones in a location of interest
- We can get back information on the likelihood of these events
- By using the probability flow ODE to estimate how much we have oversampled
- WIP to improve cBottle TC predictions

NVIDIA’s Earth-2 Tools Are Open Source & Permissively Licensed



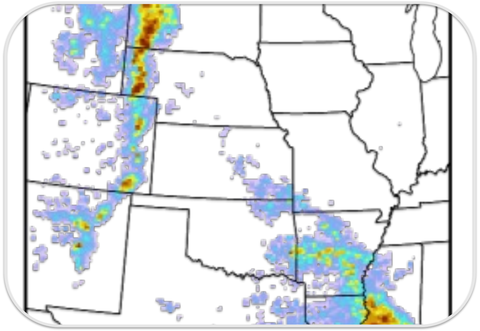
FourCastNet3

NWP: 1000+ Member AI Ensembles



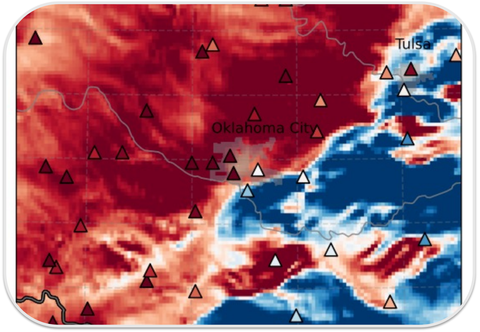
CorrDiff

AI Downscaling 25→ 3km



