Uncertainty Quantification for Forecasting Tasks Using Conditional Flow Matching

Emma Hart[1], Cici Wang[2], Ben Erichson[2], Soon Hoe Lim[3], Sherry Li[3]

[1]Emory University, [2]International Computer Science Institute, [3]KTH Institute, [4]Lawrence Berkeley National Laboratory

Scientific Goals

- Extend existing generative AI methods to scientific data and relevant scientific tasks
- Improve training efficiency
- Develop methods for uncertainty quantification

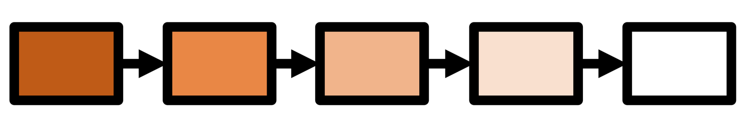
Background: Generative AI

Learn to sample from a complex distribution using a simple one

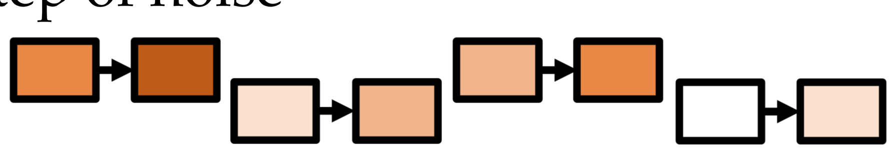
Example: Diffusion Models

Backbone of popular image generation models (e.g., DALL-E, Imagen, Stable Diffusion)

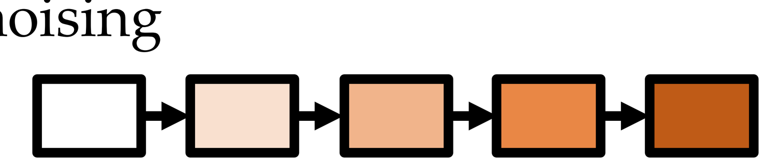
1. Add noise to images in steps until all information is corrupted



2. Train to identify and remove one step of noise



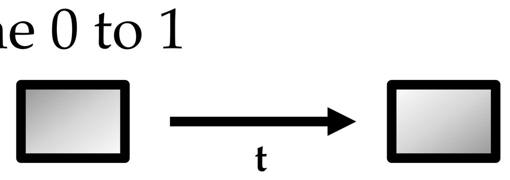
3. Sample by generating pure noise and denoising



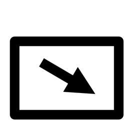
Example: Flow Matching

Easier to train, use continuous normalizing flows, motivated by diffusion models

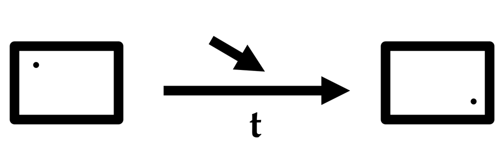
1. Consider distribution transforming from time 0 to 1



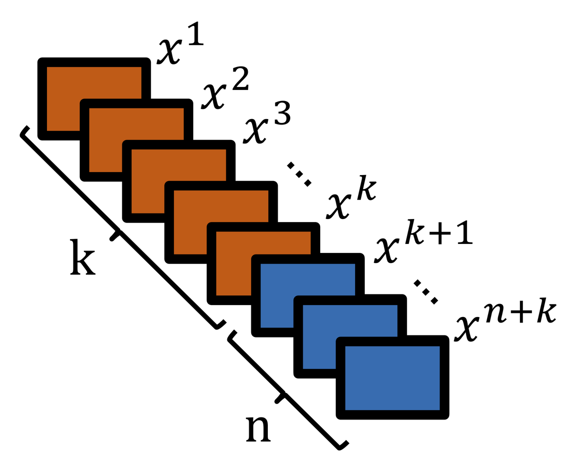
2. Train to learn flow from one distribution to another



3. Sample from initial distribution and propagate the learned flow



Problem: Forecasting



Given k past frames

Predict n next frames

Generative AI frameworks can address forecasting tasks by conditioning generation on past frames to predict next ones

