Probabilistic Forecasting

with Stochastic Interpolants and Föllmer Processes

Yifan Chen

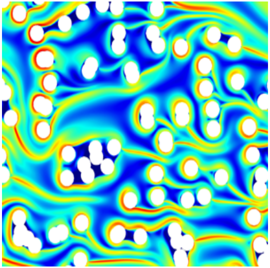
Courant Institute, New York University

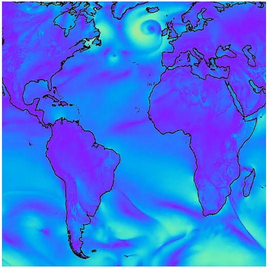
joint with Michael Albergo, Nicholas Boffi, Mark Goldstein, Mengjian Hua, Eric Vanden-Eijnden

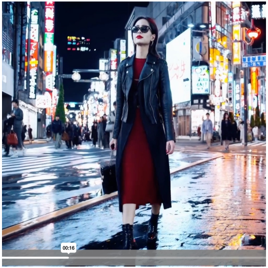
Context

Forecasting Problem

Given time series $(y_{k\tau})_{k\in\mathbb{Z}}$, predict $y_{(k+1)\tau}$ from new $y_{k\tau}$







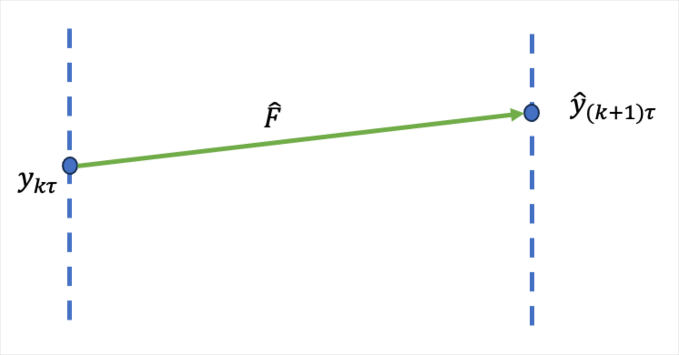
- Examples: fluids, daily weather measurements, video frames
- Assume successive observations $\sim$ joint PDF $\mu(y_{k\tau}, y_{(k+1)\tau})$
- Goal is conditional sampling $y_{(k+1)\tau} \sim \mu(\cdot|y_{k\tau})$

Figure credited to Google online search

Deterministic Forecasting

Goal of Deterministic Forecasting

Output a single forecast by learning a function $\hat{F}$



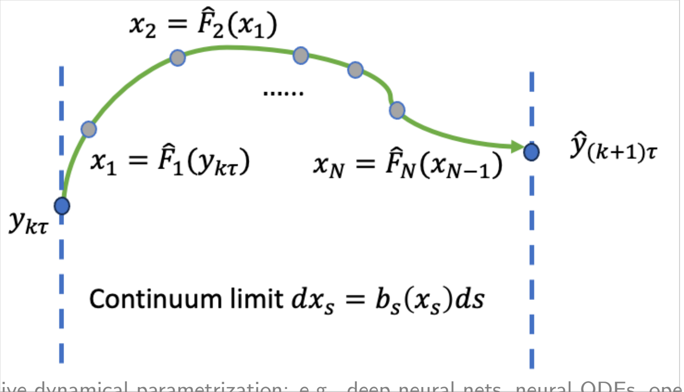
Linear regression, kernel regression, Koopman operator, ...

e.g., [Dellnitz, Junge 1999], [Berry, Giannakis, Harlim 2015], [Kutz, Brunton, Brunton, Proctor 2016], [Alexander, Giannakis 2020], ...

Deterministic Forecasting

Goal of Deterministic Forecasting

Output a single forecast by learning a function $\hat{F}$

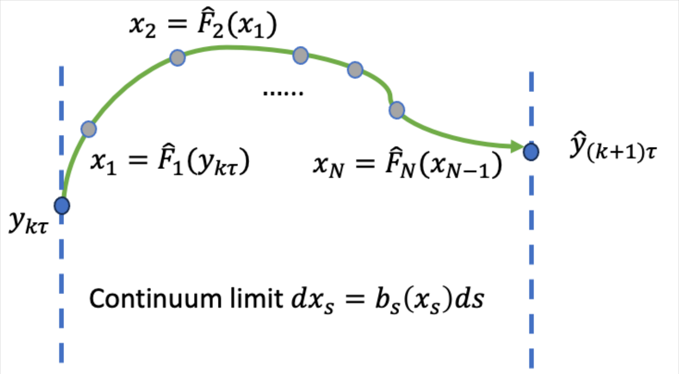


Expressive dynamical parametrization: e.g., deep neural nets, neural ODEs, operators
e.g. [Li et al, 2021], [Jiang, Lu, Orlova, Willett, 2023], ...

