S3RP: Self-Supervised Super-Resolution and Prediction for Advection-Diffusion Process

Chulin Wang; Kyongmin Yeo; Xiao Jin; Andres Codas; Levente J. Klein; Bruce Elmegreen

IBM T.J. Watson Research Center
Yorktown Heights, New York 10598
wangc@ibm.com; kyeo@us.ibm.com; Xiao.Jin@ibm.com; andrescodas@br.ibm.com; {kleinl, bge}@us.ibm.com

Abstract

We present a super-resolution model for an advection-diffusion process with limited information. While most of the super-resolution models assume high-resolution (HR) ground-truth data in the training, in many cases such HR dataset is not readily accessible. Here, we show that a Recurrent Convolutional Network trained with physics-based regularizations is able to reconstruct the HR information without having the HR ground-truth data. Moreover, considering the ill-posed nature of a super-resolution problem, we employ the Recurrent Wasserstein Autoencoder to model the uncertainty.

1 Introduction

Super-resolution (SR) reconstruction of an advection-diffusion process has a direct relevance to many important applications in atmospheric and environmental problems. SR and prediction are two of the most important aspects of down scaling climate/weather modelling and satellite observations, where ground truth is not readily available. Generating SR from low resolution data is an ill-posed problem as multiple high resolution solutions may exist corresponding to a low resolution data.

SR has been studied in the machine learning community with various methods. Deterministic methods include SRCNN[1] and more recent development ESRGAN[2], among others. However such methods are inherently deterministic, which predict the *mean* of all possible HR and tend to lose fine random features. Probabilistic models such as SRFlow[3] based on Normalizing Flow[4] and PULSE[5] based on a pretrained StyleGAN[6] have been successful in generating a distribution of HR, but are difficult to generalize to an arbitrary dataset.

Probabilistic time series predictions have been studied in a number of architectures, for example, MoCoGAN[7], S3VAE[8], and VideoFlow[9]. Most of the prior works have pleasing results with toy examples, but perform poorly in complex scenarios due to a lack of domain knowledge.

Multiple physics-informed neural networks have been proposed for SR[10, 11, 12], time series generation[13, 14, 15] and spatial-temporal super-resolution[16, 17]. To the best of our knowledge, none of the prior works have combined probabilistic SR and time series prediction in physics-informed neural networks.

We present our work of self-supervised super-resolution and prediction (S3RP) neural networks that address all of the above mentioned issues in one physics-informed neural network architecture. Our method has the following advantages:

- Address the common problem of the lack of ground-truth HR data.
- Model the uncertainty with a probabilistic model.

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

- Achieve spatial SR and temporal prediction in the same neural network architecture that is constrained by physics equation.

2 Method

2.1 Problem setup

Here, we consider the following 2-dimensional advection-diffusion equation;

$$\partial_t c + \nabla \cdot c\boldsymbol{u} = \nabla \cdot (\boldsymbol{K} \cdot \nabla c) + Q, \tag{1}$$

where c is the concentration, $\boldsymbol{u}$ is the velocity field, $\boldsymbol{K}$ is the eddy-diffusivity tensor, and Q is the source term. The wind field is generated by a multiscale Langevin process as described in [18]. The wind velocity satisfies the following mass conservation equation;

$$\nabla \cdot \boldsymbol{u} = 0. \tag{2}$$

In many real-life problems, $\boldsymbol{u}$ and c can be obtained from satellite images or weather models. However, Q and $\boldsymbol{K}$ are generally unknown. Hence, we also assume that we only have the low-resolution (LR) data for c and $\boldsymbol{u}$, not Q and $\boldsymbol{K}$.