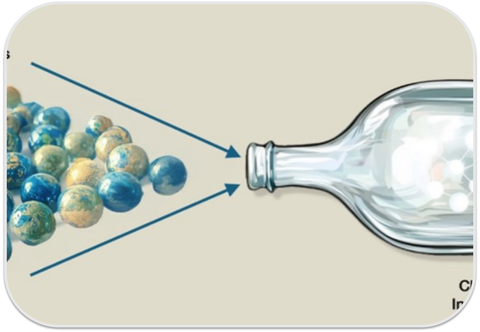
StormCast

3km Regional AI Forecasts



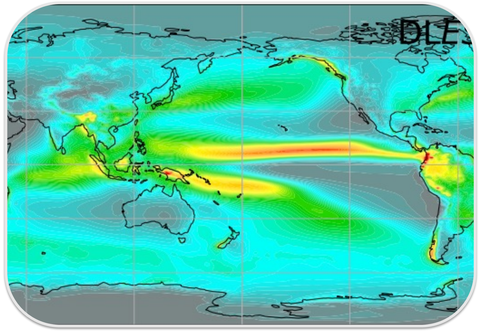
SDA

AI Data Assimilation



Climate in a Bottle

Dataset compressed into portable AI model

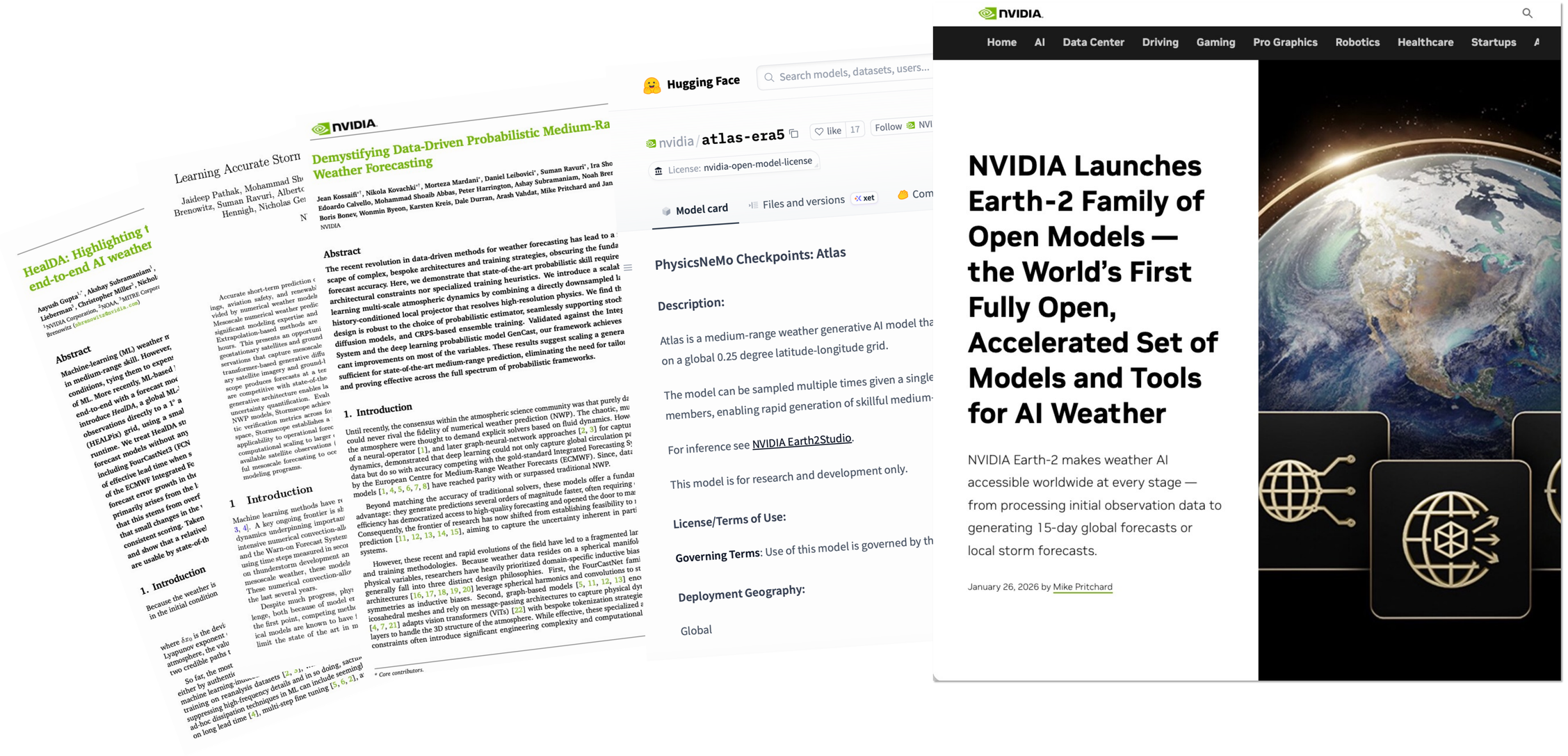


DLESyM

AI Earth System Model Prototype

Earth-2 AI

Check out three new models for nowcasting, medium range, and global data assimilation (all released in Jan)



