Computing Residual Diffusivity by Adaptive Basis Learning via Super-Resolution Deep Neural Networks

Jiancheng Lyu(✉), Jack Xin, and Yifeng Yu

UC Irvine, Irvine, CA 92697, USA
{jianchel,jack.xin,yifengy}@uci.edu

Abstract. It is expensive to compute residual diffusivity in chaotic incompressible flows by solving advection-diffusion equation due to the formation of sharp internal layers in the advection dominated regime. Proper orthogonal decomposition (POD) is a classical method to construct a small number of adaptive orthogonal basis vectors for low cost computation based on snapshots of fully resolved solutions at a particular molecular diffusivity D_0^*. The quality of POD basis deteriorates if it is applied to $D_0 \ll D_0^*$. To improve POD, we adapt a super-resolution generative adversarial deep neural network (SRGAN) to train a nonlinear mapping based on snapshot data at two values of D_0^*. The mapping models the sharpening effect on internal layers as D_0 becomes smaller. We show through numerical experiments that after applying such a mapping to snapshots, the prediction accuracy of residual diffusivity improves considerably that of the standard POD.

Keywords: Advection dominated diffusion · Residual diffusivity · Adaptive basis learning · Super-resolution deep neural networks

1 Introduction

It has been a fundamental problem to characterize the large scale effective diffusion in fluid flows containing complex and turbulent streamlines [17]. In this paper, we consider the passive scalar model [11]:

$$T_t + (\boldsymbol{v} \cdot D)\, T = D_0\, \Delta T, \tag{1}$$

where T is a scalar function, $D_0 > 0$ is a constant (the so called molecular diffusivity), $\boldsymbol{v}(\boldsymbol{x}, t)$ is a incompressible velocity field, D and Δ are the spatial gradient and Laplacian operators. In two dimension, the effective diffusivity tensor is given by [1]:

$$D_{ij}^E = D_0 \left(\delta_{ij} + \langle Dw_i \cdot Dw_j \rangle\right), \tag{2}$$

H. A. Le Thi et al. (Eds.): ICCSAMA 2019, AISC 1121, pp. 279–290, 2020.
https://doi.org/10.1007/978-3-030-38364-0_25

where $w = (w_1, w_2)$ is the unique mean zero space-time periodic vector solution of the *cell problem* [1]:

$$w_t + (\boldsymbol{v} \cdot Dw) - D_0 \Delta w = -\boldsymbol{v}, \tag{3}$$

and $\langle \cdot \rangle$ denotes space-time average over the periods. The term $\langle Dw_i \cdot Dw_j \rangle$ in (2) is a positive definite correction to $D_0\, \delta_{ij}$.

Asymptotic behavior of D_{ij}^E can be solved when the flow is steady and periodic. For instance, the time independent cellular flow [4,5,13,19,20]

$$\boldsymbol{v} = (-H_y, H_x), \quad H = \sin x \sin y,$$

has been proved to generate effective diffusion following the square root law with dominated advection [5,6]:

$$D_{ij}^E = O(\sqrt{D_0}) \gg D_0, \quad D_0 \downarrow 0, \quad \forall i, j.$$

However, when the streamlines are time-dependent or fully **chaotic**, the enhancement can be quite different and difficult to solve analytically. A simple example is

$$\boldsymbol{v} = (\cos(y), \cos(x)) + \theta\, \cos(t)\, (\sin(y), \sin(x)), \quad \theta \in (0, 1], \tag{4}$$

where the first term is a steady cellular flow with a $\pi/4$ rotation and is perturbed by the a time-periodic flow. At $\theta = 1$, the flow (4) is fully chaotic [21] and was investigated in the Rayleigh-Bénard experiment [3]. Numerical simulations of the fully chaotic model [2,10,18] suggest that

$$D_{11}^E = O(1), \quad D_0 \downarrow 0, \tag{5}$$

hence the *residual diffusivity* phenomenon emerges.

As in (4)$\boldsymbol{v}$ is periodic in time, the solution of cell problem (3) subject to periodic boundary condition in space can be computed accurately by spectral method in Fourier basis. By expanding both w and $\boldsymbol{v}$ as Fourier series, (3) is equivalent to an ordinary differential equation (ODE) system. To solve the system numerically, one can approximate w using finitely many Fourier modes thereby the problem is reduced to solving for the periodic solution to a linear ODE system. The corresponding Poincaré map is constructed in [10] and the solution is found as the unique fixed point of it. The effective diffusivity D^E can be finally computed by (2). The drawback of the spectral approach is that the number of Fourier modes required by the truncated problem grows rapidly as $D_0 \downarrow 0$ due to the sharp gradient of the solution.

