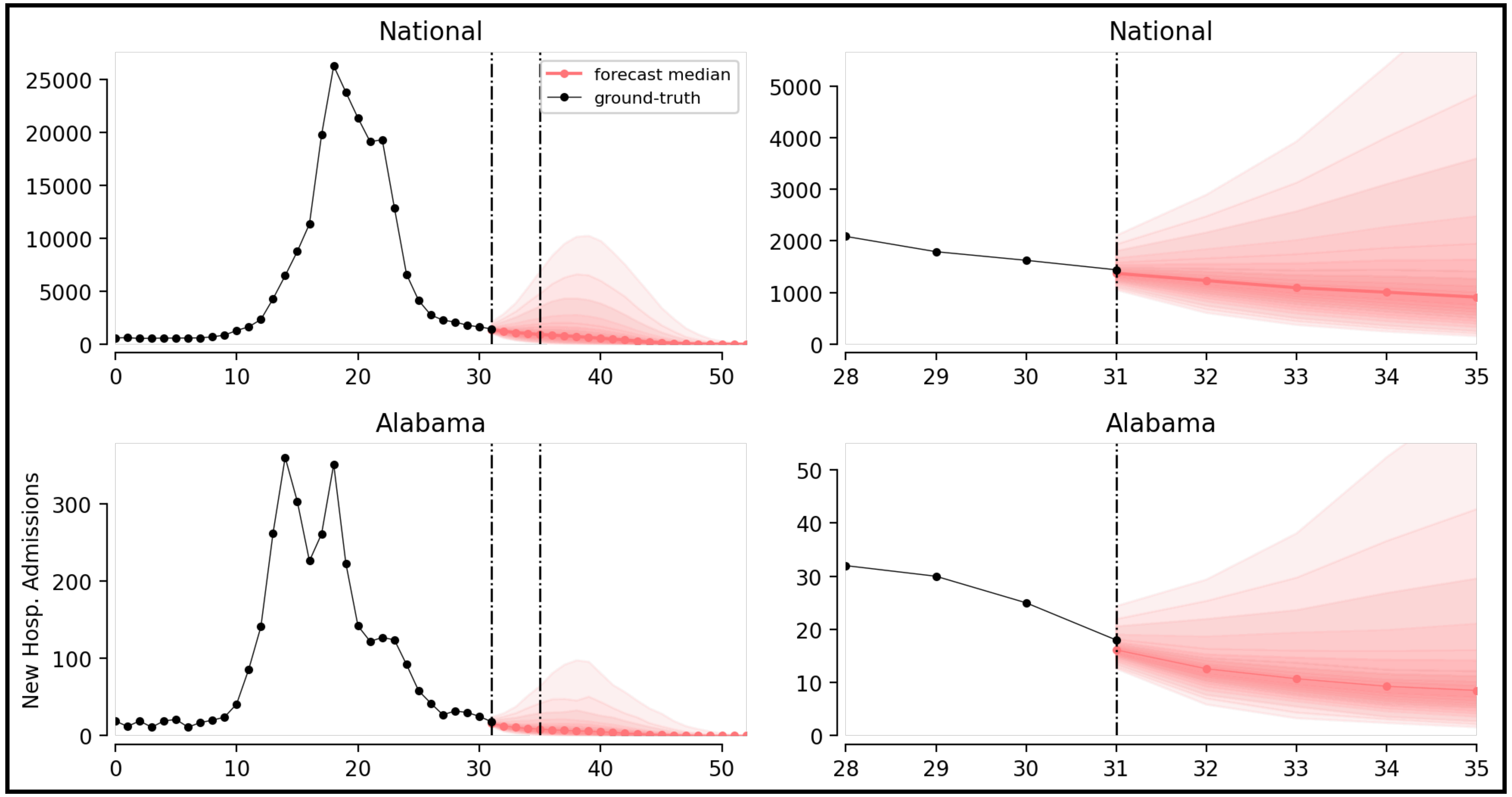
INFLUPAINT: GENERATIVE ARTIFICIAL INTELLIGENCE TO FORECAST SEASONAL INFLUENZA IN THE U.S.

Joseph Lemaitre, Jess K. Edwards, Justin Lessler

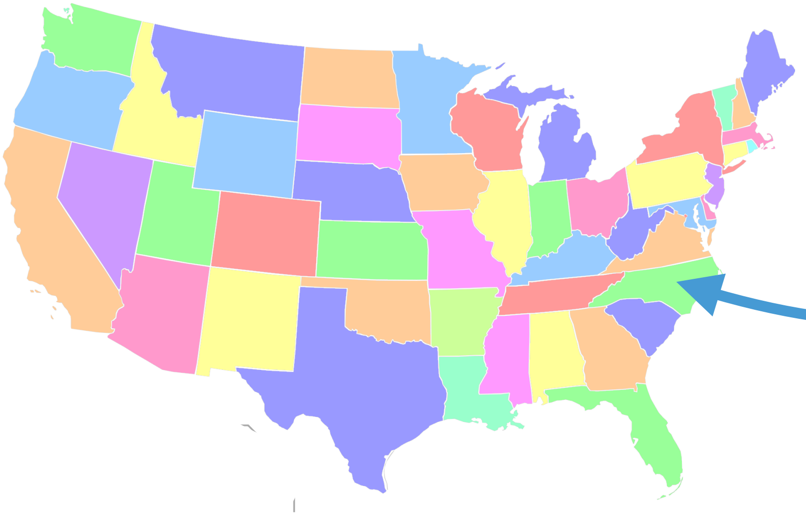
The University of North Carolina at Chapel Hill

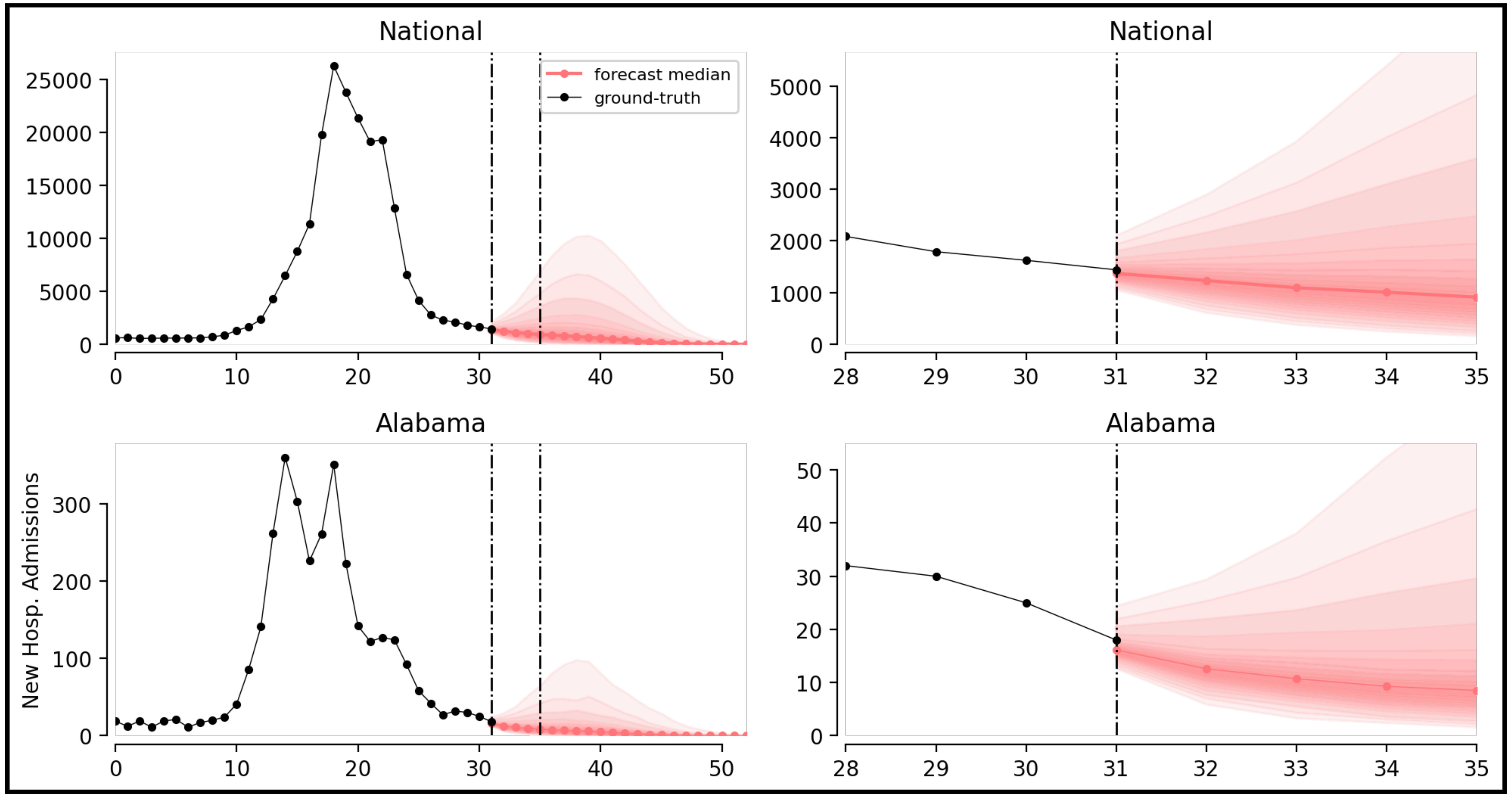


INFLUPAINT: GENERATIVE ARTIFICIAL INTELLIGENCE TO FORECAST SEASONAL INFLUENZA IN THE U.S.

JOSEPH LEMAITRE, JESS K. EDWARDS, JUSTIN LESSLER

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL





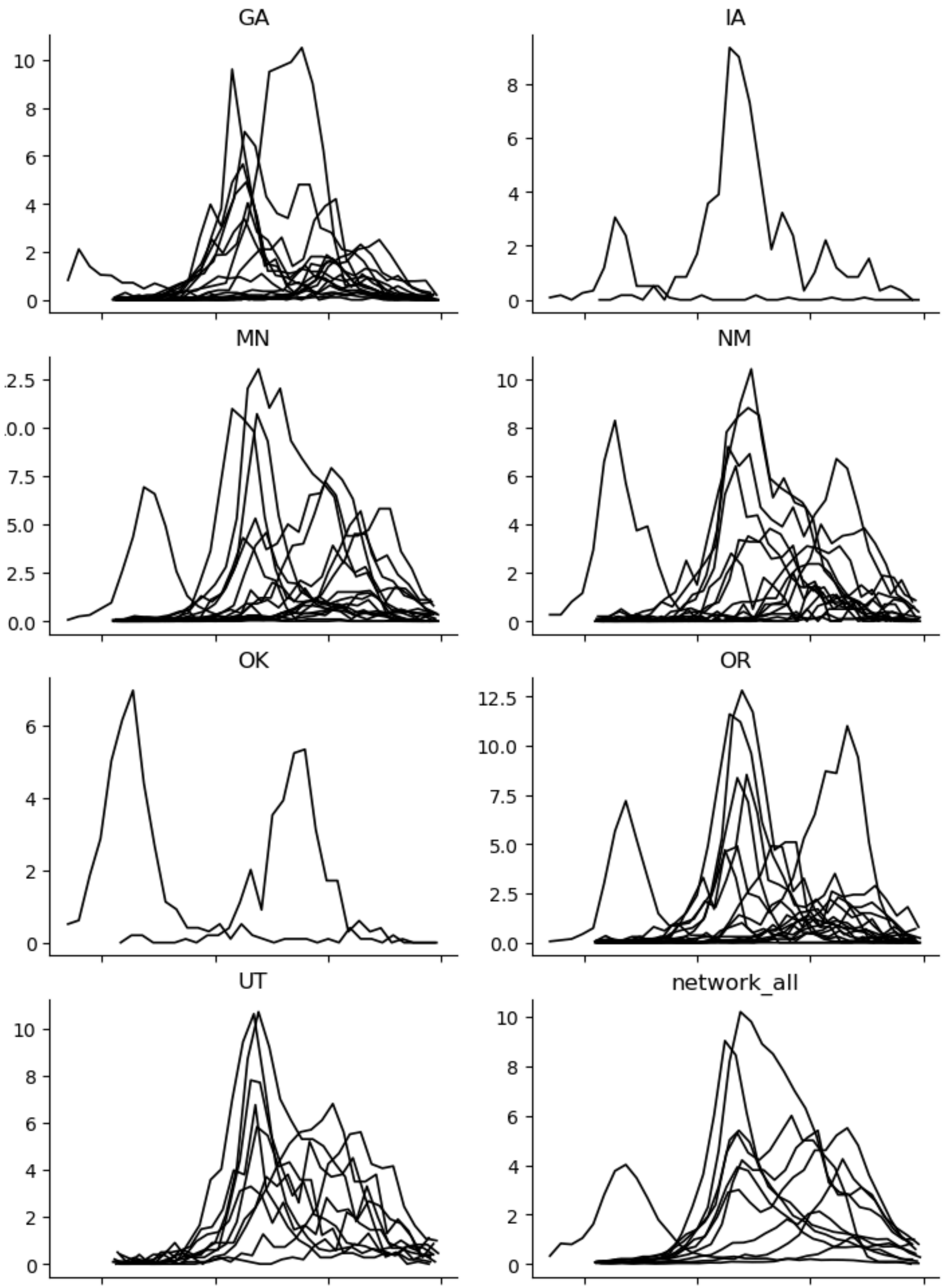
BROADER GOALS

A. **Produce forecasts of infectious disease transmission that reflect the diversity of the observed patterns**

- How many “modes” has the FluSurv data →

B. **Build hybrid statistical/mechanistic models**

- So flexible statistical modeling framework



DEEP GENERATIVE ~~AI~~ MODELING

- **Generative models:** generate new data; builds a representation of the data generating process

- Denoising Diffusion Probabilistic Models (DDPM) provide high-quality samples with **good diversity and mode coverage** (no mode collapse like GANs). Think DALL-E 2 🎨

- Other methods: variational autoencoders (VAE), autoregressive models (ARM), generative adversarial networks (GAN)

$$\underbrace{\Pr(\boldsymbol{\theta}|\boldsymbol{x})}_{\text{posterior}} = \frac{\overbrace{\Pr(\boldsymbol{x}|\boldsymbol{\theta})}^{\text{likelihood}}\ \overbrace{\Pr(\boldsymbol{\theta})}^{\text{prior}}}{\Pr(\boldsymbol{x})}$$