Please See Earth-2 Studio: Production-Ready Python APIs

Open-Source Integration with 40+ AI Weather Models

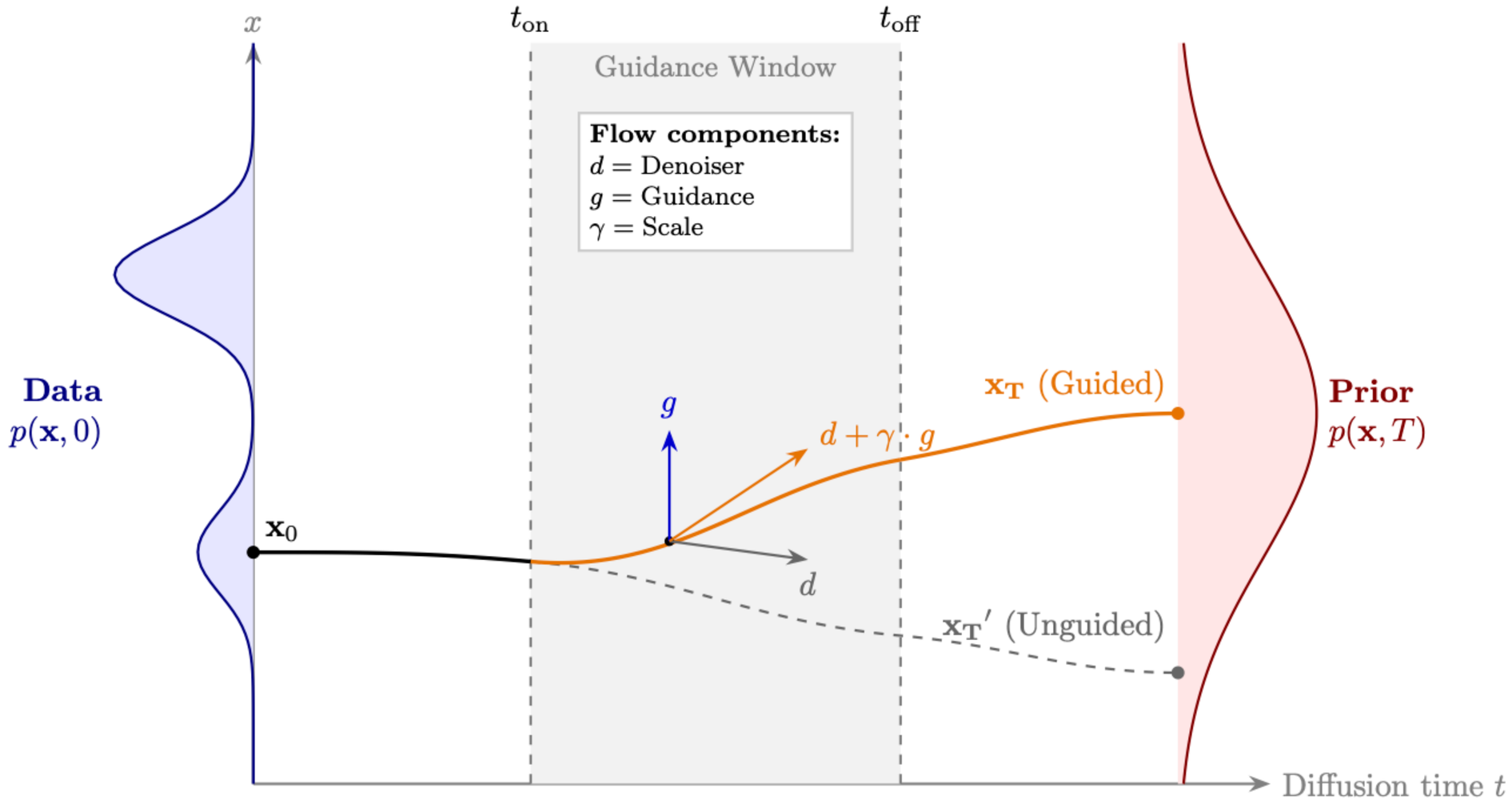
- **earth2studio.models: Models**
  - earth2studio.models.px.AIFS
  - earth2studio.models.px.Aurora
  - earth2studio.models.px.CBottleVideo
  - earth2studio.models.px.DLESyM
  - earth2studio.models.px.DLESyMLatLon
  - earth2studio.models.px.DLWP
  - earth2studio.models.px.FCN
  - earth2studio.models.px.FCN3
  - earth2studio.models.px.FengWu
  - earth2studio.models.px.FuXi
  - earth2studio.models.px.GraphCastOperational
  - earth2studio.models.px.GraphCastSmall
  - earth2studio.models.px.InterpModAFNO
  - earth2studio.models.px.Pangu24
  - earth2studio.models.px.Pangu6
  - earth2studio.models.px.Pangu3
  - earth2studio.models.px.SFNO
  - earth2studio.models.px.StormCast
  - earth2studio.models.dx.CBottleInfill
  - earth2studio.models.dx.CBottleSR
  - earth2studio.models.dx.CBottleTCGuidance
  - earth2studio.models.dx.CorrDiffTaiwan
  - earth2studio.models.dx.ClimateNet
  - earth2studio.models.dx.DerivedRH
  - earth2studio.models.dx.DerivedRHDewpoint
  - earth2studio.models.dx.DerivedVPD
  - earth2studio.models.dx.DerivedWS
  - earth2studio.models.dx.PrecipitationAFNO
  - earth2studio.models.dx.PrecipitationAFNOv2
  - earth2studio.models.dx.SolarRadiationAFNO1H
  - earth2studio.models.dx.SolarRadiationAFNO6H
  - earth2studio.models.dx.TCTrackerWuDuan
  - earth2studio.models.dx.TCTrackerVitart
  - earth2studio.models.dx.WindgustAFNO

```python
# Direct UFS integration with Earth-2 Studio
import earth2studio as e2s

# Load UFS initial conditions
ufs_state = e2s.data.UFS(path="gfs.t00z.pgrb2")

# Generate 1000-member ensemble with FourCastNet 3
fcn3_ensemble = e2s.FourCastNet3(
    initial_conditions=ufs_state,
    members=1000
)

# Downscale with CorrDiff-US
high_res = e2s.CorrDiffUS(fcn3_ensemble)

# Output in UFS-compatible format
e2s.io.save_grib2(high_res, "enhanced_forecast.grib2")
```

Thanks.