We assume the LR data is defined in a rectangular mesh with a uniform spacing; $\mathcal{W}^L = \boldsymbol{s}_1^L \bigotimes \boldsymbol{s}_2^L$, where $\boldsymbol{s}_i^L = \{s_i^0 + j\delta s_L,\ j = 0, \cdots, N_L - 1\}$. Let $\boldsymbol{x}_t \in \mathbb{R}^{N_L \times N_L \times 3}$ be the LR data used for training, $\boldsymbol{y}_t \in \mathbb{R}^{N_H \times N_H \times 3}$ be the HR ground-truth, and $\hat{\boldsymbol{y}}_t \in \mathbb{R}^{N_H \times N_H \times 3}$ be the SR solution by the neural network. Note that $\boldsymbol{x}_t$, $\boldsymbol{y}_t$ and $\hat{\boldsymbol{y}}_t$ consist of three channels; two for the velocity components and one for the concentration. We assume that the SR image downsamples correctly with respect to the data, *i.e.*, $\mathrm{DS}(\hat{y}_t) \approx x_t$ [5], where $\mathrm{DS}(\cdot)$ is the downsampling process of natural images. If we assume a simple spatial average, the high-resolution grid system, $\mathcal{W}^H$, is naturally defined by DS.

2.2 Deep Learning Model

Consider the following stochastic advection equation,

$$\partial_t c = -\nabla \cdot c\boldsymbol{u} + \epsilon. \tag{3}$$

Here, ϵ is a random variable due to the uncertainty in Q and $\boldsymbol{K}$. We impose regularizations for the deterministic terms and employ the Recurrent Wasserstein Autoencoder (WAE) to model the stochasticity.

Physics regularization For the physics regularization, we consider the deterministic terms of Eq. (3) as well as the mass conservation of the ambient fluid;

$$\mathcal{L}_{\text{phy}} = \|\partial_t c + \nabla \cdot c\boldsymbol{u}\|_2^2 + \beta \cdot \|\nabla \cdot \boldsymbol{u}\|_2^2. \tag{4}$$

Recurrent WAE Here, we consider the following data generating distribution,

$$p(\boldsymbol{x}_{0:t}, \boldsymbol{z}_{0:t}) = \prod_{i=0}^{t} p_\theta(\boldsymbol{x}_i|\boldsymbol{z}_i) p(\boldsymbol{z}_i|\boldsymbol{z}_{0:i-1}), \tag{5}$$

in which $\boldsymbol{z}_t$ is a latent variable. Then, as shown in [19], minimizing the following loss function corresponds to minimizing a Wasserstein distance between the data generating distribution and the distribution of a deep generative model,

$$\mathcal{L}_{\text{WAE}} = \sum_{i=1}^{t} \mathbb{E}_{q(\boldsymbol{z}_i|\boldsymbol{z}_{0:i-1},\boldsymbol{x}_{i-1})}[l(\boldsymbol{x}_i, G(\boldsymbol{z}_i))] + \lambda \sum_{i=0}^{t} \mathcal{D}\left(q_{\boldsymbol{z}_i|\boldsymbol{z}_{0:i-1}}, p_{\boldsymbol{z}_i|\boldsymbol{z}_{0:i-1}}\right), \tag{6}$$

in which $l(\cdot,\cdot)$ denotes a cost function, $\mathcal{D}(q,p)$ is a divergence between distributions, and $p_{\boldsymbol{z}_i|\boldsymbol{z}_{0:i-1}}$ is a prior distribution. Here, the posterior distribution, $q(\boldsymbol{z}_i|\boldsymbol{z}_{0:i-1}, \boldsymbol{x}_i)$, and the generator, $G(\boldsymbol{z}_i)$, are parameterized by neural networks. The l^2-norm is used for the cost function, $l(\boldsymbol{x}_{t+1}, G(\boldsymbol{z}_t)) = \|\boldsymbol{x}_{t+1} - G(\boldsymbol{z}_t)\|_2^2$. $G(\boldsymbol{z}_t)$ consists of a super-resolution decoder followed by the downsampling,

$$G(\boldsymbol{z}_t) = \mathrm{DS}(\hat{\boldsymbol{y}}_t),$$

in which $\hat{\boldsymbol{y}}_t = \Psi_{\text{dec}}(\boldsymbol{z}_t)$ and Ψ_{dec} denotes a decoder neural network. The maximum mean discrepancy[20] is used for the divergence, $\mathcal{D}$.