In [10], adaptive orthogonal basis vectors are constructed from snapshots of spectral solutions to handle the near singular solutions of (3) at small D_0. Particularly, at certain D_0 and θ sample, snapshots of the solution in one time period form a solution matrix and the adaptive basis consists of singular vectors for the top singular values of the matrix. Hence the linear ODE system with D_0 or θ that are close to the sampled values can be rewritten in terms of adaptive basis

and solved similarly with Poincaré map. The number of adaptive basis functions is one or two hundred, far less than that of Fourier basis which is usually at least a few thousand. With the reduced adaptive basis, the relative error of computing residual diffusivity is no more than 6.5% with carefully selected samples.

The above procedure to generate adaptive basis vectors by taking snapshots of solutions and solving singular value decomposition (SVD) is the so called reduced order modeling [14,16]. In fluid dynamics literature [7,9], such technique is also referred to by proper orthogonal decomposition (POD). However, POD method relies heavily on collection of good snapshots data. Snapshots data with less representation capability would hardly recover the fully resolved solution. As shown in the test results of computing residual diffusivity, when snapshots are collected at certain D_0 and the solutions at a much smaller $D_1 \ll D_0$ are to be computed with reduced basis constructed at D_0, the errors can rapidly increase.

In this paper, we study a deep neural network (DNN) approach to alleviate the accuracy loss of POD and improve the error of reduced basis computation at D_1 based on prediction from snapshots at two values D_0^1 and D_0^2 (both above D_1). The idea is to train a mapping from snapshots (images) at D_0^1 to those at D_0^2 ($D_0^2 < D_0^1$). The mapping sharpens the images similar to what happens to solutions of (1) as D_0 becomes smaller. The mapping is applied to snapshots at D_0^2 for improving POD basis construction. As a proof of concept, we select a super-resolution DNN in the form of a generative adversarial network (SRGAN, [8]). We show that it serves our purpose well through numerical experiments where the mapping is constructed (trained) based on snapshots at D_0^1 and D_0^2, then tested (applied) at $D_1 < D_0^2$.

The paper is organized as follows. In Sect. 2, we describe the POD basis construction for (2), SRGAN architecture and its training objective. In Sect. 3, we present computational results on predicted residual diffusivity from POD basis with and without SRGAN. Concluding remarks are in Sect. 4.

2 Construction of Adaptive Basis via DNNs

2.1 Learning Thinner Structures

Consider spectral method for solving the cell problem below for D_{11}^E:

$$w_t + (\boldsymbol{v} \cdot \nabla)\, w - D_0\, \Delta\, w = -v, \tag{6}$$

where $\boldsymbol{v} = (v, \tilde{v})$, and

$$\begin{aligned} v\,(x,t) &= \cos\,(x_2) + \sin\,(x_2)\cos\,(t)\,, \\ \tilde{v}\,(x,t) &= \cos\,(x_1) + \sin\,(x_1)\cos\,(t)\,. \end{aligned} \tag{7}$$

Let w_k^N be the k-th mode of a $(2N+1)^2$ term Fourier approximation of w on the $[0, 2\pi]^2$ periodic domain [10]. Let v_k and $\tilde{v}_k$ be the k-th Fourier modes of v and $\tilde{v}$

respectively. In view of (7), v_k ($\tilde{v}_k$) equals zero unless $k = (0, \pm 1)$ ($k = (\pm 1, 0)$). The truncated ODE system on w_k^N is:

$$\frac{dw_k^N}{dt} + D_0 |k|^2 w_k^N + i \sum_{\|k-j\| \le N} \left[(k_1 - j_1) v_j(t) + (k_2 - j_2) \tilde{v}_j(t) \right] w_{k-j}^N = -v_k(t), \tag{8}$$

which reads in vector-matrix form: $d\mathbf{w}/dt = A(t)\mathbf{w} + \mathbf{v}(t)$. For a time discretization with N_t grid points on $[0, 2\pi]$, denote by $\{\hat{\mathbf{w}}_n^*\}_{n=0}^{N_t}$ a numerical time periodic solution to (8) for $D_0 = D_0^*$, a value where snapshots data are collected. Such a solution can be found by solving for the unique fixed point of the Poincaré map [10]. Define the solution matrix of size $(2N+1)^2 \times N_t$:

$$W = \left[\hat{\mathbf{w}}_0^* \quad \hat{\mathbf{w}}_1^* \quad \dots \quad \hat{\mathbf{w}}_{N_t}^* \right], \tag{9}$$

