GenCast: Diffusion-based ensemble forecasting for medium-range weather

Leonhard Döring

Motivation

Why would anyone predict the weather using machine learning?

Importance of Weather Forecasts

Put simply:

- Know whether or not to take an umbrella

But also:

- Prepare more accurately for extreme events (cyclones, floods, droughts)
- Forecast energy demand and renewable energy generation

Ultimately:

- Make informed decisions

Weather Forecast Basics

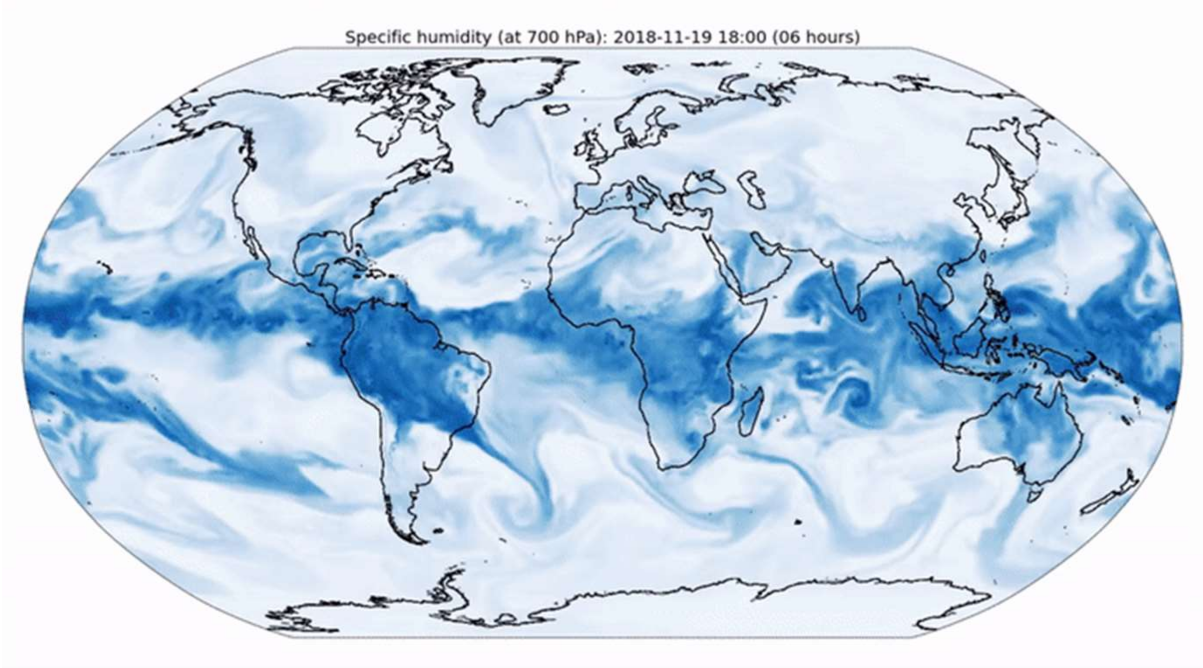
State of the Art:

- Numerical Weather Prediction (NWP) by **E**uropean **C**entre for **M**edium-Range **W**eather **F**orecasts (ECMWF)
  - Solving differential equations that describe physics of the atmosphere
- Accurate
- Computationally expensive
- Slow

can we change that?

ML Weather Prediction: Previous Works

- **GraphCast**
  - Graph neural network
  - by Google DeepMind
  - Tries to predict the actual weather (deterministic forecast)
  - Trained with RMSE → results are blurry
- **Pangu-Weather**
  - Deep neural network
  - by Huawei Cloud
- **...**



https://deepmind.google/discover/blog/graphcast-ai-model-for-faster-and-more-accurate-global-weather-forecasting/

Main Challenge for Weather Forecasting

- Atmosphere is a chaotic system

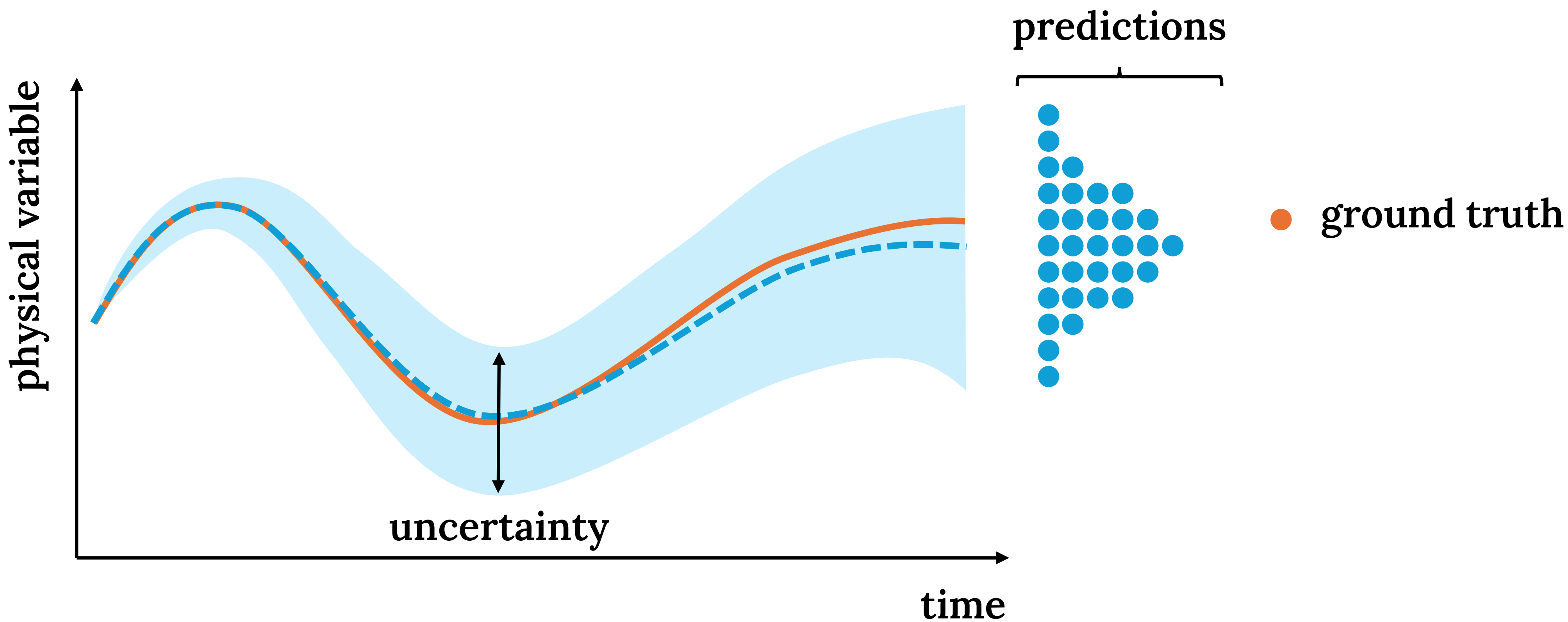
> **Chaotic System:** small perturbations in initial conditions lead to massive changes in outcome (**Butterfly Effect**).

- Impossible to perfectly observe initial conditions
- We want to know, how sure we are with our predictions

→ **Uncertainty measure needed**

Ensembles

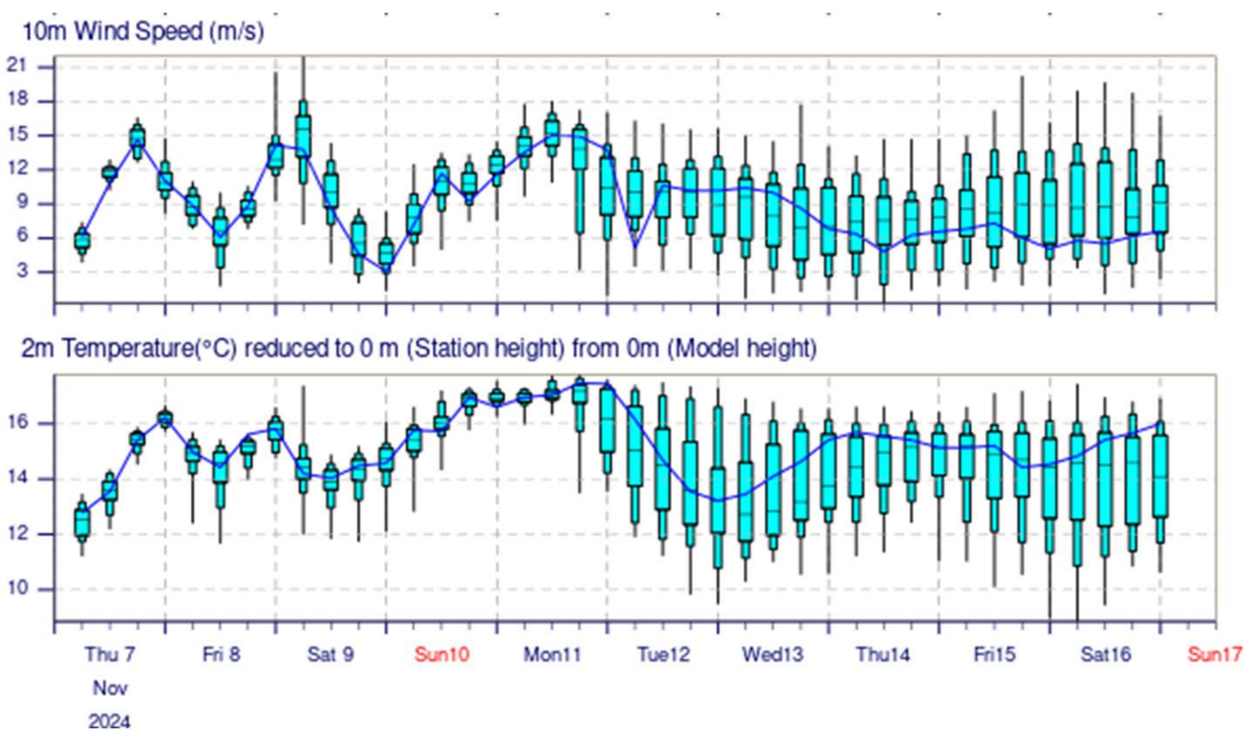
Basic Idea: Generate multiple predictions to approximate the actual distribution

