DYffusion: A Dynamics-informed Diffusion Model for Spatiotemporal Forecasting

Salva Rühling Cachay, Bo Zhao, Hailey Joren, Rose Yu

University of California San Diego, La Jolla, CA

{sruhlingcachay, bozhao, hjoren, roseyu}@ucsd.edu

Motivation and Key Idea

- Diffusion models are mostly designed for static data
  - ⋆ How can we design a *diffusion model for temporal data*?
- Many spatiotemporal forecasting methods are deterministic
  - ⋆ How can we effectively use *generative modeling for probabilistic forecasting* problems?
- Autoregressive forecasting methods may produce unstable rollouts and poor long-range forecasts
  - ⋆ How to close the gap between training and evaluation and perform *efficient multi-step training*?

Key idea: Replace the forward and reverse processes of standard diffusion models with dynamics-informed interpolation and forecasting.

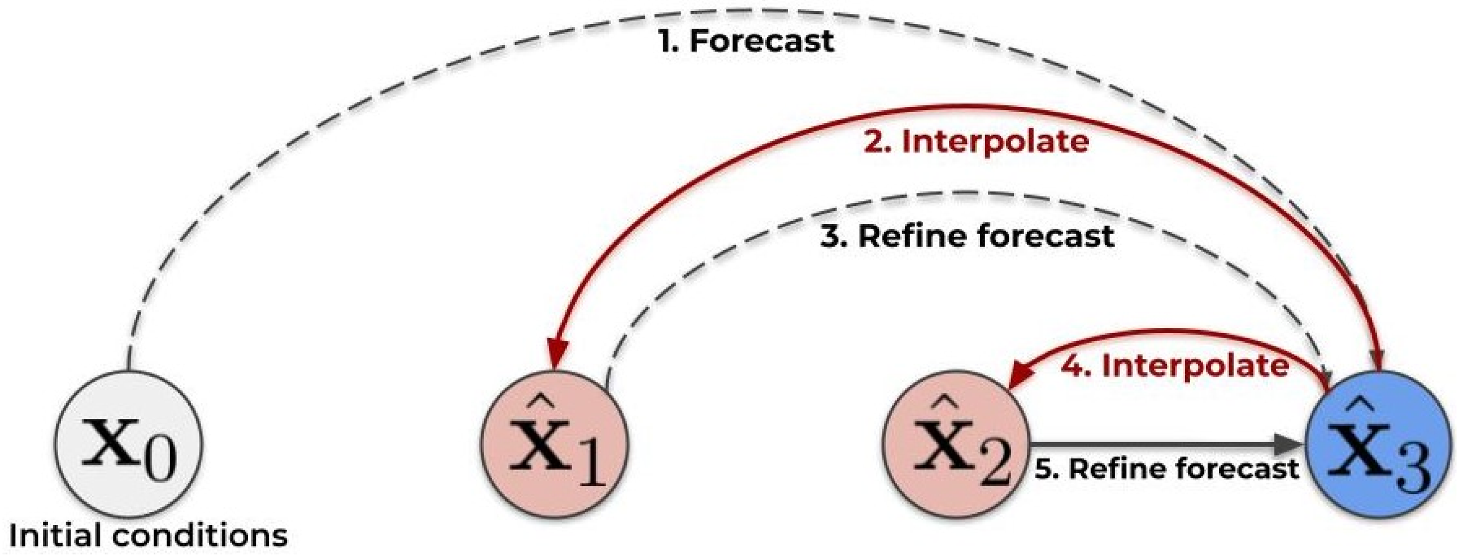
Our Results

- First study on diffusion models for spatiotemporal forecasting
- Novel adaptation of diffusion models to ensemble-based probabilistic forecasting
- Effective training approach for multi-step and long-range forecasting with low memory needs
- Competitive performance on probabilistic evaluations for forecasting complex dynamics in sea surface temperatures, Navier-Stokes flows, and spring mesh systems
- Code: `https://github.com/Rose-STL-Lab/dyffusion`