WAE has a similar structure with the variational autoencoder (VAE). However, unlike VAE, WAE does not require explicitly defining the emission probability, $p(\boldsymbol{x}_t|\boldsymbol{z}_t)$, which is advantageous when $\boldsymbol{x}_t$ has a complex correlation structure, such as the physical constraints in the current problem setup. Finally, the loss function is given as the sum of WAE and the physics loss;

$$\mathcal{L} = \mathcal{L}_{\text{WAE}} + \gamma \mathcal{L}_{\text{phy}}. \tag{7}$$

Neural network architecture. A diagram of our neural network is shown in Fig. 1. The conditional probabilities are approximated by RNNs in both the encoding and decoding process. The initial condition, $\hat{y}_0$, is generated by a bilinear upsampling of the LR data. We use a LSTM for the variational encoder, and we use a PhyCell (see Section A) in conjunction with a LSTM for the decoder.

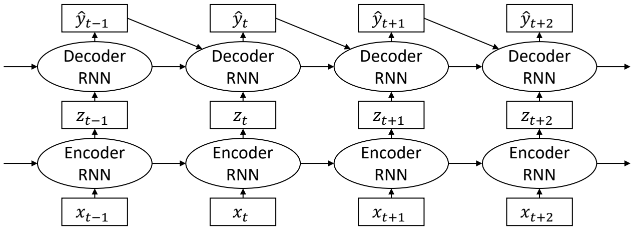

Figure 1: A sketch of our Recurrent WAE neural network. The recurrent encoder is probabilistic, representing the conditional distribution of $\boldsymbol{z}_t$ given the past LR data $\boldsymbol{x}_{0:t}$. The recurrent decoder is deterministic, and is a function of $\boldsymbol{z}_{0:t}$ and the last HR output $\hat{y}_{t-1}$, and generates the new output $\hat{y}_t$.

3 Experiments

3.1 Dataset

Following Section 2.1, we generate 10 simulations with different $\boldsymbol{u}$ and $\boldsymbol{K}$. Within each simulation, 16 concentration fields are computed for 16 source locations, separately. The final concentration data is generated by a random combination of the concentration fields due to the linearity. The HR simulations is performed for the mesh size $N_H = 128$. The LR training data set $\boldsymbol{x}_t$ is subsequently generated by a spatial average over 8×8 pixels in HR, resulting in a input data $\boldsymbol{x}_t \in \mathbb{R}^{16\times16\times3}$.

3.2 Baseline and Models

Bicubic. We use bicubic upsampling to generate SR images as the baseline.
S3RP model (Interpolation model). We use $x_{0:t}$ for SR reconstruction of $y_{0:t}$ as shown in Eq. (5)
S3RP model (Other variations). The same model can be reconfigured to perform SR of $\boldsymbol{u}$ as well as SR and forecast of c (c-only model). Alternatively, the same model can perform SR and forecast of $(c, \boldsymbol{u})$ (full extrapolation model). Detailed discussion can be found in Appendix C

3.3 Metric

We evaluate the model by a Monte Carlo (MC) simulation with 100 samples. Then, the expectation and the prediction intervals are estimated. The mean squared error (MSE) is evaluated between the expectation and the HR ground truth. The empirical coverage probability is also shown to provide a quantitative assessment of the probabilistic model. Finally, we also evaluate the physics error of the model output and verify that the predictions generated by the model is indeed physically consistent.

3.4 Results

We evaluate the model performance by using a hold-out LR data for 90 time steps, $\boldsymbol{x} \in \mathbb{R}^{90\times16\times16\times3}$, and comparing the output with the ground-truth HR. The results of the interpolation model are summarized in Table 1. It is shown that our model outperforms the baseline. Since our model is a generative model that computes the probability distribution, we also showed the empirical coverage probability (ECP) for a quantitative comparison in Table 1. More discussions about the estimated probability distribution is provided in Appendix E. In Fig. 2, the model output of concentration c is visualized, together with the estimated standard deviation. The model also outperforms the baseline in capturing the physics process. Fig. 3 shows a snapshot of the physics errors. In particular, it is shown that the SR solution of our model well satisfies the mass conservation condition shown in Eq. (2), compared to the baseline. In Table 1, it is shown that ϵ_{div} of our model is an order of magnitude smaller than the baseline. As mentioned in Section 3.2, the models can be set up to accommodate for different input settings. We show the range of outputs for an a given coordinate generated by all 3 variations of the probabilistic model in Fig. 4. In Fig 4, for c-only and extrapolation models, the LR data is provided only up to $t = 90$, and the model makes a probabilistic forecast by a Monte Carlo simulation for the next 30 steps. It is shown that the prediction interval of the extrapolation mode becomes much larger than that of the c-only mode due to the uncertainties in the future wind condition.