Deterministic Forecasting

Goal of Deterministic Forecasting

Output a single forecast by learning a function $\hat{F}$



Expressive dynamical parametrization: e.g., deep neural nets, neural ODEs, operators

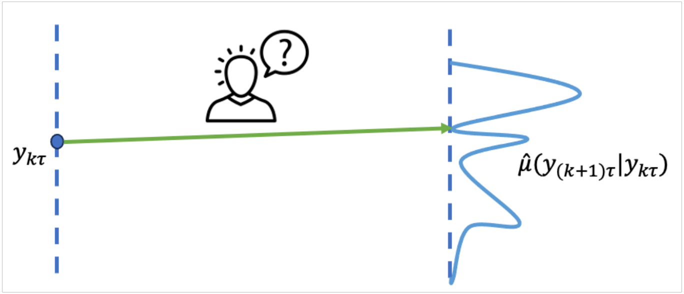
e.g. [Li et al, 2021], [Jiang, Lu, Orlova, Willett, 2023], ...

however deterministic forecast overlooks uncertainties :(

Probabilistic Forecasting

Goal of Probabilistic Forecasting

Output an ensemble of forecasts by learning a distribution

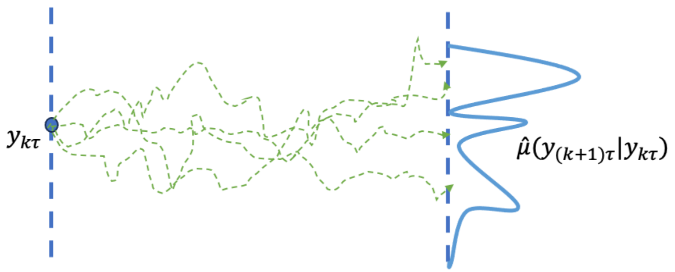


Stochastic Koopman operators e.g., [Wanner, Mezic 2022], [Zhao, Jiang 2023]
Learning SDEs and probabilistic models e.g., Gaussians, neural SDEs, ...

Probabilistic Forecasting

Goal of Probabilistic Forecasting

Output an ensemble of forecasts by learning a distribution



Goal: Learn an SDE that maps a Diracs at $y_{k\tau}$ to $\hat{\mu}(y_{(k+1)\tau}|y_{k\tau})$

Roadmap of This Talk

1. Building the SDE with Stochastic Interpolants
2. Tunnable Diffusions, KL Optimization and Föllmer's Processes
3. Forecasting Stochastic NSE and Videos

Roadmap of This Talk

1. Building the SDE with Stochastic Interpolants
2. Tunnable Diffusions, KL Optimization and Föllmer's Processes
3. Forecasting Stochastic NSE and Videos

Stochastic Interpolants

Let x_0 and x_1 denote the current and forecasting state

Stochastic Interpolants

Define the stochastic process $I_s = \alpha_s x_0 + \beta_s x_1 + \sigma_s W_s$

- $\alpha_0 = \beta_1 = 1$ and $\alpha_1 = \beta_0 = \sigma_1 = 0 \rightsquigarrow I_0 = x_0, I_1 = x_1$
- $(x_0, x_1) \sim \mu(x_0, x_1)$ joint distribution
- $W = (W_s)_{s\in[0,1]}$ is a Wiener process with $W \perp (x_0, x_1)$

- Fact: $dI_s = (\dot{\alpha}_s x_0 + \dot{\beta}_s x_1 + \dot{\sigma}_s W_s)ds + \sigma_s dW_s$
- Define the SDE

$$dX_s = b_s(X_s, x_0)ds + \sigma_s dW_s, \;\; X_{s=0} = x_0$$

  where $b_s(x, x_0) = \mathbb{E}[\dot{\alpha}_s x_0 + \dot{\beta}_s x_1 + \dot{\sigma}_s W_s | I_s = x, x_0]$
- It holds Law(X_s) = Law$(I_s|x_0)$. In particular $X_{s=1} \sim \mu(\cdot|x_0)$

[Albergo, Vanden-Eijnden, 2022], [Albergo, Boffi, Vanden-Eijnden 2023]
See also [Liu, Gong, Liu 2022], [Lipman et al 2022], ...