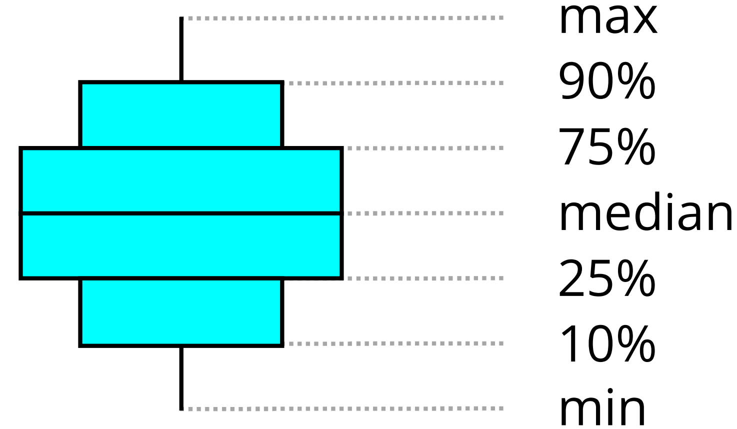


Ensembles

- ECMWF's ensemble forecast is called **ENS**
  - Consists of one "best guess" based on best available input data and 50 additional predictions based on perturbed inputs and model assumptions

"Gold Standard" of medium-ranged weather forecasts





https://charts.ecmwf.int/products/medium-2mt-wind30?base_time=202411070000&projection=opencharts_europe&valid_time=202411101500

Comparison of Approaches

	NWP (e.g. ENS by ECMWF)	Deterministic MLWP (e.g. GraphCast)	?
accuracy	high	high	high
speed	low	high[1]	high[1]
(computational) cost	high	low[1]	low[1]
uncertainty measure	yes	no	yes

[1]apart from the training

GenCast: Overview

- **Conditional diffusion model** sampling a prediction of the weather in a **12h** timestep for the entire earth from the approximated **forecast distribution**
- Conditioned on the **two previous states** as input
- **8 min inference time** for 30 timesteps (equivalent to 15-day forecast) on a TPUv5 device

Method

Aren’t diffusion models for image and media generation?

Basic Idea

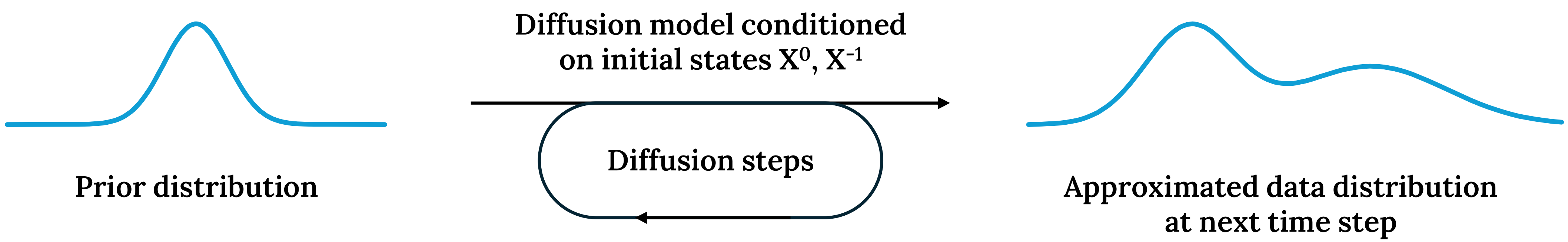
- Model joint probability distribution

approximated by neural network (GenCast)

$$p(\bar{X}^{0:T}|O^{\le 0}) = \underbrace{p(\bar{X}^0|O^{\le 0})}_{\text{State Inference}} \; \underbrace{p(\bar{X}^{1:T}|\bar{X}^0)}_{\text{Forecast Model}} = p(\bar{X}^0|O^{\le 0}) \prod_{t=1}^{T} p(\bar{X}^t|\bar{X}^{t-1})$$

$\bar{X}^t$: atmospheric state at time t
$O^{\le t}$: observations up to time t

- Use conditional diffusion model to generate samples of the approximated actual distribution of the atmospheric state for the entire globe



Dataset

$$p(\bar{X}^{0:T}|O^{\leq 0}) = \underbrace{p(\bar{X}^{0}|O^{\leq 0})}_{\text{State Inference}} \underbrace{p(\bar{X}^{1:T}|\bar{X}^{0})}_{\text{Forecast Model}}$$

- Reanalysis data (ERA5) from ECMWF including ERA5 EDA (ensemble of data assimilations) for initial conditions

Timespan	**1940 – present** (84 years)
Spatial Resolution	**0.25°** (approx. 28km)
Temporal Resolution	**1h** time steps
Pressure Levels	**37** (up to 80 km height)

divides the globe into 1440 x 720 cells

Input and Output Parameters

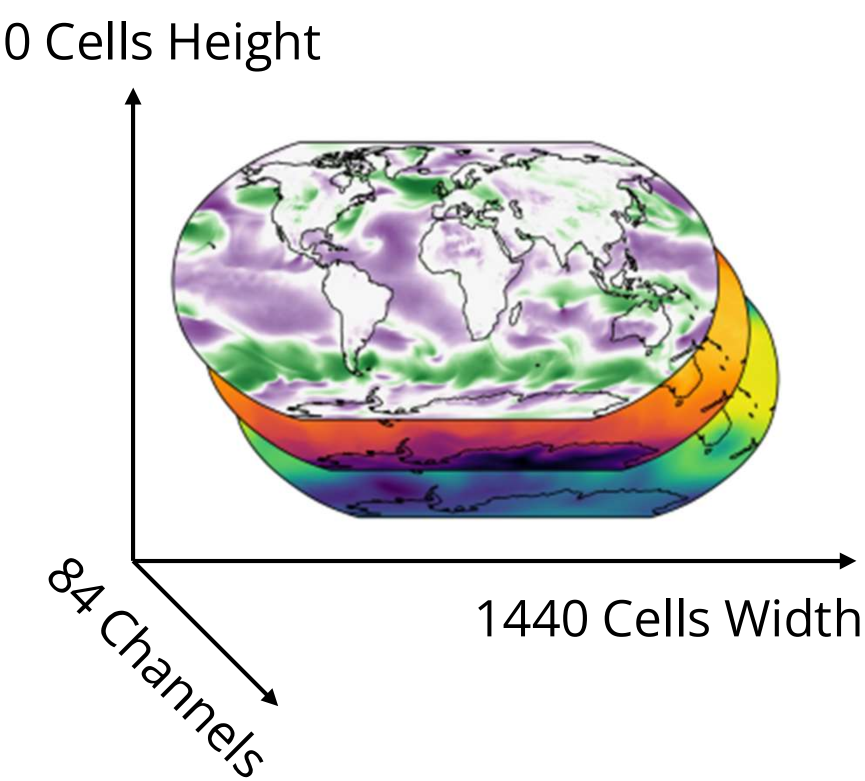
Type	Variable name	Short name	ECMWF Parameter ID	Role (accumulation period, if applicable)
Atmospheric	Geopotential	z	129	Input/Predicted
Atmospheric	Specific humidity	q	133	Input/Predicted
Atmospheric	Temperature	t	130	Input/Predicted
Atmospheric	U component of wind	u	131	Input/Predicted
Atmospheric	V component of wind	v	132	Input/Predicted
Atmospheric	Vertical velocity	w	135	Input/Predicted
Single	2 metre temperature	2t	167	Input/Predicted
Single	10 metre u wind component	10u	165	Input/Predicted
Single	10 metre v wind component	10v	166	Input/Predicted
Single	Mean sea level pressure	msl	151	Input/Predicted
Single	Sea Surface Temperature	sst	34	Input/Predicted
Single	Total precipitation	tp	228	Predicted (12h)
Static	Geopotential at surface	z	129	Input
Static	Land-sea mask	lsm	172	Input
Static	Latitude	n/a	n/a	Input
Static	Longitude	n/a	n/a	Input
Clock	Local time of day	n/a	n/a	Input
Clock	Elapsed year progress	n/a	n/a	Input

6 atmospheric variables for each of the 13 pressure levels used by GenCast

Price, I. et al. (2023). *GenCast: Diffusion-based ensemble forecasting for medium-range weather*. Table B1.

Data

- 40 years of the ERA5 dataset
- Each sample's dimensions:



Spatial dimensions representing the entire earth as a 0.25° lat-lon-grid

$$\underbrace{84}_{\text{6 surface variables + 13 vertical pressure levels times 6 atmospheric variables}} \times \overbrace{720 \times 1440}^{\text{spatial dimensions}} \approx \underbrace{87\,\text{Mio.}}_{\text{variables}}$$