Table 1: Comparison of our S3RP model (interpolation mode) and baseline. The 1σ and 2σ coverage denote 68% and 95% prediction intervals. The physics errors, $\epsilon_{\text{adv-diff}}$ and ϵ_{div}, are defined in Appendix D. The best results are in boldface.

	MSE	1σ coverage	2σ coverage	$\epsilon_{\text{adv-diff}}$	ϵ_{div}
S3RP	**1.38E-4**	**65.5%**	**82.2%**	**1.69e-6**	**6.28E-6**
Bicubic	3.97E-4	-	-	1.83E-6	6.20E-5

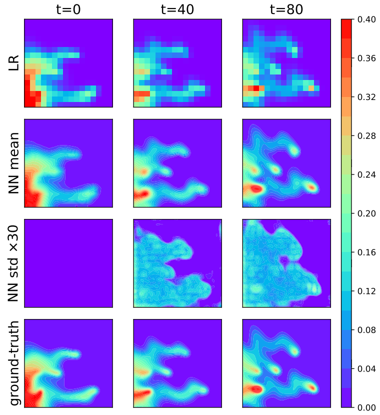

Figure 2: Visualization of the S3RP model at different timesteps. The model (2nd row) is able to perform $8\times$ SR accurately as well as capture the uncertainty (3rd row).

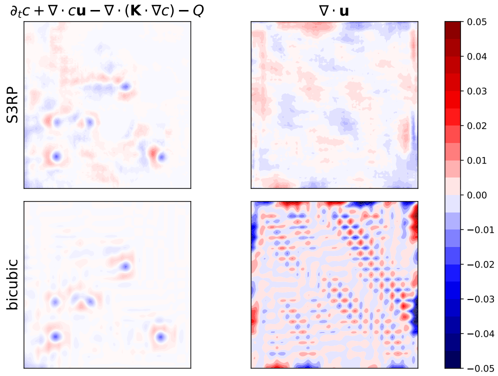

Figure 3: A snapshot of the spatial distribution of the physics errors. The first and second rows correspond to S3RP and the bicubic interpolations, respectively.

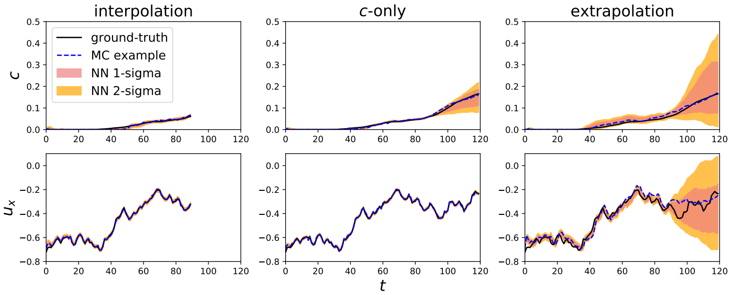

Figure 4: Comparison between the ground-truth and the model output at an arbitrary coordinate. A sample from the MC simulation is shown as the dashed line. Each column represent one of the variations of the model.

4 Conclusions

We propose a deep learning model, S3RP, which achieves both super-resolution and prediction for advection-diffusion process. The training is self-supervised assuming no access to HR ground-truth. The uncertainty is estimated by using WAE. The model also embeds physics equations by using a physics regularization and a specially designed module, PhyCell, in the network. As the result shows, the model performs better in all metrics against the baseline. We expect the same framework can be applied to many real-world physics problems, especially where only LR data is available.