Learning the Drift via Square Loss Regression

- $I_s = \alpha_s x_0 + \beta_s x_1 + \sigma_s W_s$
- $b_s(x, x_0) = \mathbb{E}[\dot{\alpha}_s x_0 + \dot{\beta}_s x_1 + \dot{\sigma}_s W_s | I_s = x, x_0]$

- Conditional expectation $\rightsquigarrow$ square loss regression
- The drift $b_s(x, x_0)$ is the unique minimizer of

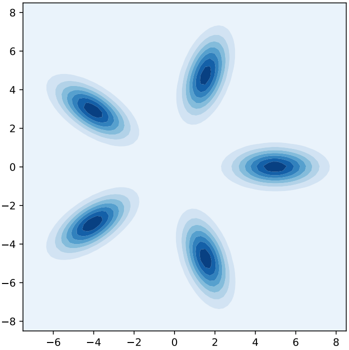
$$L_b[\hat{b}_s] = \int_0^1 \mathbb{E}\big[|\hat{b}_s(I_s, x_0) - \dot{\alpha}_s x_0 - \dot{\beta}_s x_1 - \dot{\sigma}_s W_s|^2\big] ds$$

- Loss function is simulation-free: $W_s \overset{d}{=} \sqrt{s} z$ with $z \sim \mathsf{N}(0, \mathsf{I})$
- Parametrize $\hat{b}_s$ by neural nets and optimize L_b via SGD

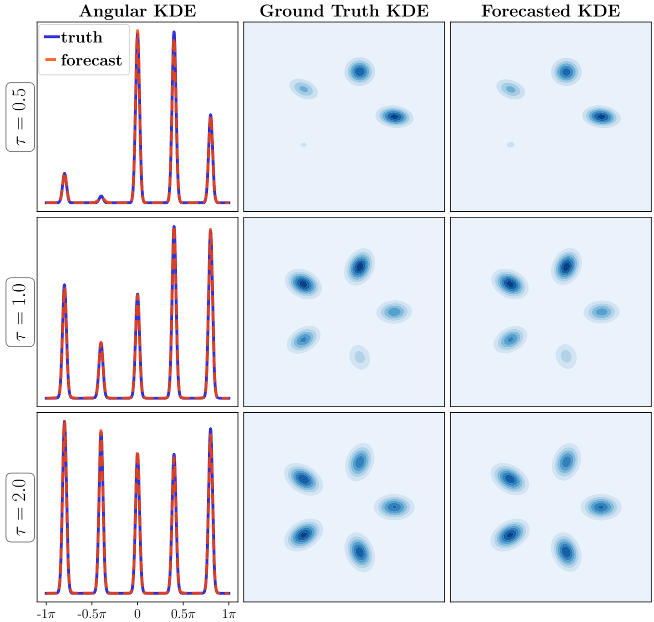
A Synthetic Example: Multimodal Jump Processes

2D particle jump-diffusion dynamics:

- Between the jumps, the particle moves according to the Langevin dynamics $dx_t = \nabla \log \rho_{\mathsf{GMM}}(x_t)dt + \sqrt{2}dW_t$
- At jump times specified by a Poisson process with rate $\lambda = 2$, the particle is rotated counterclockwise by an angle $2\pi/5$



Forecasting A Synthetic Multimodal Jump Processes



Roadmap of This Talk

Tunnable Diffusions for SDEs

Trading drift and diffusion terms

$\nabla \cdot (\rho \nabla \log \rho) = \Delta \rho$ so can trade drift $-\nabla \log \rho$ with diffusion dW

Tunnable Diffusions for SDEs

Trading drift and diffusion terms

$\nabla \cdot (\rho \nabla \log \rho) = \Delta \rho$ so can trade drift $-\nabla \log \rho$ with diffusion dW

- For $dX_s = b_s(X_s, x_0)ds + \sigma_s dW_s$, Law($X_s$) = Law($X_s^g$) where

$$dX_s^g = b_s^g(X_s^g, x_0)ds + g_s dW_s$$

  with $b_s^g(x, x_0) = b_s(x, x_0) + \frac{1}{2}(g_s^2 - \sigma_s^2)\nabla \log \rho_s(x|x_0)$

- $\rho_s(x|x_0)$ is the PDF of $X_s \stackrel{d}{=} I_s|x_0$

Tunnable Diffusions for SDEs

Trading drift and diffusion terms

$\nabla \cdot (\rho \nabla \log \rho) = \Delta \rho$ so can trade drift $-\nabla \log \rho$ with diffusion dW