6 surface variables
+
13 vertical pressure levels times 6 atmospheric variables

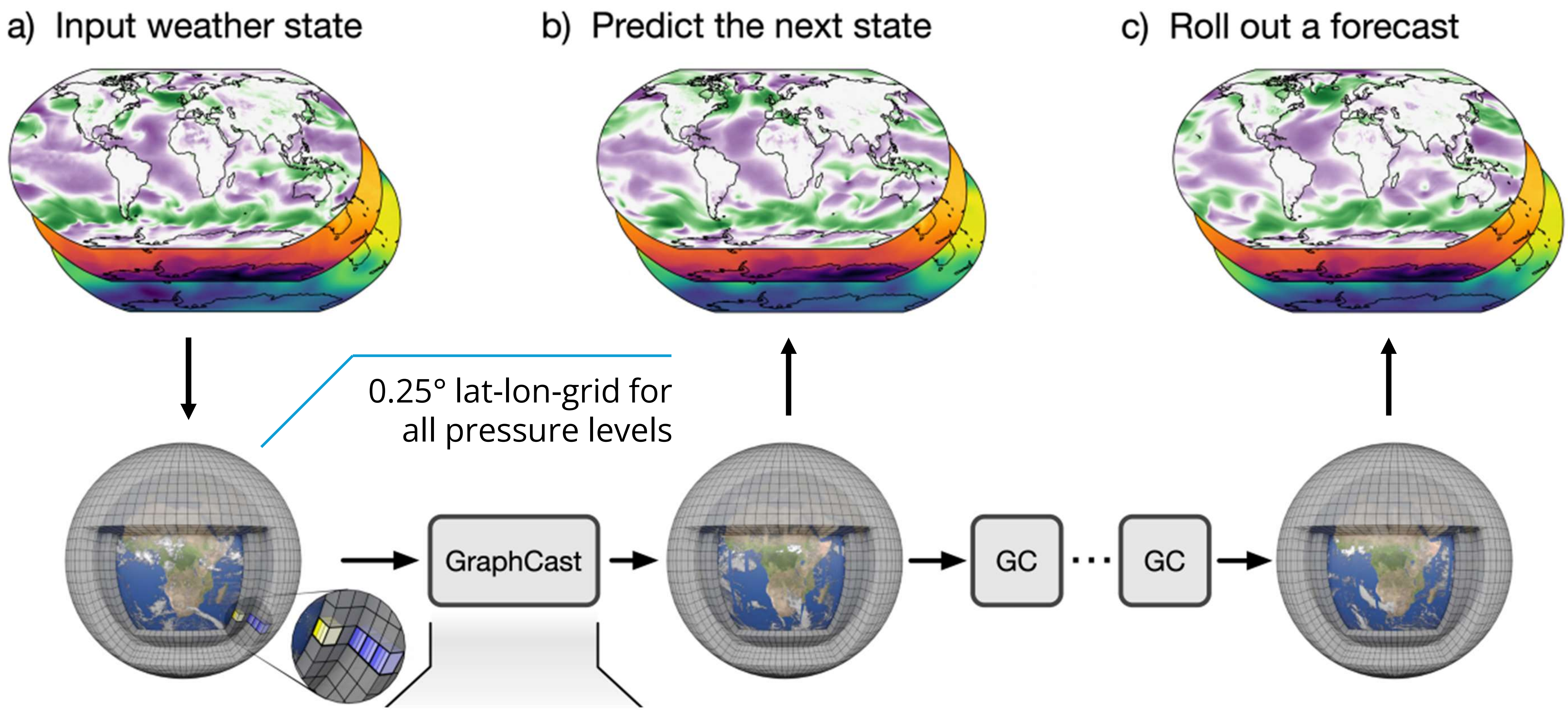
Variables to describe atmospheric state (GenCast output and major part of input)

Lam, R. et al. (2022). *GraphCast: Learning skillful medium-range global weather forecasting*. Figure 1

Quick Intro: GraphCast

Back to the roots

Compare: GraphCast Overall Idea



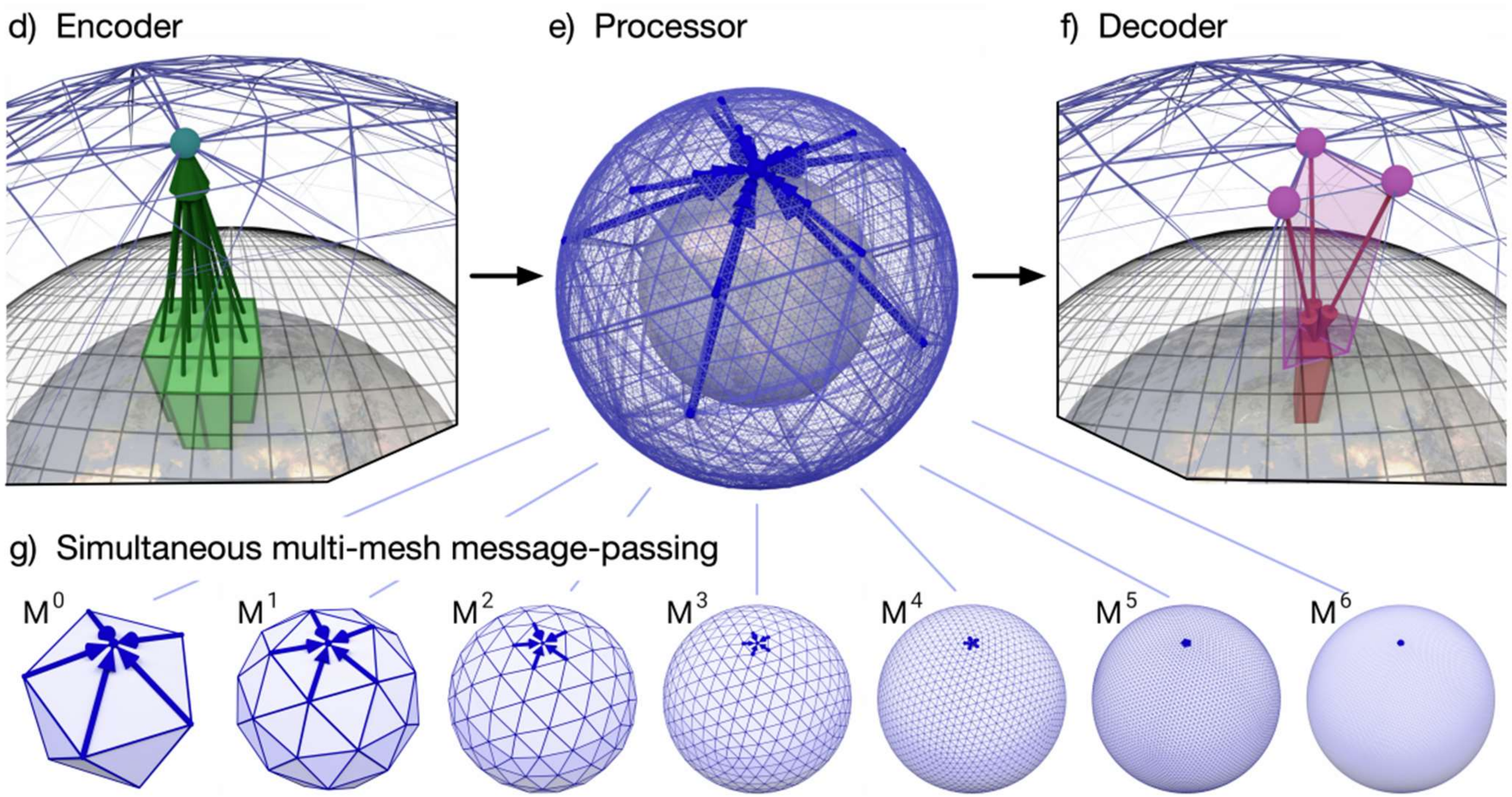
Lam, R. et al. (2022). *GraphCast: Learning skillful medium-range global weather forecasting*. Figure 1

Compare: GraphCast Architecture

approx. **1 million** lat-lon-grid cells

41 162 nodes

approx. **1 million** lat-lon-grid cells



Lam, R. et al. (2022). *GraphCast: Learning skillful medium-range global weather forecasting*. Figure 1

Back to GenCast

Diffusion is all you need

Diffusion Model Basics

Forward process: add noise

x_0 x_1 x_2 ... x_t x_N

sample

$p(X_N) \sim \mathcal{N}(0, \mathbb{I})$

x_0 x_1 x_2 ... x_t x_N

Backward process: remove noise

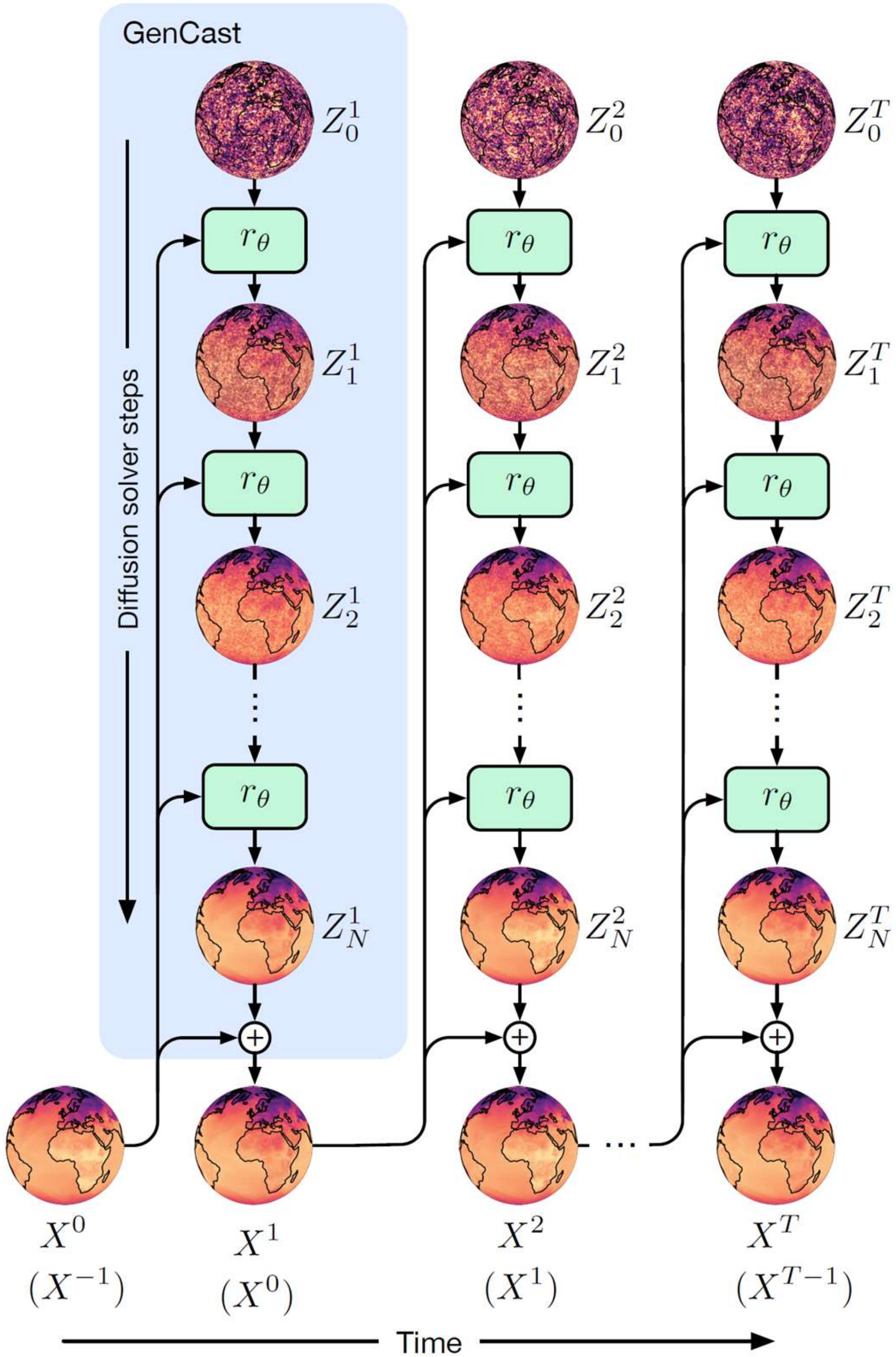
determined by noise schedule

$$p(x_0)$$
$$p(x_1 \mid x_0) = \mathcal{N}\left(x_0 \mid \sqrt{\alpha_1} x_0,\, (1-\alpha_1)\mathbb{I}\right)$$
$$p(x_2 \mid x_1, x_0) = \mathcal{N}\left(x_1 \mid \sqrt{\alpha_2} x_1,\, (1-\alpha_2)\mathbb{I}\right)$$
$$\vdots$$
$$p(x_t \mid x_0) = \mathcal{N}\left(x_t \mid \sqrt{\bar{\alpha}_t} x_0,\, (1-\bar{\alpha}_t)\mathbb{I}\right)$$