Very useful

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

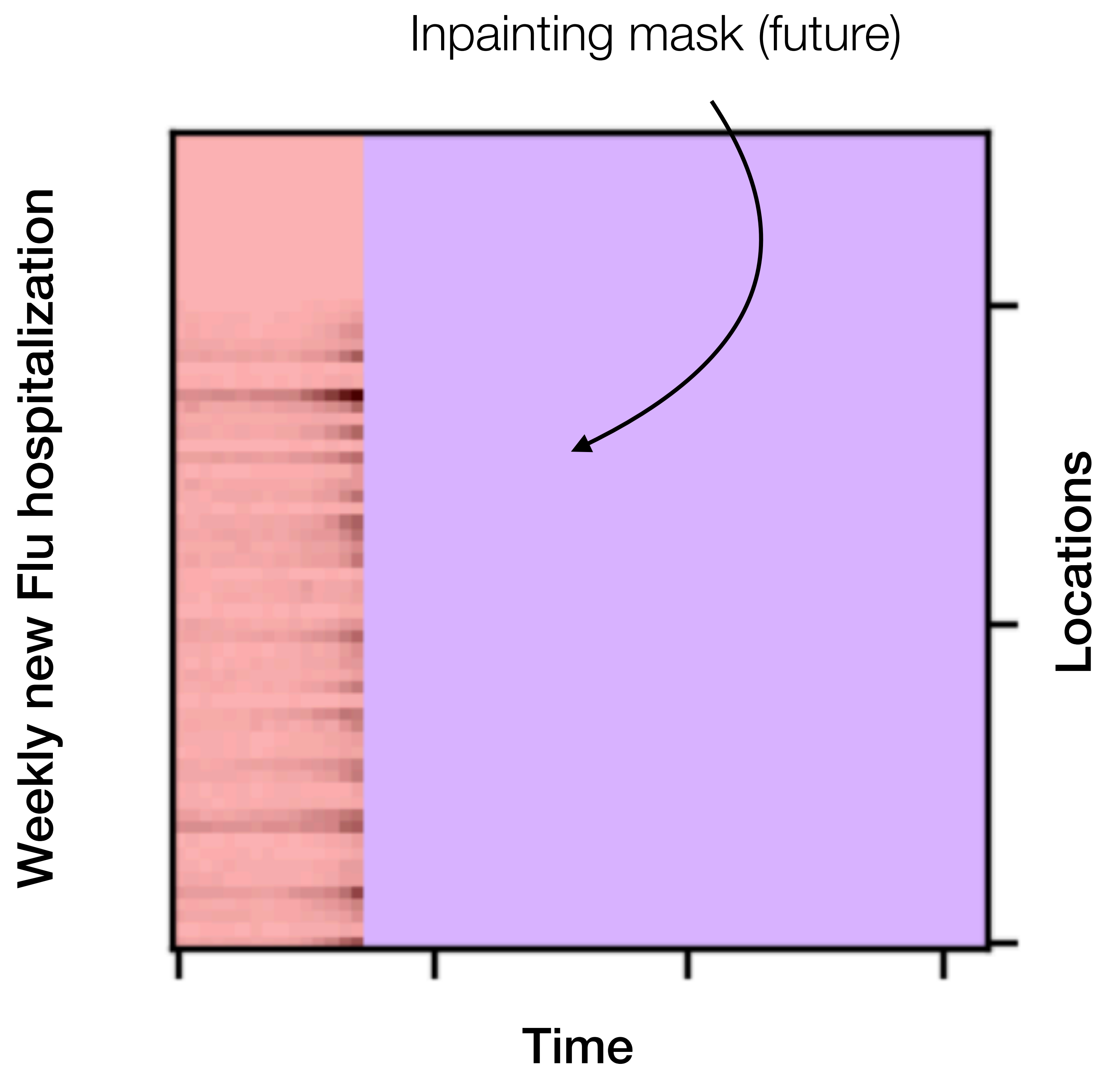
Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal*
OpenAI
prafulla@openai.com

Alex Nichol*
OpenAI
alex@openai.com

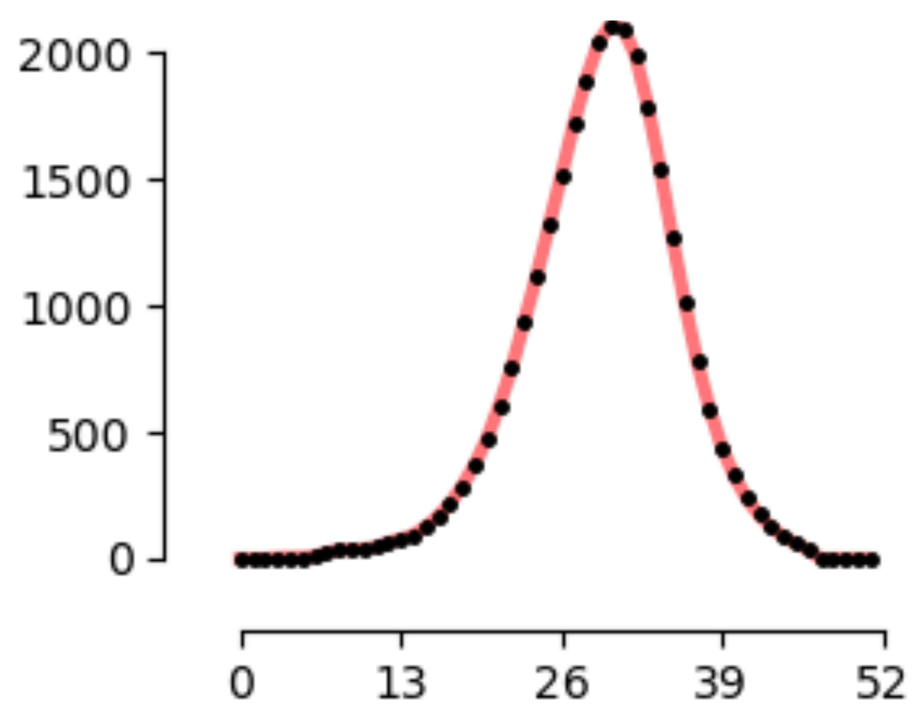
DEEP GENERATIVE ~~AI~~ MODELING

- We **treat influenza epidemic curves as images** (axes are time and location, pixel value is e.g incident hospitalization). We generate epidemic trajectories using a DDPM…



DENOISING DIFFUSION PROBABILISTIC MODELS 101

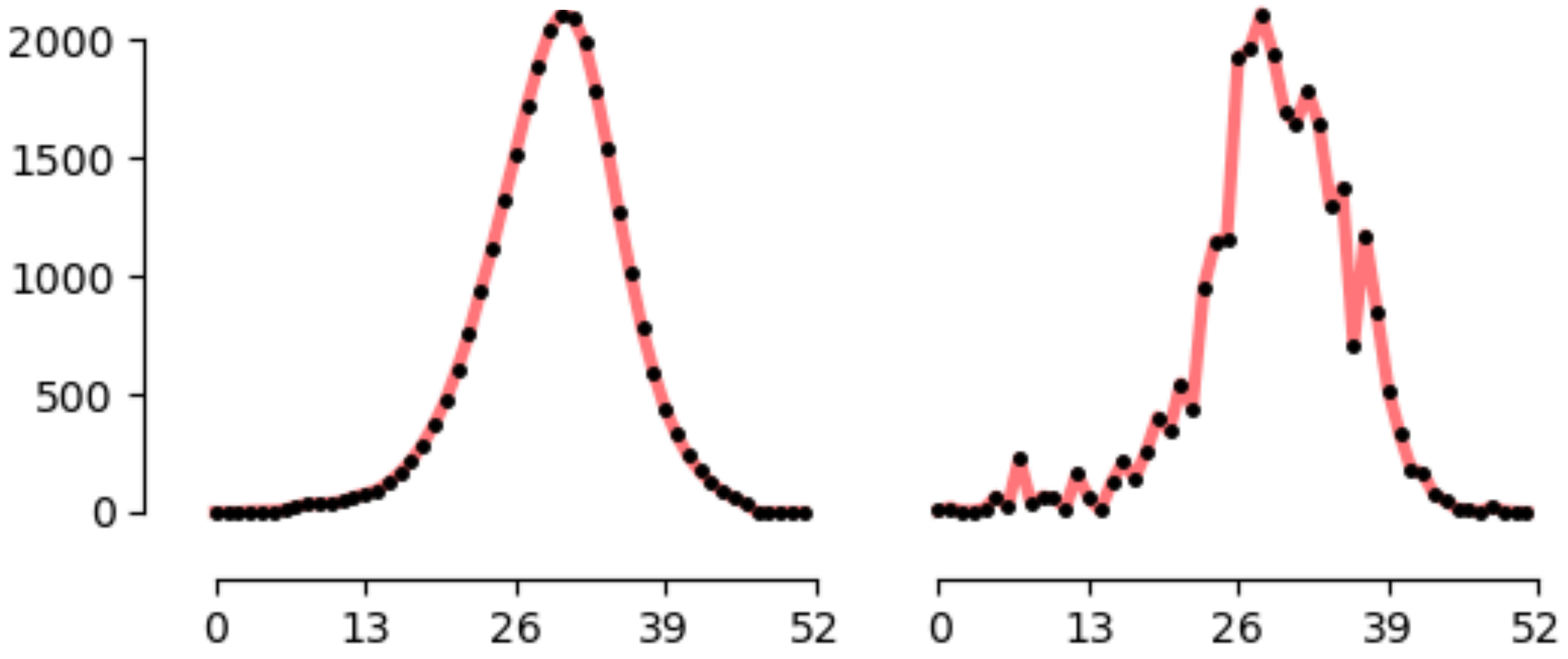
Pure signal



“diffusion time”

DENOISING DIFFUSION PROBABILISTIC MODELS 101

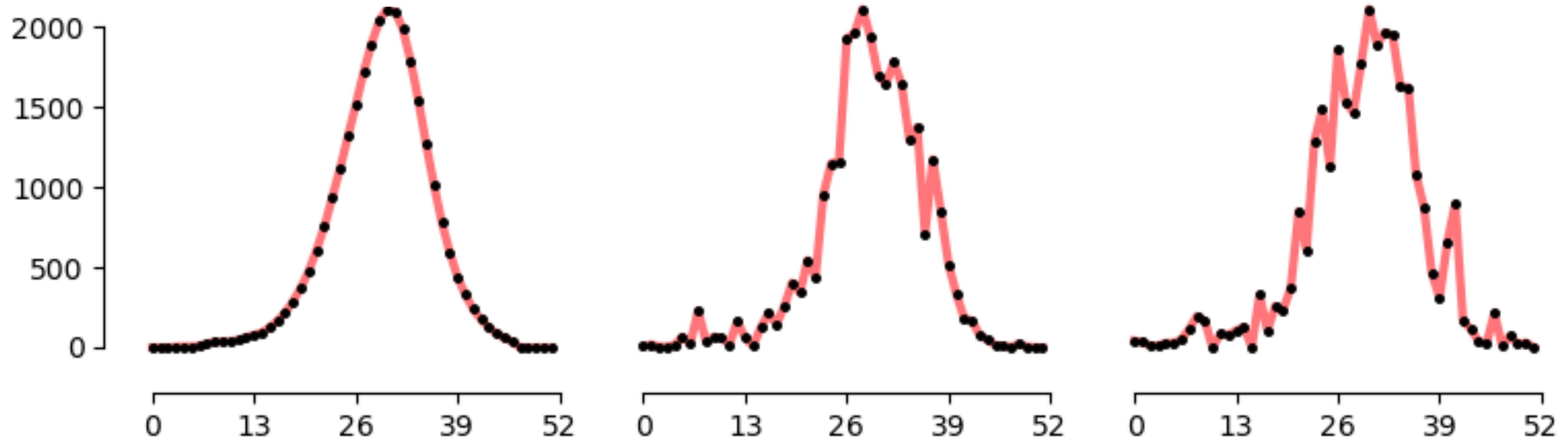
Pure signal



“diffusion time”

DENOISING DIFFUSION PROBABILISTIC MODELS 101

Pure signal

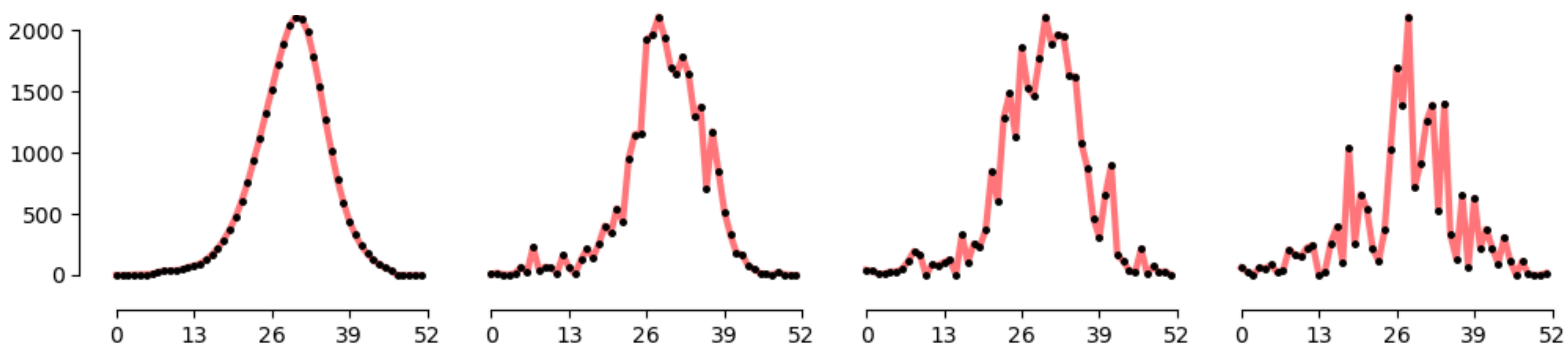


"diffusion time"

DENOISING DIFFUSION PROBABILISTIC MODELS 101

$$q(\mathbf{x_t} \mid \mathbf{x_{t-1}})$$

Pure signal

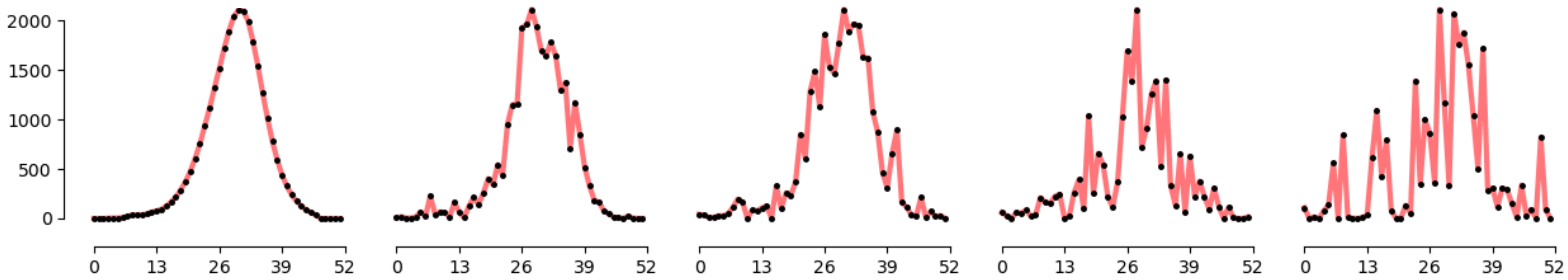


"diffusion time"