and compute the SVD factorization $W = U \Sigma V^T$. Then one extracts columns $\mathbf{u}_j$ ($j = 1, \cdots, m$) of the matrix U corresponding to the largest $m \ll O(N^2)$ singular values, to form the adaptive orthogonal basis vectors and the matrix: $U_m = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_m]$. This is the end of basis training at a sampled value D_0^*. At $D_0 \neq D_0^*$, project $\mathbf{w}(t)$ to the span of column vectors of U_m or seek a vector of the form $U_m \mathbf{a}(t)$, where $\mathbf{a}(t) \in \mathbb{R}^m$ satisfies the ODE system in a much lower dimension (bar is complex conjugate, T is transpose):

$$\frac{d\mathbf{a}}{dt} = \bar{U}_m^T A(t) U_m \mathbf{a} + \bar{U}_m^T \mathbf{v}(t). \tag{10}$$

Finding the time periodic solution to a time discrete version of (10) via Poincaré map to compute D^E by (2) in Fourier space, we completed the reduced order modeling.

The inverse Fourier transform of W gives the snapshot matrix in the physical domain:

$$S_p = \left[\mathbf{I}_0^* \quad \mathbf{I}_1^* \quad \dots \quad \mathbf{I}_{N_t}^* \right] \tag{11}$$

where each column vector (snapshot) is an image after reshaping into a square matrix. The S_p is convenient for visualization and drawing a connection with image processing. Figs. 1 and 2 illustrate that internal layers in the physical domain snapshots (column vectors of S_p) emerge and get thinner as D_0 becomes smaller. For a better prediction of the nearly singular solutions at D_1 much smaller than D_0^*, it is helpful if the adaptive basis learned at D_0^* encodes certain thinner layered structures. Particularly, given the solution matrix W at D_0^* as (9), we look for a map $\mathcal{M}$ such that the physical domain snapshots (inverse Fourier transform) of $\mathcal{M}(W)$ have sharpened internal layers.

Suppose $D_0^1 > D_0^2$ are two values $\ge D_0^*$, and W^i is the Fourier domain solution matrix at D_0^i for $i = 1, 2$. Let $\mathcal{F}$ be column-wise Fourier transform on matrices, then columns of $\mathcal{F}^{-1}\left(W^i\right)$ are physical domain snapshots of W^i. When the W^i's are available, we use DNN to train a map $\mathcal{T}$ for the following regression problem:

$$\mathcal{T} : \mathcal{F}^{-1}\left(W^1\right) \to \mathcal{F}^{-1}\left(W^2\right). \tag{12}$$

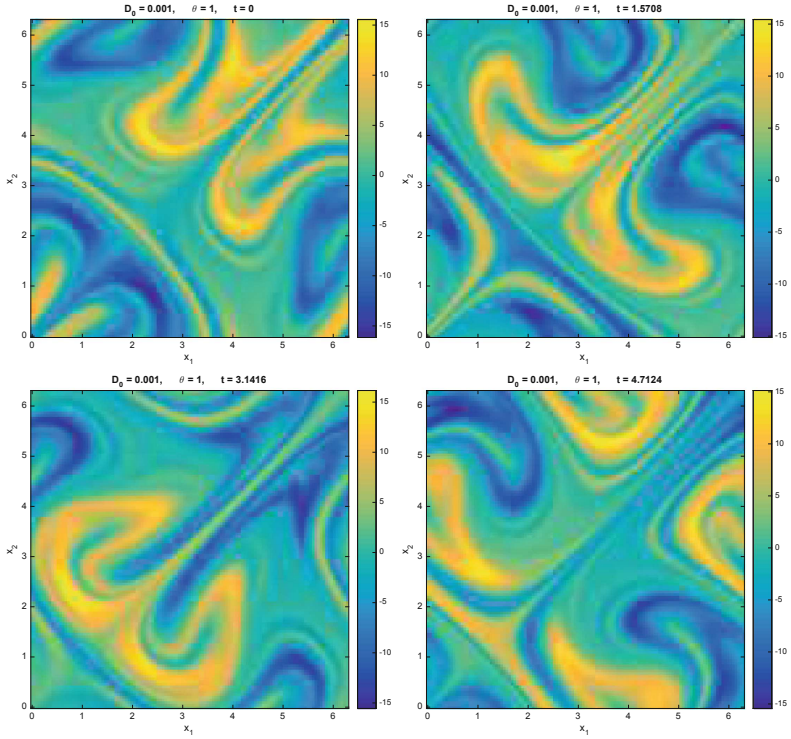

Fig. 1. Sampled snapshots of (6) at $D_0 = 10^{-3}$ with layered structures.