- For $dX_s = b_s(X_s, x_0)ds + \sigma_s dW_s$, Law($X_s$) = Law($X_s^g$) where

$$dX_s^g = b_s^g(X_s^g, x_0)ds + g_s dW_s$$

  with $b_s^g(x, x_0) = b_s(x, x_0) + \frac{1}{2}(g_s^2 - \sigma_s^2)\nabla \log \rho_s(x|x_0)$

- $\rho_s(x|x_0)$ is the PDF of $X_s \overset{d}{=} I_s|x_0$, with

$$\nabla \log \rho_s(x|x_0) = A_s \left(\beta_s b_s(x, x_0) - c_s(x, x_0)\right)$$

  - $A_s = [s\sigma_s(\dot{\beta}_s\sigma_s - \beta_s\dot{\sigma}_s)]^{-1}$
  - $c_s(x, x_0) = \dot{\beta}_s x + (\beta_s\dot{\alpha}_s - \dot{\beta}_s\alpha_s)x_0$

- A family of SDEs serve for generation purposes

Tunnable Diffusions for SDEs

Trading drift and diffusion terms

$\nabla \cdot (\rho \nabla \log \rho) = \Delta \rho$ so can trade drift $-\nabla \log \rho$ with diffusion dW

- For $dX_s = b_s(X_s, x_0)ds + \sigma_s dW_s$, Law($X_s$) = Law($X_s^g$) where

$$dX_s^g = b_s^g(X_s^g, x_0)ds + g_s dW_s$$

with $b_s^g(x, x_0) = b_s(x, x_0) + \frac{1}{2}(g_s^2 - \sigma_s^2)\nabla \log \rho_s(x|x_0)$

- $\rho_s(x|x_0)$ is the PDF of $X_s \stackrel{d}{=} I_s|x_0$, with

$$\nabla \log \rho_s(x|x_0) = A_s \left(\beta_s b_s(x, x_0) - c_s(x, x_0)\right)$$

  - $A_s = [s\sigma_s(\dot{\beta}_s \sigma_s - \beta_s \dot{\sigma}_s)]^{-1}$
  - $c_s(x, x_0) = \dot{\beta}_s x + (\beta_s \dot{\alpha}_s - \dot{\beta}_s \alpha_s)x_0$

- A family of SDEs serve for generation purposes

We can estimate b first and then adjust both the noise amplitude g_s and the drift b^g *a-posteriori* without having to retrain b

Optimize over g

Question

Any choice of g that is optimal?

Optimize over g

Question

Any choice of g that is optimal?

Criteria: Consider the KL between the path measures of

- the truth SDE solution $X^g = (X_s^g)_{s \in [0,1]}$ with drift b
- the approximation $\hat{X}^g = (\hat{X}_s^g)_{s \in [0,1]}$ with a learned $\hat{b}$

which upper bounds the KL between densities of X_1^g and $\hat{X}_1^g$

Optimize over g

Question

Any choice of g that is optimal?

Criteria: Consider the KL between the path measures of

- the truth SDE solution $X^g = (X_s^g)_{s\in[0,1]}$ with drift b
- the approximation $\hat{X}^g = (\hat{X}_s^g)_{s\in[0,1]}$ with a learned $\hat{b}$

which upper bounds the KL between densities of X_1^g and $\hat{X}_1^g$

Formula: by Girsanov's theorem

$$\mathsf{KL}(X^g||\hat{X}^g) = \int_0^1 \frac{|1+\frac{1}{2}\beta_s A_s(g_s^2-\sigma_s^2)|^2}{2|g_s|^2} L_s ds$$

where $L_s = \mathbb{E}^{x_0}\left[|\hat{b}_s(I_s,x_0) - b_s(I_s,x_0)|^2\right]$

Optimize over g

Question

Any choice of g that is optimal?