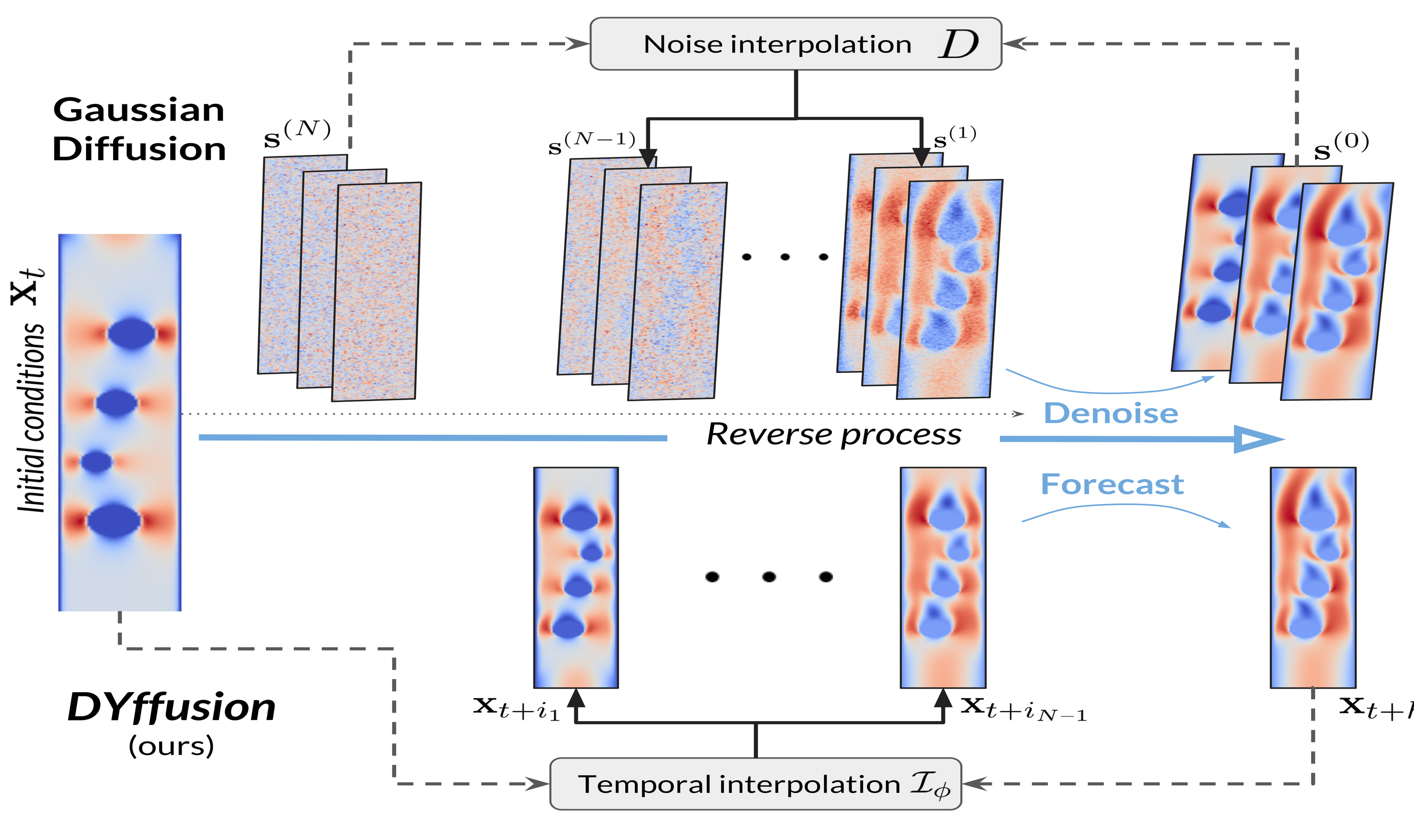
DYffusion at inference time

Spatiotemporal Forecasting: given the the initial conditions $\mathbf{x}_0$ of a dynamical system, forecast a sequence of h snapshots $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_h$

DYffusion: iteratively refines the forecast of $\mathbf{x}_h$, similarly to how standard diffusion models are used to sample from a distribution.



DYffusion vs. Standard Diffusion



Top: Standard Gaussian Diffusion, or the direct application of a video diffusion model to dynamics forecasting for a horizon of $h = 3$.

Bottom: DYffusion, which operates in the observation space at all times and does not need to model high-dimensional videos at each diffusion state.

Methods

Training

- Standard forward process → a stochastic temporal interpolation net, $\mathcal{I}_\phi$
- Standard reverse process → a deterministic forecaster network, F_θ, that predicts h steps ahead
- Train networks in two stages with simple time-conditioned objectives
- In the second stage and during sampling, use a schedule that maps diffusion steps to interpolation timesteps. In the simplest case $[i_n]_{i=0}^{N-1} = \{0, 1, \ldots, h-1\}$

Algorithm DYffusion, Two-stage Training

Input: networks $F_\theta, \mathcal{I}_\phi$, norm $||\cdot||$, horizon h, schedule $[i_n]_{i=0}^{N-1}$