Since $D_0^2 < D_0^1$, $\mathcal{F}^{-1}\left(W^2\right)$ has thinner layered structures than $\mathcal{F}^{-1}\left(W^1\right)$. By solving the regression problem (12) via DNN, our goal is that the $\mathcal{T}\left(\mathcal{F}^{-1}\left(W^1\right)\right)$ inherits the image sharpening capability. Then $\mathcal{T}$ can be applied to solution matrix W^* at $D_0^* \leq D_0^2$ and $\mathcal{T}\left(\mathcal{F}^{-1}\left(W^*\right)\right)$ is expected to have thinner structures for better prediction of residual diffusivity. Finally, the adaptive basis with thinner structures will be obtained from SVD of

$$\mathcal{M}\left(W^*\right) = \mathcal{F}\left(\mathcal{T}\left(\mathcal{F}^{-1}\left(W^*\right)\right)\right).$$

2.2 Adversarial Network

We opt for the super-resolution generative adversarial network (SRGAN) [8] to train the map $\mathcal{T}$. As a generative adversarial network (GAN), SRGAN consists of a generator network G and a discriminator network D. The two networks compete in a way that D is trained to distinguish the real high-resolution (HR) images and those generated from low-resolution (LR) images, while G is trained to create fake HR images from LR images to fool D. We train the SRGAN with $\mathcal{F}^{-1}\left(W^1\right)$ as input data and $\mathcal{F}^{-1}\left(W^2\right)$ as target data so that the generator

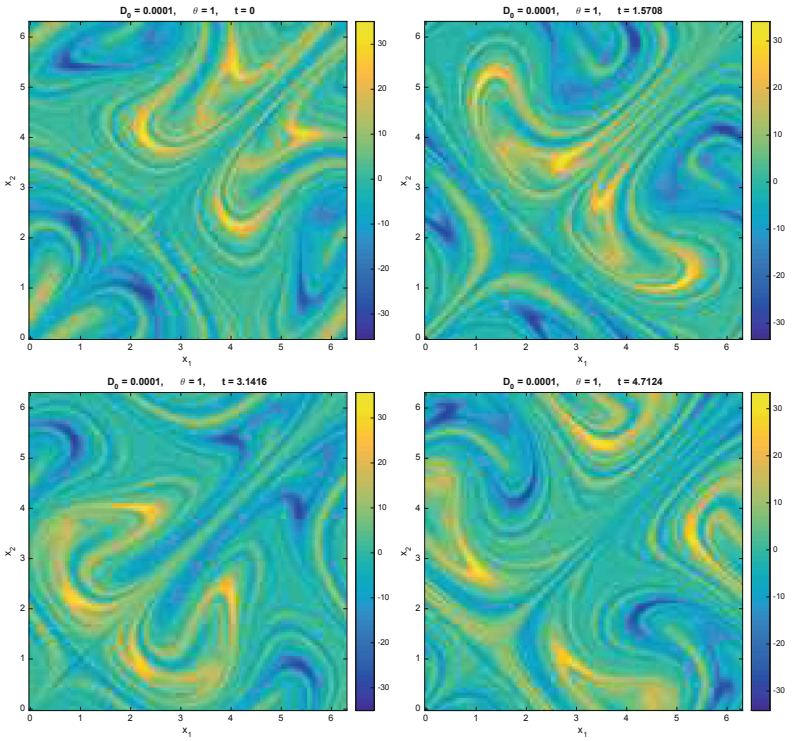

Fig. 2. Sampled snapshots of (6) at $D_0 = 10^{-4}$, formation of thinner layers.

G learn to generate thinner structures when it is fed with $\mathcal{F}^{-1}(W)$. In this approach, we realize $\mathcal{T}$ through a trained G.

The network architecture is shown in Fig. 3. The generator network G starts with a convolutional block with kernel size 9×9, followed by a few residual blocks. Here a convolutional block consists of a convolutional layer and a PReLU layer, a residual block is a convolutional block with kernel size 3×3 followed by a convolutional layer of the same kernel size and a shortcut from the input to output. There are two more convolutional layers with kernel size 3×3 and 9×9 after the residual blocks at the end of the network. The number of filters in all convolutional blocks are the same except for the last one. Note that we remove the two upscale layers in [8] since the snapshot sizes of $\mathcal{F}^{-1}\left(W^1\right)$ and $\mathcal{F}^{-1}\left(W^2\right)$ are the same.