Broader Impact

We propose to approach the SR problem from a probabilistic formulation, because in a SR problem uniqueness of the solution is not guaranteed. Moreover, the model enables dense spatial-temporal predictions when high resolution ground truth doesn't exist but the governing physics are well understood. Such problems are commonly encountered in Earth and Environmental Sciences, such as pollution prediction and source identification from satellite images or for autonomous vehicle/robot navigation under hazardous conditions.

References

[1] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[2] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.

[3] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Srflow: Learning the super-resolution space with normalizing flow. In *European Conference on Computer Vision*, pages 715–732. Springer, 2020.

[4] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[5] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 2437–2445, 2020.

[6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[7] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

[8] Yizhe Zhu, Martin Renqiang Min, Asim Kadav, and Hans Peter Graf. S3vae: Self-supervised sequential vae for representation disentanglement and data generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6538–6547, 2020.

[9] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A conditional flow-based model for stochastic video generation. *arXiv preprint arXiv:1903.01434*, 2019.

[10] Thomas Vandal, Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, and Auroop R Ganguly. Deepsd: Generating high resolution climate change projections through single image super-resolution. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, pages 1663–1672, 2017.

[11] Chulin Wang, Eloisa Bentivegna, Wang Zhou, Levente Klein, and Bruce Elmegreen. Physics-informed neural network super resolution for advection-diffusion models. *arXiv preprint arXiv:2011.02519*, 2020.

[12] Han Gao, Luning Sun, and Jian-Xun Wang. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Physics of Fluids*, 33(7):073603, 2021.

[13] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11474–11484, 2020.

[14] Casper Kaae Sønderby, Lasse Espeholt, Jonathan Heek, Mostafa Dehghani, Avital Oliver, Tim Salimans, Shreya Agrawal, Jason Hickey, and Nal Kalchbrenner. Metnet: A neural weather model for precipitation forecasting. *arXiv preprint arXiv:2003.12140*, 2020.

[15] Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1457–1466, 2020.

[16] Soheil Esmaeilzadeh, Kamyar Azizzadenesheli, Karthik Kashinath, Mustafa Mustafa, Hamdi A Tchelepi, Philip Marcus, Mr Prabhat, Anima Anandkumar, et al. Meshfreeflownet: a physics-constrained deep continuous space-time super-resolution framework. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15. IEEE, 2020.

[17] You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Transactions on Graphics (TOG)*, 37(4):1–15, 2018.

[18] Kyongmin Yeo, Youngdeok Hwang, Xiao Liu, and Jayant Kalagnanama. Development of hp-inverse model by using generalized polynomial chaos. *Computer Methods in Applied Mechanics and Engineering*, 347:1–20, 2019.

[19] Jun Han, Martin Renqiang Min, Ligong Han, Li Erran Li, and Xuan Zhang. Disentangled recurrent wasserstein autoencoder. *arXiv preprint arXiv:2101.07496*, 2021.

[20] Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*, 2015.

Appendices

A PhyCell

As studied in [13], recurrent neural networks (RNN) is not efficient in video prediction. PhyCell is proposed as a special module that encodes an arbitrary partial differential equation (PDE). We implement the same structure in the decoder Ψ_{dec} and observed better extrapolation results of Eq. 1.

B Architecture Details

The detailed encoder and decoder structure is illustrated in Figs. 5 and 6.

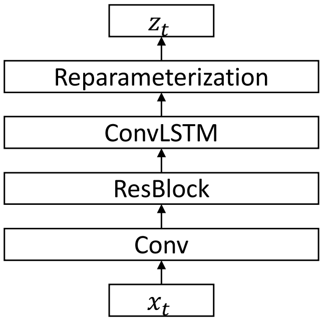

Figure 5: The variational encoder is a probabilistic model. It takes as input the LR data, x_t, and outputs a sample of the latent variable z_t, which is done by reparameterization a unit gaussian with the mean and standard deviation predicted by the model. The ConvLSTM handles the memory of previous LR data $x_{0:t-1}$.