Stage 1: Train interpolator network, $\mathcal{I}_\phi$
1. Sample $i \sim \texttt{Uniform}(\{1, \ldots, h-1\})$
2. Sample $\mathbf{x}_t, \mathbf{x}_{t+i}, \mathbf{x}_{t+h} \sim \mathcal{X}$
3. Optimize $\min_\phi ||\mathcal{I}_\phi(\mathbf{x}_t, \mathbf{x}_{t+h}, i) - \mathbf{x}_{t+i}||^2$

Stage 2: Train forecaster network (diffusion model backbone), F_θ
1. Freeze $\mathcal{I}_\phi$ and enable inference stochasticity (e.g. dropout)
2. Sample $n \sim \texttt{Uniform}(\{0, \ldots, N-1\})$ and $\mathbf{x}_t, \mathbf{x}_{t+h} \sim \mathcal{X}$
3. Optimize $\min_\theta ||F_\theta(\mathcal{I}_\phi(\mathbf{x}_t, \mathbf{x}_{t+h}, i_n), i_n) - \mathbf{x}_{t+h}||^2$

Sampling

- DYffusion models the dynamics $\mathbf{x}(s)$ as follows, given initial conditions $\mathbf{x}(t) = \mathbf{x}_t$:

$$\mathbf{x}(s) = \mathbf{x}(t) + \int_t^s \frac{d\mathcal{I}_\phi(\mathbf{x}_t, F_\theta(\mathbf{x}, s), s)}{ds} ds \qquad \text{for } s \in (t, t+h].$$

- At inference time, we evaluate the integral using cold sampling [1].

Proposition 1. *Cold Sampling is an approximation of the Euler method.*

Proposition 2. *In Cold Sampling, the discretization error per step is bounded by $O(\Delta s)$. Naive sampling does not have this property.*

- Different discretizations are allowed: flexible sampling schedules at inference time

Algorithm Adapted Cold Sampling [1] for DYffusion

```
Input: Initial conditions x̂_t := x_t, schedule [i_n]_{i=0}^{N-1}, output timesteps J (by default J = {1, ..., h-1})
for n = 0, 1, ..., N-1 do
    x̂_{t+h} ← F_θ(x̂_{t+i_n}, i_n)
    x̂_{t+i_{n+1}} = I_φ(x_t, x̂_{t+h}, i_{n+1}) − I_φ(x_t, x̂_{t+h}, i_n) + x̂_{t+i_n}
end for
x̂_{t+j} ← I_φ(x_t, x̂_{t+h}, j), ∀j ∈ J    # Optional refinement
Return: {x̂_{t+j} | j ∈ J} ∪ {x̂_{t+h}}
```

Experimental Setup

Dataset	Spatial grid	Training horizon	Evaluation horizon
Sea surface temperature (SST; daily, tropical Pacific)	60×60	7	7
Navier-Stokes with 4 obstacles [3]	221×42	16	64
Spring-mesh [3]	10×10	134	804

Baselines:

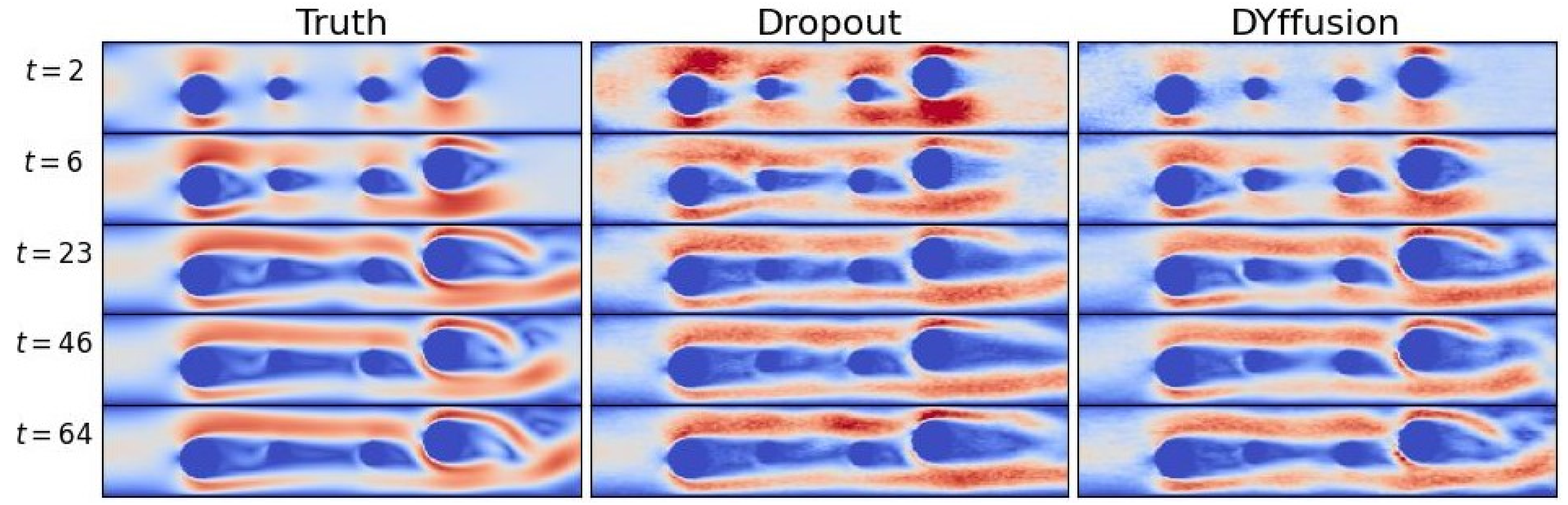
- *Perturbation:* Ensemble diffusion model backbone via input perturbations
- *Dropout:* Ensemble diffusion model backbone via enabling inference dropout
- *MCVD* and *DDPM:* Standard video [5] and denoising [2] diffusion models