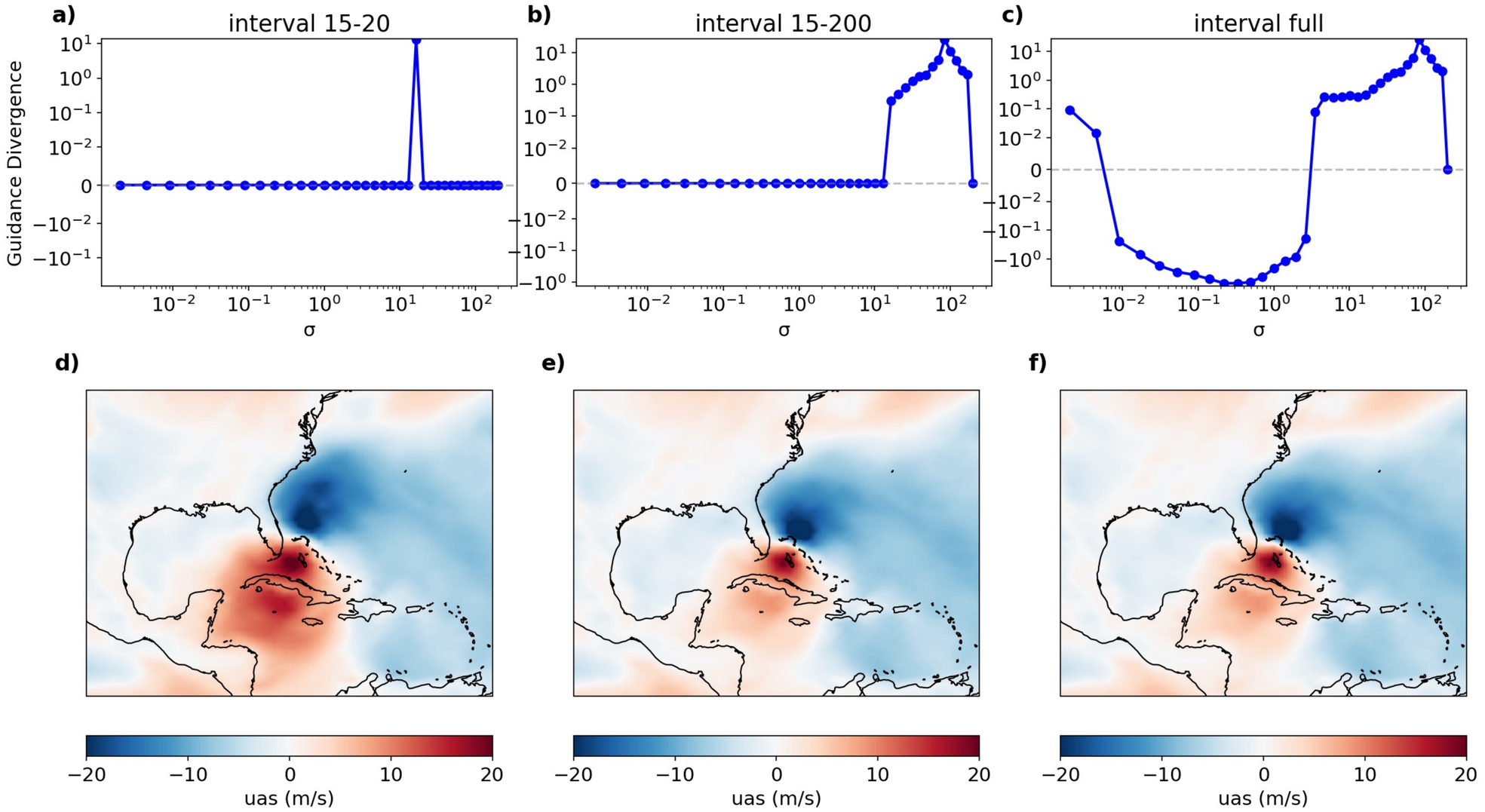
How do we find odds ratios?