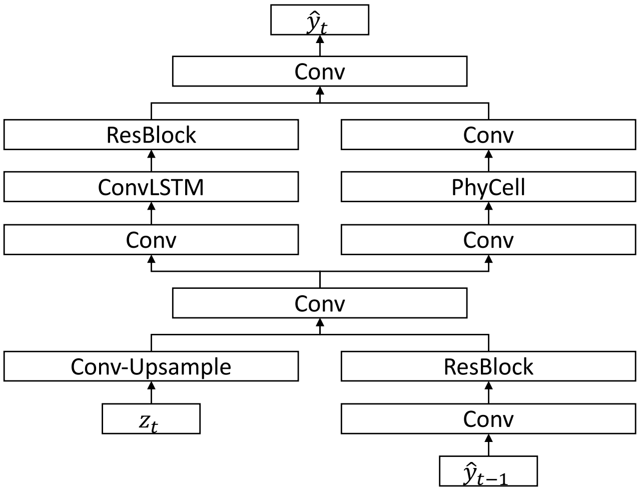

Figure 6: The variational decoder is deterministic (Dirac delta probability distribution). The input of the decoder is the latent variable z_t and the previous HR output $\hat{y}_{t-1}$, and outputs the new HR output $\hat{y}_t$. The decoder is comprised of two branches, the PhyCell that captures the PDE, and the ConvLSTM that takes care of the residual dynamics.

C Other Model Variations

C.1 Full extrapolation model

As illustrated in Section 2.2 and Fig. 1, our model is set up to generate the super-resolution without prediction by default. However, with slight change of the mapping from $\boldsymbol{x}$ to $\boldsymbol{z}$ to $\boldsymbol{y}$, The model can be rerouted for for 1-step prediction. Specifically, instead of the following setup, the following data generating distribution is defined,

$$p(\boldsymbol{x}_{1:t+1}, \boldsymbol{z}_{0:t}) = \prod_{i=0}^{t} p_\theta(\boldsymbol{x}_{i+1}|\boldsymbol{z}_i)p(\boldsymbol{z}_i|\boldsymbol{z}_{0:i-1}). \tag{8}$$

We can then define a *extrapolation* variant of the model. In other words, in the *extrapolation model*, we aim to compute the probability distribution of the SR prediction $(c_{t+1}, \boldsymbol{u}_{t+1})$, given the low resolution data at time t. An illustration is shown in Fig. 7

C.2 $\boldsymbol{c}$-only model

In certain circumstances, we might want to do prediction of concentration given a reliable future wind forecasting. Such a configuration can be set up by repurposing the encoder to take as input

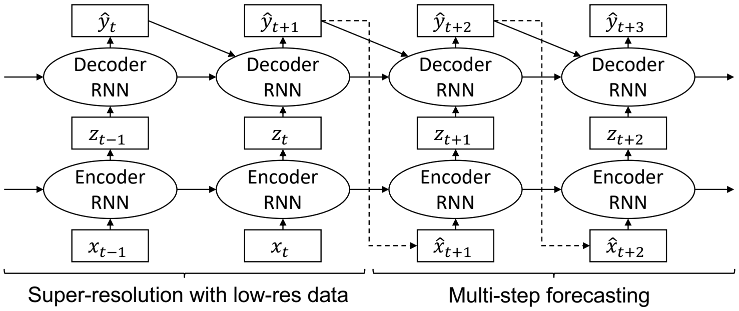

Figure 7: Full extrapolation model. The recurrent decoder takes as input the history of latent variables $z_{0:t}$ and the last HR output $\hat{y}_t$, and generates the new output $\hat{y}_{t+1}$.

$x'_t = (c_t, \boldsymbol{u}_{t+1})$, such that the data generating process can be written as,

$$p(\boldsymbol{x'}_{1:t}, \boldsymbol{z}_{0:t}) = \prod_{i=0}^{t} p_\theta(\boldsymbol{x'}_i|\boldsymbol{z}_i)p(\boldsymbol{z}_i|\boldsymbol{z}_{0:i-1}). \tag{9}$$

In this problem setup, we are interested in the joint probability distribution of the SR reconstruction of $\boldsymbol{u}_{t+1}$ given the low resolution velocity field at $t+1$ and the SR prediction of c_{t+1} given the low resolution concentration at time t.

