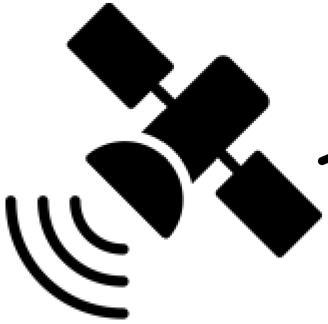
DiffDA: a Diffusion Model for Weather-scale Data Assimilation

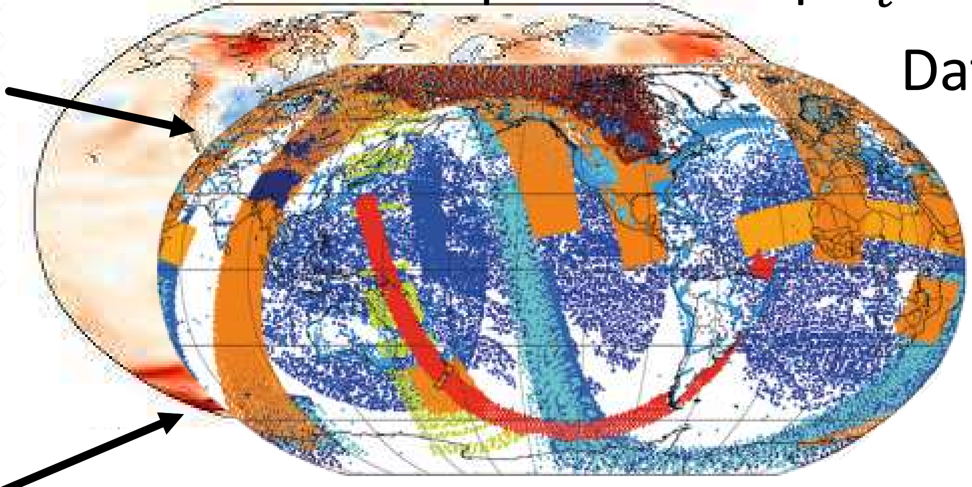
Langwen Huang, Lukas Gianinazzi, Yuejiang Yu, Peter Dominik Dueben, Torsten Hoefler

Motivation

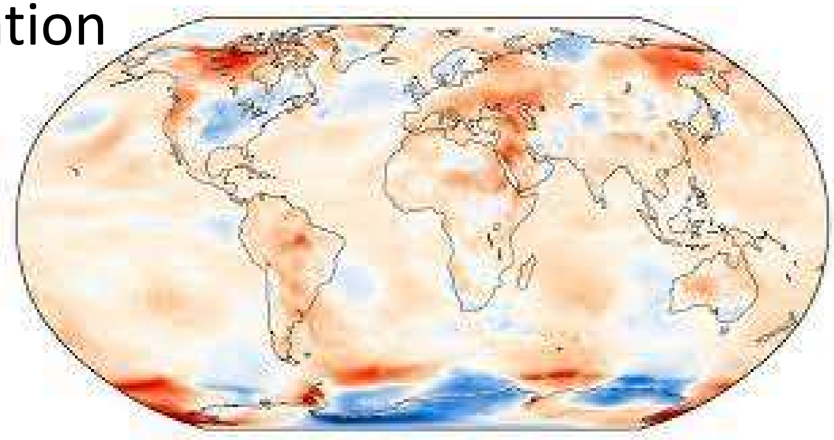
- **Traditional DA methods are slow.**
- **Traditional DA methods make point estimatimation for posterior distribution.**
- **DA tools are not easily available**
- **AI weather models rely on reanalysis datasets.**

Forecast based on previous step $\hat{x}_t = F(x_{t-1})$





Data Assimilation



Observation: $\boldsymbol{y}_t = h(\boldsymbol{x}_t^*)$

Assimilated state: $\boldsymbol{x}_t \sim p(\boldsymbol{x}_t | \hat{\boldsymbol{x}}_t, \boldsymbol{y}_t)$

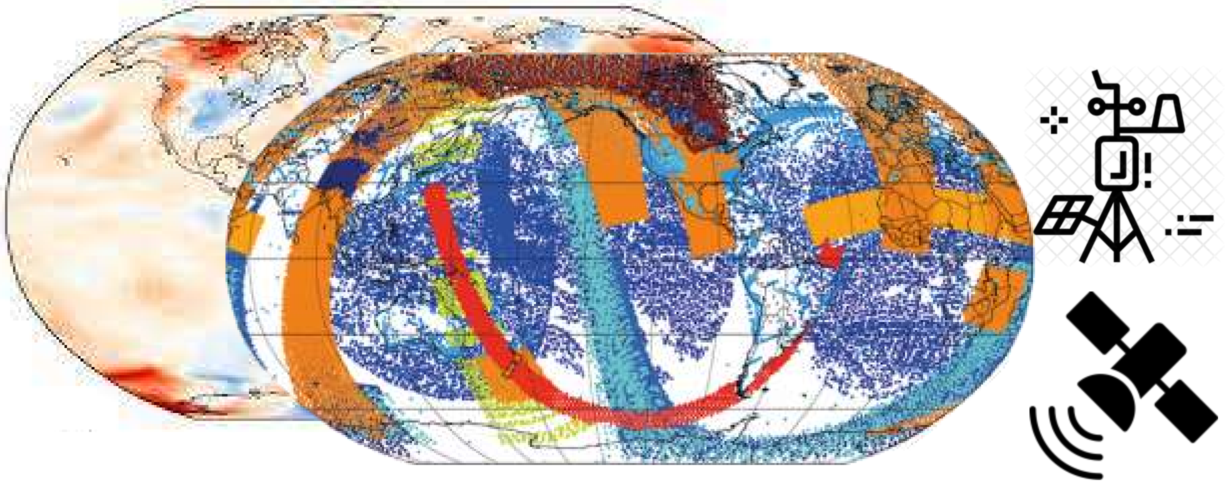
$h(\boldsymbol{x})$: Observation operator
$F(\boldsymbol{x})$: Forecast operator

Ground truth state at time step t

https://www.ecmwf.int/en/about/media-centre/news/2019/forecasting-system-upgrade-set-improve-global-weather-forecasts

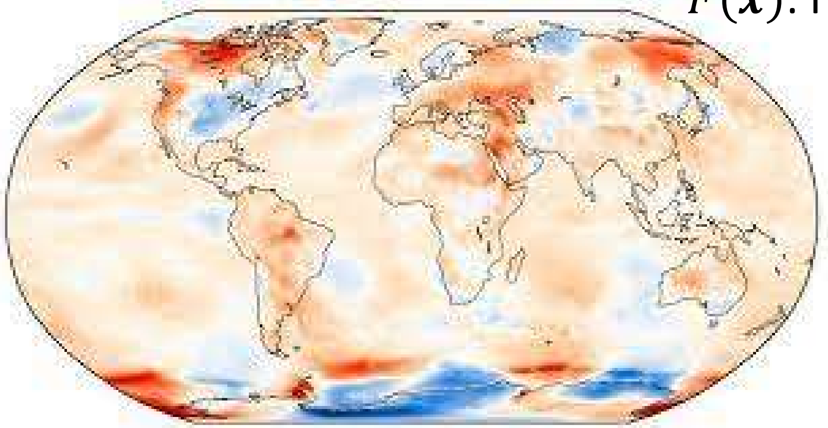
Motivation

Predicted state $\widehat{\boldsymbol{x}}_t = F(\boldsymbol{x}_{t-1})$ Observation: $\boldsymbol{y}_t = h(\boldsymbol{x}_t^*)$