Metrics (computed using a 50-member ensemble):

- Continuous ranked probability score (CRPS), lower is better
- Ensemble-mean MSE
- Spread-skill ratio (SSR) = ensemble standard deviation / RMSE. Measures reliability of the ensemble; Closer to 1 is better

Experiments

Qualitative results. Navier-Stokes velocity norm forecasts by the best baseline for the dataset and DYffusion. Our method (right column) can reproduce fine-scale details visibly better than the baseline (see e.g. right sides of the snapshots). Scan the QR code to view the full video.



Main benchmark results. Evaluation with 50-member ensembles for sea surface temperature forecasting of 1 to 7 days ahead, and Navier-Stokes flow full trajectory forecasting of 64 timesteps. Numbers are averaged out over the evaluation horizon. **Bold** indicates best, blue second best. Lower is better for CRPS and MSE; Closer to 1 is better for SSR.

Method	SST				Navier-Stokes		
	CRPS	MSE	SSR	Time [s]	CRPS	MSE	SSR
Perturbation	0.281 ± 0.004	0.180 ± 0.011	0.411 ± 0.046	0.4241	0.090 ± 0.001	0.028 ± 0.000	0.448 ± 0.002
Dropout	0.267 ± 0.003	0.164 ± 0.004	0.406 ± 0.042	0.4241	0.078 ± 0.001	0.027 ± 0.001	0.715 ± 0.005
DDPM	0.246 ± 0.005	0.177 ± 0.005	0.674 ± 0.011	0.3054	0.180 ± 0.004	0.105 ± 0.010	0.573 ± 0.001
MCVD	**0.216**	**0.161**	0.926	79.167	0.154 ± 0.043	0.070 ± 0.033	0.524 ± 0.064
DYffusion	0.224 ± 0.001	0.173 ± 0.001	**1.033 ± 0.005**	4.6722	**0.067 ± 0.003**	**0.022 ± 0.002**	**0.877 ± 0.006**

Increasing the forecasted resolution. DYffusion can be used for continuous-time forecasts and temporal super-resolution. Here, we forecast the same Navier-Stokes trajectory shown in the figure above but at 8× resolution. That is 512 timesteps instead of 64 are forecasted in total. Scan the QR code to view the full video.

Ablations

- Requires fewer diffusion steps than standard diffusion models
- Using non-integer timesteps beyond the data resolution improves performance on SST dataset
- Sampling can be accelerated by skipping intermediate sampling states, similar to DDIM [4]
- The predictions of the forecaster net for $\mathbf{x}_{t+h}$ iteratively improve with each sampling step in terms of probabilistic scores (CRPS and SSR).
- Lower sample complexity than standard diffusion models

References

[1] A. Bansal, E. Borgnia, H.-M. Chu, J. S. Li, H. Kazemi, F. Huang, M. Goldblum, J. Geiping, and T. Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *Advances in Neural Information Processing Systems*, 2023.

[2] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

[3] K. Otness, A. Gjoka, J. Bruna, D. Panozzo, B. Peherstorfer, T. Schneider, and D. Zorin. An extensible benchmark suite for learning to simulate physical systems. *Advances in Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks*, 2021.

[4] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[5] V. Voleti, A. Jolicoeur-Martineau, and C. Pal. MCVD: Masked conditional video diffusion for prediction, generation, and interpolation. *Advances in Neural Information Processing Systems*, 2022.

Acknowledgments This work was supported in part by the U.S. Army Research Office under Army-ECASE award W911NF-07-R-0003-03, the U.S. Department Of Energy, Office of Science, IARPA HAYSTAC Program, NSF Grants #2205093, #2146343, and #2134274.