$$x_{t-1} \sim q_\theta(x_{t-1} \mid x_t)$$

$$\boxed{q_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1} \mid \underbrace{\mu_\theta(x_t, t),\, \Sigma_\theta(x_t, t)}_{\text{learned by NN}}\right)}$$

https://scholar.harvard.edu/binxuw/classes/machine-learning-scratch/materials/foundation-diffusion-generative-models



GenCast: Autoregressive Diffusion Forecast

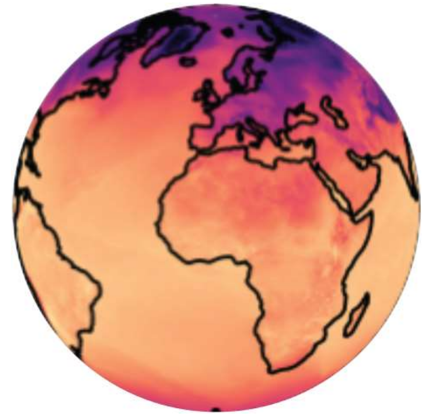
1. Sample residue Z_0^1 from an isotropic normalized Gaussian
2. Denoise Z_0^1 with one solver step r_θ calling the denoiser D_θ conditioned on the previous states X^0 and X^{-1} to receive Z_1^1
3. Repeat until fully denoised residue Z_N^1 with $N = 20$
4. Invert normalization of residue by SZ_{20}^1 and add to previous state X^0 to receive the next state X^1
5. Repeat steps 1. to 4. for $T = 30$ timesteps (12h each) to receive a 15-day forecast

X^t: atmospheric state at time $t \in [-1, T]$
Z_n^t: normalized sample after $n \in [0, N]$ denoising steps at time $t \in [-1, T]$
r_θ: solver step calling denoiser D_θ

Price, I. et al. (2023). GenCast: Diffusion-based ensemble forecasting for medium-range weather. Figure 1

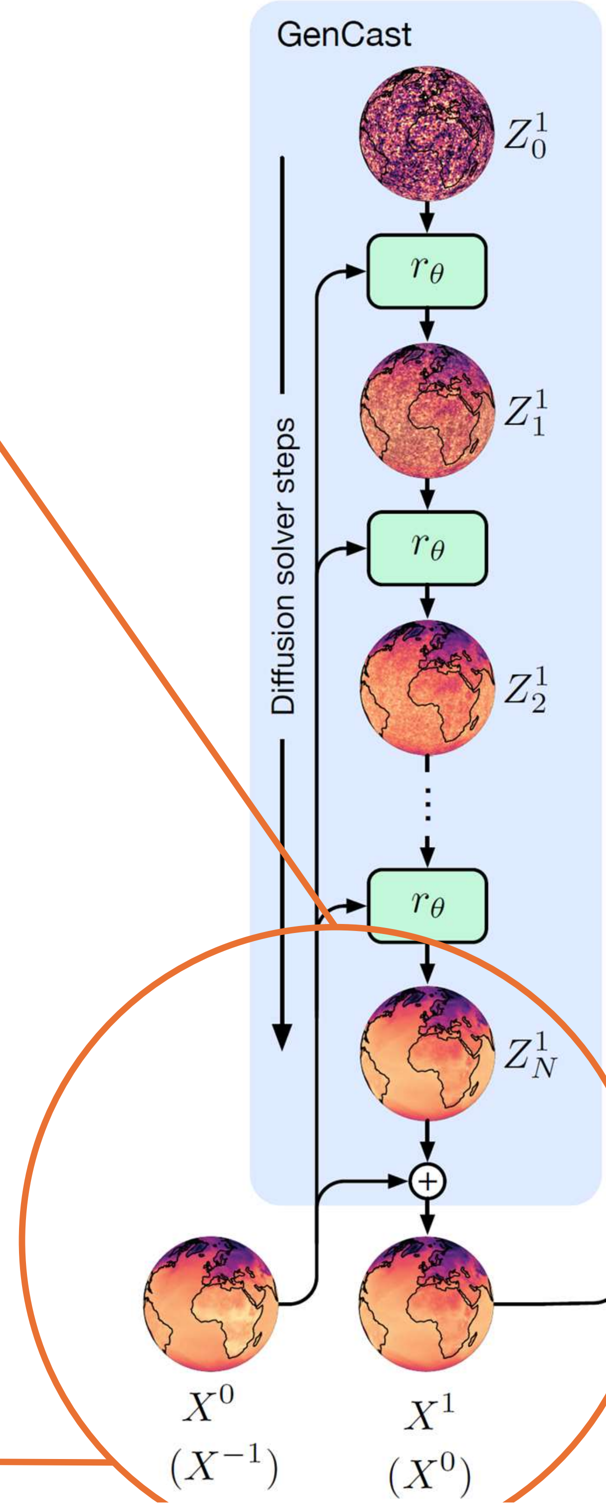
Predicting the Delta





$$\underbrace{X^{t+1}}_{\text{predicted next atmospheric state}} = \underbrace{X^{t}}_{\text{current atmospheric state}} + \underbrace{S}_{\text{inverts normalization}}\boxed{Z_N^{t+1}}$$

change of atmospheric state predicted by solver step r_θ

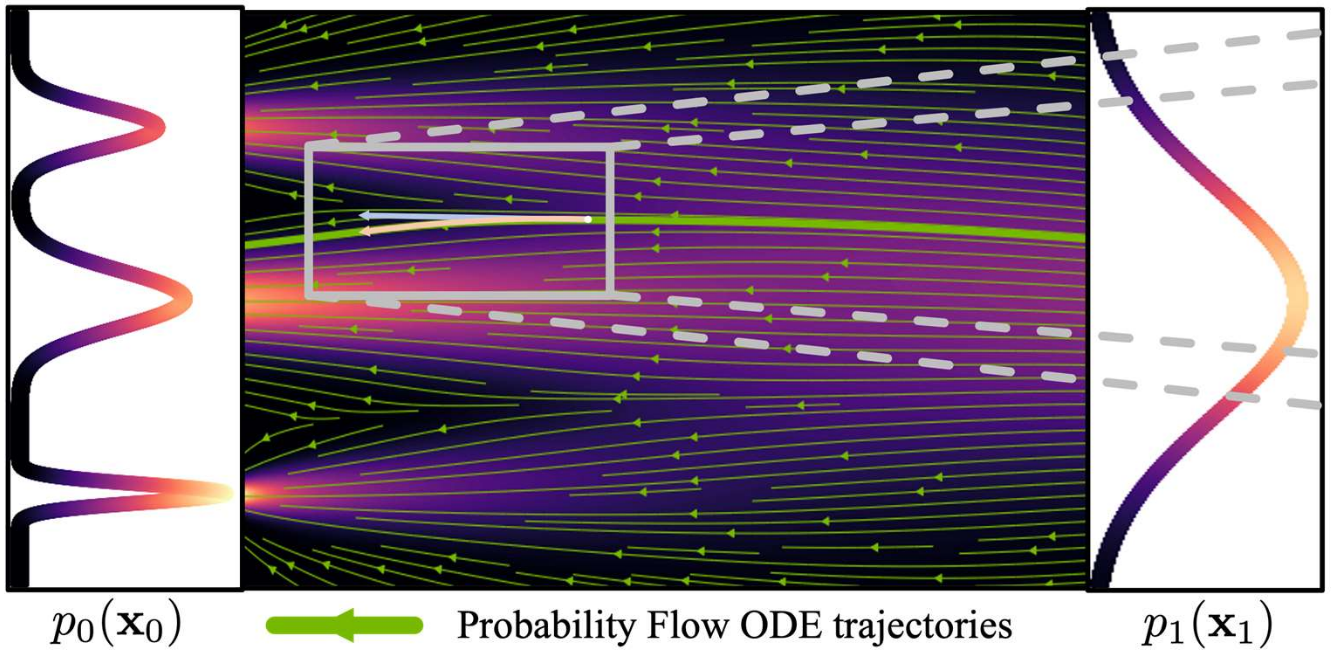


Price, I. et al. (2023). *GenCast: Diffusion-based ensemble forecasting for medium-range weather*. Figure 1