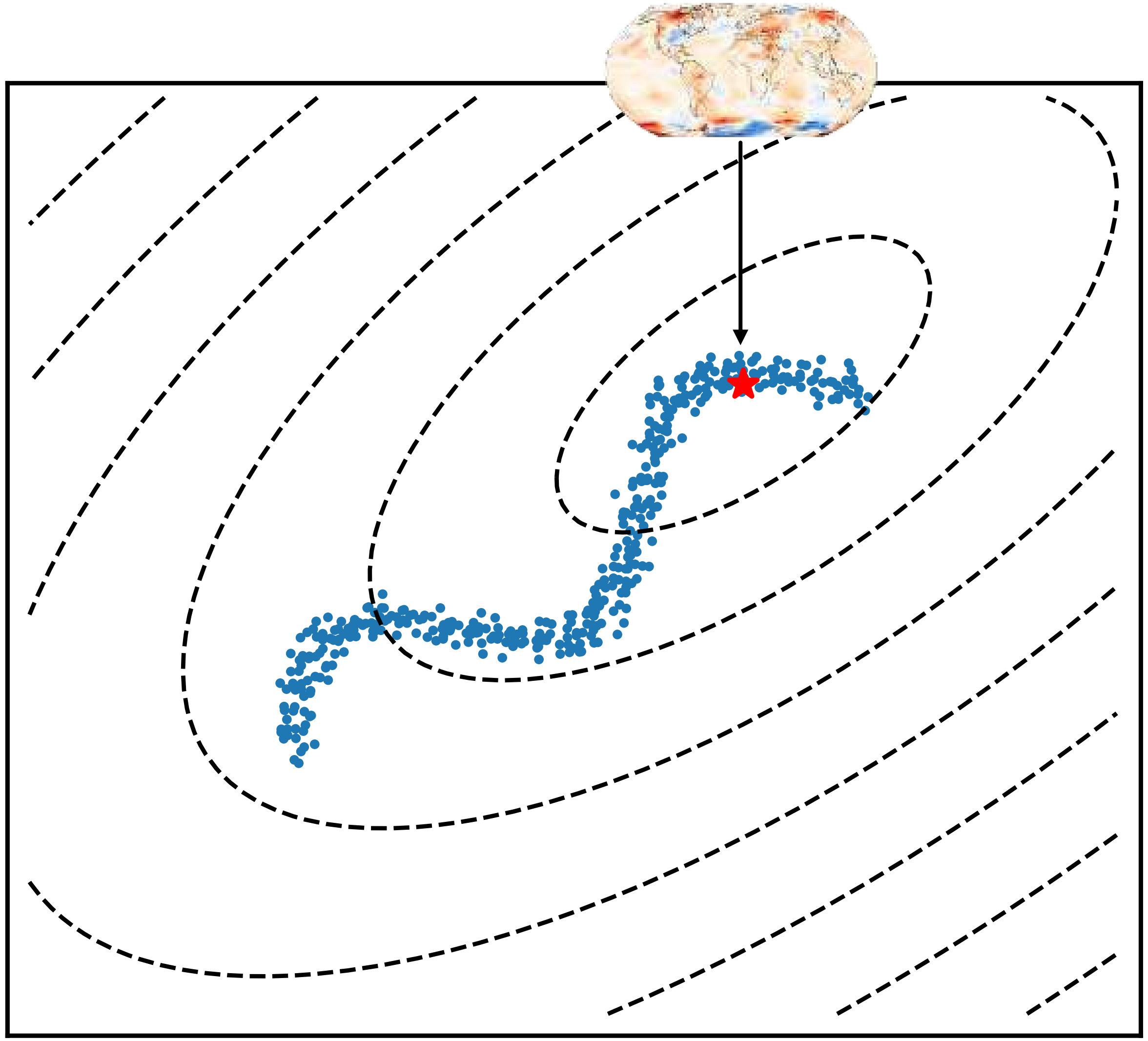
Data Assimilation

$h(\boldsymbol{x})$: Observation operator
$F(\boldsymbol{x})$: Forecast operator



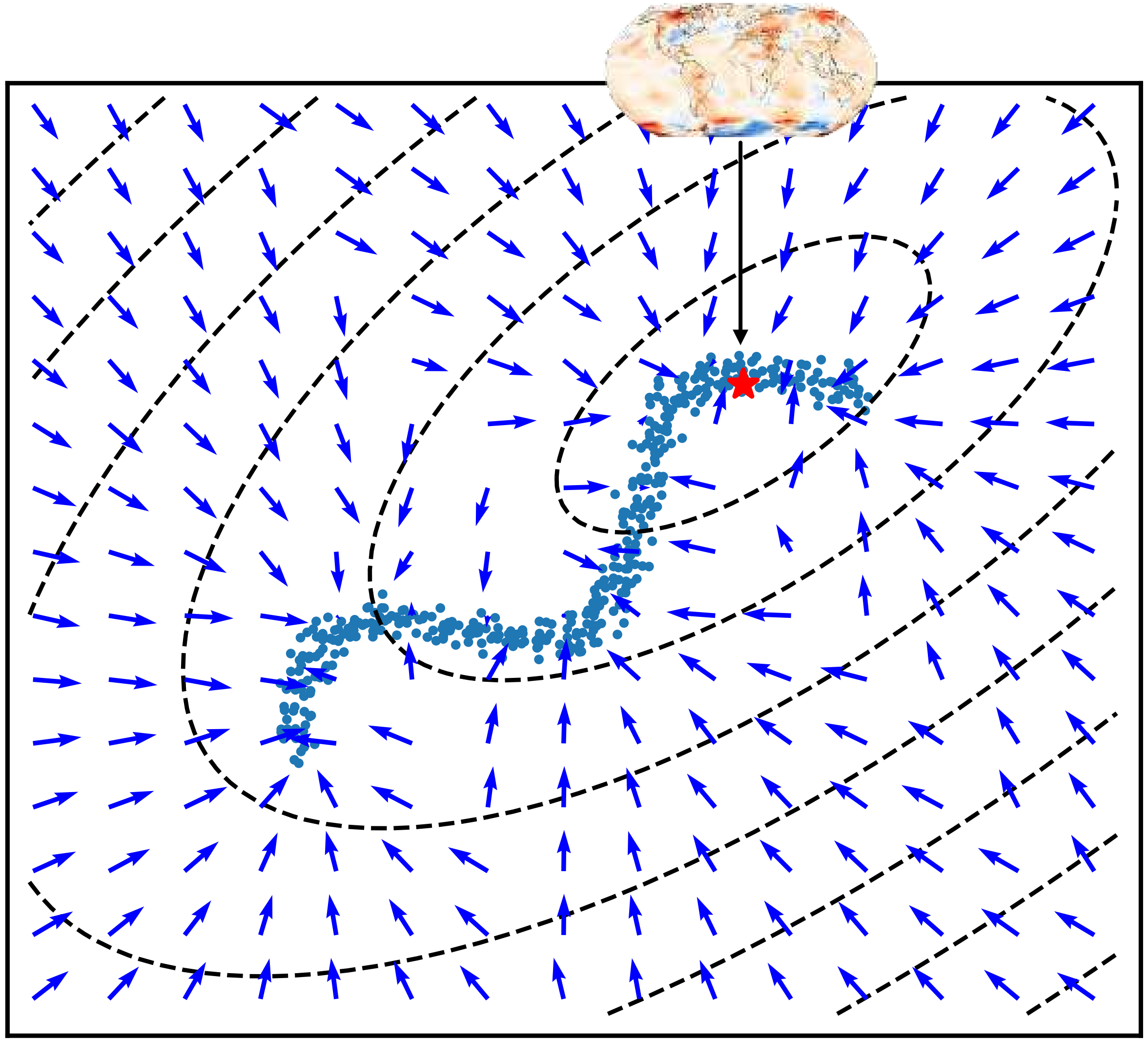
Assimilated state: $\boldsymbol{x}_t \sim p(\boldsymbol{x}_t | \widehat{\boldsymbol{x}}_t, \boldsymbol{y}_t)$

General Idea



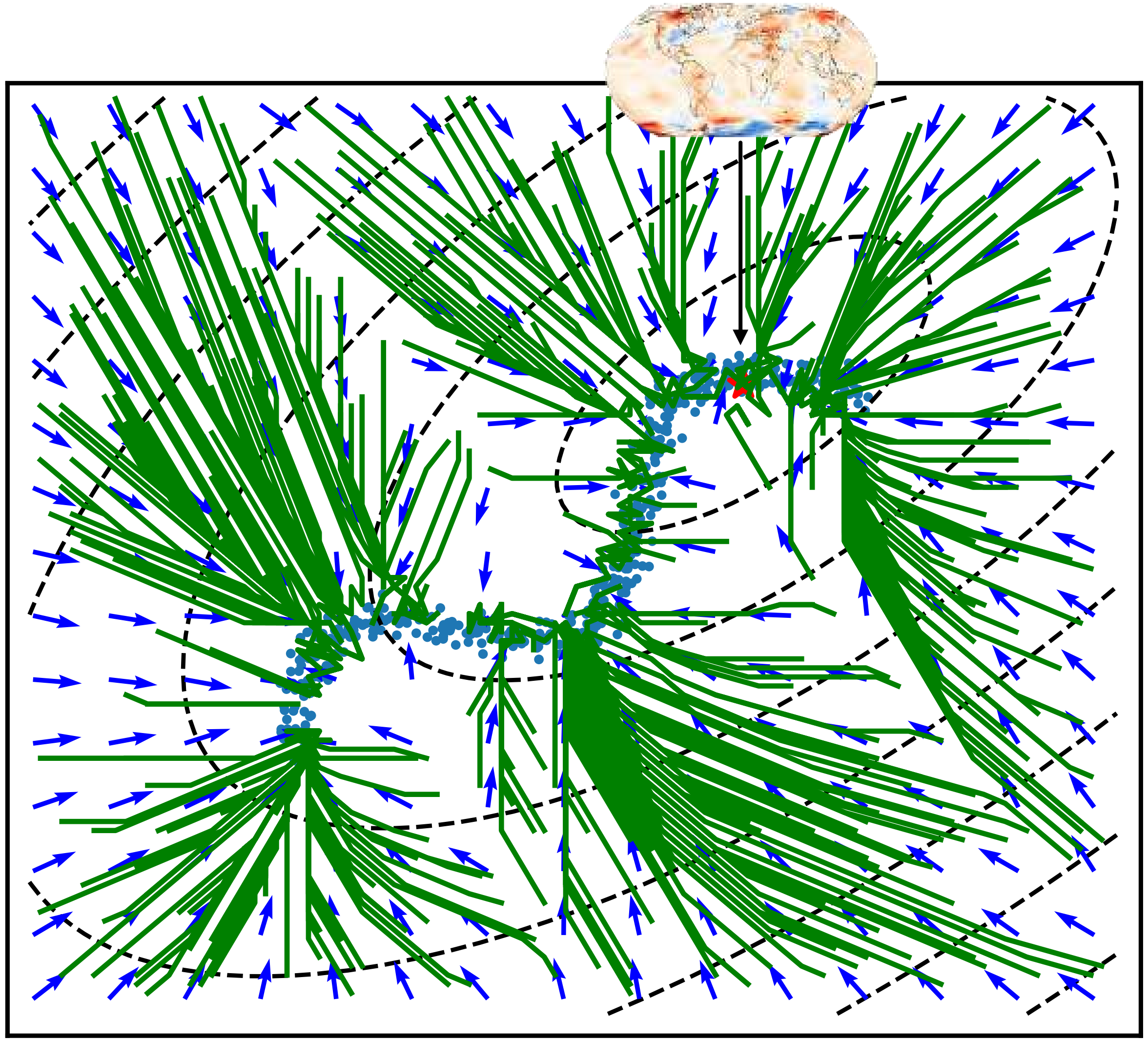
- Valid atmosphere states forms a (high-dimensional) manifold
- 3DVar performs maximum likelihood estimation of posterior distribution through minimizing a quadrate loss function
  - Assume Gaussian Process
  - Need to design covariance matrix
  - Need to “invert” covariance matrix when minimizing
  - Perform gradient descent / Newton’s method to numerically find minima

General Idea



- Generalize gradient of loss function into a **learned** vector field
- Denoising diffusion model defines the vector field via the reverse of adding gaussian white noise

General Idea



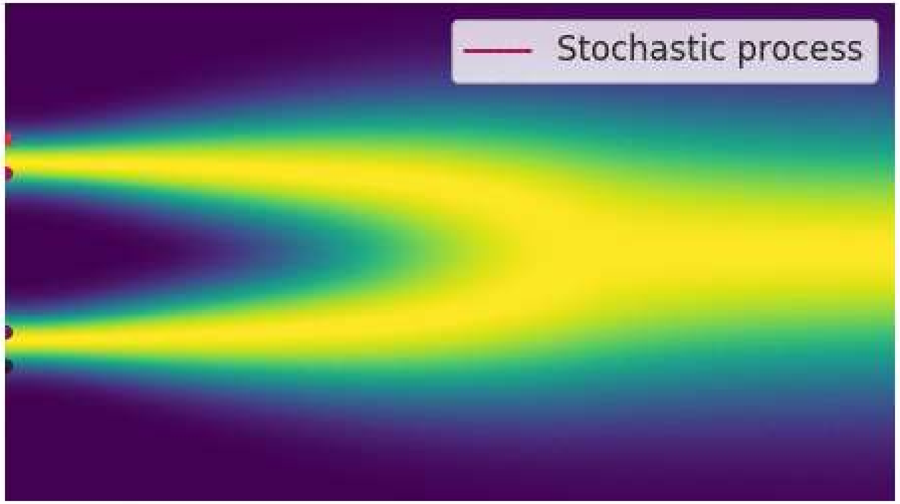
- Generalize gradient of loss function into a **learned** vector field
- Denoising diffusion model defines the vector field via the reverse of adding gaussian white noise
- A diffusion model generates (unconditional) samples of possible atmosphere state from randomly generated states

General Idea

- **Sampling from probability distribution with denoising diffusion model**

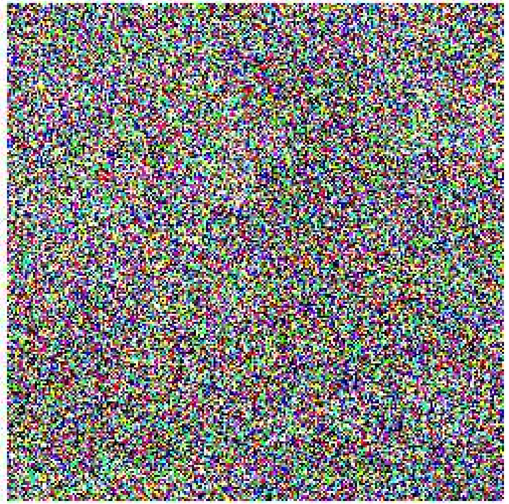
Forward Process:

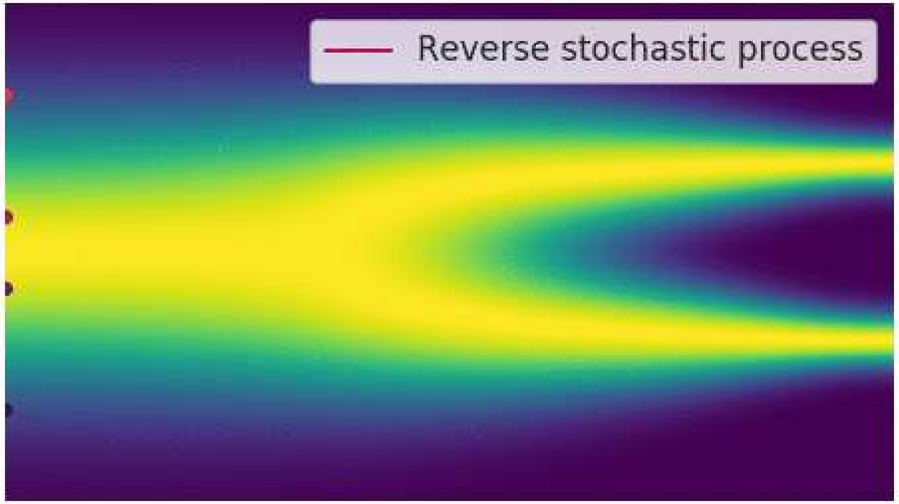
complex distribution ->
simple realizable distribution



Backward Process:

simple realizable distribution
-> complex distribution

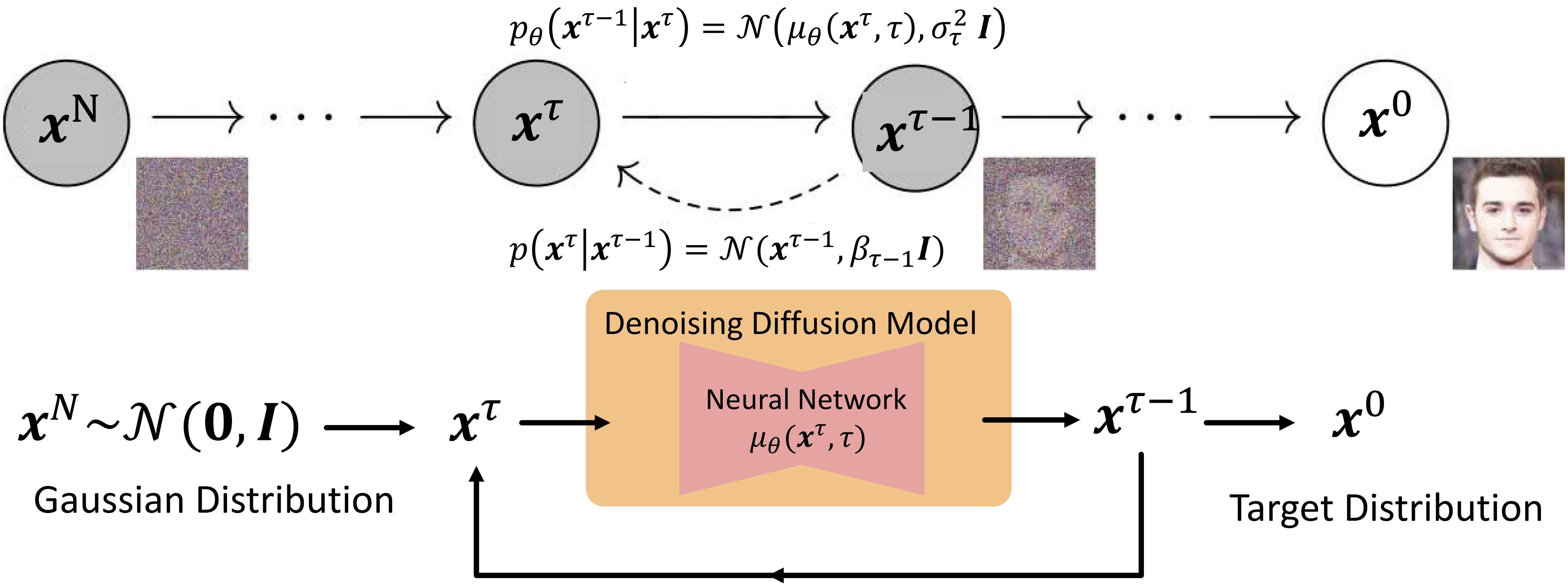




https://yang-song.net/blog/2021/score/

General Idea

- **Sampling from probability distribution with denoising diffusion model**



Ho, J., Jain, A. and Abbeel, P., 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, *33*, pp.6840-6851.