Criteria: Consider the KL between the path measures of
- the truth SDE solution $X^g = (X^g_s)_{s\in[0,1]}$ with drift b
- the approximation $\hat{X}^g = (\hat{X}^g_s)_{s\in[0,1]}$ with a learned $\hat{b}$

which upper bounds the KL between densities of X^g_1 and $\hat{X}^g_1$

Formula: by Girsanov's theorem

$$\mathsf{KL}(X^g \| \hat{X}^g) = \int_0^1 \frac{|1 + \frac{1}{2}\beta_s A_s (g_s^2 - \sigma_s^2)|^2}{2|g_s|^2} L_s ds$$

where $L_s = \mathbb{E}^{x_0}\left[|\hat{b}_s(I_s, x_0) - b_s(I_s, x_0)|^2\right]$

Claim: KL is minimized if we set $g_s = g_s^{\mathsf{F}}$ with

$$g_s^{\mathsf{F}} = \left| 2s\sigma_s^2 \frac{d}{ds} \log \frac{\beta_s}{\sqrt{s}\sigma_s} \right|^{1/2}$$

Föllmer's Processes

Theorem

If $\beta_s/(\sqrt{s}\sigma_s)$ is non-decreasing, then $X^{g^{\mathsf{F}}}$ is an **Föllmer process**

- **Föllmer processes** solve the Schrödinger bridge problem when one endpoint is a point mass, offering an entropy-regularized solution to optimal transport
- Usually defined by minimizing KL against the Wiener process subject to constraints on the endpoints
- Our result offers a generalization and new interpretation of Föllmer as the minimizer of the KL of the exact forecasting process from the estimated one, which is more tailored to statistical inference

Föllmer process [Föllmer, 1986] wide applications
In functional inequality [Lehec 2013], [Eldan, Lehec, Shenfeld 2020], ...
In sampling: [Zhang, Chen 2021], [Wang, Jiao, Xu, Wang, Yang 2021], [Huang et al, 2021], [Vargas et al, 2023], [Liu et al, 2023], ...

Other Design Considerations

Behavior of Drift at $s = 0$

Assume the density of $\mu(\cdot|x_0)$ is upper bounded by an exponential tailed density, and $\sigma_0 > 0$, then $\dot{\beta}_0 = 0$ is the sufficient and necessary condition for $\lim_{s\to 0} |b_s(x, x_0)| < \infty$, for any x, x_0

- When $\dot{\beta}_0 = 0$, $\lim_{s\to 0} |\nabla b_s(x, x_0)| < \infty$ as well
- Thus $\dot{\beta}_0 = 0$ can be beneficial for the Lipschitz bound of b
- Practical significance: $\beta_s = s^2$ lead to more stable training than $\beta_s = s$
- We take $\beta_s = s^2$ throughout our experiments

Roadmap of This Talk

Forecasting 2D Stochastically Forced Navier Stokes

2d NSE with Stochastic Forcing

$$\mathrm{d}\omega + v \cdot \nabla\omega \mathrm{d}t = \nu\Delta\omega \mathrm{d}t - \alpha\omega \mathrm{d}t + \epsilon \mathrm{d}\eta \quad \text{on} \quad \mathbb{T}^2$$

- $v = \nabla^{\perp}\psi = (-\partial_y\psi, \partial_x\psi)$ is the velocity
- ψ is the stream function, solution to $-\Delta\psi = \omega$
- $\mathrm{d}\eta$ is white-in-time random forcing on a few Fourier modes
- $\nu = 10^{-3}, \alpha = 0.1, \epsilon = 1$
- Ergodicity shown in [Hairer, Mattingly, 2006]

Goal: Forecast $\omega_{t+\tau}$ from ω_t under stationarity

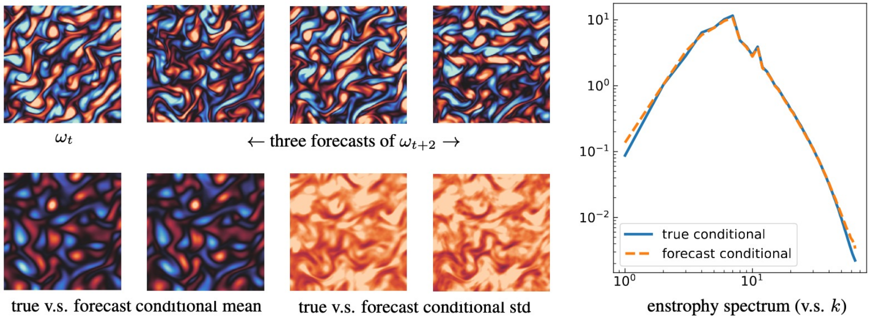


Figure: Probabilistic forecasting with lag $\tau = 2$ (autocorrelation 10%). Resolution 128×128, using $200K$ data pairs for training 2M-parameter-Unet for 50 epochs

- Necessity of probabilistic over deterministic forecasting
- **Forecasting efficiency**: for this example 100 times faster than running the PDE simulation