Probability Flow ODE

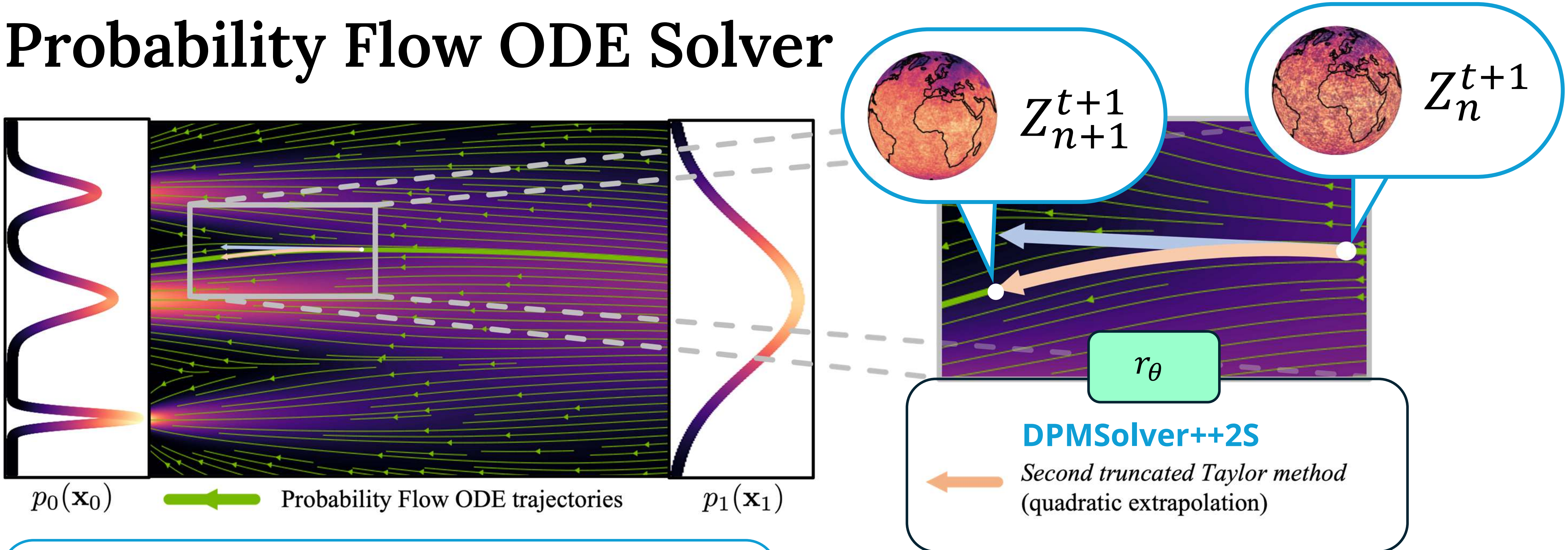
Basic Diffusion Model:
- adding controlled noise at every iteration in forward process
- in backward process remove noise that is again sampled
- stochastic process (non-deterministic)
- can be described by **stochastic differential equation (SDE)**
- comparably slow

GenCast:
- Every diffusion SDE has a corresponding **probability flow ODE** describing the denoising process continuously
- can be solved with numerical ODE solvers
- GenCast uses **DPMSolver++2S**
- deterministic
- computationally efficient



https://research.nvidia.com/labs/toronto-ai/GENIE/assets/genie_pipeline.png

Probability Flow ODE Solver



r_θ solves the **probability flow ODE numerically**:

$$\mathrm{d}\boldsymbol{x} = -\dot{\sigma}(t)\sigma(t)\ \underbrace{\nabla_{\boldsymbol{x}} \log p\left(\boldsymbol{x}; \sigma(t)\right)}_{\text{score function}}\mathrm{d}t$$

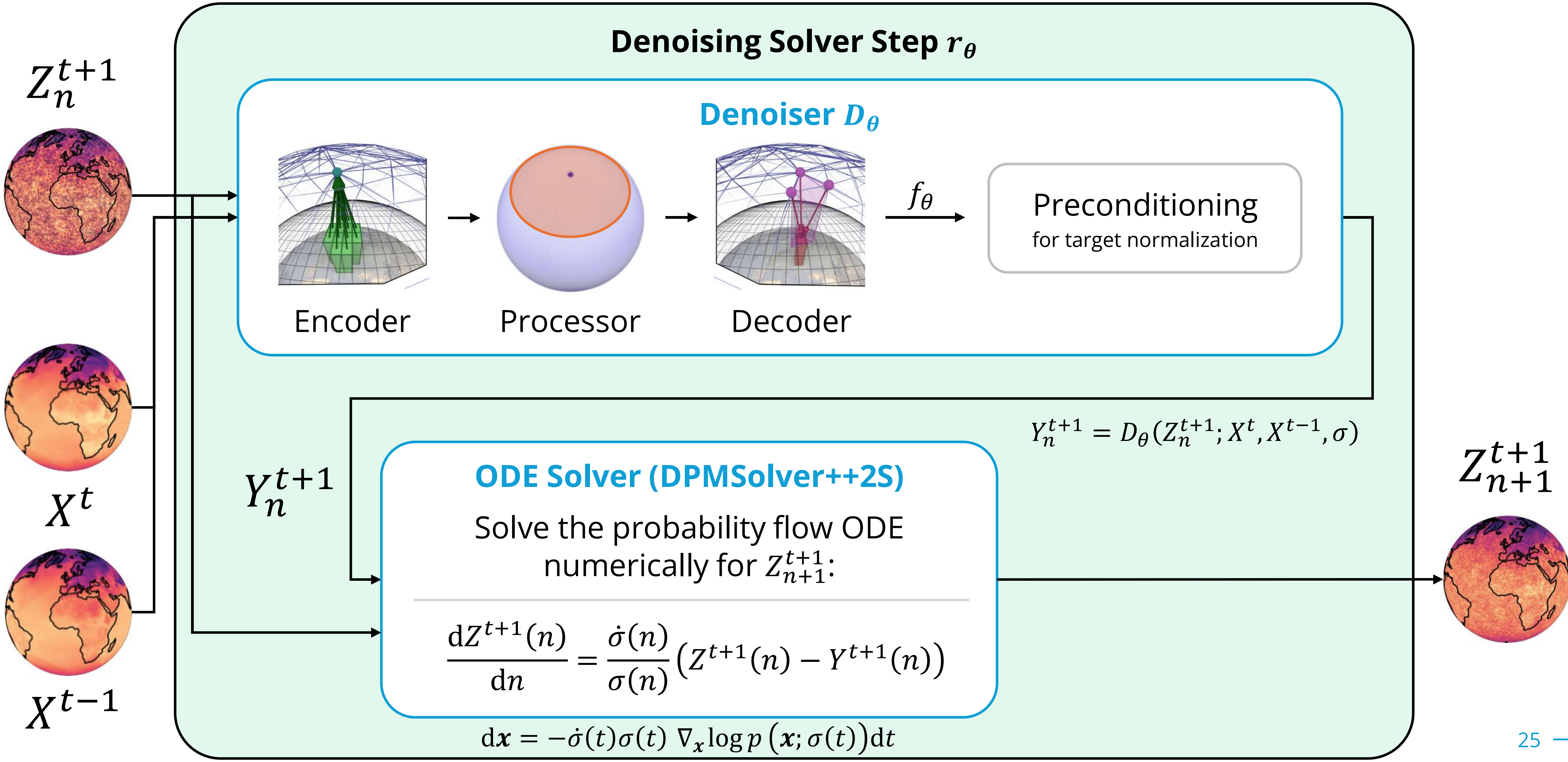
score function
learned by the denoiser →

$$\nabla_{\boldsymbol{x}} \log p\left(\boldsymbol{x}; \sigma(t)\right) = \big(\overbrace{D_\theta(\boldsymbol{x}; \sigma)}^{\text{prediction by Denoiser Neural Network}} - \boldsymbol{x}\big)/\sigma^2$$

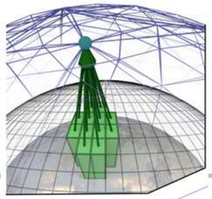
prediction by
Denoiser Neural Network

https://research.nvidia.com/labs/toronto-ai/GENIE/assets/genie_pipeline.png

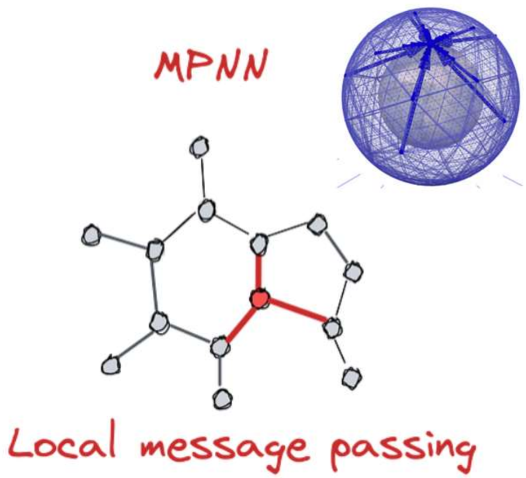
GenCast: One Full Denoising Step



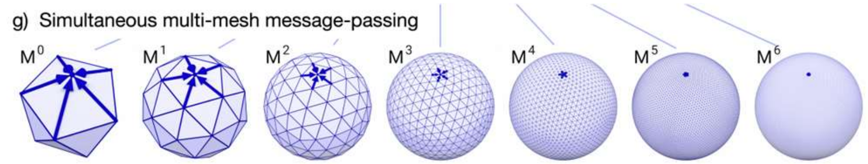
Processor: GraphCast vs. GenCast



Encoder

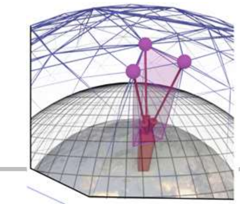


Processor



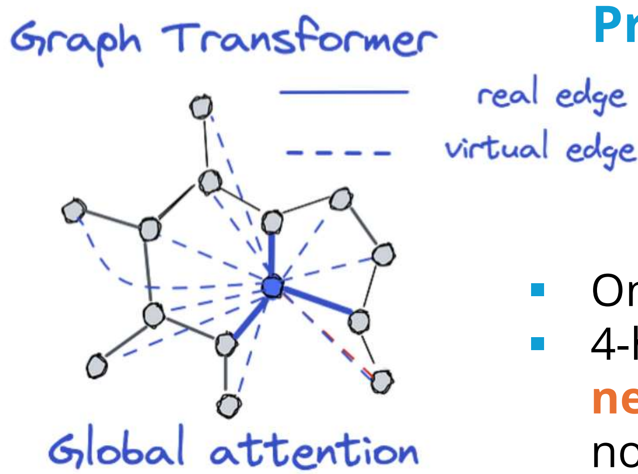
- Iteratively refined multi-mesh
- Allows for long-distance node communication besides standard message passing