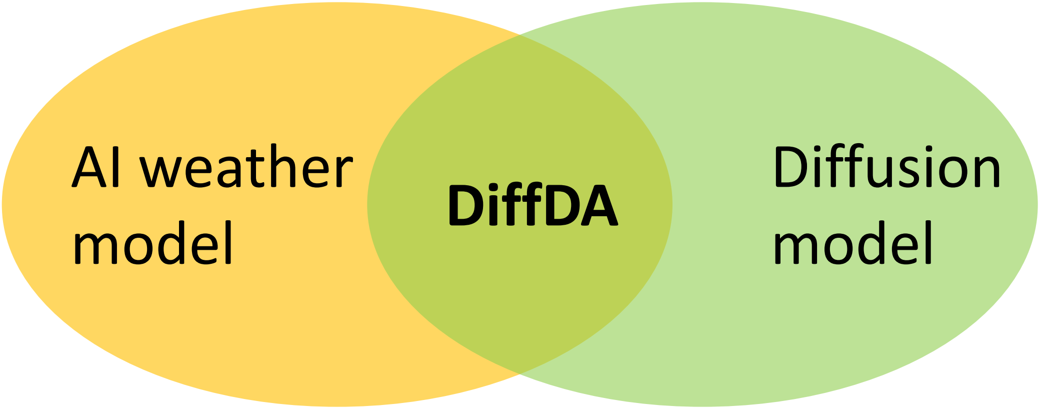
Challenge 1: How to diffuse on high dimensional fields?

- **Normal input shape for diffusion model: (3, 512, 512)**
- **Shape of atmosphere state: (6x13+5, 721, 1440)**
  - High spatial resolution
  - Dimension size is not power of 2

=> Need special treatment!

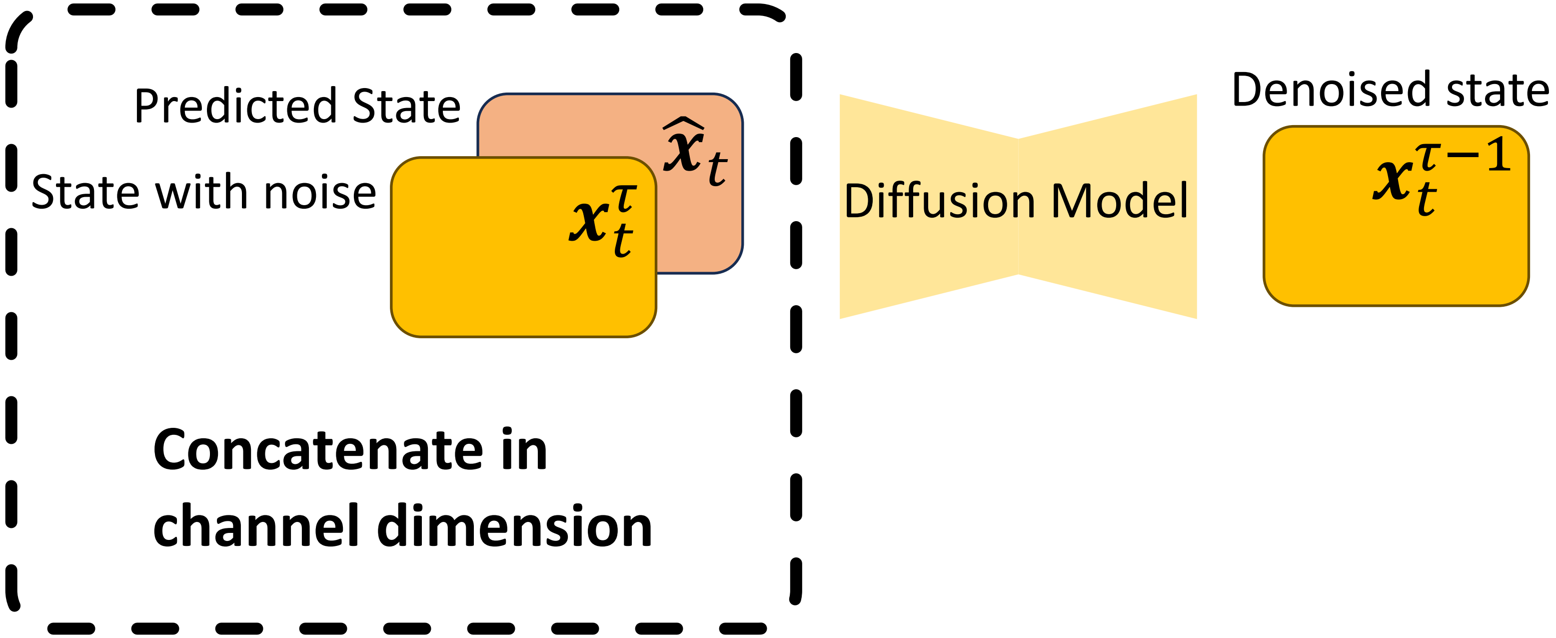
Options:

- Develop a new dedicated network structure
- **Use the structure from AI weather model having similar input/output shape!**

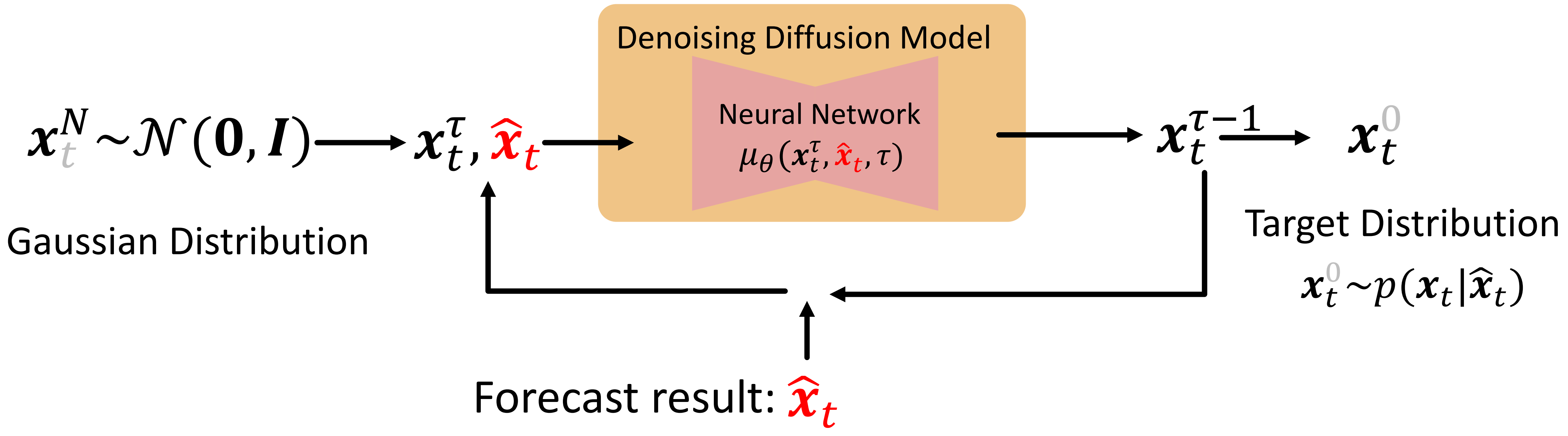


Challenge 2: How to add conditioning?

- **Conditioning for predicted state $\hat{\boldsymbol{x}}_t$**
  - $\hat{\boldsymbol{x}}_t$ has the same shape as the assimilated state $\boldsymbol{x}_t^\tau$
  - Replace $\mu_\theta(\boldsymbol{x}_t^\tau, \tau)$ (unconditional) with $\mu_\theta(\boldsymbol{x}_t^\tau, \hat{\boldsymbol{x}}_t, \tau)$ (conditional)

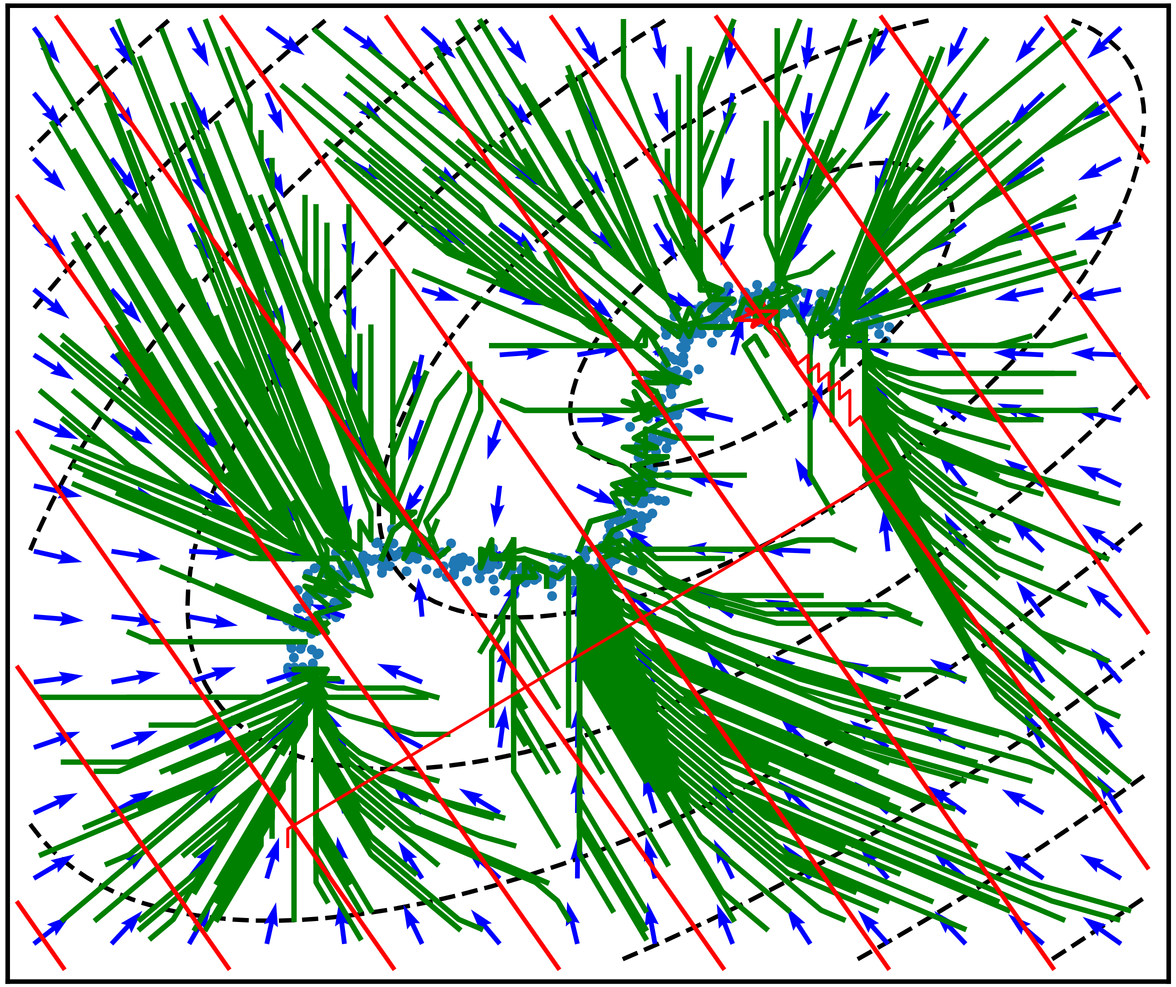


Overall Process



Challenge 2: How to add conditioning?

- **Conditioning for observations y_t: An inpainting approach**



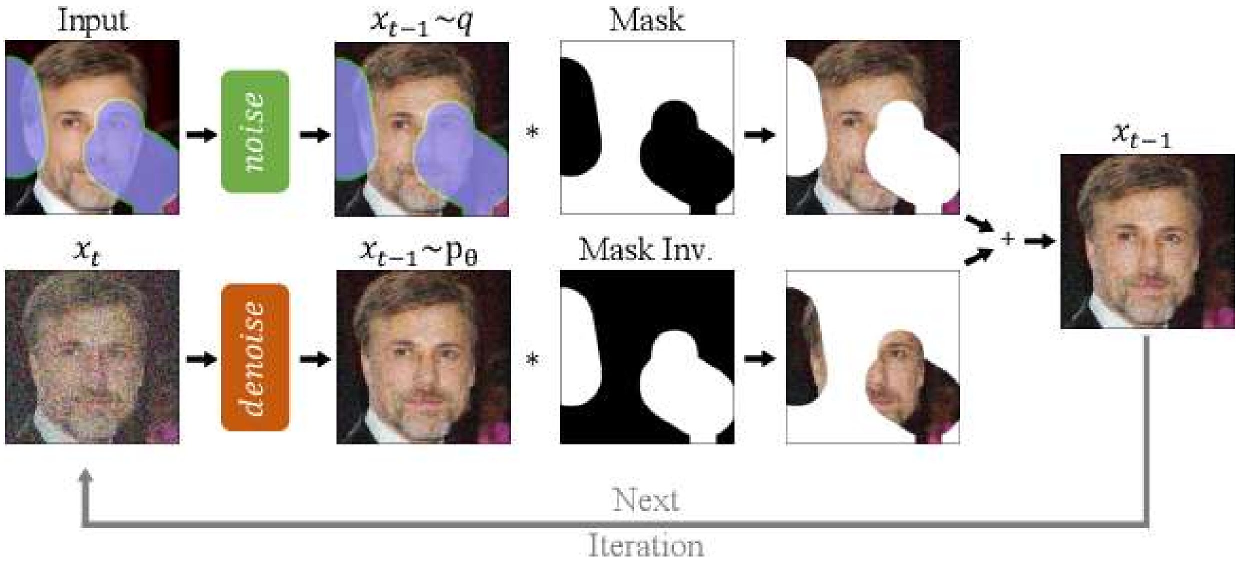
- Add additional pentalty to guide generation
  - Simpler than 3Dvar loss function
  - $||\boldsymbol{y} - h(\boldsymbol{x})||^2$
- Operator splitting
- One step solution to penalty term

Challenge 2: How to add conditioning?