DENOISING DIFFUSION PROBABILISTIC MODELS 101

$$q(\mathbf{x_t} \mid \mathbf{x_{t-1}})$$

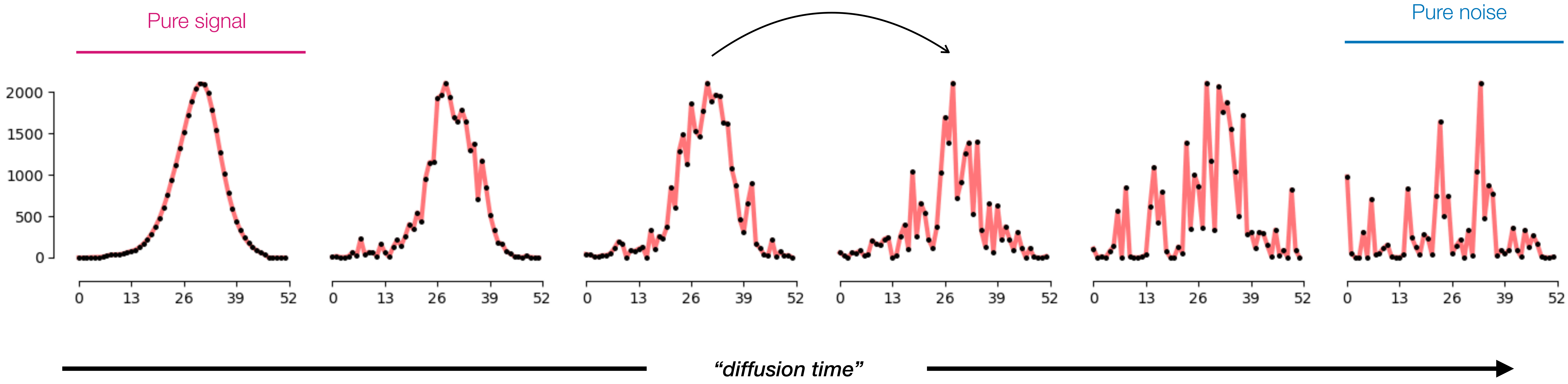
Pure signal



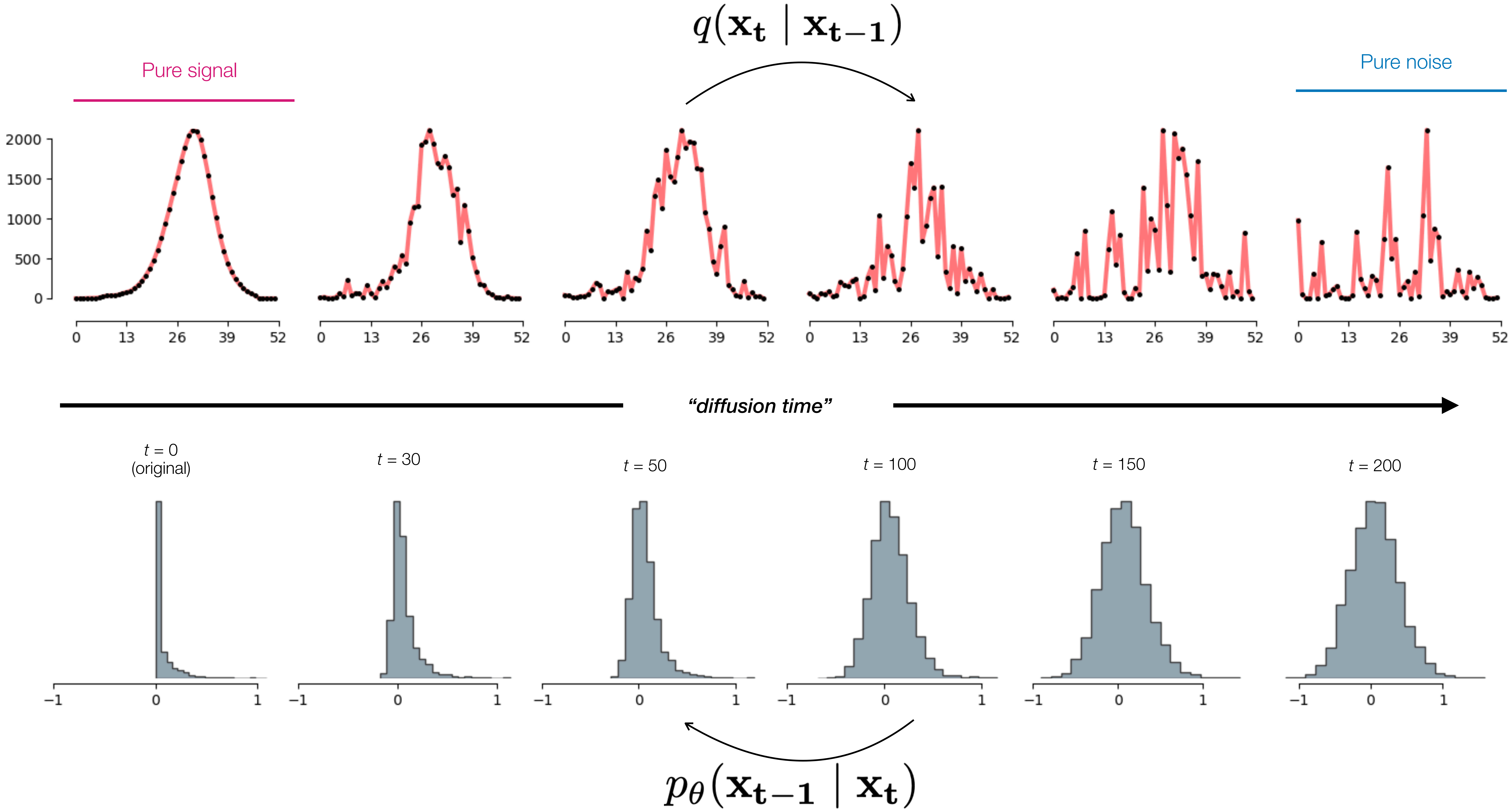
"diffusion time"

DENOISING DIFFUSION PROBABILISTIC MODELS 101

$$q(\mathbf{x_t} \mid \mathbf{x_{t-1}})$$

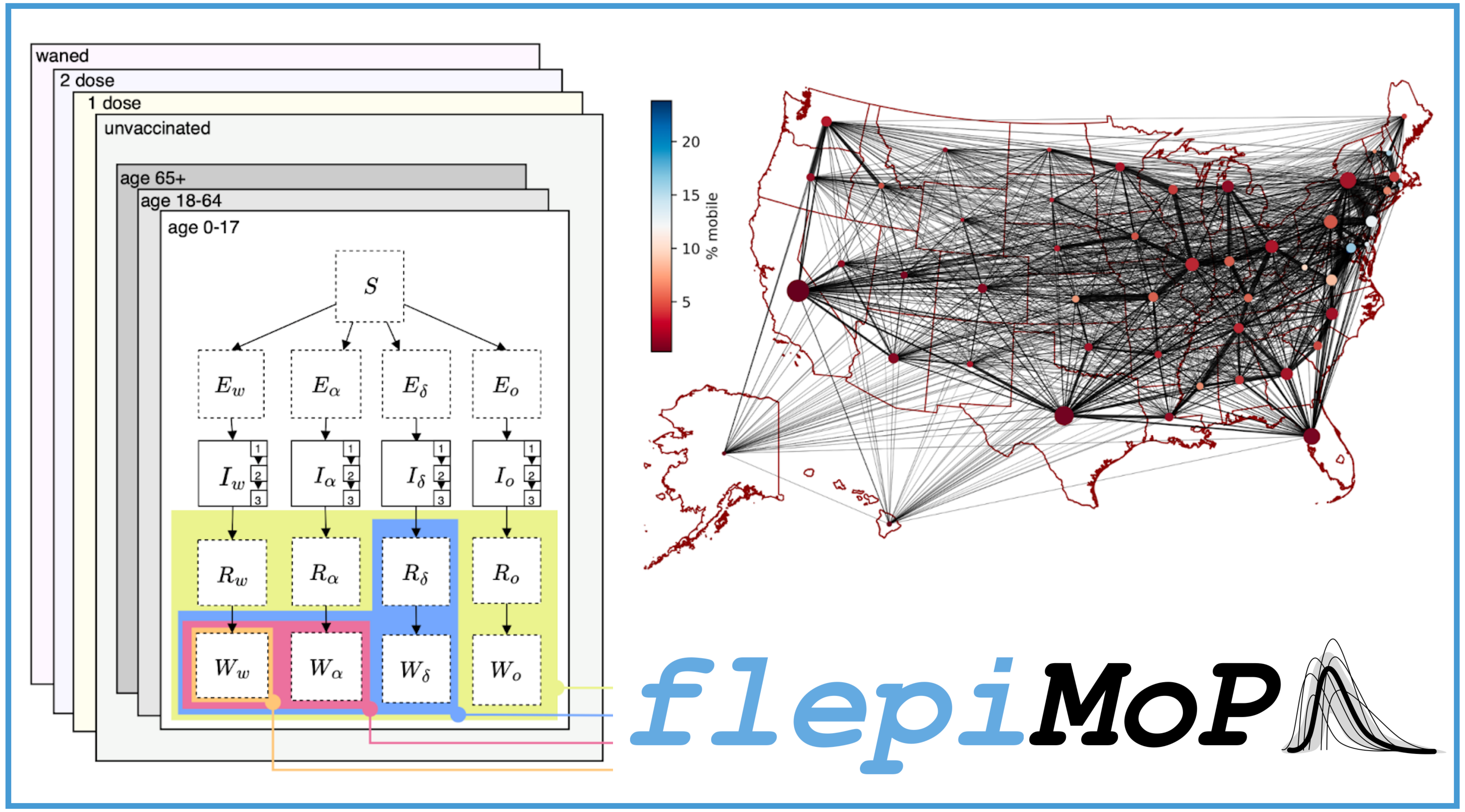


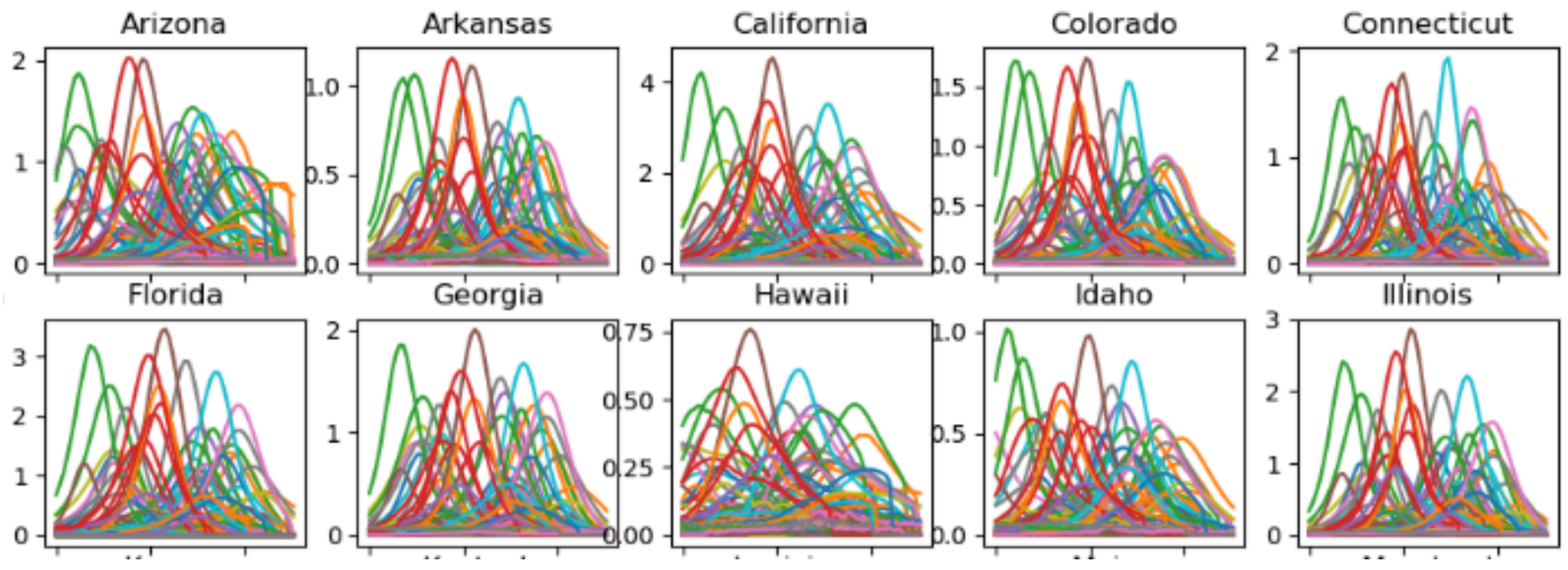
DENOISING DIFFUSION PROBABILISTIC MODELS 101



TRAINING DATA

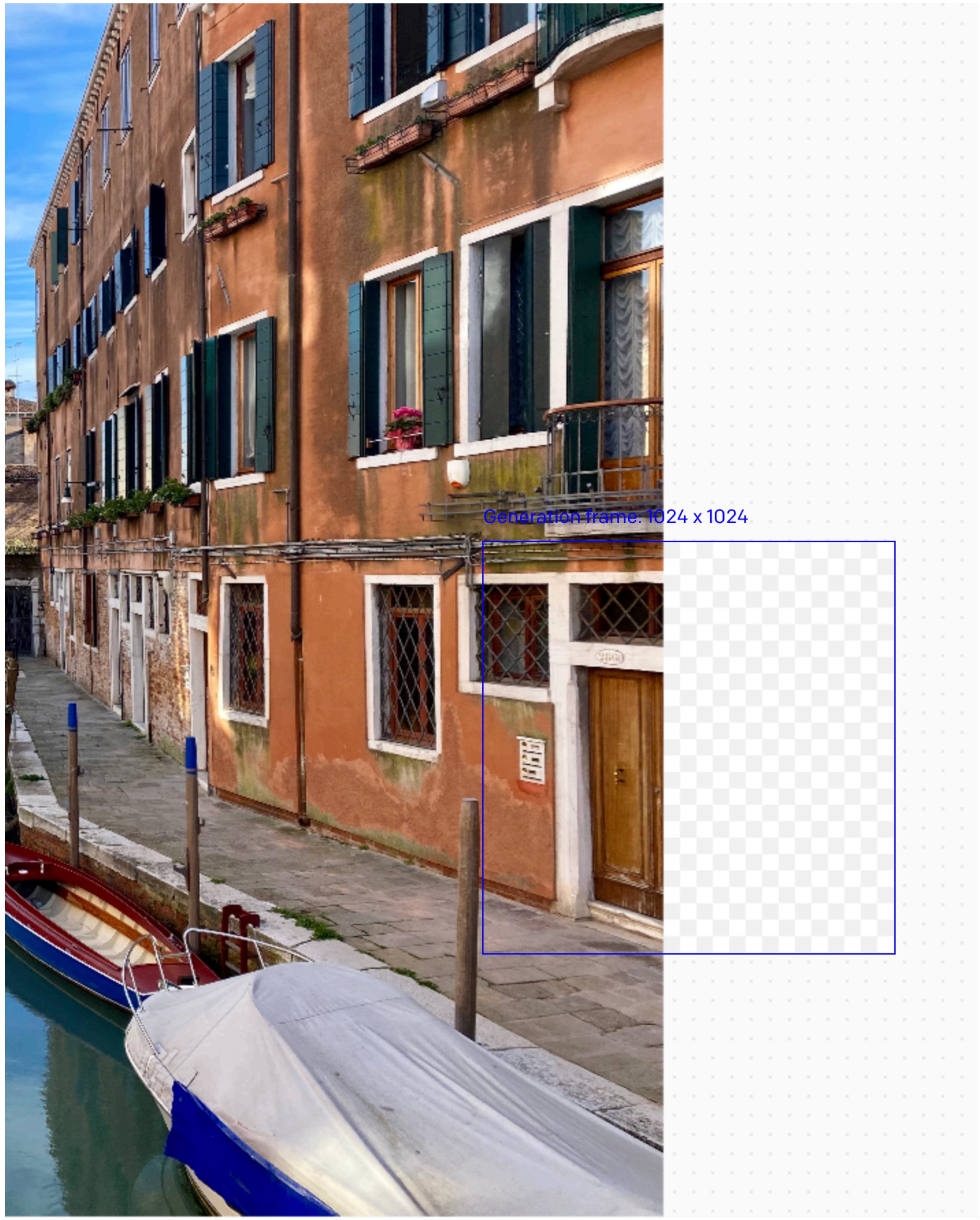
- **No dataset corresponds to what we want to model** (New Hospital Admission at the state level). We mix 👨🏻‍🍳:
    - Outputs from a **mechanistic influenza transmission** model (flepiMoP.org 👀)
    - **Reported US influenza data** (Fluview, FluSurv) at different locations (very small datasets, dozen of samples),
- **Augmented** & scaled