Effects of Tuning g

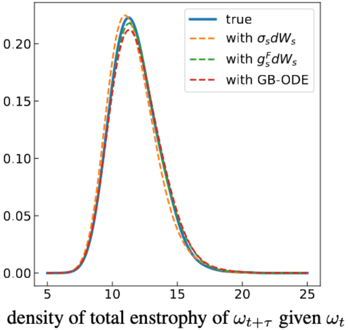


Figure: The 1D conditional distributions of total enstrophy and total energy of $\omega_{t+\tau}$, given a fixed initial vorticity field ω_t and $\tau = 1$. Here we compare between the truth, generated samples via SDEs with $\sigma_s dW_s$, via SDE with $g_s^{\mathrm{F}} dW_s$ which corresponds to a Föllmer process, and via ODEs with Gaussian bases a.k.a. conditional flow matching

Forecasting with Incomplete Observation

Let ω_t be of 32×32 while $\omega_{t+\tau}$ is of 128×128

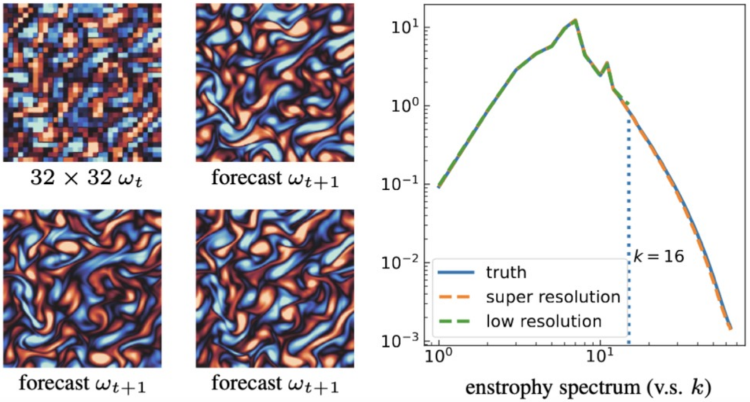


Figure: Probabilistic forecasting with low resolution input, using $200K$ data pairs for training 2M-parameter-Unet for 50 epochs

Forecasting Videos: CLEVER Datasets

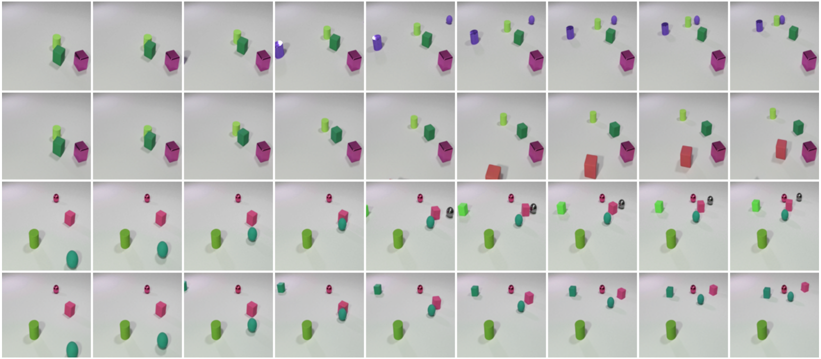


Figure: **Top row:** Real trajectory. **Second row:** Generated trajectory. A new, red cube enters the scene. **Third row:** Real trajectory. **Fourth row:** Generated trajectory. A new green cube enters the scene, and collision physics is respected (green ball hits red cube).

Quantitative Results

	KTH		*CLEVRER*	
Method	100k	250k	100k	250k
RIVER	46.69	41.88	60.40	48.96
PFI (ours)	**44.38**	**39.13**	**54.7**	**39.31**
Auto-enc.	33.45	33.45	2.79	2.79

Table: FVD computed on 256 test set videos, with the model generating 100 completions for each video. Results are reported for 100k grad steps and 250k. The auto-enc represents the FVD of the pretrained encoder-decoder vs the real data. It serves as a bound on the possible model performance, as the modeling is done in the latent space of a pre-trained VQGAN.

RIVER [Davtyan, Sameni, Favaro 2023]

Summary

Probabilistic forecasting with stochastic generative dynamics

- Learn dynamics from point mass to conditional distribution
- Build SDE dynamics with stochastic interpolants
- Tune diffusion coefficients to optimize KL estimation error
- Optimized processes are Föllmer processes, which are also entropy minimizing Schrödinger bridges
- Design choices of interpolants for improved regularity
- High-Dim experiments: 2D stochastic Navier-Stokes, videos
- Future work: further design using connections to renormalizing group flows, and generative modeling in function space

Thank You!