- **Conditioning for observations $\boldsymbol{y}_t$: An inpainting approach**
  - Assuming $\boldsymbol{y}_t$ is sparse measurement of $\boldsymbol{x}_t^*$: $\boldsymbol{y}_t = \boldsymbol{H}\boldsymbol{x}_t^*$, where 0,1 matrix $\boldsymbol{H}$ has only one nonzero value in each row
  - proof: similar to "classifier" guidance $\nabla_{\boldsymbol{x}_t^\tau} \log p(\boldsymbol{x}_t^\tau | \widehat{\boldsymbol{x}}_t, \boldsymbol{y}_t) = \nabla_{\boldsymbol{x}_t^\tau} \log p(\boldsymbol{x}_t^\tau | \widehat{\boldsymbol{x}}_t) + \nabla_{\boldsymbol{x}_t^\tau} \log p(\boldsymbol{y}_t | \boldsymbol{x}_t^\tau)$

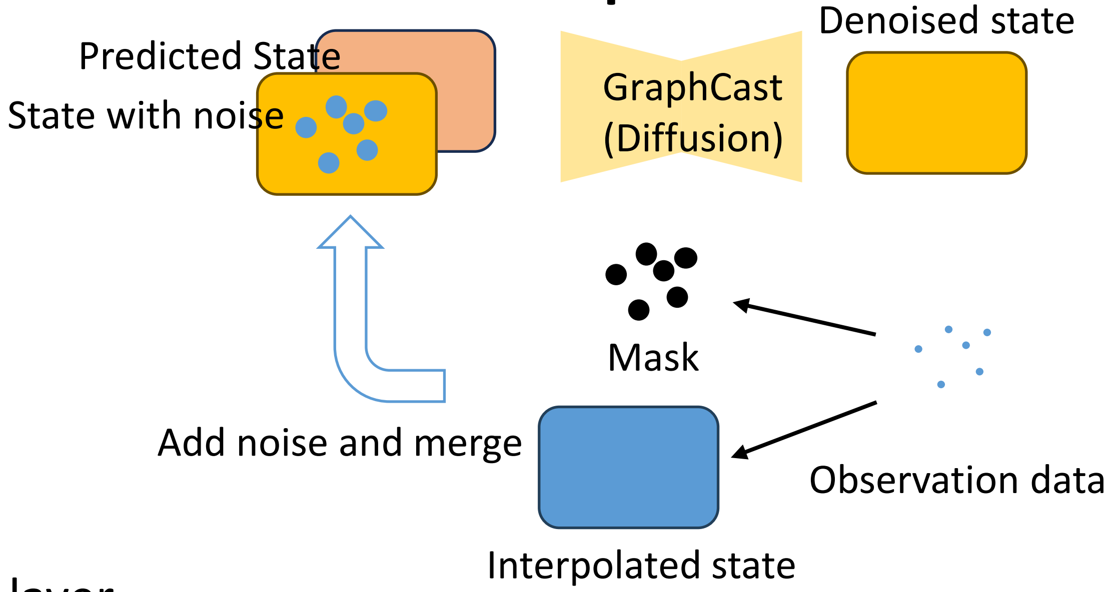
$$p(\boldsymbol{y}_t | \boldsymbol{x}_t^\tau) \approx \mathcal{N}(\boldsymbol{y}_t | \boldsymbol{H}\mathbb{E}[\boldsymbol{x}_t^0 | \widehat{\boldsymbol{x}}_t], \Sigma_y) \Rightarrow \nabla_{\boldsymbol{x}_t^\tau} \log p(\boldsymbol{y}_t | \boldsymbol{x}_t^\tau) \approx \nabla_{\boldsymbol{x}_t^\tau} \left\| \boldsymbol{y}_t - \boldsymbol{H}\mathbb{E}[\boldsymbol{x}_t^0 | \widehat{\boldsymbol{x}}_t] \right\|_{\Sigma_y}^2$$

Inpainting Pipeline



Problem: sparse signal often suppressed in the downsampling layer

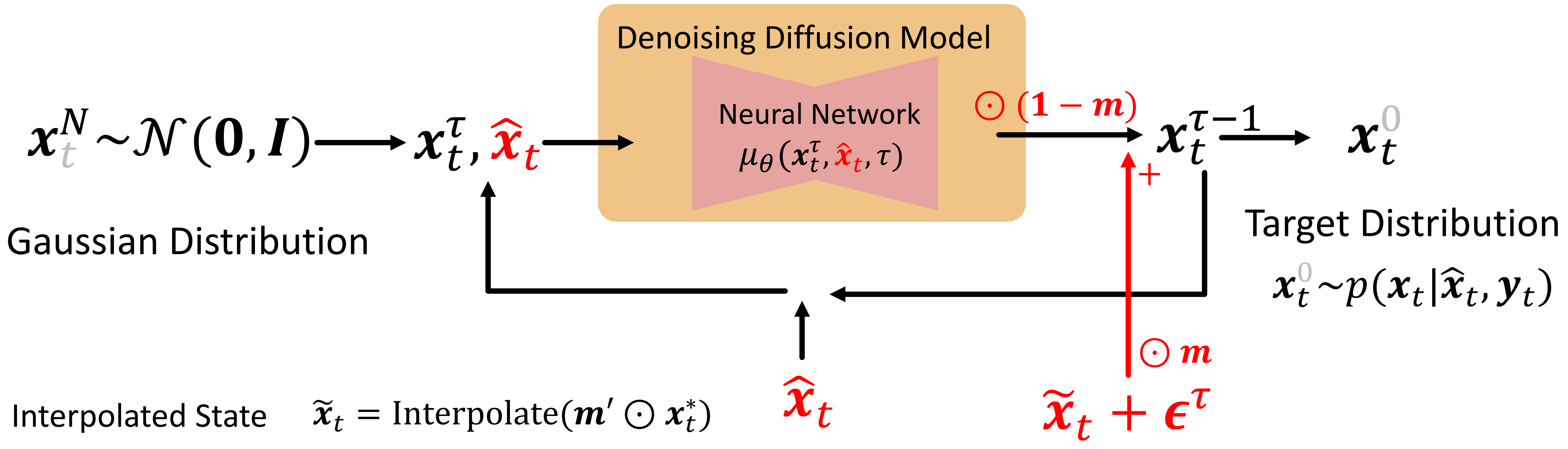
Treatment of Sparse Mask



Solution: enlarge the mask with interpolated data (assuming data is smooth)

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R. and Van Gool, L., 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11461-11471).

Overall Process

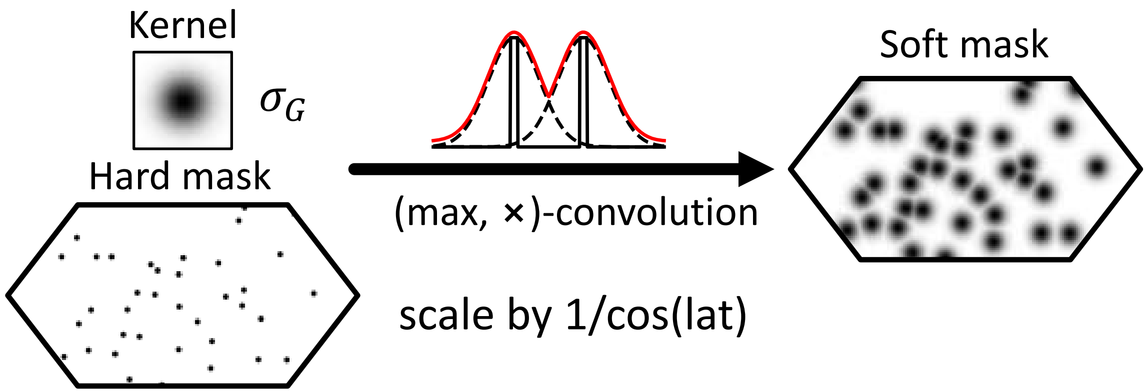


Interpolated State $\tilde{x}_t = \mathrm{Interpolate}(\boldsymbol{m}' \odot x_t^*)$

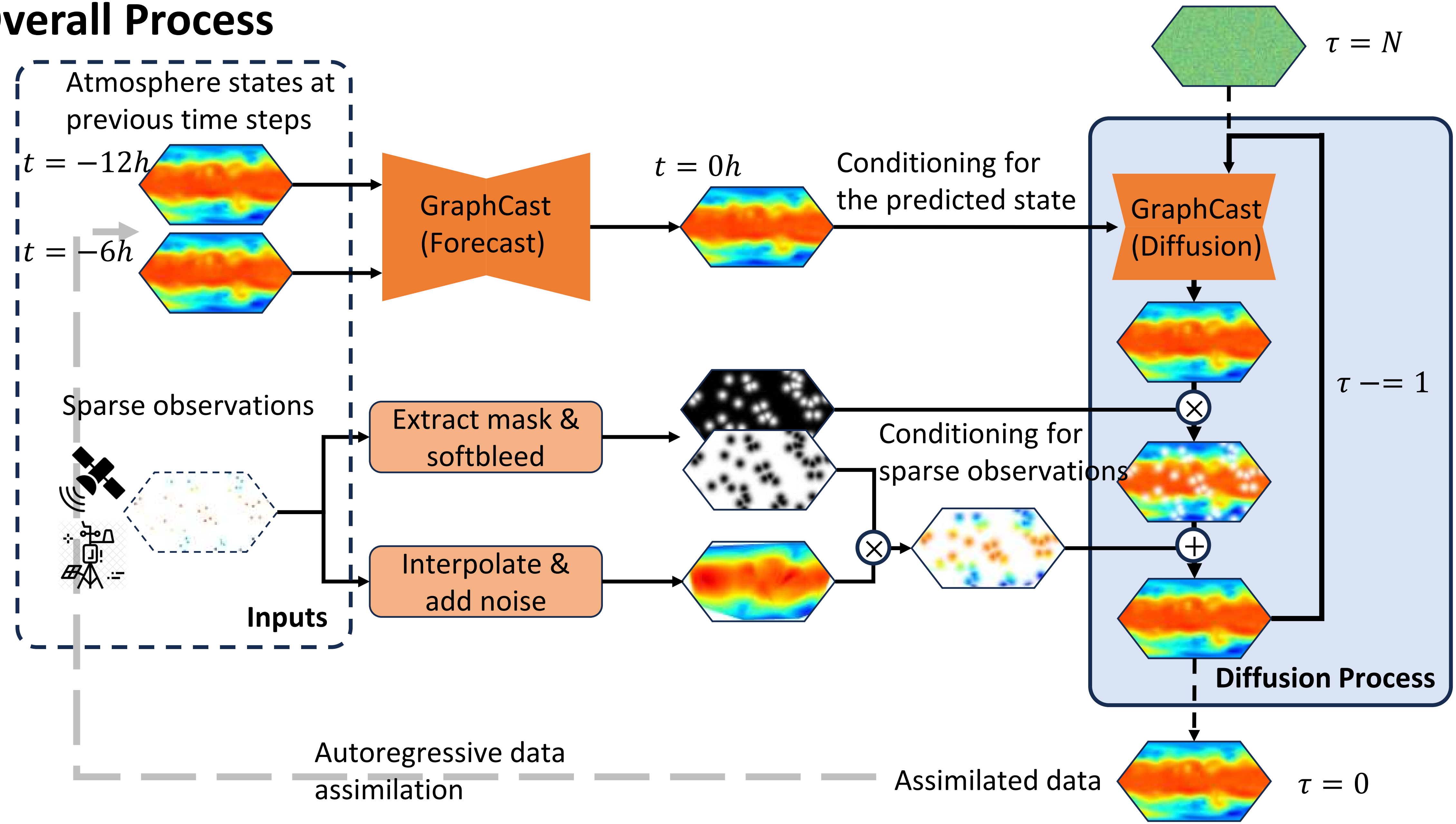
Soft Mask $\boldsymbol{m} = \mathrm{Softbleed}(\boldsymbol{m}', \sigma_G)$

Hard Mask $\boldsymbol{m}_i' = \sum_j \boldsymbol{H}_{ji}, \boldsymbol{y} = \boldsymbol{H}x_t^*$