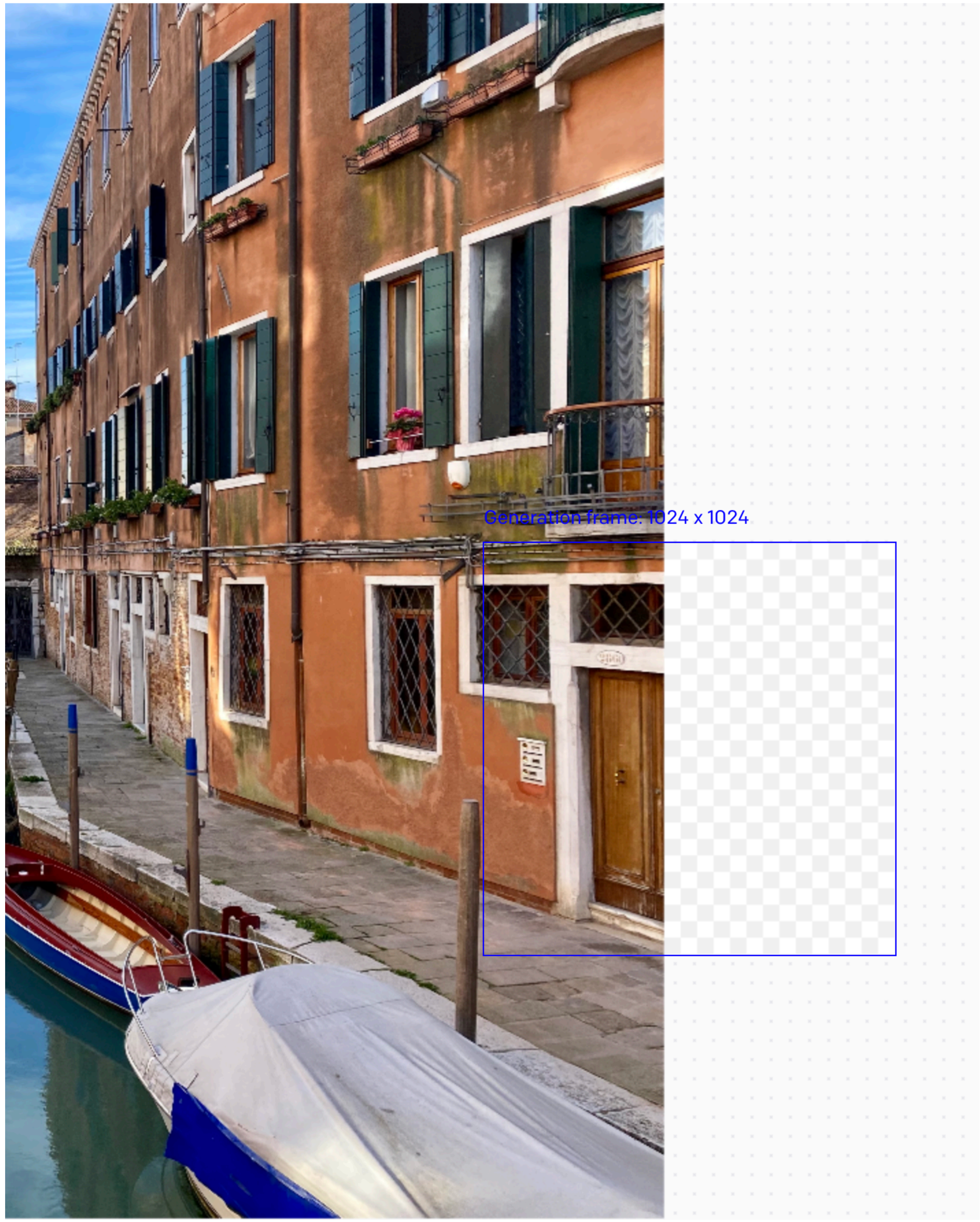
Transformed training data sample

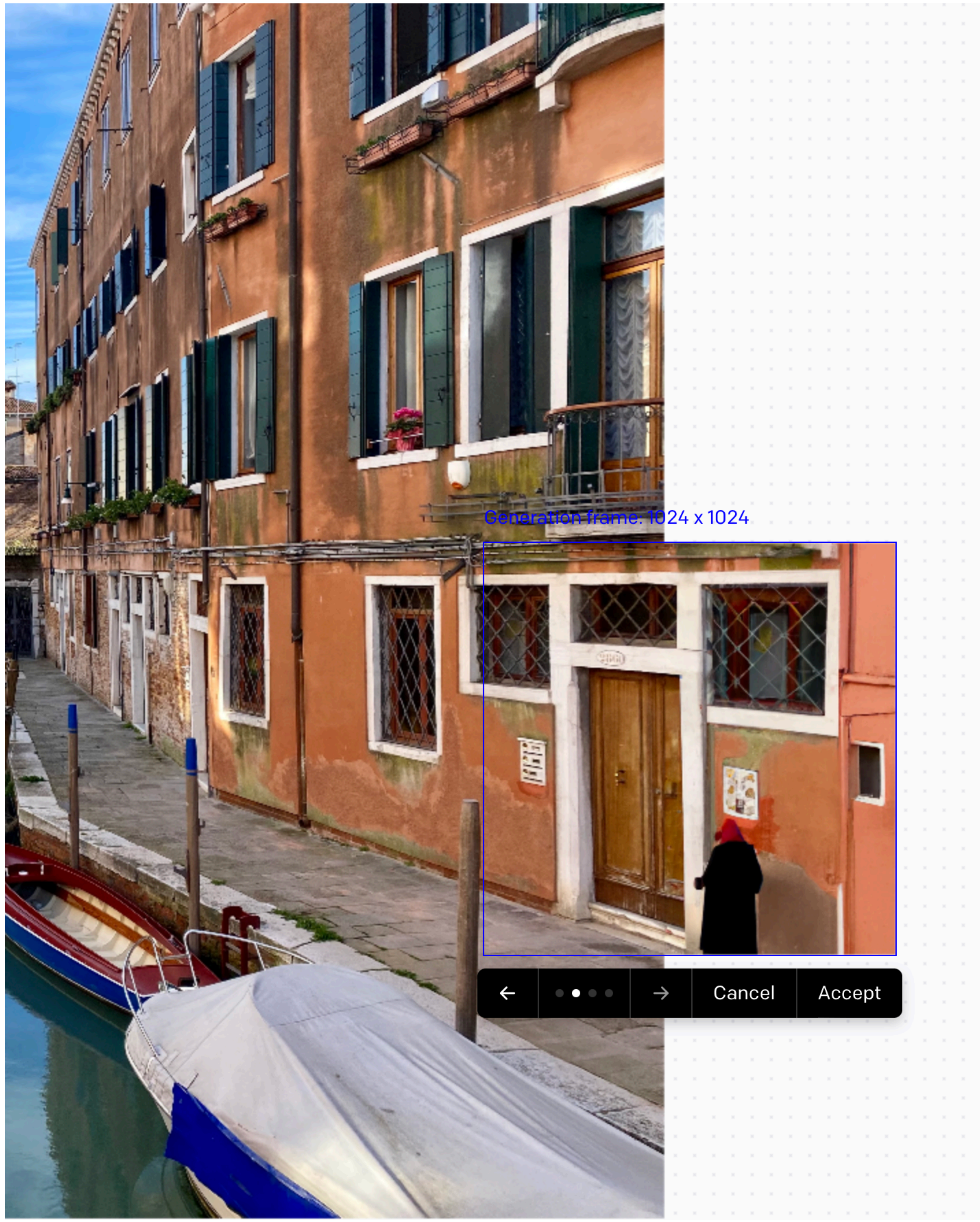
CONDITIONING DENOISING DIFFUSION MODELS



DALL-E 2 from OpenAI, 2022. Try it !