Decoder



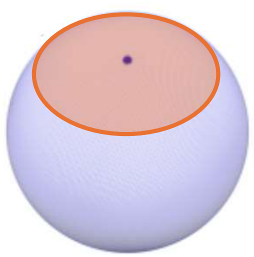
GraphCast

GenCast



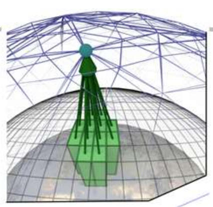
Processor

M^6

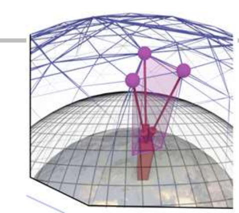


- One single 6-refined icosahedral mesh
- 4-headed self-attention over **32-hop neighborhood** allows for long-distance node communication

Encoder

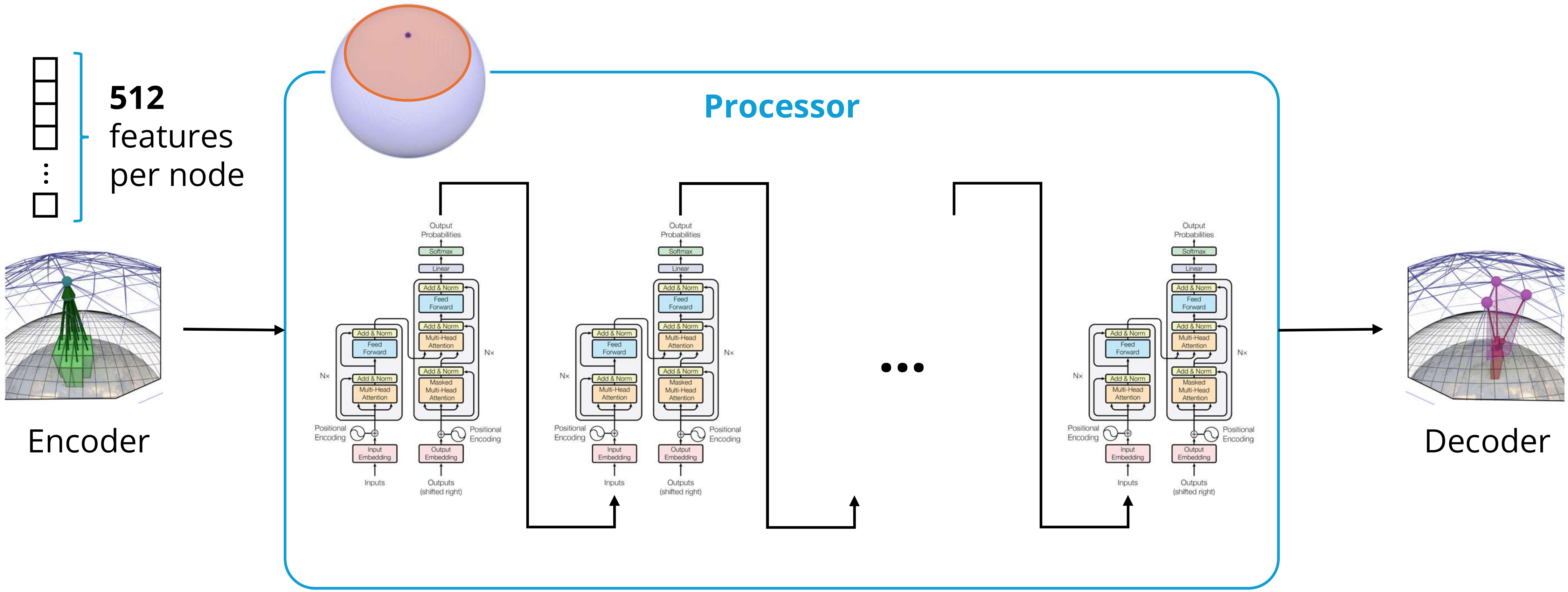


Decoder



https://miro.medium.com/v2/resize:fit:1200/1*gOUffNZRkXJB5utkz6hlWg.png

Processor: Graph Transformer Architecture



16x consecutive standard transformer blocks with
4-head self-attention and feature dimension of 512

Vaswani, A. et al. (2017). *Attention Is All You Need*.

Training the Denoiser D_θ: Loss Function

- Denoiser trained to predict Y^t as expectation of noise-free target Z^t through minimization of loss function:

$$\sum_{t \in \mathcal{D}_{\text{train}}} \mathbb{E}\left[\lambda(\sigma)\frac{1}{|G||J|}\sum_{i \in G}\sum_{j \in J} w_j a_i (Y_{i,j}^t - Z_{i,j}^t)^2\right]$$

$$\begin{aligned}
t &: \text{ timestep index of training set } D_{\text{train}} \\
j \in J &: \text{ variable index (includes pressure level)} \\
i \in G &: \text{ location index (lat \& lon)} \\
w_j &: \text{ loss weight for variable } j \\
a_i &: \text{ area of lat-lon grid cell } i \\
\lambda(\sigma) &: \text{ loss weight for noise level } \sigma
\end{aligned}$$

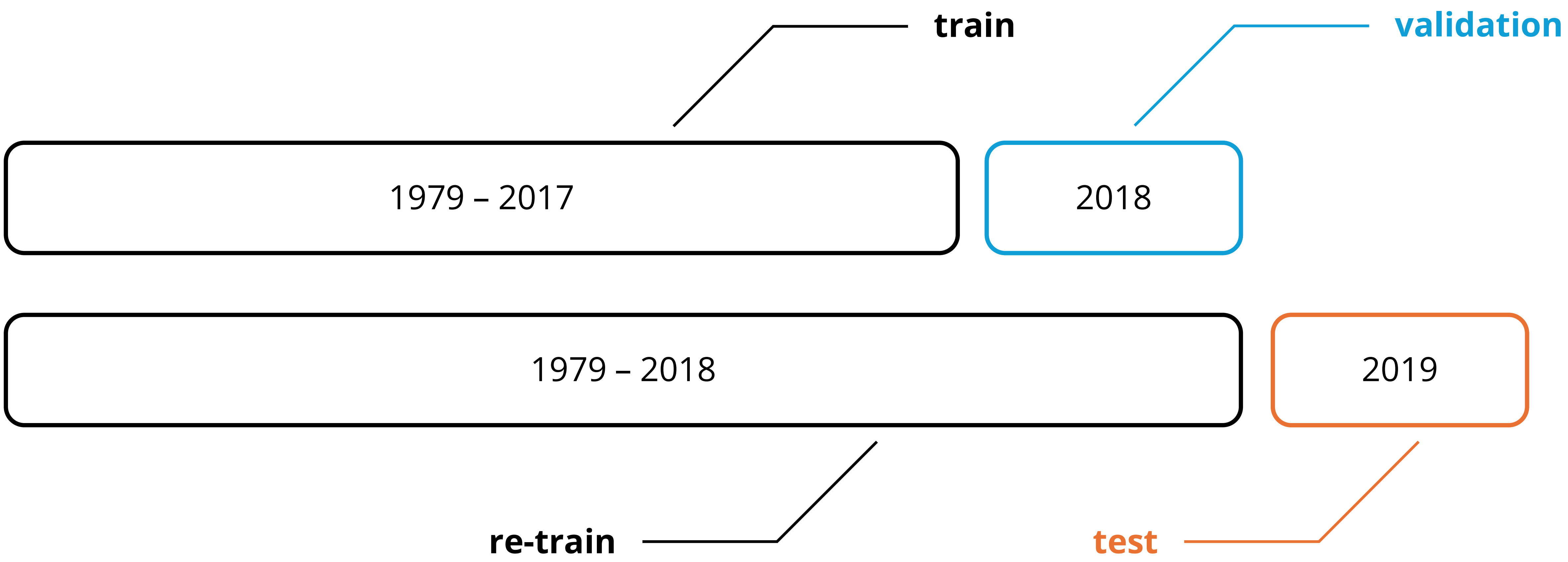
Price, I. et al. (2023). *GenCast: Diffusion-based ensemble forecasting for medium-range weather*. D4

Training Schedule

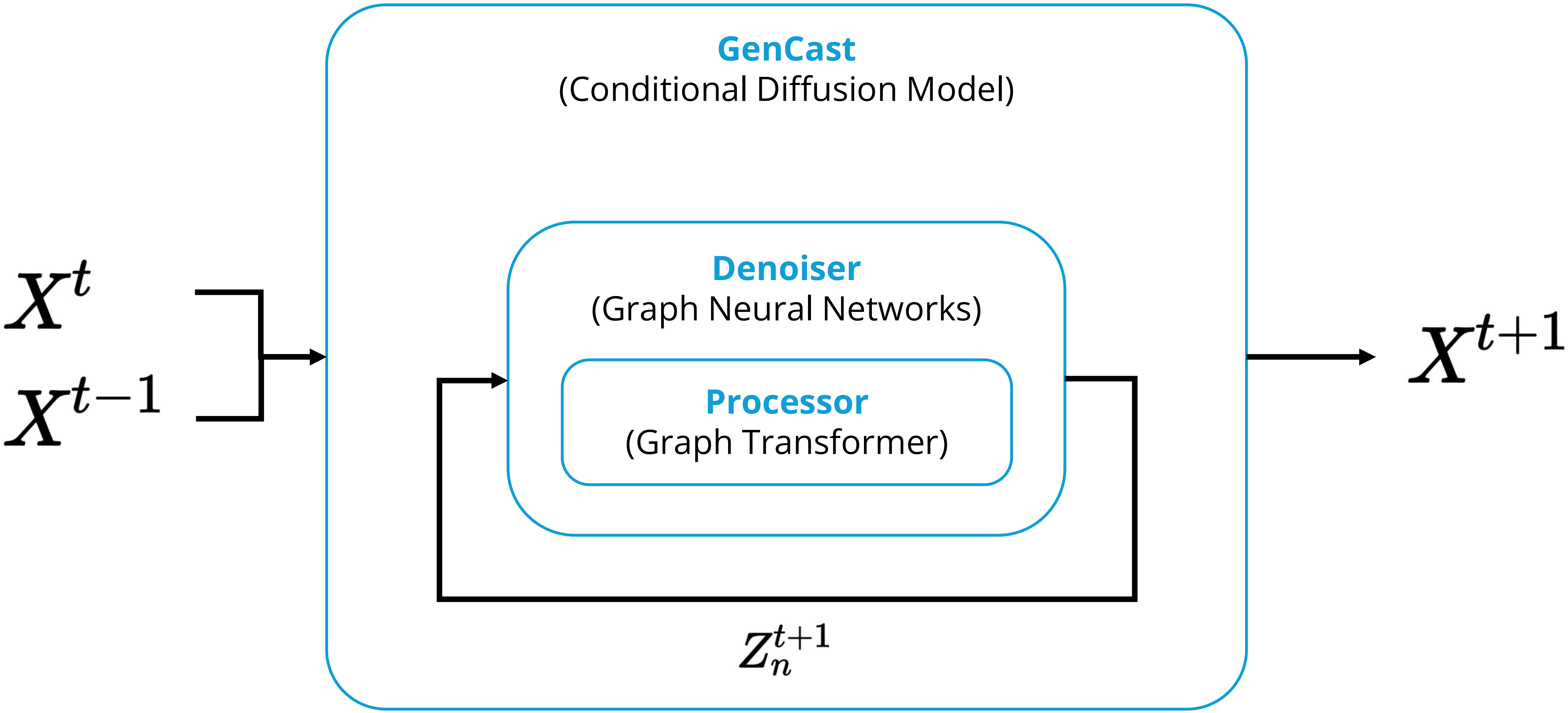
- The model is pre-trained at lower resolution before fine-tuning at actual resolution during the final training