Noise $\epsilon^\tau = \left(1 - \prod_{s=1}^{j-1} \beta_s\right)\epsilon', \epsilon' \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$



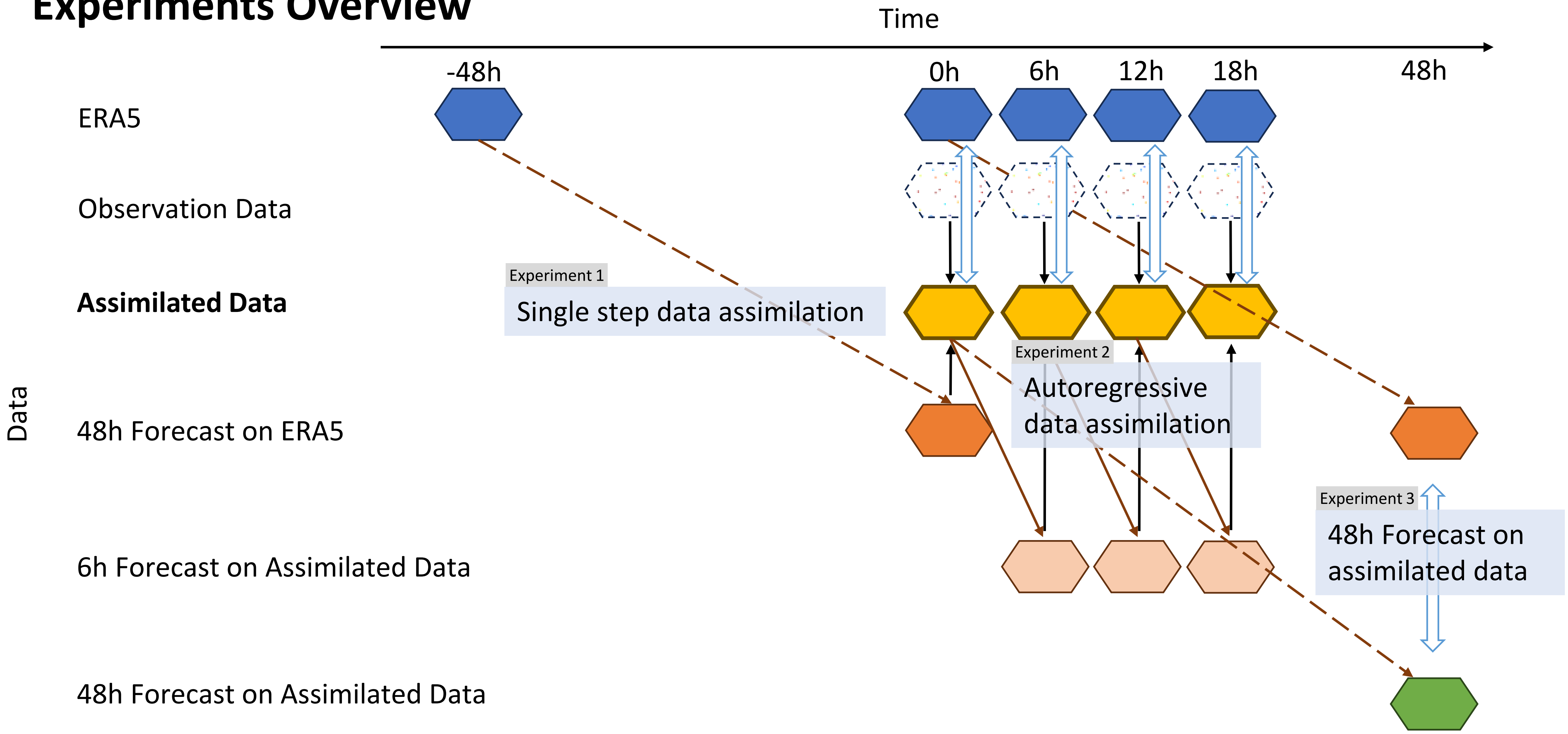
Overall Process



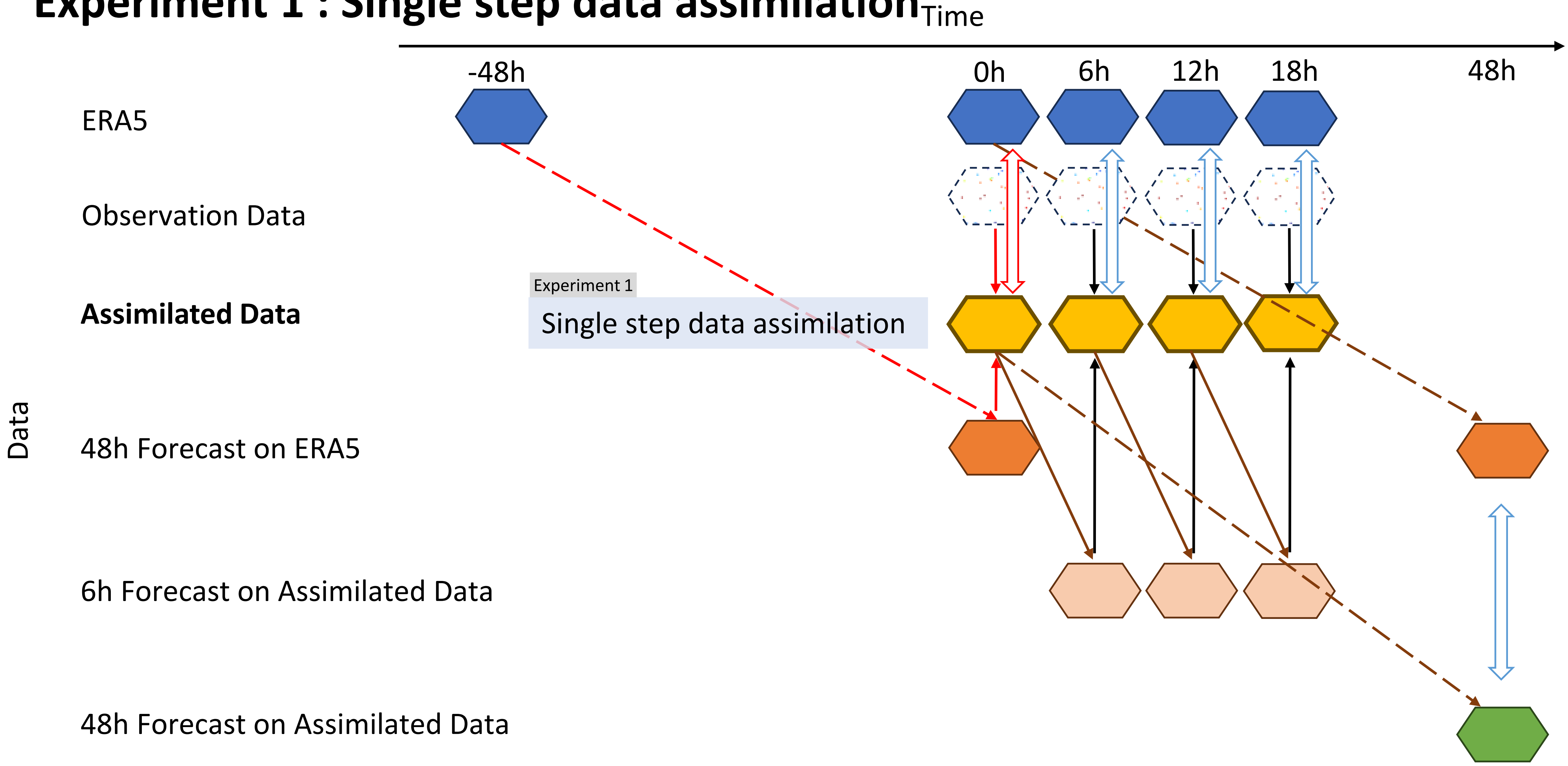
Experiment Settings

- Backbone model: GraphCast operational (0.25deg 721x1440, 13 levels)
- Training data: ERA5 1979 - 2016 6hour resolution
- Emulate observations from ERA5: randomly sample horizontal coordinates + take all vertical levels & variables
- Batch size: 48 (global), 1 (local)
- Num epochs: 20
- Optimizer: Adam, LR scheduler: warmup_cos_decay
- LR: 1e-5 (0%) -> 1e-4 (12%) -> 3e-6 (100%)
- $\sigma_G = 1.5$
- Compute resources: 48 A100 80G, 4 GPUs per node, 2 days

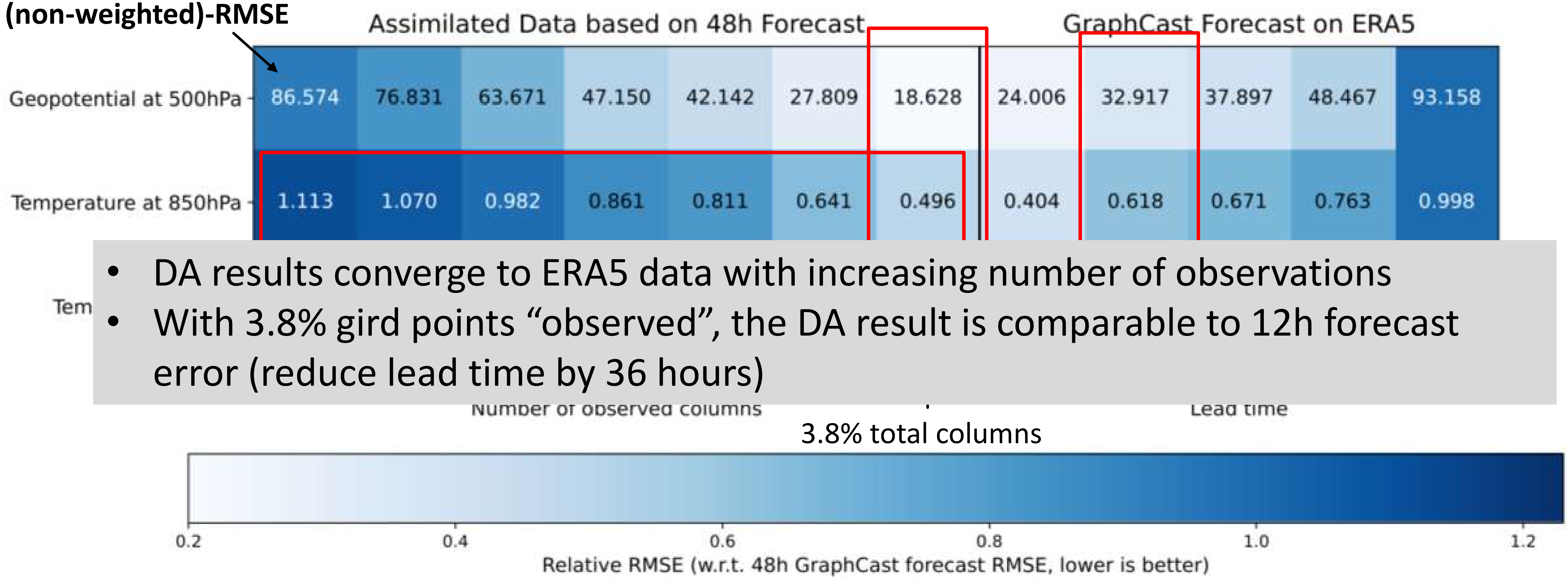
Experiments Overview



Experiment 1 : Single step data assimilation



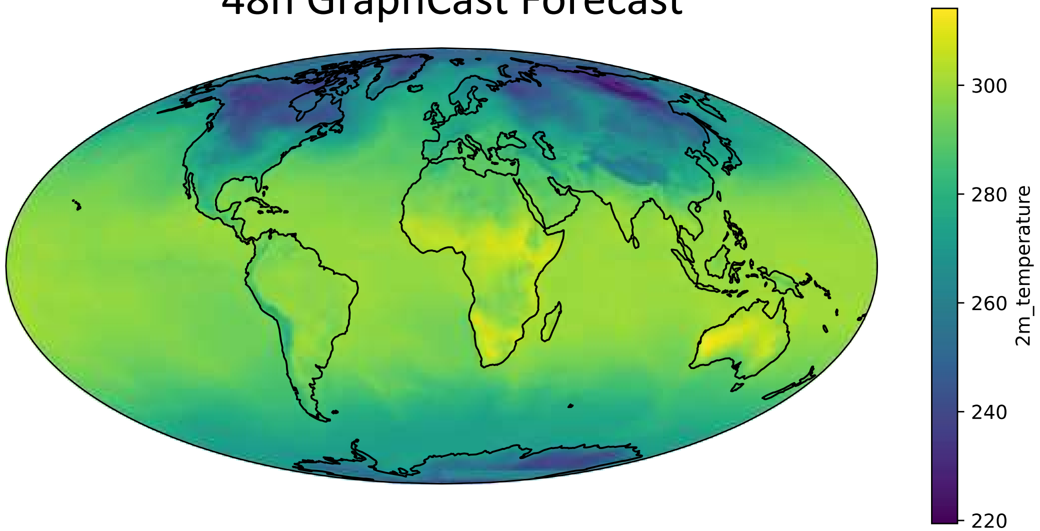
Experiment 1: Result



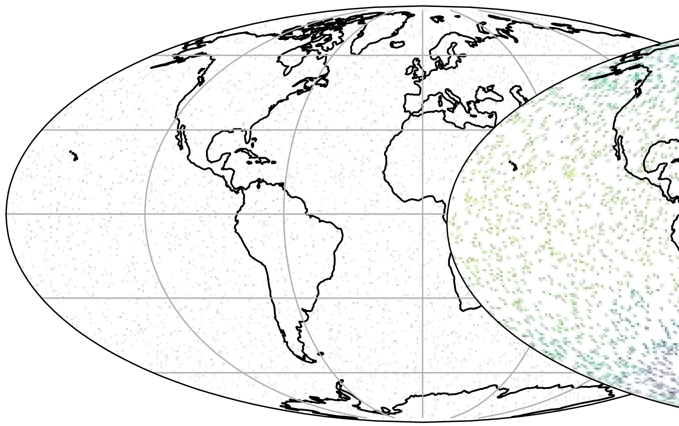
- DA results converge to ERA5 data with increasing number of observations
- With 3.8% gird points “observed”, the DA result is comparable to 12h forecast error (reduce lead time by 36 hours)