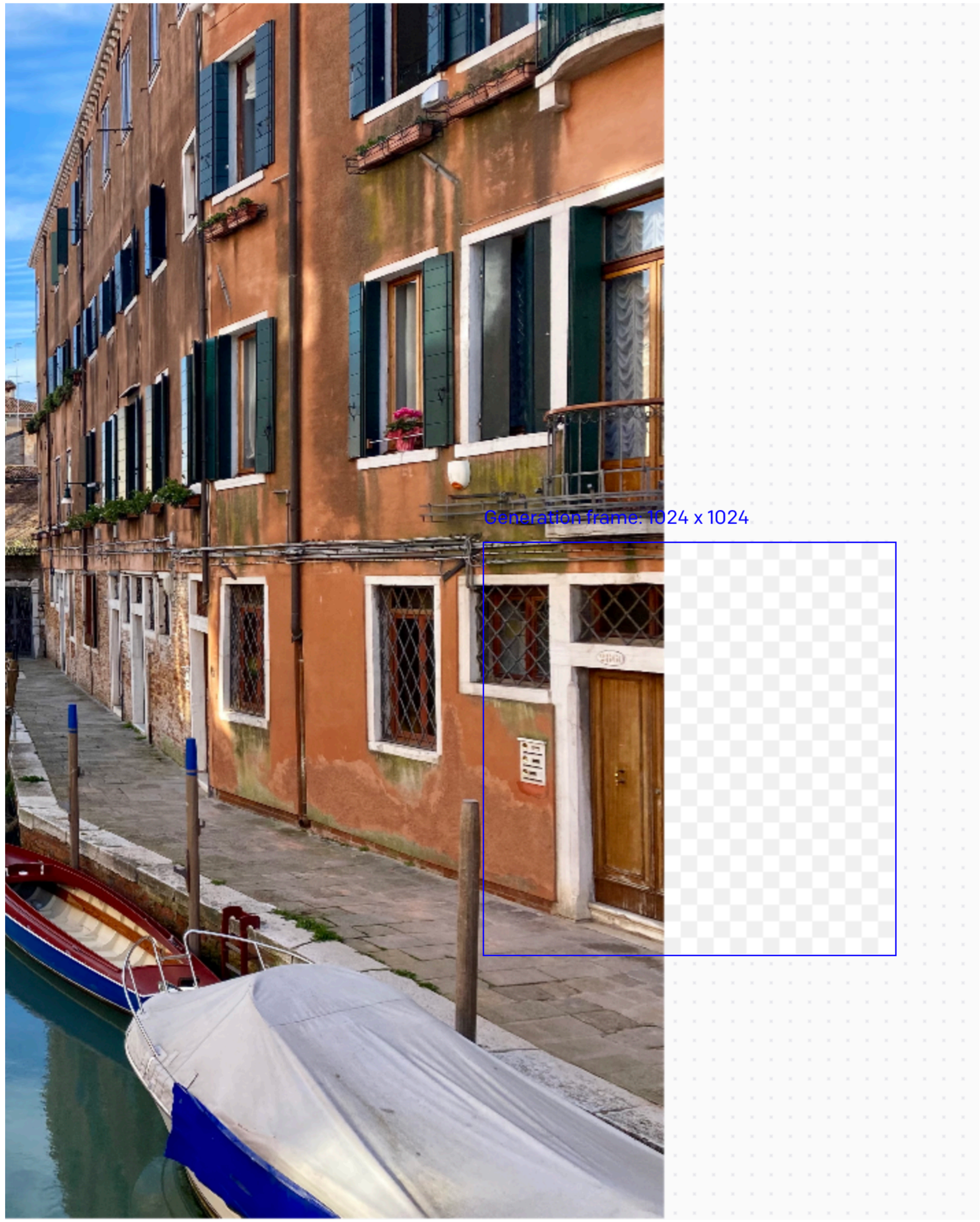
CONDITIONING DENOISING DIFFUSION MODELS

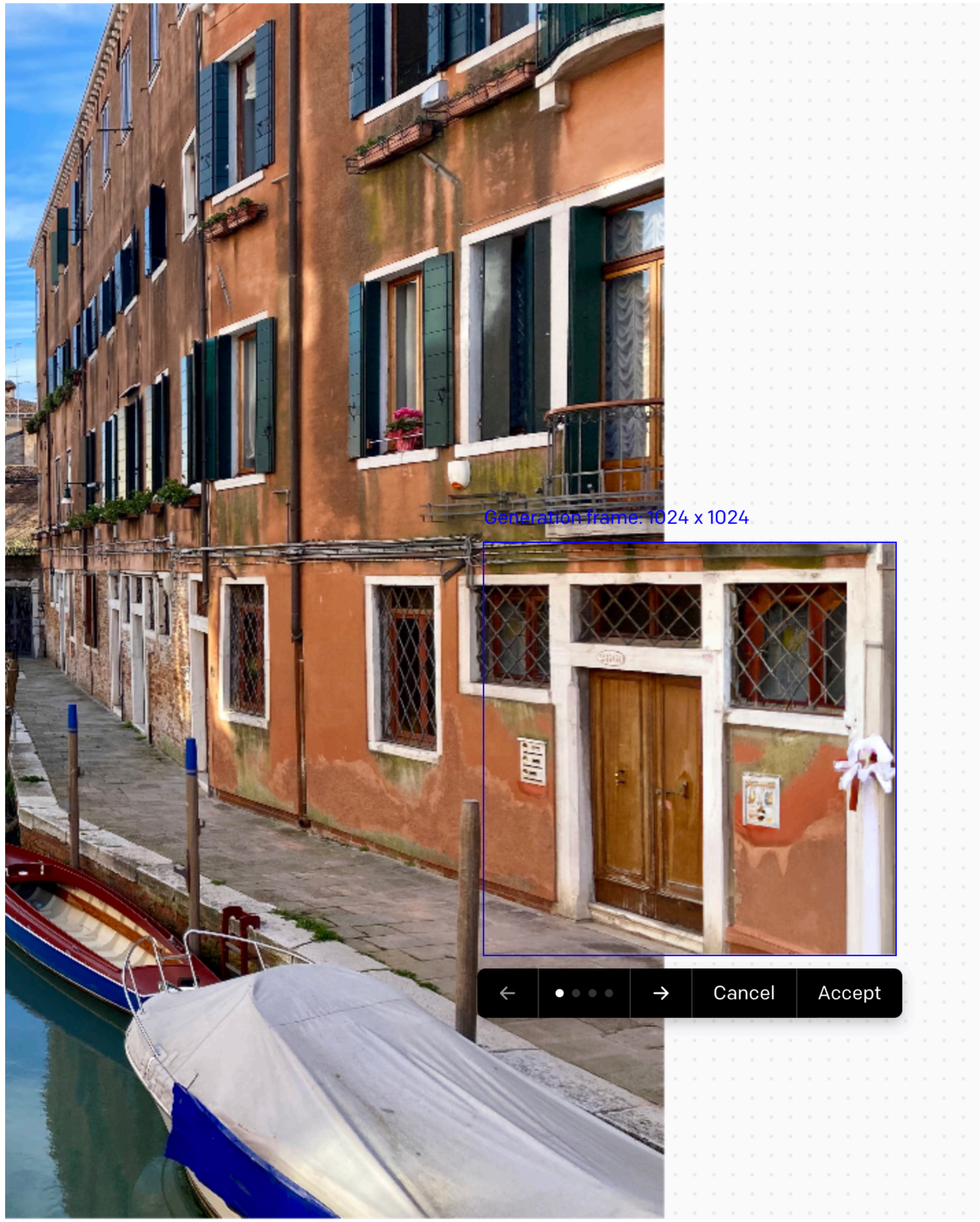




DALL-E 2 from OpenAI, 2022. Try it !

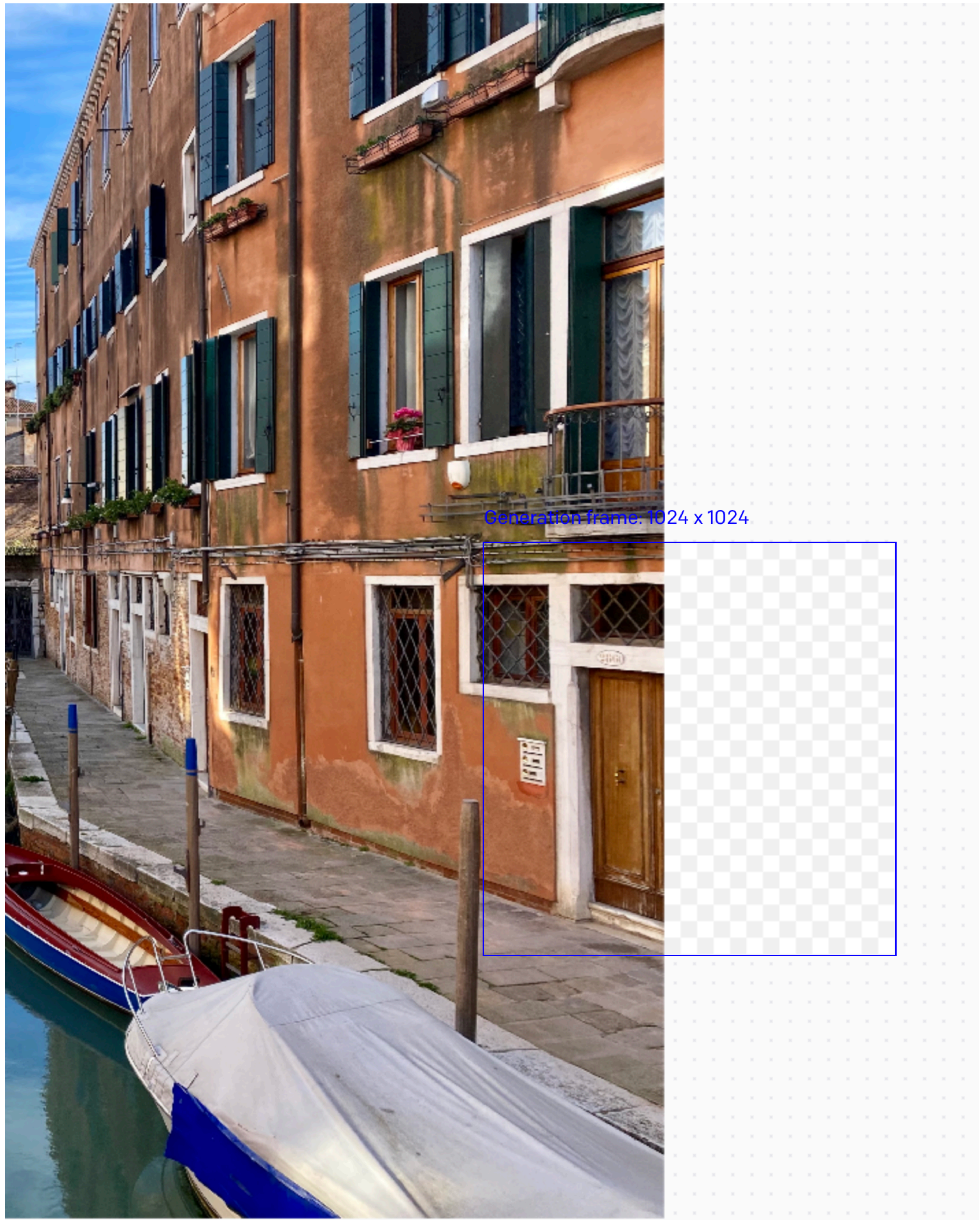
CONDITIONING DENOISING DIFFUSION MODELS

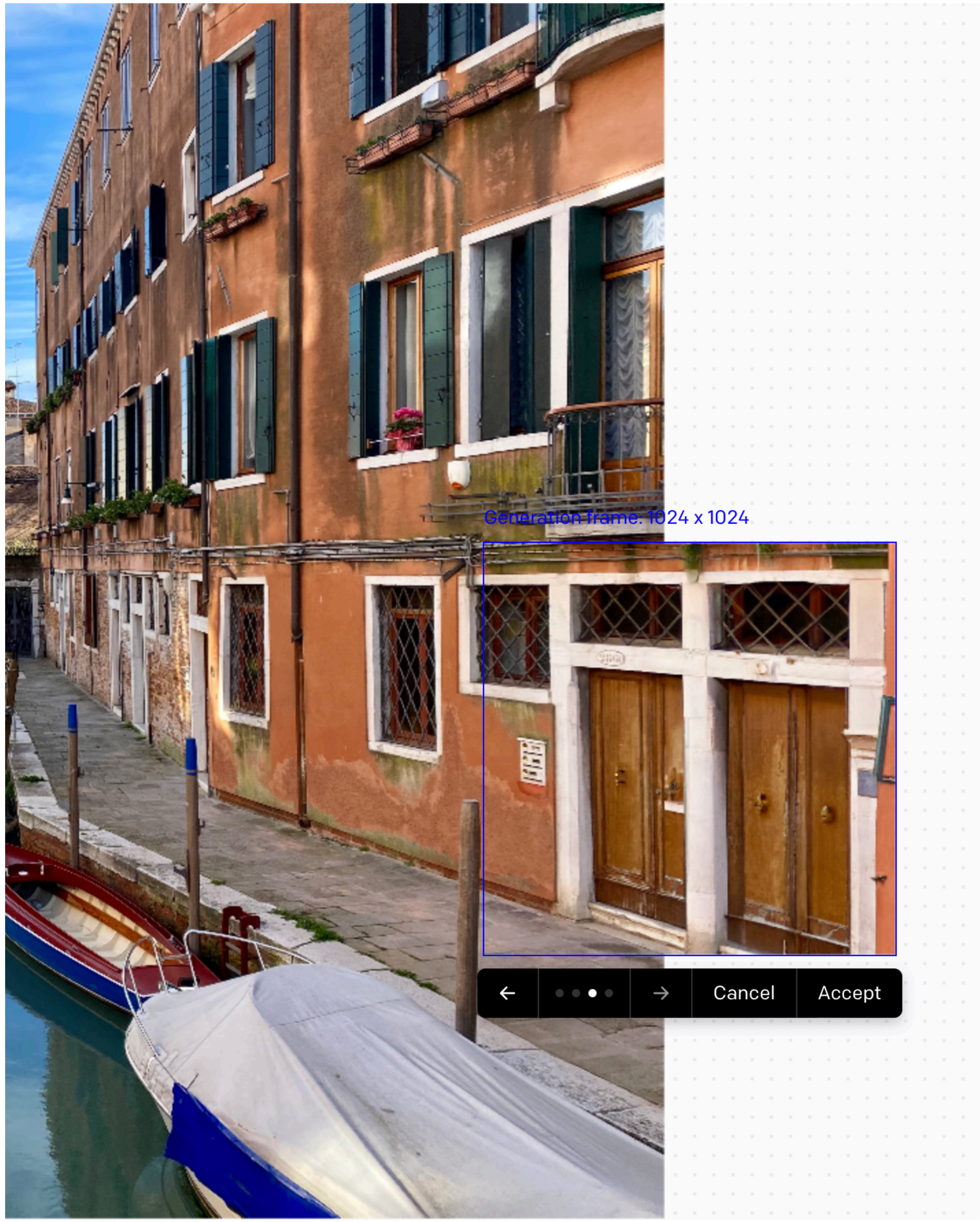




DALL-E 2 from OpenAI, 2022. Try it !

CONDITIONING DENOISING DIFFUSION MODELS





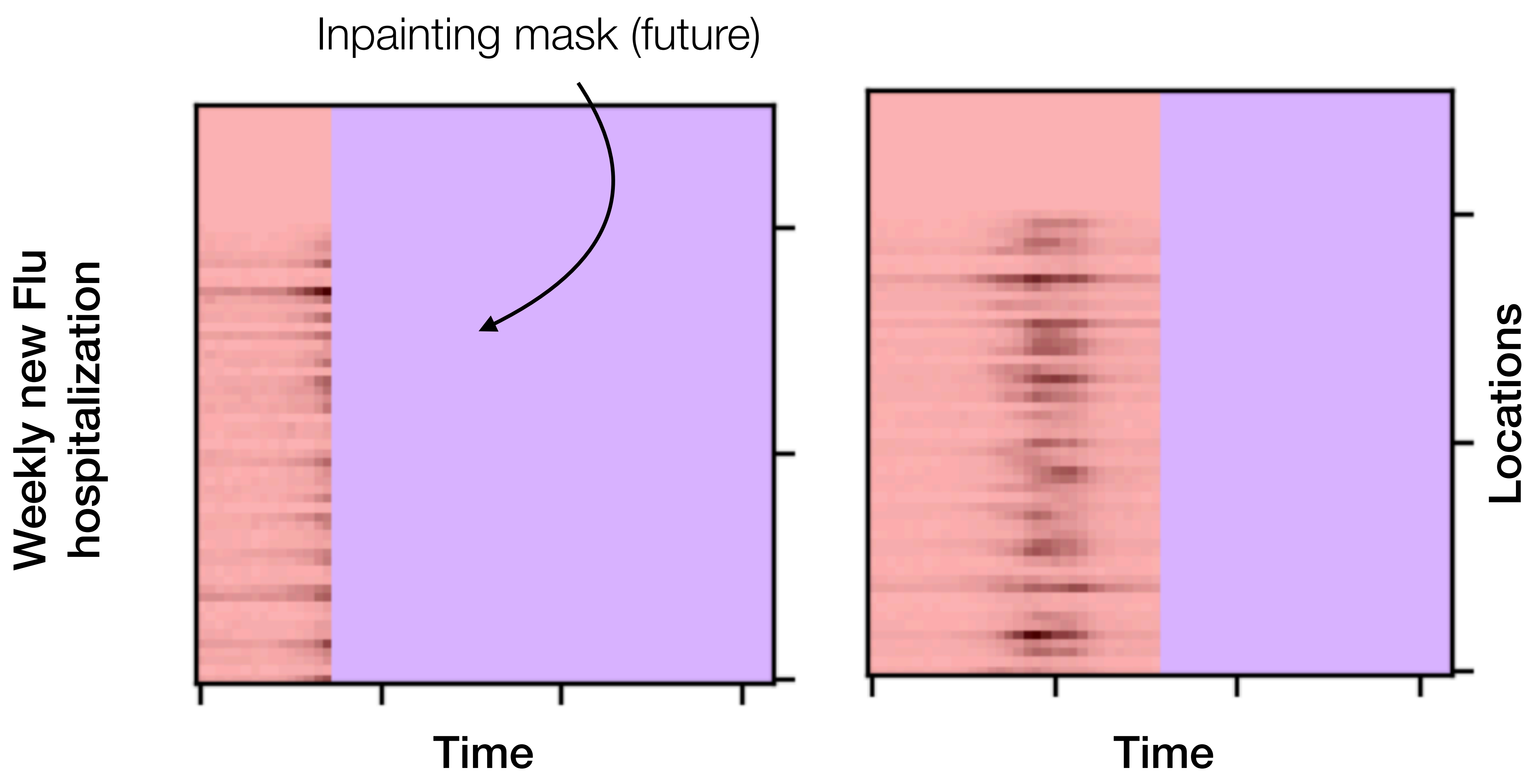
DALL-E 2 from OpenAI, 2022. Try it !