	Pre-Training	**Final Training**
Number training steps	2 million	64 000
Spatial resolution	1°	0.25°
Denoiser mesh	5-refined icosahedral	6-refined icosahedral
Training hardware	32x TPUv5	32x TPUv5
Training duration	3.5 days	1.5 days

Data Split


train
validation
1979 – 2017
2018
1979 – 2018
2019
re-train
test

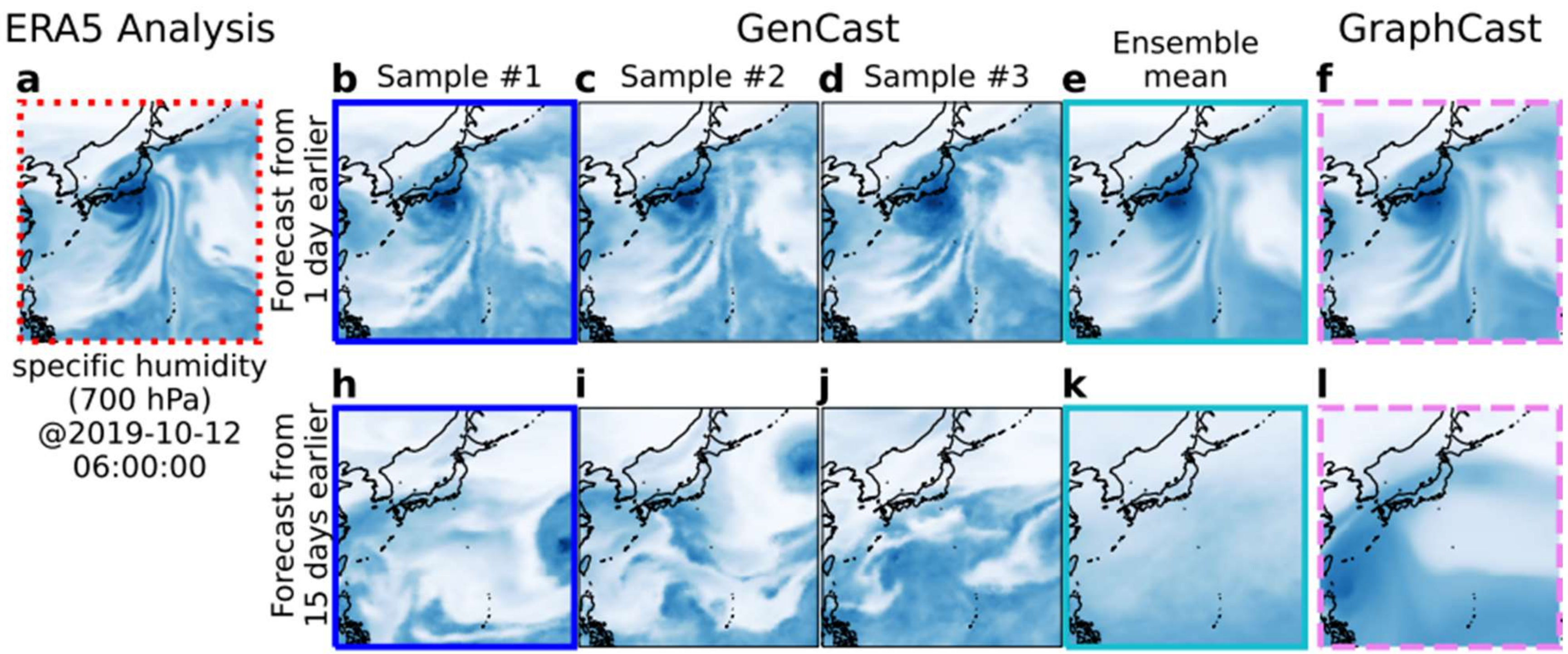
Overall Architecture



Experiments and Results

Let’s benchmark

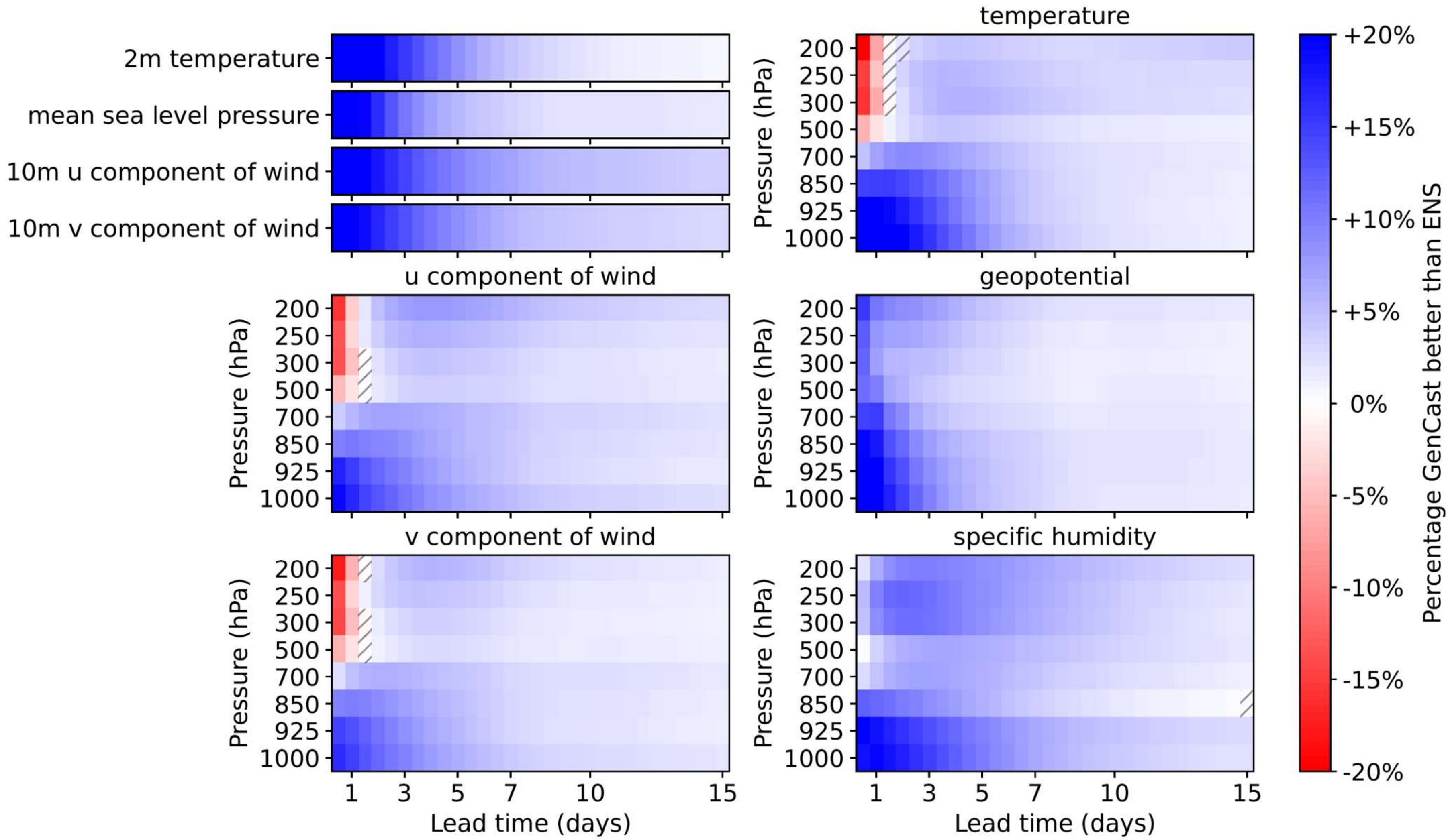
GenCast vs GraphCast: Typhoon Hagibis 2019



- **GenCast** generates crisp samples even 15 days ahead
- **GraphCast** generates blurry forecasts resembling more the ensemble mean

Price, I. et al. (2023). *GenCast: Diffusion-based ensemble forecasting for medium-range weather*. Figure 2

GenCast vs ENS



1320

variable combinations

97.4%

of variable combinations, GenCast outperforms ENS

Price, I. et al. (2023). *GenCast: Diffusion-based ensemble forecasting for medium-range weather*. Figure 3

Strengths and Limitations

\# Mighty but not almighty

The Good …

- very accurate
  - beats ENS in 97.4% of tested variable combinations
- fast
  - only 8 minutes inference time for a single 15-day forecast (30 timesteps of 12h each) on a Cloud TPUv5 device
- lower (computational) cost than ENS
- inherent uncertainty measure

… and the "Bad"

- predicting only one sample is not a "good" forecast as it is randomly sampled from distribution
- computationally more expensive than deterministic MLWP models (like GraphCast)
- still relies on NWP ensemble data assimilation for initial conditions
- temporal resolution limited: only 12h steps (compared to 6h steps for ENS)
- underlying dataset ERA5 is lower bound for spatial and temporal resolution
  - compare to: ENS recently got updated to 0.1° spatial resolution
- physical behavior only incorporated in initial condition
- Diffusion model only approximates underlying distribution

Conclusion

So what?