Case study: Assimilated 2m temperature at 2022-01-03 06z

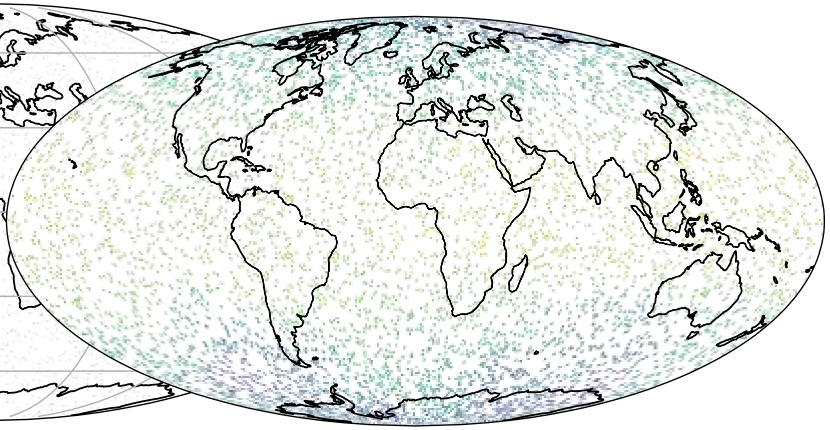
48h GraphCast Forecast



Raw Observations

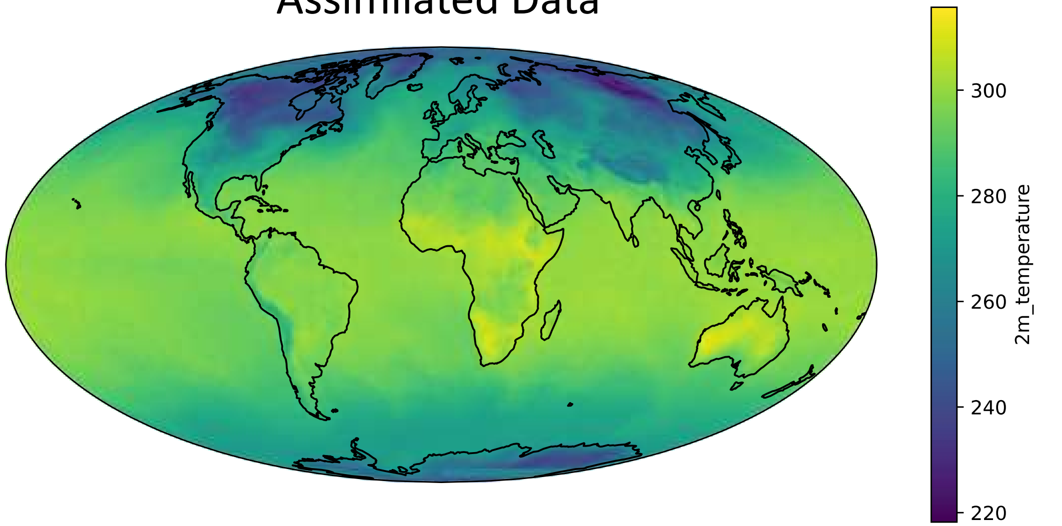


Interpolated Observations with softmask



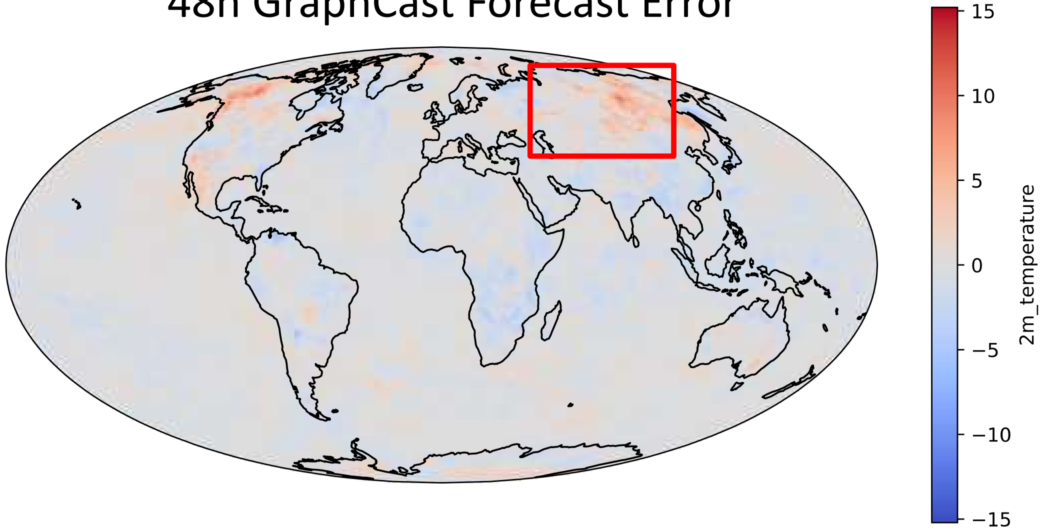
DiffDA

Assimilated Data

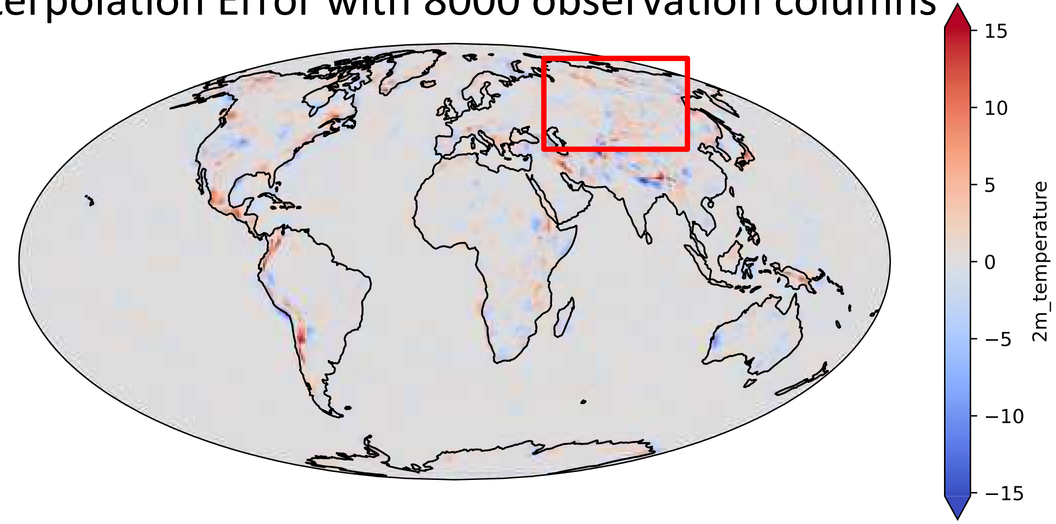


Case study: Assimilated 2m temperature at 2022-01-03 06z

48h GraphCast Forecast Error

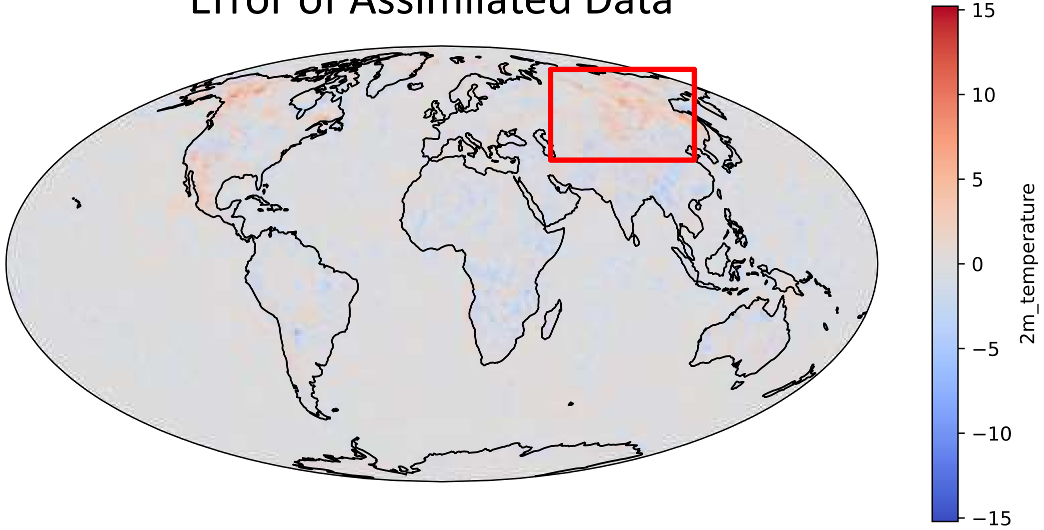


Interpolation Error with 8000 observation columns



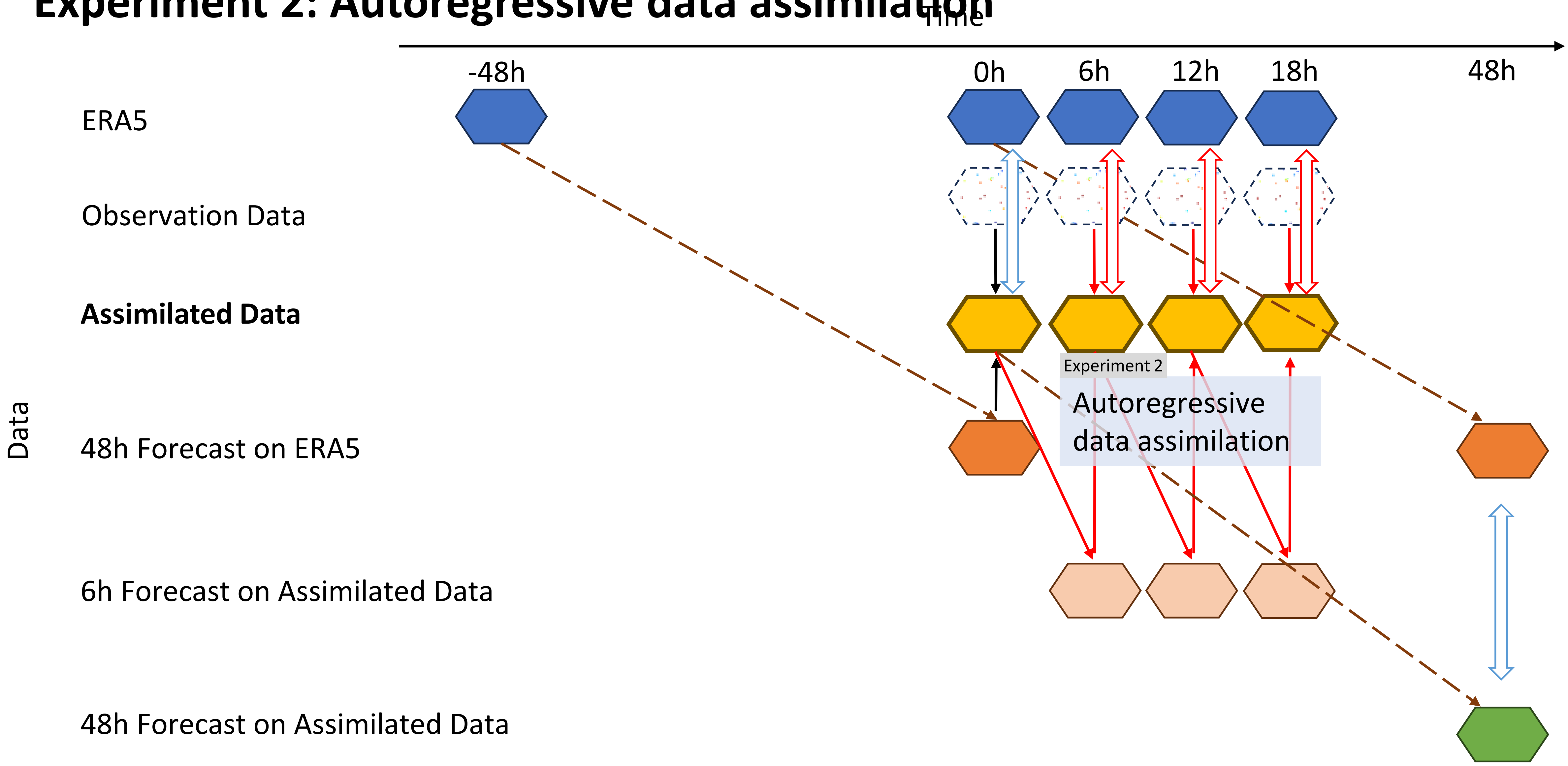
DiffDA

Error of Assimilated Data



Assimilated data is better than both inputs
(48h forecast and interpolated observations)