Involves modeling

$$p(x^{k+1}, x^{k+2}, \dots, x^{k+n} | x^1, \dots, x^k) = \prod_{i=1}^{n} p(x^{k+i} | x^1, \dots, x^{k+i-1})$$

Explicit conditioning is

- Computationally expensive
- Highly demanding in memory

Our Methodology

Built on RIVER Flow Matching method from Davtyan et al. [1]

Main Ideas

- Condition generation on random previous frames to *reduce computation* while still *capturing relevant information* from past frames
- Start with noisy previous frame as a *good guess* for next frame
- Work in the *smaller-scale* latent space of an autoencoder to *reduce costs*
- Generate many samples and compare to *quantify uncertainty*

Methodology: Training

Aim to learn vector field in latent space

$$v_t(z): [0,1] \times \mathbb{R}^d \to \mathbb{R}^d, t \in [0,1]$$

such that

$$\dot{\phi}_t(z) = v_t(\phi_t(z)), \ \phi_0(z) = z$$

defines a flow that pushes

$$\underbrace{p_0(z) = \mathcal{N}(z|0,1)}_{\text{white noise}} \quad \text{to} \quad \underbrace{p_1(z) \approx \mathrm{q}(z)}_{\text{unknown data distribution}}$$

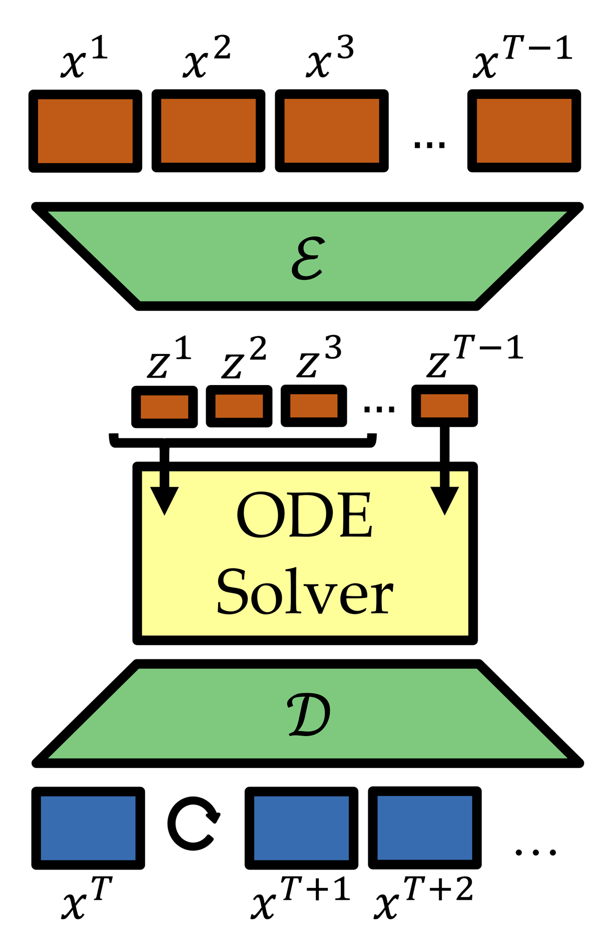
Learn v_t conditioned on *context* and *reference* frames, z^c and $z^{\tau-1}$

Flow matching loss

$$\mathcal{L}_{FM}(\theta) = \| v_t(z|z^{\tau-1}, z^c, \tau - c; \underbrace{\theta}_{\text{network parameters}}) - \overbrace{u_t(z|z^t)}^{\text{target vector field}} \|$$

Methodology: Inference

Encode original sequence $z^1 = \mathcal{E}(x^1), \dots$



Generate initial guess $z_0^T \sim \mathcal{N}(0,1)$

For each integration step:

- Sample *context frame* z^c, $c \sim \mathcal{U}\{1, \dots, T-2\}$
- Step forward following learned vector field $v_t(z|z^{T-1}, z^c, T-c)$