INPAINTING

- Inpainting alter the reverse diffusion by sampling the unmasked regions using ground-truth
- REpaint outperforms SOTA for inpainting
- Parameters: the global harmonization is defined by the resampling schedule
- The mask can have any shape

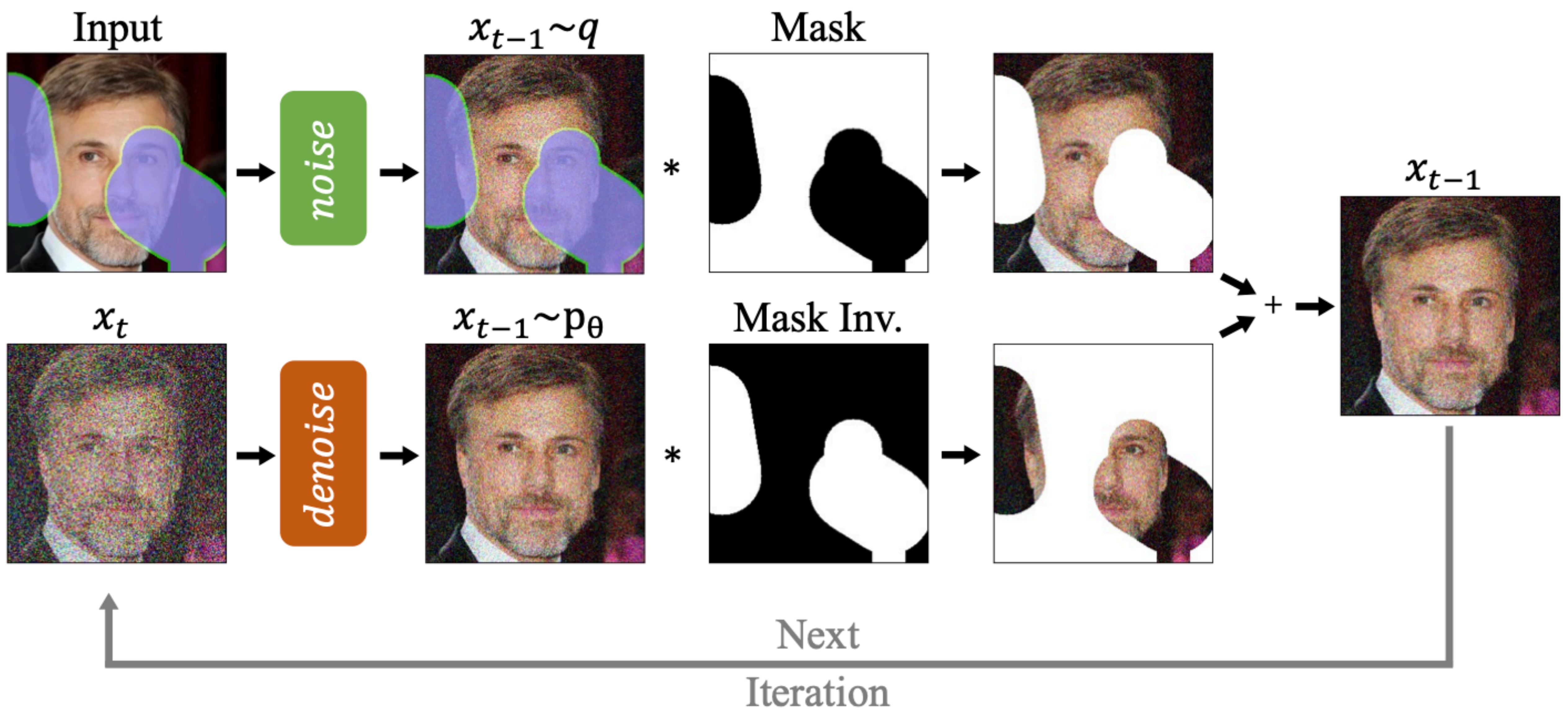
RePaint: Inpainting using Denoising Diffusion Probabilistic Models

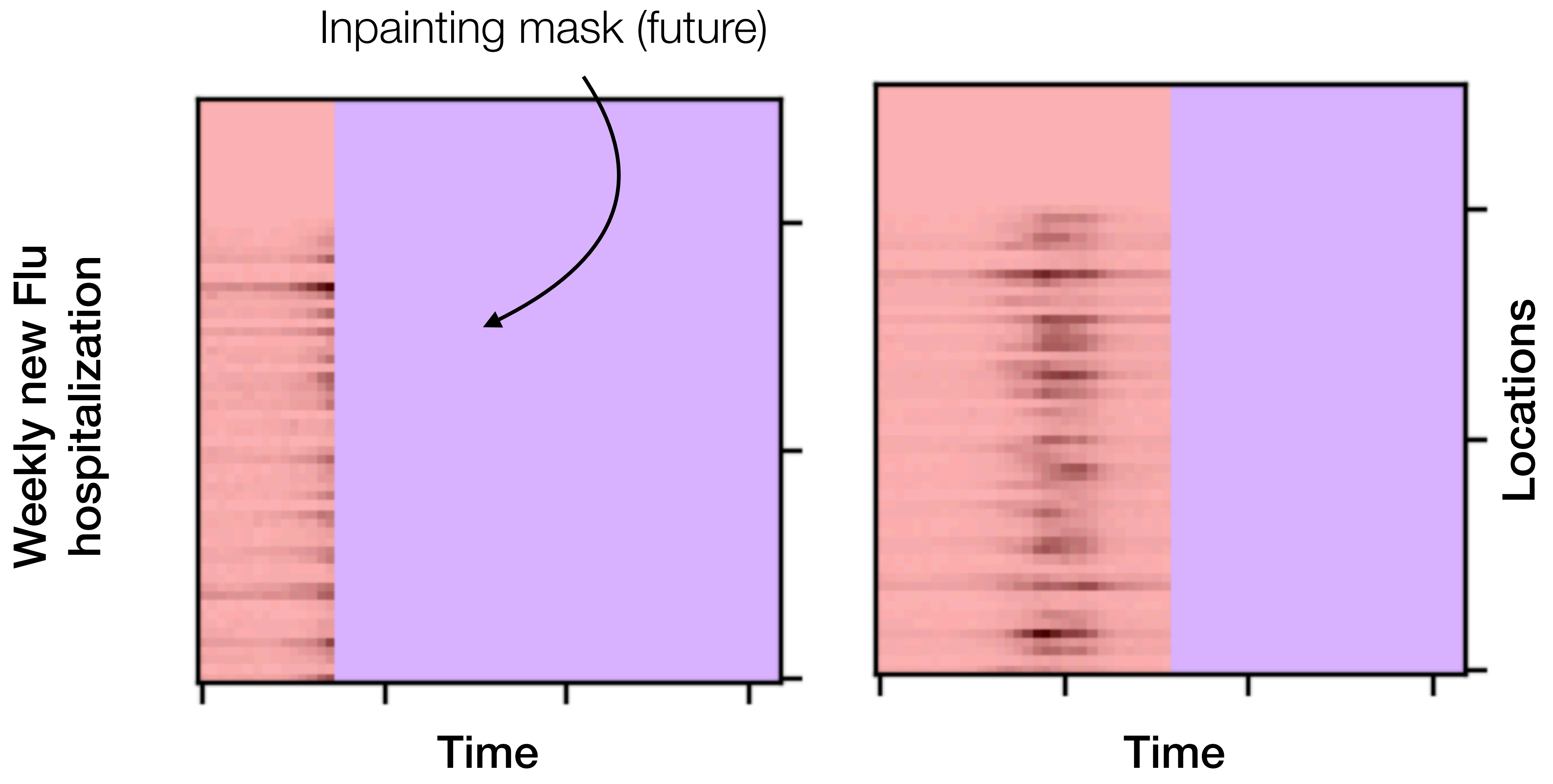
Andreas Lugmayr Martin Danelljan Andres Romero Fisher Yu Radu Timofte Luc Van Gool
Computer Vision Lab
ETH Zürich, Switzerland



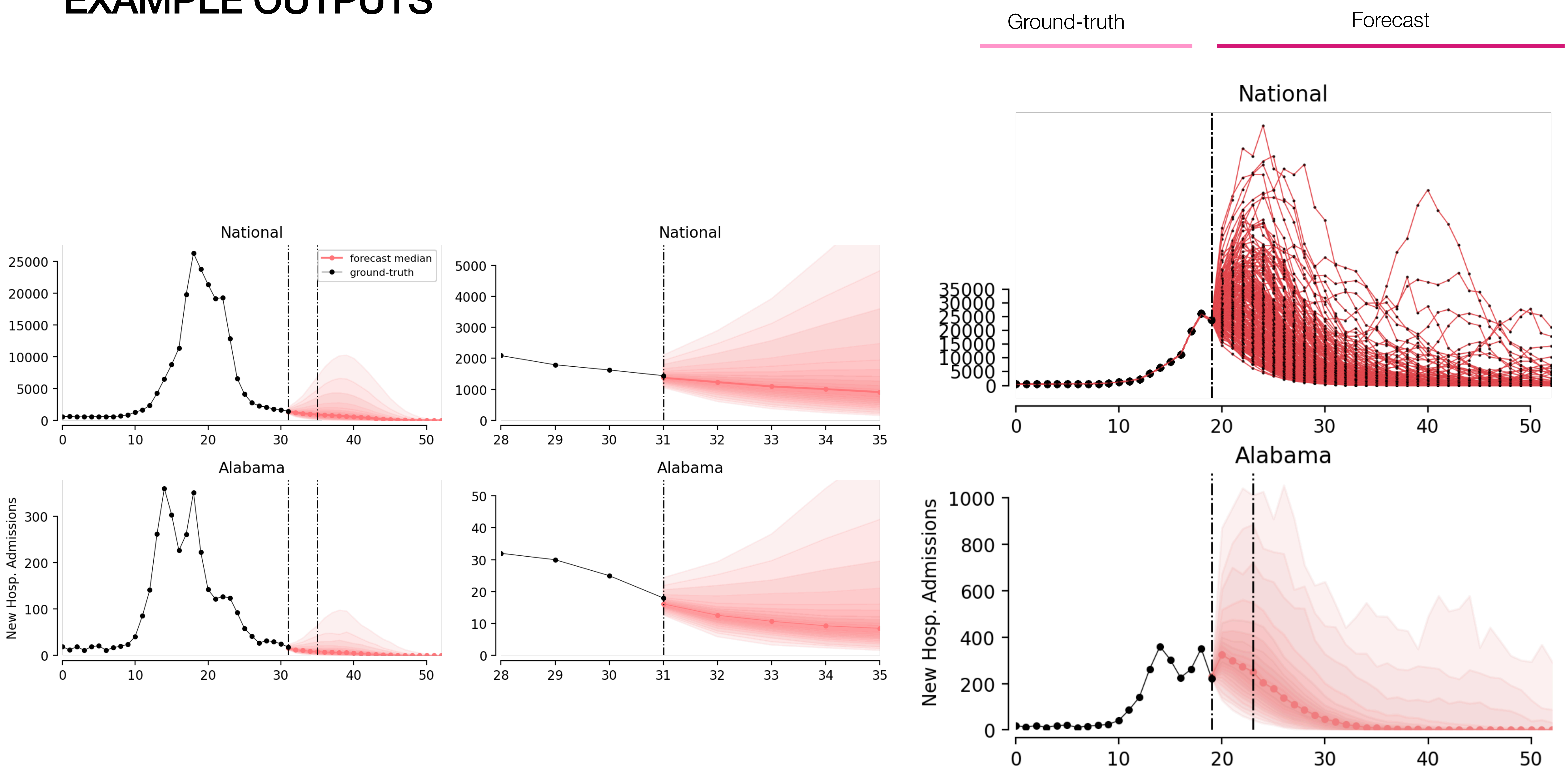
INPAINTING

- **Inpainting** alter the reverse diffusion by sampling the unmasked regions using ground-truth
- **REpaint** outperforms SOTA for inpainting
- Parameters: the **global harmonization** is defined by the resampling schedule
- The mask can have any shape