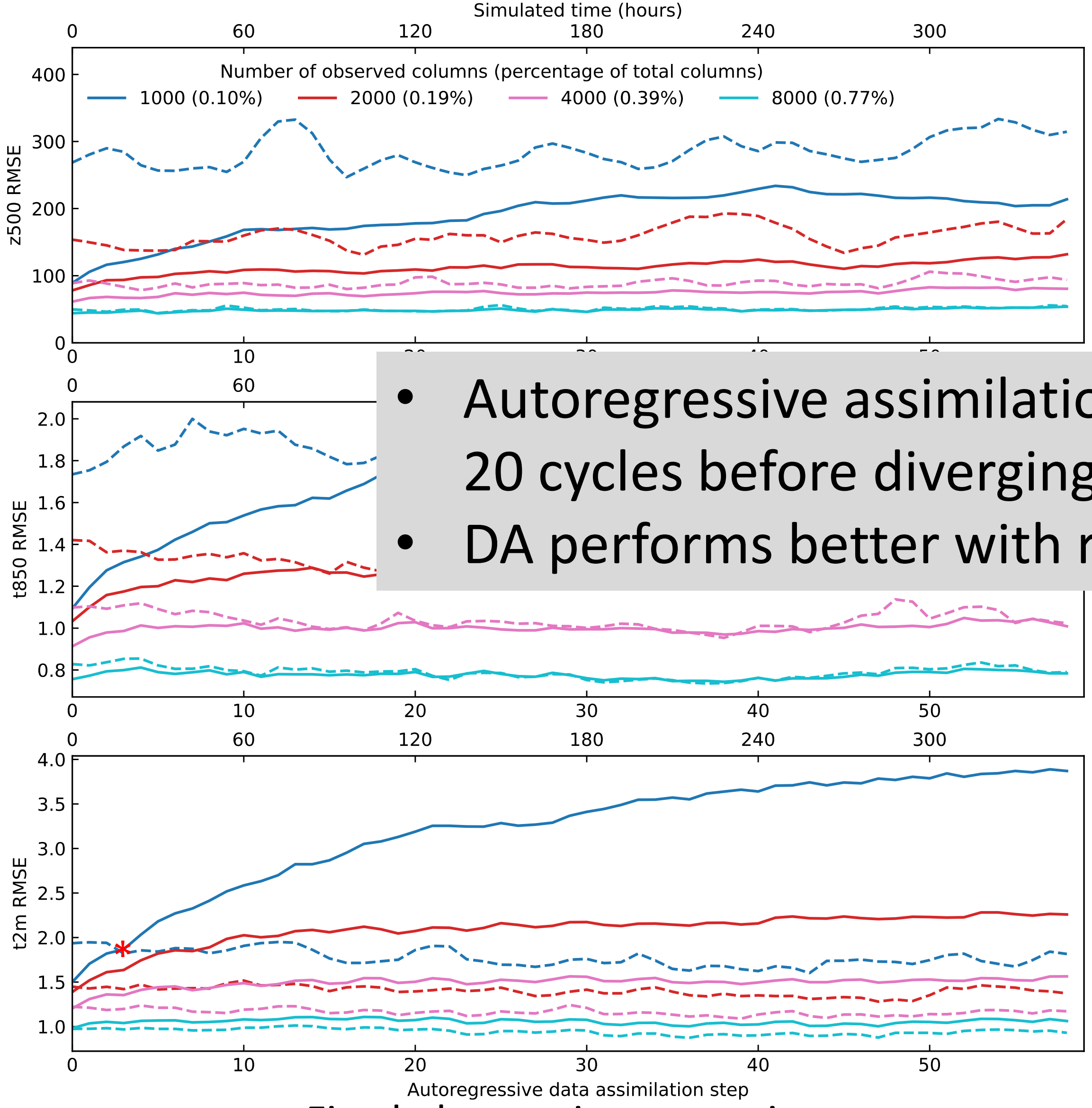
Experiment 2: Autoregressive data assimilation



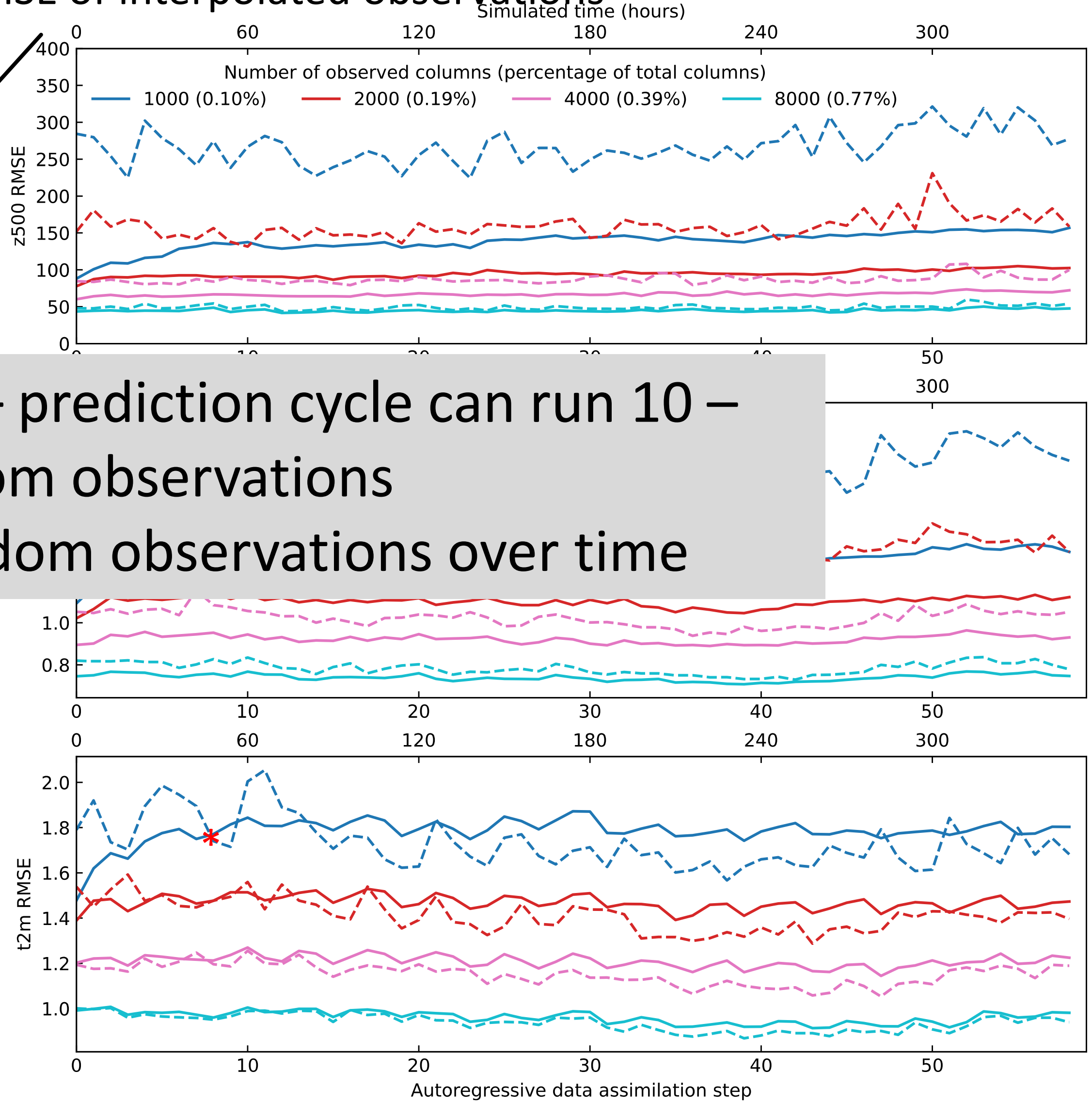
Experiment 2: Result

RMSE of Autoregressive assimilated data

RMSE of interpolated observations



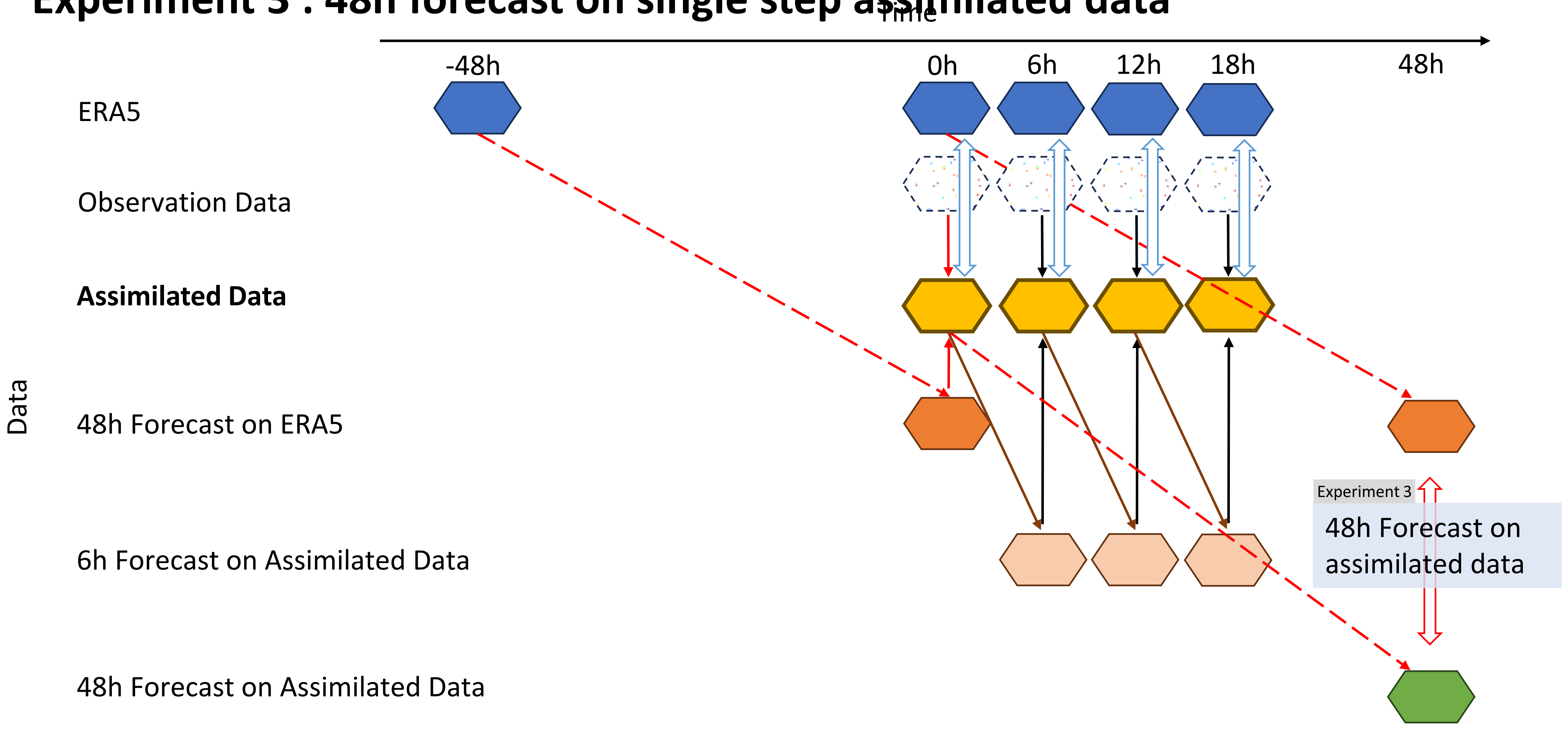
Fixed observations over time



Random observations over time

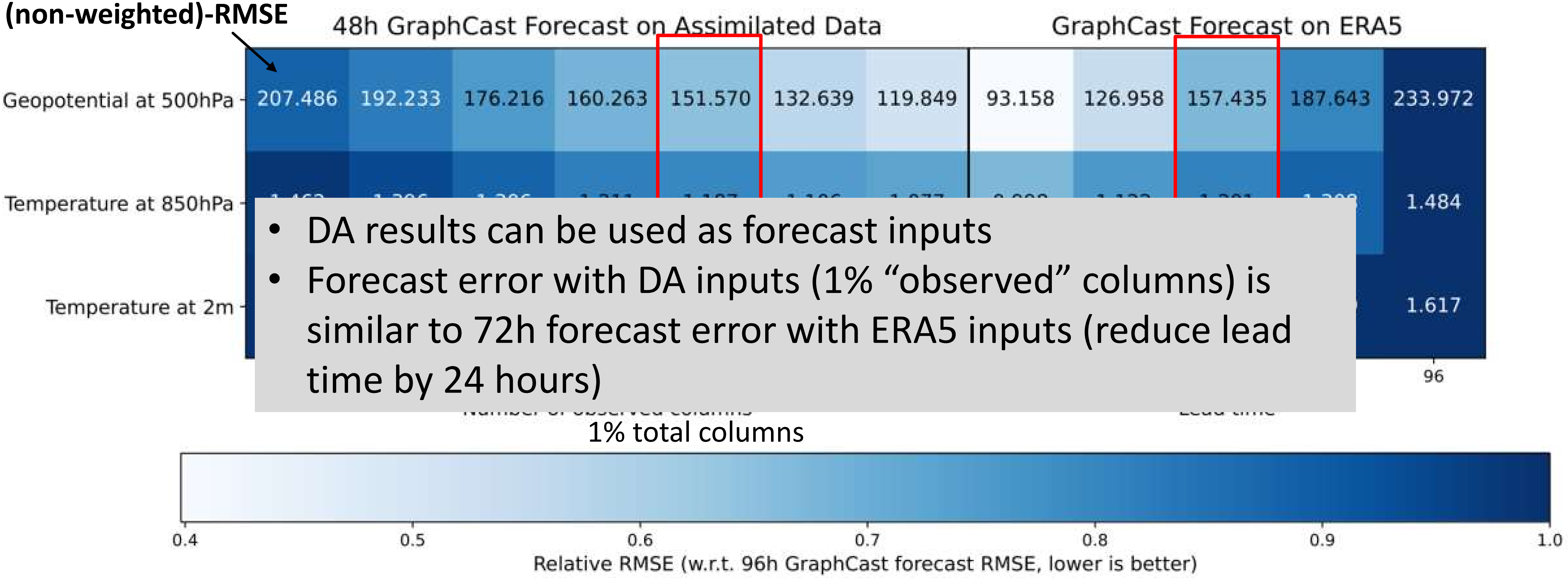
- Autoregressive assimilation – prediction cycle can run 10 – 20 cycles before diverging from observations
- DA performs better with random observations over time