Decode z_1^T for $x^T \approx \mathcal{D}(z_1^T)$

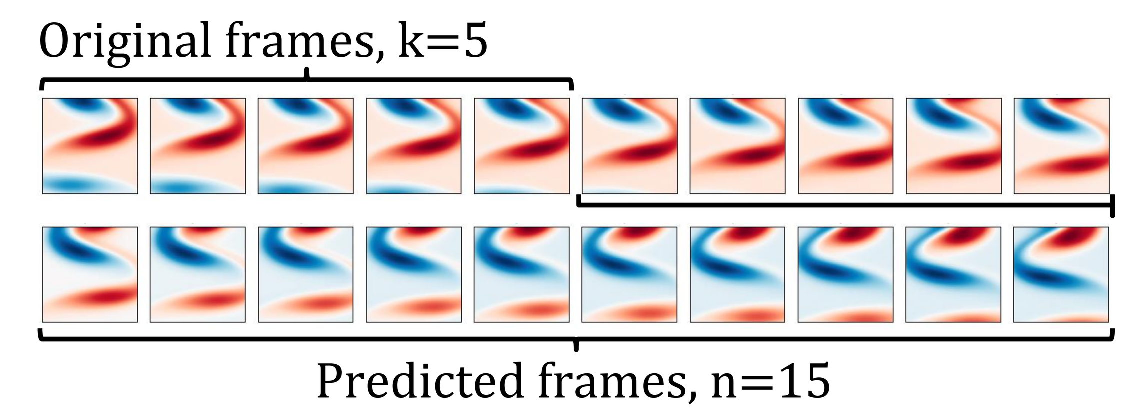
Add predicted x^T to original sequence and repeat process to generate $x^{T+1}, x^{T+2} \dots$

References

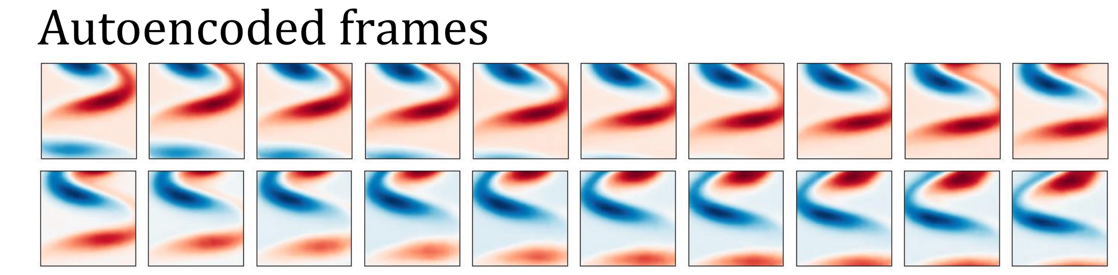
[1] A. Davtyan, S. Sameni and P. Favaro, "Efficient Video Prediction via Sparsely Conditioned Flow Matching," in 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2023 pp. 23206-23217.

Preliminary Results

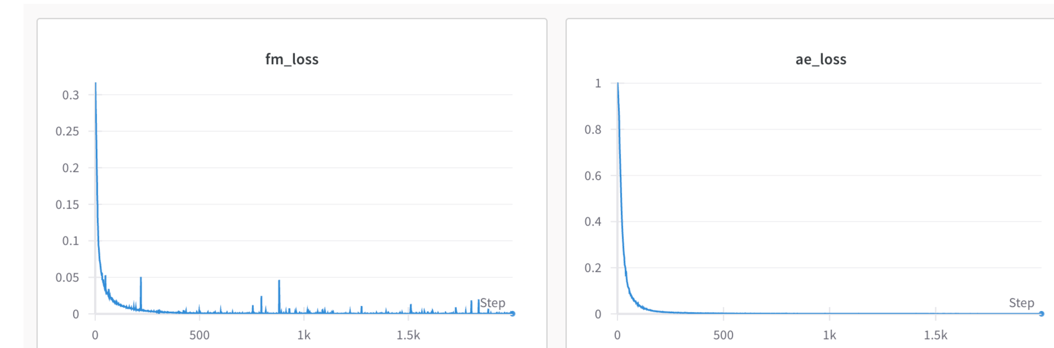
Simple Heat Flow Example:



Autoencoded frames



Losses



Continuing Work

- Implement changes to base model and tune hyperparameters
- Experiment with larger and more complicated scientific data
- Add uncertainty quantification by sampling many times
- Explore alternate probability paths

Acknowledgements

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0024386.