Using the log probability calculated from the Probability Flow ODE



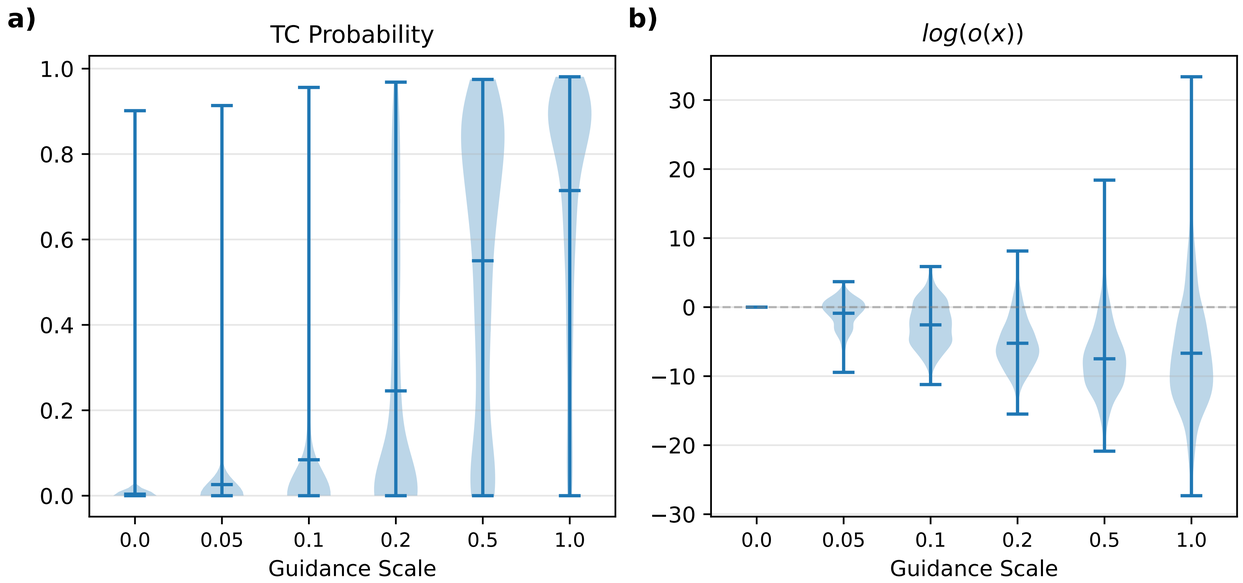
Guidance can be applied in a small interval

The denoising is most sensitive to certain phenomena at certain noise levels



For strong guidance, large variance in odds ratios

Proceed with smaller guidance so weights don’t explode



Odds ratio calculation

Log probability of samples under unguided minus log probability under guided model

- Odds ratio as difference between log probs of the same sample under the guided vs unguided model
- Difference in latent probabilities plus difference in flow divergence

$$\log o(\mathbf{x}) = \log(p_T(\mathbf{x}'_T))) - \log(p_T(\mathbf{x}_T))$$

$$+ \int_0^T \nabla \cdot [\mathbf{v}'_\theta(\mathbf{x}(t), t, A) - \mathbf{v}_\theta(\mathbf{x}(t), t)]\, dt$$

$$= \underbrace{\log(p_T(\mathbf{x}'_T))) - \log(p_T(\mathbf{x}_T))}_{\text{probability of latent under Gaussian prior}}$$

$$+ \int_0^T \underbrace{\nabla \cdot \left(\frac{\mathbf{x}' - \mathcal{D}_\theta(\mathbf{x}', \sigma)}{\sigma} - \frac{\mathbf{x} - \mathcal{D}_\theta(\mathbf{x}, \sigma)}{\sigma} \right)}_{\text{divergence of denoiser terms along different trajectories}} dt$$

$$- \int_0^T \underbrace{\nabla \cdot \gamma \nabla_{\mathbf{x}} \mathcal{L}_{BCE}(\mathcal{C}(\mathbf{x}), y)}_{\text{divergence of guidance term}}\, dt$$