Experiment 3 : 48h forecast on single step assimilated data



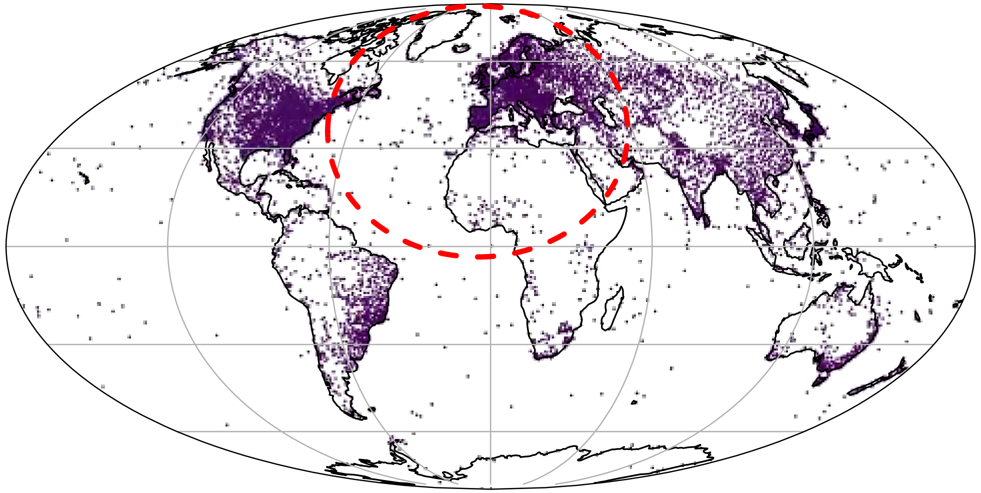
Experiment 3: Result

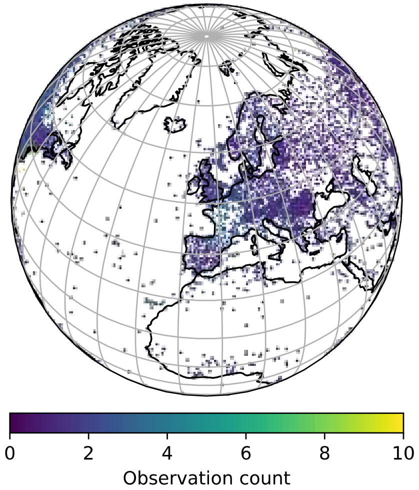
(non-weighted)-RMSE



- DA results can be used as forecast inputs
- Forecast error with DA inputs (1% “observed” columns) is similar to 72h forecast error with ERA5 inputs (reduce lead time by 24 hours)

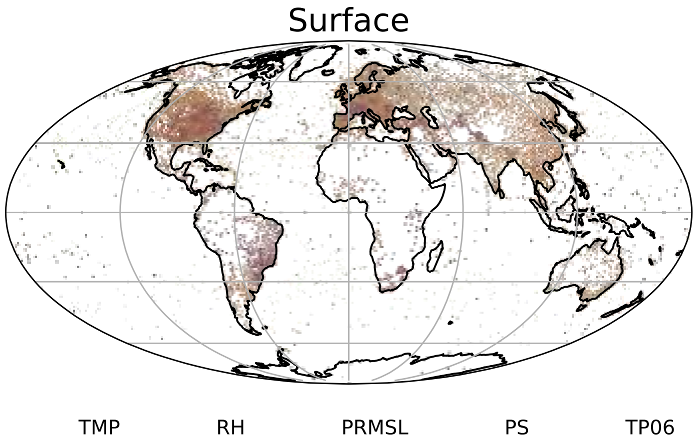
Next Step: Towards Assimilating Real-World Observations

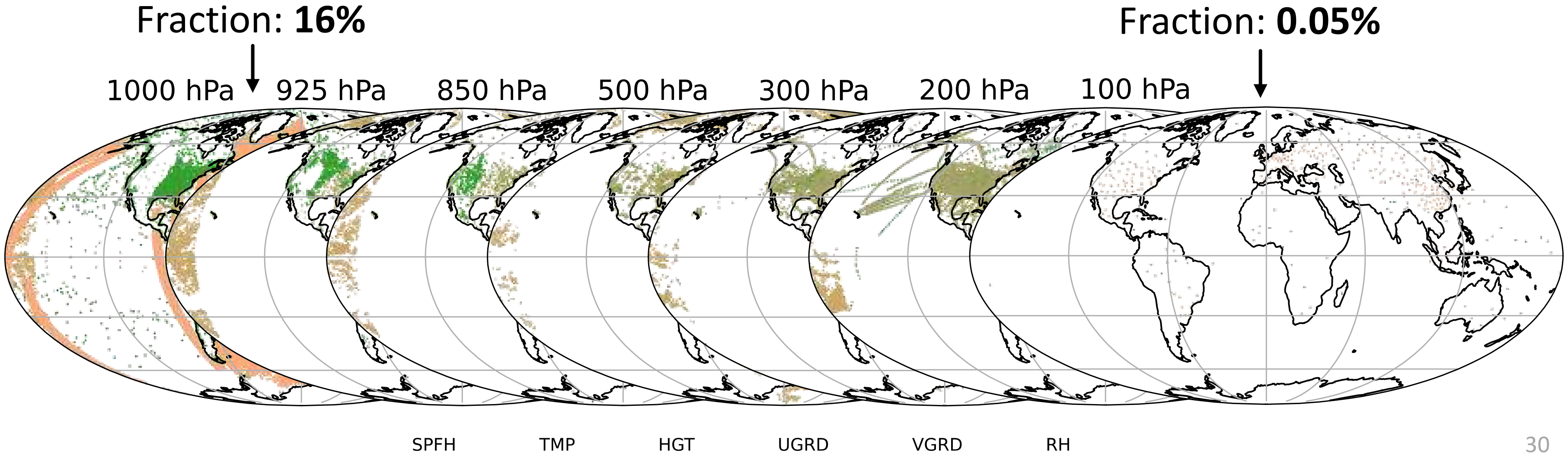




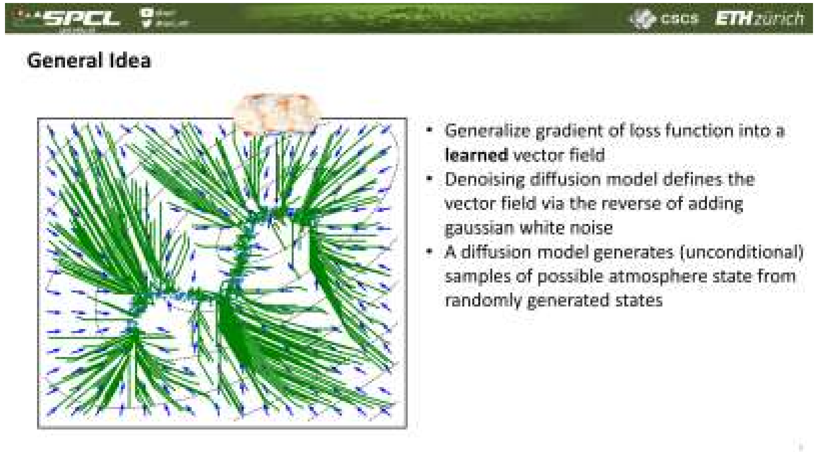
Challenges:
- Non-uniform distribution
- Only a subset of variables are measured
- Less observations at higher levels
- Observations are collected in a time window, e.g. : (-3h, 3h)
- Need quality control

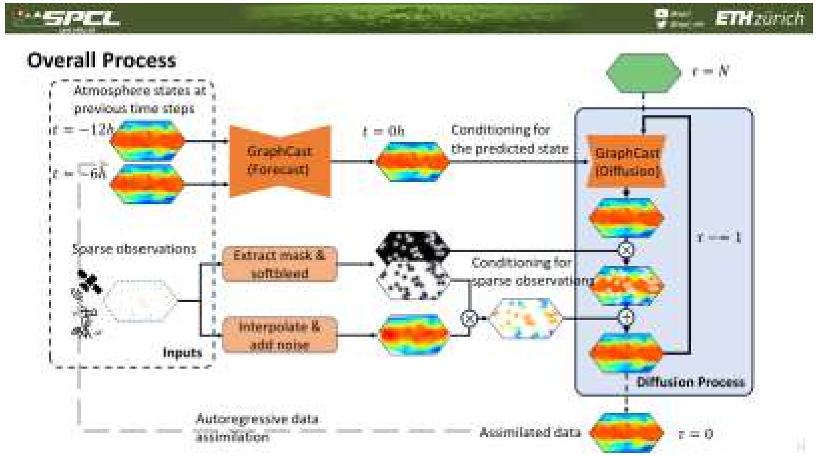
Observations of 2m temperature at 30.12.2022 00z from GDAS
Measurements/Total Grid Points: 10054/1036800
Fraction: **0.97 %**

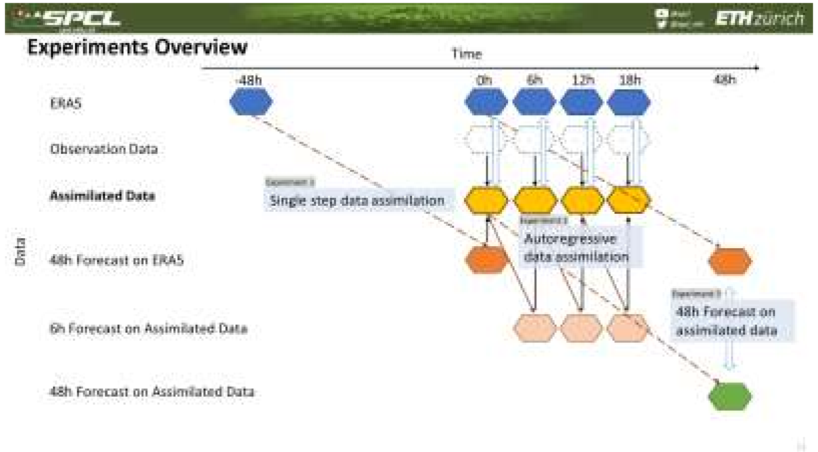


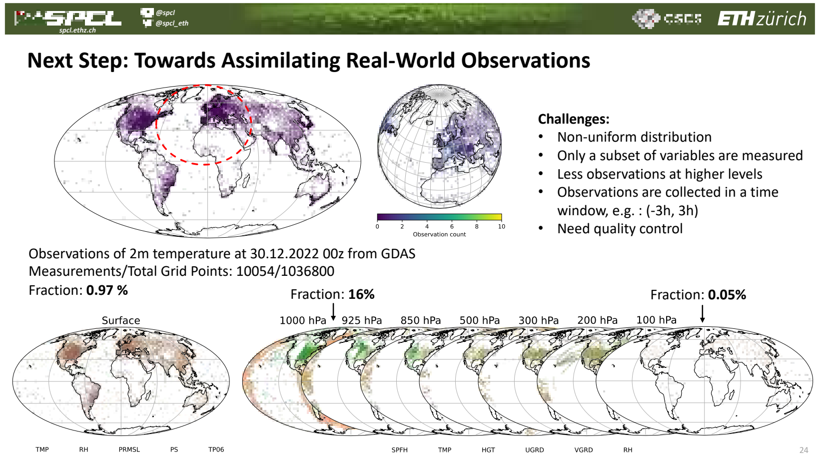


Conclusions









More of SPCL's research:

- youtube.com/@spcl — 150+ Talks
- twitter.com/spcl_eth — 1.2K+ Followers
- github.com/spcl — 2K+ Stars

… or spcl.ethz.ch

Next:

- assimilate real-world observations
- 4D Assimilation
- Incorporate errors in observations
- Incorporate non sparse observations