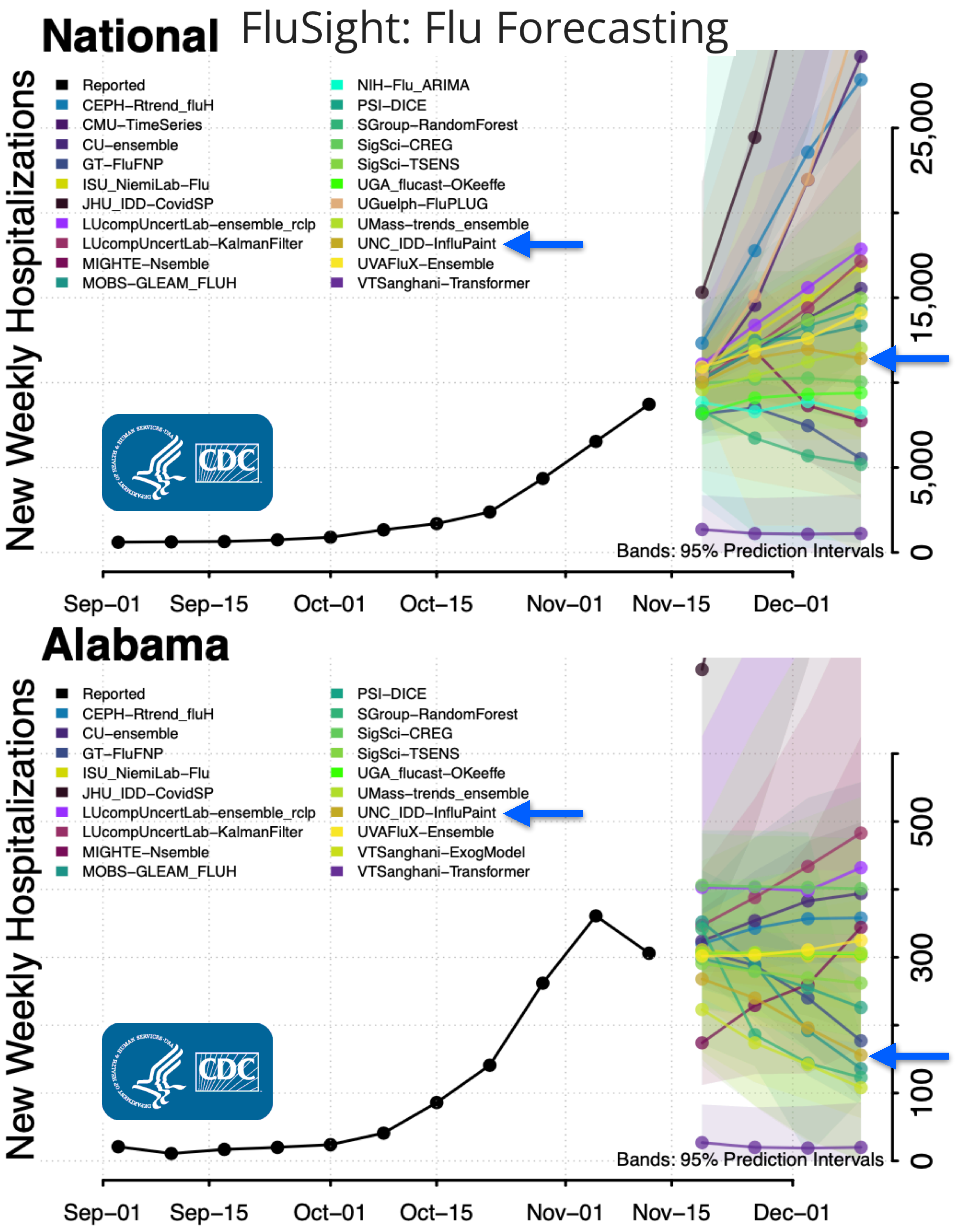




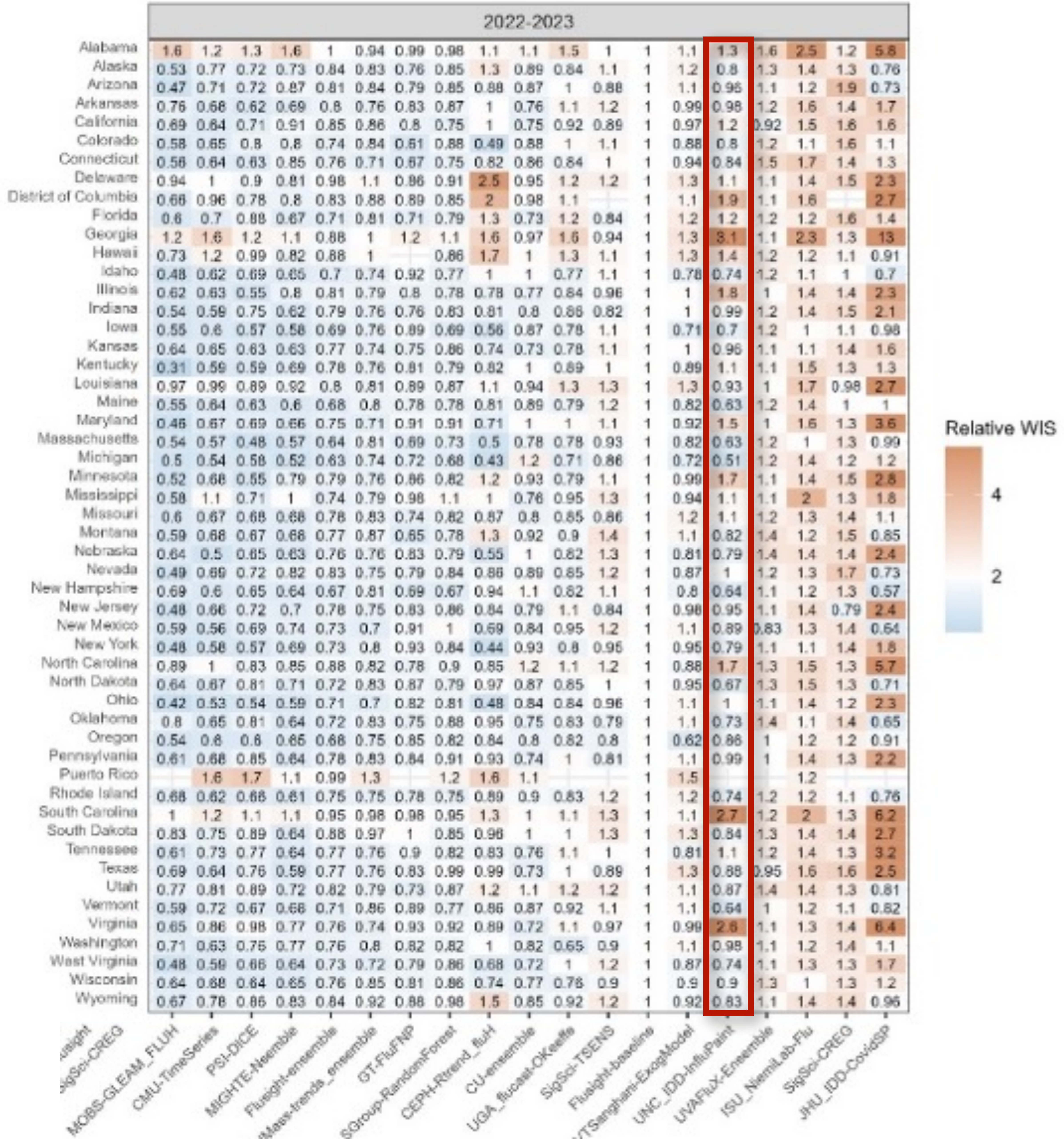
EXAMPLE OUTPUTS



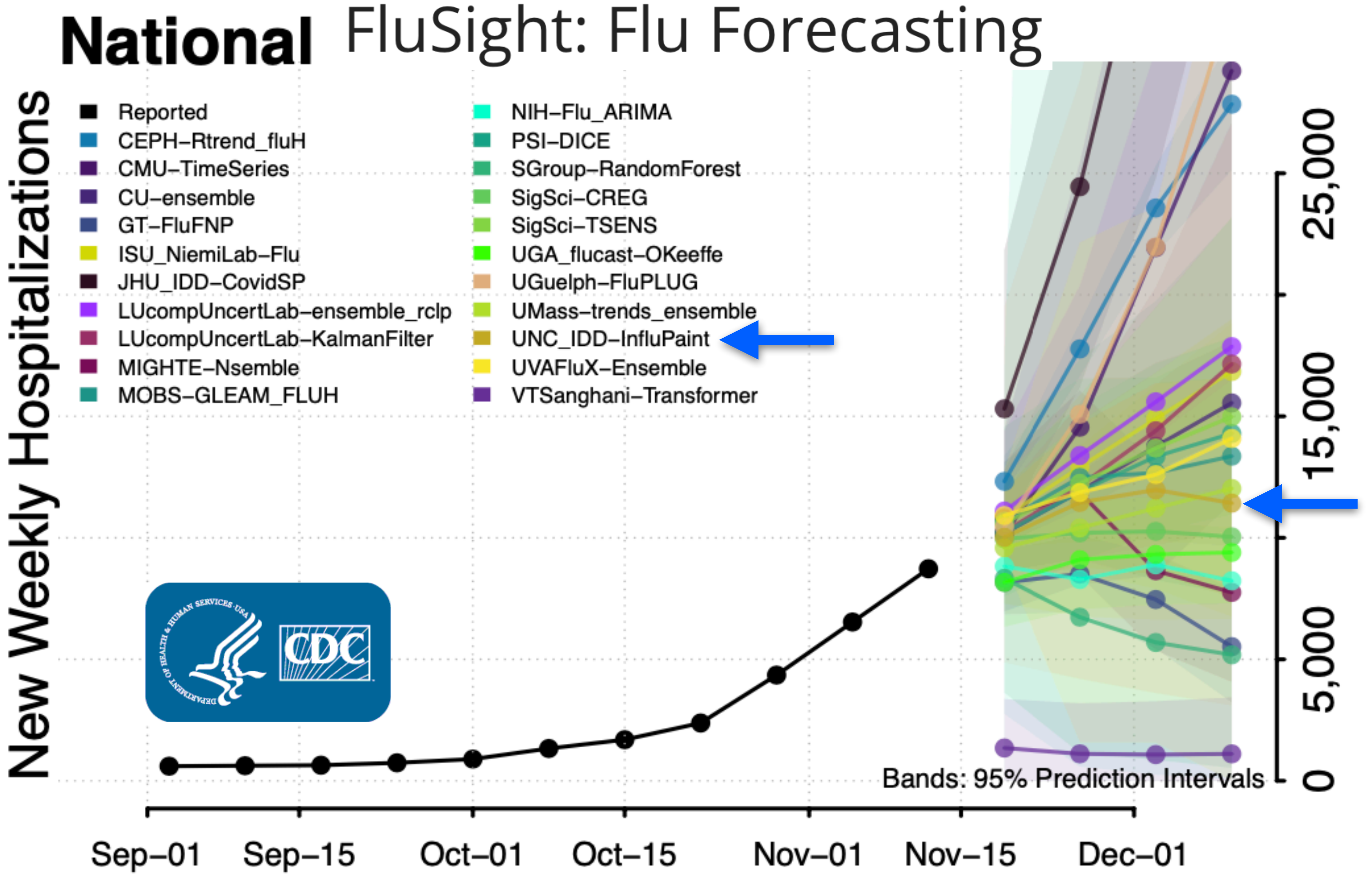
FLUSIGHT FORECASTING CHALLENGE

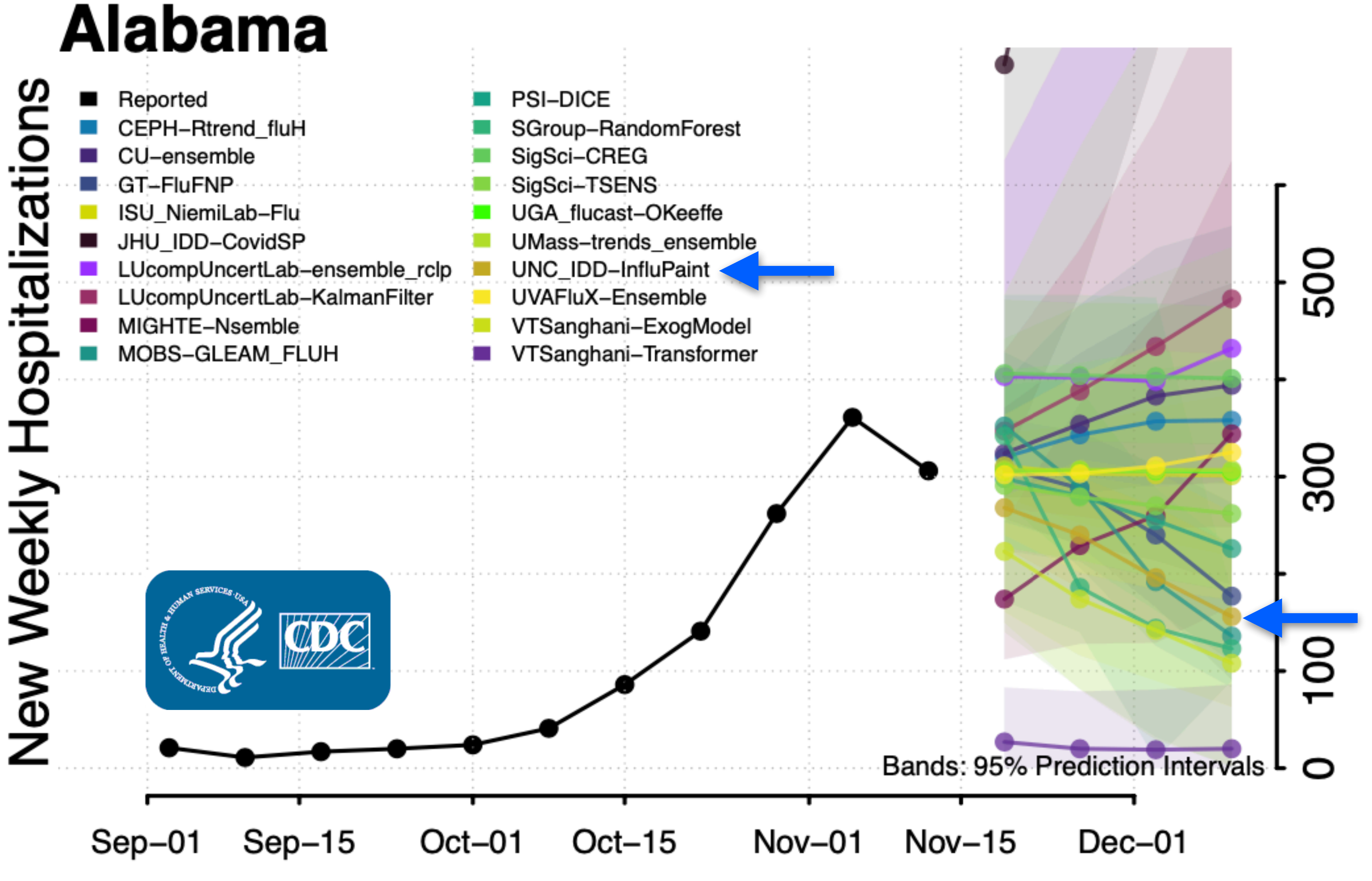


FLUSIGHT FORECASTING CHALLENGE



FluSight: Flu Forecasting





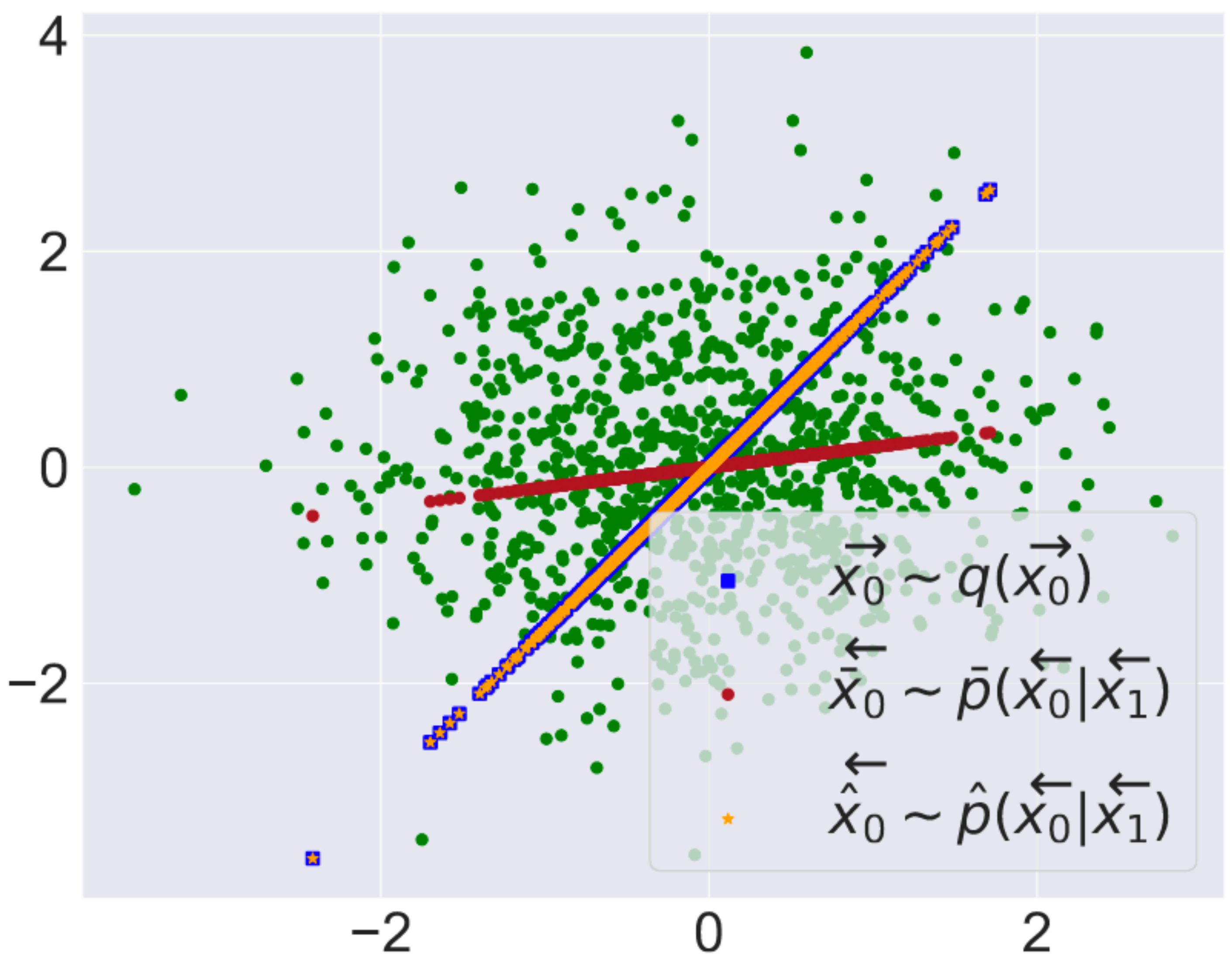
SO WHAT ?