D Physics error

To evaluate the model's ability to make physically consistent predictions, we calculate the physics errors based on Eqs. (1) and (2). Concretely, the advection-diffusion error $\epsilon_{\text{adv-diff}}$ and the divergence-free error ϵ_{div} is defined as follows,

$$\epsilon_{\text{adv-diff}} = \|\partial_t c + \nabla \cdot c\boldsymbol{u} - \nabla \cdot (\boldsymbol{K} \cdot \nabla c) - Q\|_2^2 \tag{10}$$

$$\epsilon_{\text{div}} = \|\nabla \cdot \boldsymbol{u}\|_2^2 \tag{11}$$

For the S3RP model and the bicubic baseline, the physics errors are visualized in Fig. 3. The 1st column is used calculate $\epsilon_{\text{adv-diff}}$ as described in Eq. (10), and the 2nd column is used to calculate ϵ_{div} described in Eq. (11). The model only slightly outperforms the baseline in $\epsilon_{\text{adv-diff}}$ due to the lack of knowledge of $\boldsymbol{K}$ and Q. Whereas it significantly outperforms the baseline in ϵ_{div}.

E Uncertainty and error

Our model is a generative model of which output is a sample from the probability distribution. Unlike a deterministic model, it is not straightforward to assess the accuracy of the probabilistic model for a high-dimensional spatio-temporal process. We have provided a quantitative measure in Table 1 in terms of the empirical coverage probability (ECP). It is shown that, while ECP for 1σ has a small error (only about 2.5%), for 2σ, there is about 12% error in the coverage. This may due to the fact that the probability distribution is inferred from the LR data, while the comparison is made against the HR ground truth.

Here, we provide another method to assess the estimated probability distribution from S3RP. Here, the probability distribution captures the uncertainty of the model prediction. In general, we expect that the width of the probability distribution increases when the model is uncertain about the state of the process. In other words, we expect to see a wider distribution, where the model error is larger. In Fig. 8, a two-dimensional histogram is plotted to compare a point-wise absolute difference, $|c_t(\boldsymbol{s}) - c_t^*(\boldsymbol{s})|$, in which $c_t(\boldsymbol{s})$ and $c_t^*(\boldsymbol{s})$ are, respectively, the model output and the HR ground truth at a spatial location $(\boldsymbol{s})$ at time t, and the width of the probability distribution, denoted by the standard deviation. It is shown that in general as $|c_t(\boldsymbol{s}) - c_t^*(\boldsymbol{s})|$ increases, the estimated probability distribution becomes wider. There is a faint band at around standard deviation between 0.003 and 0.004, which is not well understood. It may come from the error in the estimation of the probability distribution. Again, here we aim to estimate the probability distribution of the SR solution solely from the LR data, which makes the problem challenging.

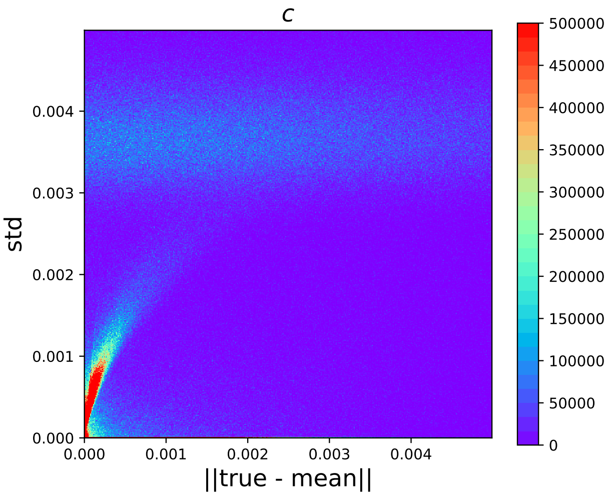

Figure 8: The standard deviation of model prediction plotted against the error in a 2D histogram. The colors correspond to the frequency density. An approximate trend that larger error corresponding to larger standard deviation indicates that the model performs well in confidence interval estimation.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This is a fundamental research.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [No] Space limitations, and complete proofs are beyond the scope of this paper.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code and the data are proprietary.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [No] Did not detail the hyperparameters.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] This data is proprietary
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] Assets are Open Source
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] The new assets are proprietary.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We generated our own data.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]