Potential and Outlook

Potential:

- Application in industry promising (e.g. energy trading)
- Long-term forecasts potentially interesting

Outlook:

- Papers in AI-based weather forecasting are skyrocketing
- Many different architectures led to promising results
- ECMWF started adopting AI models
- Exploiting spherical properties
- Higher resolution data for re-training

Questions?

Let's talk about it

Sources

#

Sources

- I. Price, A. Sanchez-Gonzalez, F. Alet, T. R. Andersson, A. El-Kadi, D. Masters, T. Ewalds, J. Stott, S. Mohamed, P. Battaglia, R. Lam, M. Willson. GenCast: Diffusion-based ensemble forecasting for medium-range weather. *arXiv*, 2023.
- R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu, A. Merose, S. Hoyer, G. Holland, O. Vinyals, J. Stott, A. Pritzel, S. Mohamed, P. Battaglia. GraphCast: Learning skillful medium-range global weather forecasting. *Science*, 382, 2023.
- T. Karras, M. Aittala, T. Aila, S. Laine. Elucidating the Design Space of Diffusion-Based Generative Models. *Conference on Neural Information Processing Systems*, 36, 2022.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- G. Nikolentzos, G. Dasoulas, M. Vazirgiannis. k-hop graph neural networks. *Neural Networks*, Volume 130, 2020. Pages 195-205.
- Video from the author: https://www.youtube.com/watch?v=ez1pIFcU52s

Annex

Further information needed?

Hyperparameters: Diffusion Model Training

Optimiser	**AdamW** (Loshchilov and Hutter, 2018)
LR decay schedule	Cosine
Stage 1: Batch size	32
Stage 1: Warm-up steps	1e3
Stage 1: Total train steps	2e6
Stage 1: Peak LR	1e-3
Stage 1: Weight decay	0.1
Stage 2: Batch size	32
Stage 2: Warm-up steps	5e3
Stage 2: Total train steps	64000
Stage 2: Peak LR	1e-4
Stage 2: Weight decay	0.1

Price, I. et al. (2023). *GenCast: Diffusion-based ensemble forecasting for medium-range weather*. Table D1.

Denoiser Preconditioning

$$D_\theta\left(Z_\sigma^t;\ X^{t-1}, X^{t-2},\ \sigma\right) := c_{skip}(\sigma) \cdot Z_\sigma^t\ +\ c_{out}(\sigma) \cdot f_\theta\Big(c_{in}(\sigma) Z_\sigma^t;\ X^{t-1}, X^{t-2},\ c_{noise}(\sigma)\Big)$$

- f_θ is the neural network function
- Z_σ^t is a noise-corrupted version of target Z^t at noise level σ
- c_{in}, c_{out}, c_{skip}, c_{noise} are preconditioning functions

Skip scaling	$c_{\text{skip}}(\sigma)$	$\sigma_{\text{data}}^2 / \left(\sigma^2 + \sigma_{\text{data}}^2\right)$
Output scaling	$c_{\text{out}}(\sigma)$	$\sigma \cdot \sigma_{\text{data}} / \sqrt{\sigma_{\text{data}}^2 + \sigma^2}$
Input scaling	$c_{\text{in}}(\sigma)$	$1 / \sqrt{\sigma^2 + \sigma_{\text{data}}^2}$
Noise cond.	$c_{\text{noise}}(\sigma)$	$\frac{1}{4} \ln(\sigma)$

Karras, T. et al. (2022). *Elucidating the Design Space of Diffusion-Based Generative Models*. Table 1

Noise Schedule

$$\sigma_i := \left(\sigma_{max}^{\frac{1}{\rho}} + \frac{i}{N-1}\left(\sigma_{min}^{\frac{1}{\rho}} - \sigma_{max}^{\frac{1}{\rho}}\right)\right)^{\rho} \quad \text{for } i \in \{0 \ldots N-1\}$$

- ρ controls shortening of noising steps near σ_{min} in exchange for longer steps near σ_{max}
- $\sigma_{max} = \sigma_0$, $\sigma_{min} = \sigma_{N-1}$, are hyperparameters for the highest and lowest noise level

Name	Notation	Value, sampling	Value, training
Maximum noise level	σ_{max}	80	88
Minimum noise level	σ_{min}	0.03	0.02
Shape of noise distribution	ρ	7	7
Number of noise levels	N	20	
Stochastic churn rate	S_{churn}	2.5	
Churn maximum noise level	S_{tmax}	80	
Churn minimum noise level	S_{tmin}	0.75	
Noise level inflation factor	S_{noise}	1.05	

Karras, T. et al. (2022). *Elucidating the Design Space of Diffusion-Based Generative Models*. Table 1

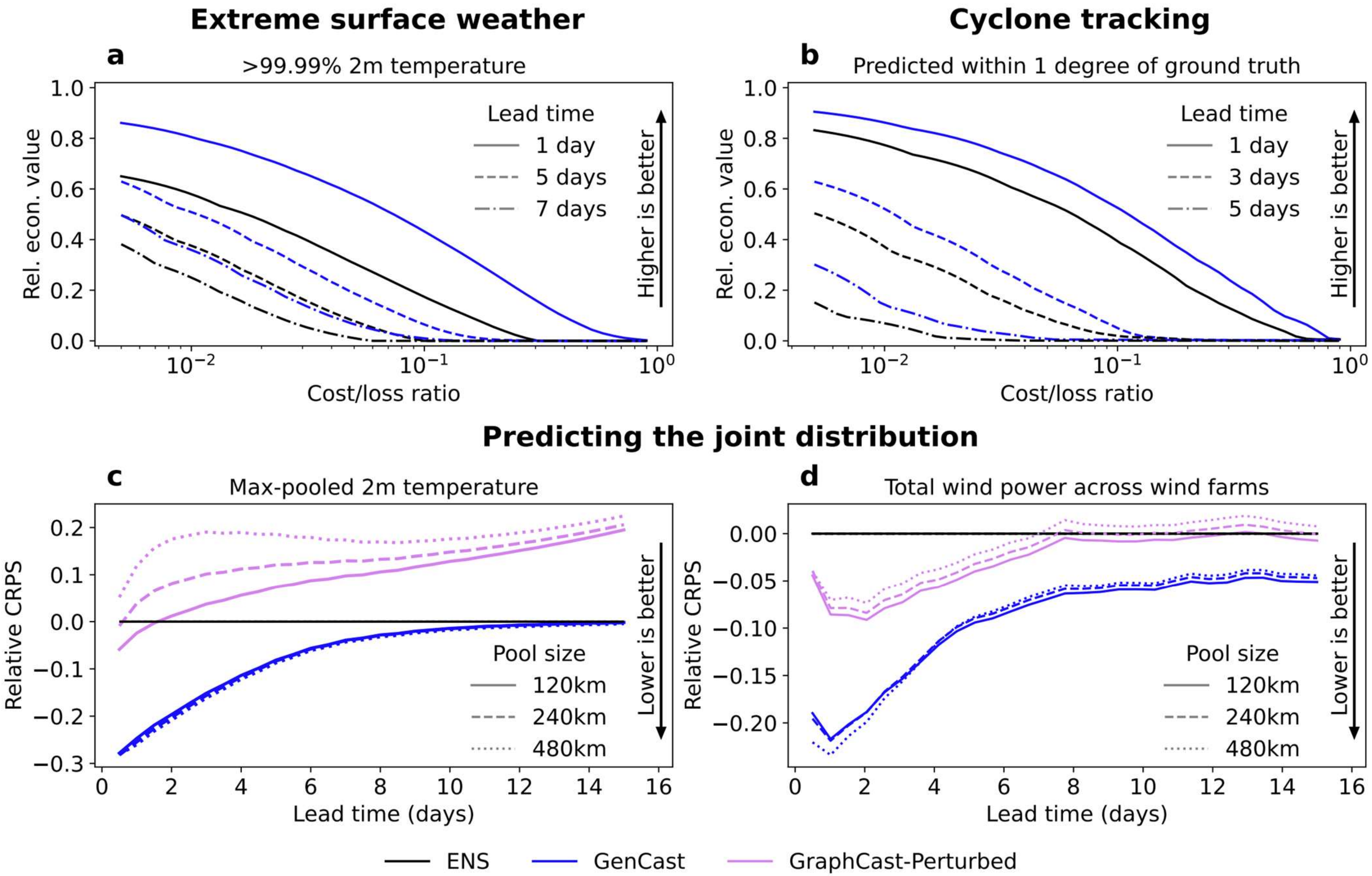
CRPS (Continuously Ranked Probability Score)

The CRPS tries to measure goodness of probabilistic forecast by comparing the expected value of the forecast with the ground truth while incorporating the forecast's uncertainty.

$$\mathrm{CRPS} := \frac{1}{K}\sum_{k}\frac{1}{|G|}\sum_{i} a_i\left(\frac{1}{M}\sum_{m}|x_{i,k}^{m} - y_{i,k}| - \frac{1}{2M^2}\sum_{m,m'}|x_{i,k}^{m} - x_{i,k}^{m'}|\right)$$

Price, I. et al. (2023). GenCast: Diffusion-based ensemble forecasting for medium-range weather. E.2.1.

GenCast vs ENS: Extreme Weather



Price, I. et al. (2023). GenCast: Diffusion-based ensemble forecasting for medium-range weather. Figure 3

Denoiser Distribution

$$F^{-1}(u) = \left(\sigma_{max}^{\frac{1}{\rho}} + u(\sigma_{min}^{\frac{1}{\rho}} - \sigma_{max}^{\frac{1}{\rho}})\right)^{\rho}$$

- $F^{-1}(u)$ is inverse CDF of noise schedule → sample by drawing $u \sim U[0,1]$
- ρ controls shortening of noising steps near σ_{min} in exchange for longer steps near σ_{max}
- $\sigma_{max} = \sigma_0$, $\sigma_{min} = \sigma_{N-1}$, are hyperparameters for the highest and lowest noise level

Karras, T. et al. (2022). *Elucidating the Design Space of Diffusion-Based Generative Models*. Table 1