- **Theory**
  - *“You can't just treat forecasts as images, the incidence and time axes have completely different semantics (even if e.g locality is important)”*
  - Some very nice **theoretical properties** — e.g DDPMs as SDEs, proof that *Repaint* generalizes well to unseen masks without retraining
- **Practice**
  - Surprisingly works
  - Good diversity: double peaks and single peaks, little bumps …
  - **Poor forecasting performance**

A Theoretical Justification for Image Inpainting using Denoising Diffusion Probabilistic Models

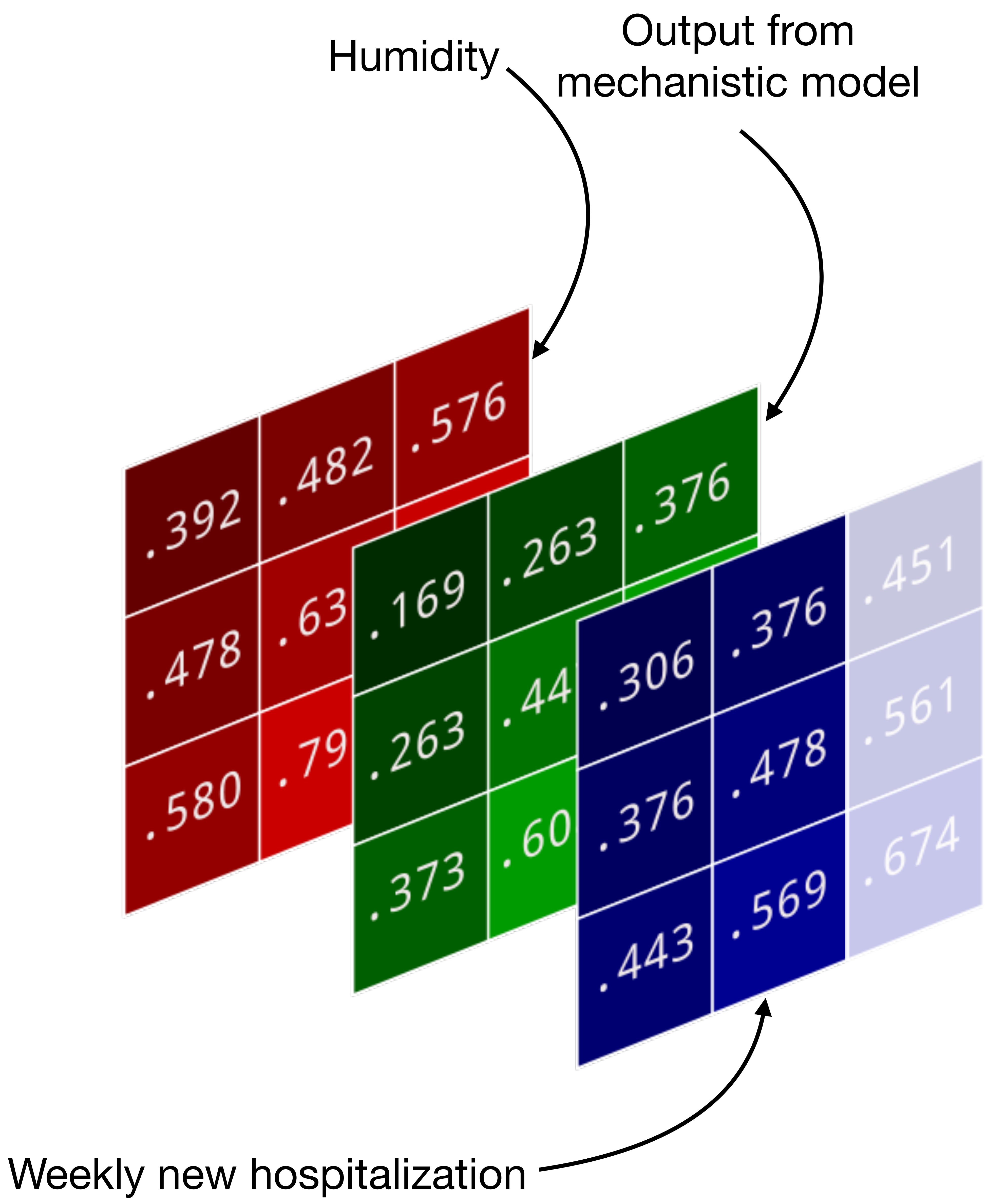
Litu Rout*, Advait Parulekar†, Constantine Caramanis‡ and Sanjay Shakkottai§

The University of Texas at Austin



CURRENT WORK

- Covariates (humidity, Flu strains…) as additional image channels.
  - But also **mechanistic model outputs**, leveraging the masking flexibility
- Broader applicability of knowing p(**x**): imputation, structure discovery, density estimation



THANK YOU FOR YOUR ATTENTION 👀

💻 **Code** github.com/jcblemai/inpainting-idforecasts

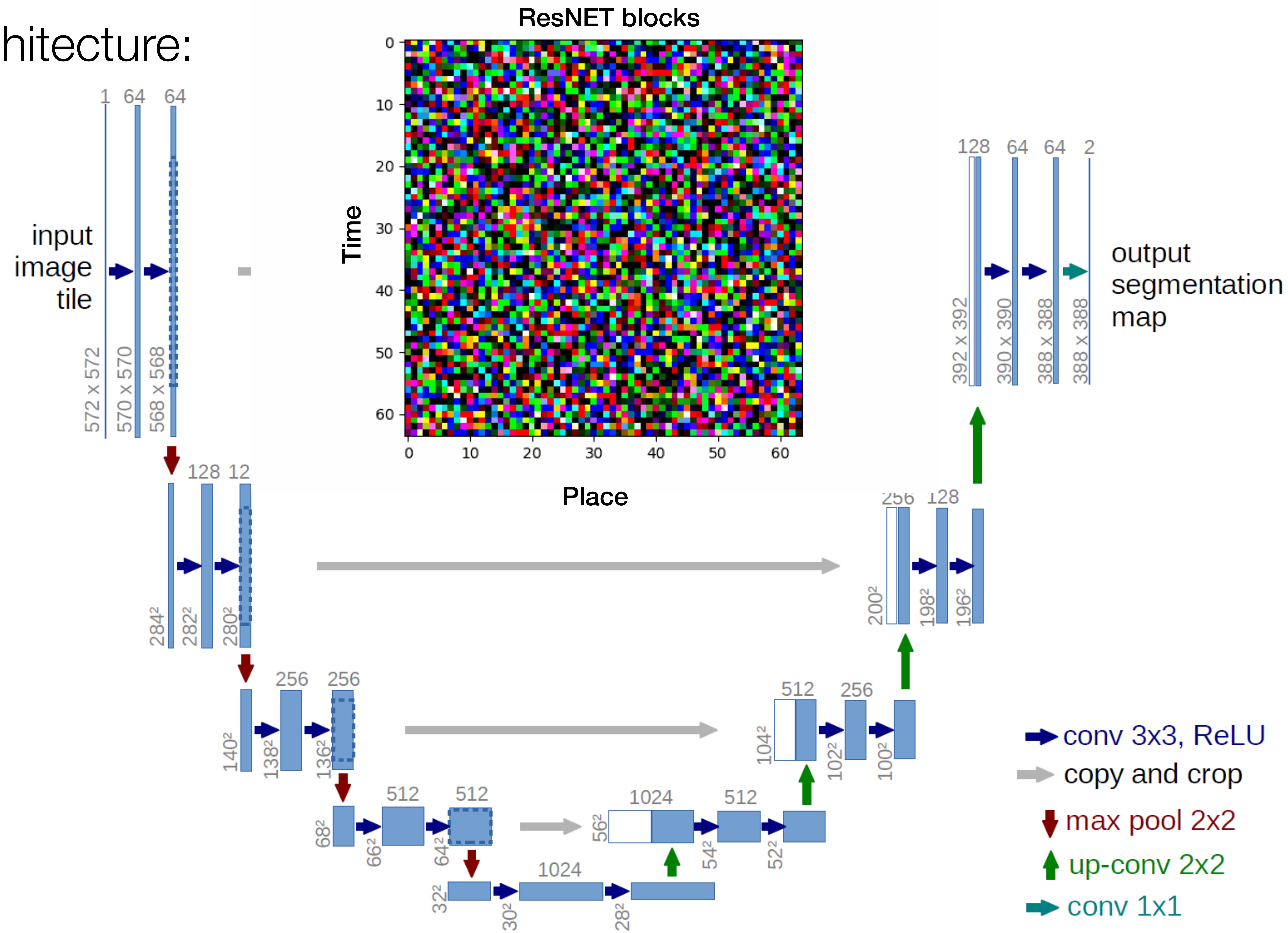
📈 **Forecasts** github.com/cdcepi/Flusight-forecast-data

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

@JosephLemaitre_

A DDPM ARCHITECTURE FOR INFECTIOUS DISEASE

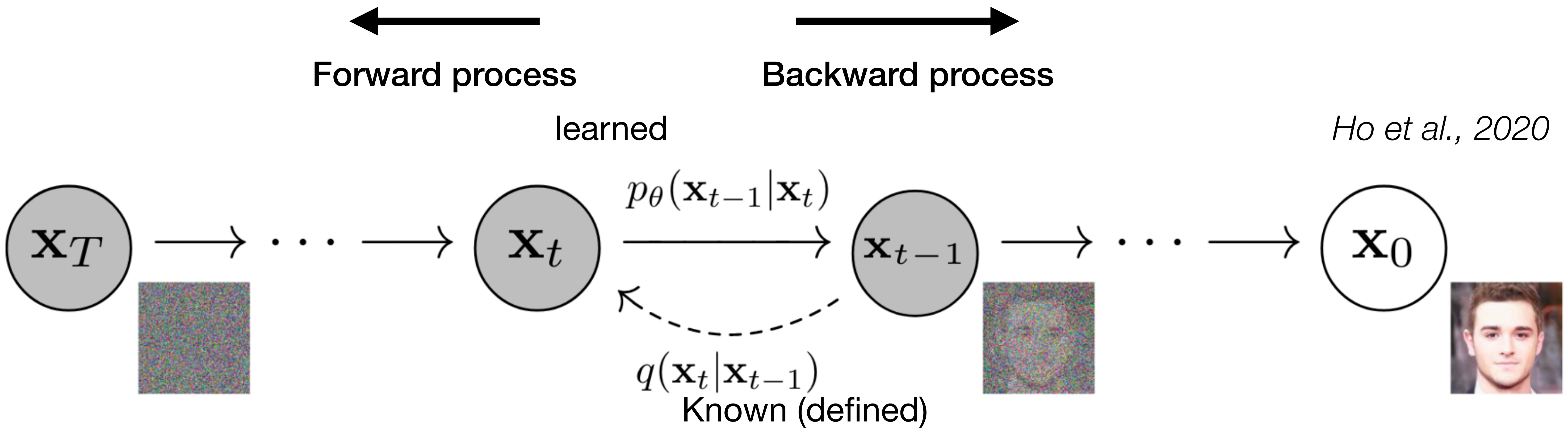
- Backward process network architecture:
  - **U-net** with
    - Wide ResNet block
    - + Attention module
    - + Group normalization
    - + Residual connections
    - + **many tricks** 🪄



DENOISING DIFFUSION PROBABILISTIC MODELS

- DDPM consists of two transforms:
  - **Forward**: Markov chain that gradually adds noise to the data until the signal is destroyed;
  - **Backward**: trained neural network that denoises an image for one step.
- **Training** of the neural net is done on forward-transformed samples from the dataset.
- **Sampling** consists of generating gaussian random noise images and transforming them with the backward transform.
- We stay very close to the literature for image generation.

Ho et al., 2020



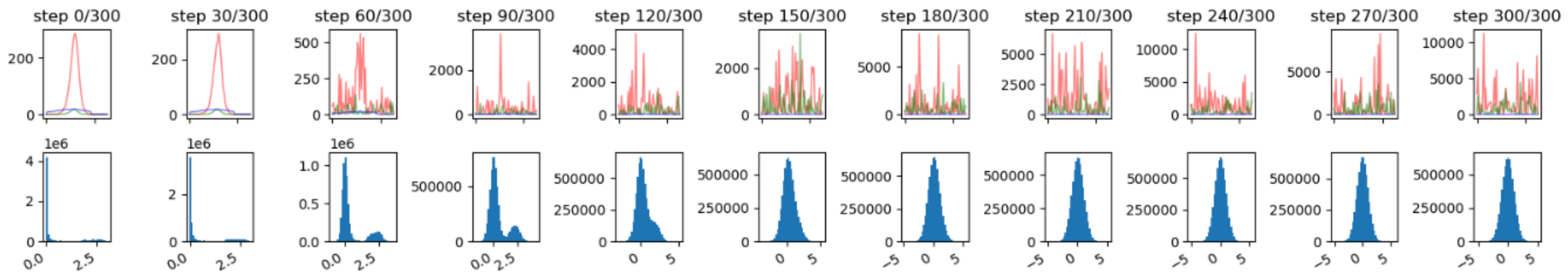
Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
    $\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
until converged

Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$

Example forward diffusion process of an epidemic curve:



DENOISING DIFFUSION PROBABILISTIC MODELS 101

$$q(\mathbf{x_t} \mid \mathbf{x_{t-1}})$$

Pure signal

Pure noise

🚂 Training Algorithm

repeat

$\mathbf{x}_0 \sim q(\mathbf{x}_0)$

$t \sim \mathrm{Uniform}(\{1, \ldots, T\})$

$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Take gradient descent step on

$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$

until converged

✨ Sampling Algorithm

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

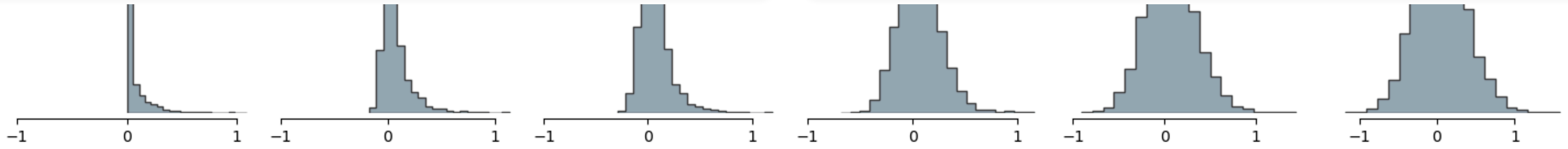
for $t = T, \ldots, 1$ **do**

$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$$

end for

return $\mathbf{x}_0$



$$p_\theta(\mathbf{x_{t-1}} \mid \mathbf{x_t})$$