The discriminator network D is defined by the architectural guidelines summarized in [12], see Fig. 3. It has eight convolutional blocks with PReLU layers replaced by LeakyReLU layers with slope parameter $\alpha = 0.2$. Moreover, there is a batch normalization layer before each LeakyReLU in the convolutional blocks. The kernel size is 3×3 in all convolutional blocks and the number of filters is doubled in the 3rd, 5th and 7th block. Those blocks are followed by a fully

connected layer, a LeakyReLU layer and one more fully connected layer. Finally the feature map is fed in a sigmoid layer which gives the probability of real HR image and the reconstructed image.

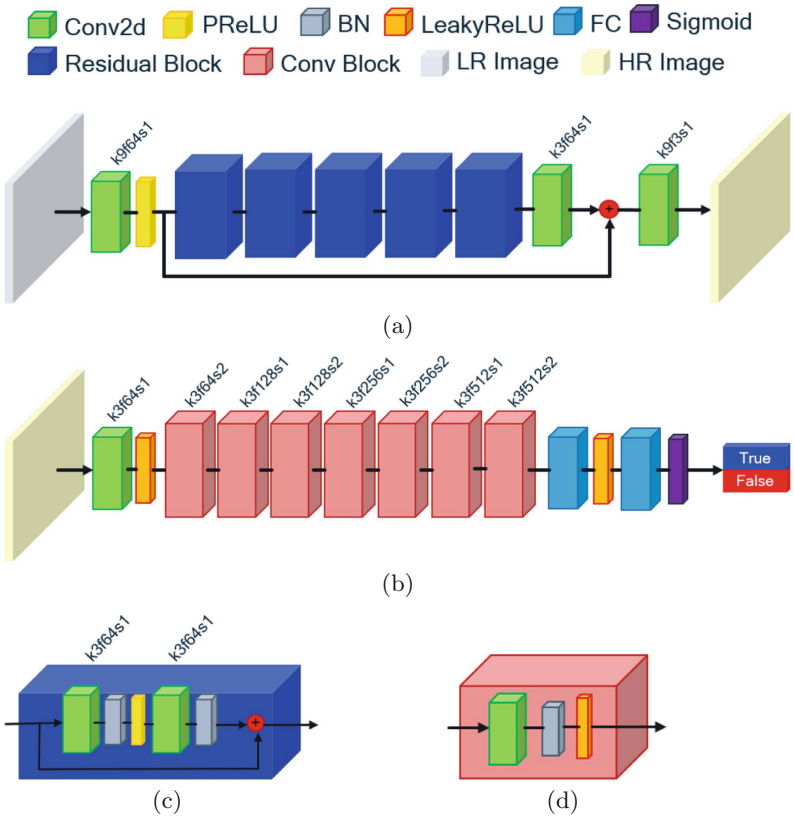

Fig. 3. Architecture of the generator and discriminator networks. (a) The generator network. (b) The discriminator network. (c) Residual block in the generator network. (d) Convolutional block in the discriminator network. We use a simple notation to indicate the Conv2d layer. For example, k9f64s1 indicates a convolutional layer with kernel size 9, number of filters 64 and stride 1.

As a binary classifier, the discriminator network is equipped with the cross entropy loss. Let us focus on the loss function of the generator network. Suppose $\mathcal{F}^{-1}\left(W^1\right)$ and $\mathcal{F}^{-1}\left(W^2\right)$ are real matrices of dimension $(2N+1)^2 \times N_t$, and the columns of $\mathcal{F}^{-1}\left(W^1\right)$ and $\mathcal{F}^{-1}\left(W^2\right)$ are x_i and y_i, $i = 1, 2, \ldots, N_t$. Following the formulation in [8], we define the loss function of the generator network as

$$l\left(G\right) = l_{MSE}\left(G\right) + 10^{-2}\, l_{VGG}\left(G\right) + 10^{-3}\, l_{Gen}\left(G\right). \tag{13}$$

In (13), l_{MSE} is the pixel-wise **MSE loss** defined as the sum of the squares of error at each pixel,

$$l_{MSE}(G) = \sum_{i=1}^{N_t} \|y_i - G(x_i)\|_2^2 .$$

The l_{VGG} is the **VGG loss** based on layers of the pre-trained VGG-19 network [15]. Let ϕ be a feature map of VGG-19 and s_ϕ be its size, then the VGG loss is the average of squares of Euclidean distances between the feature representations of y_i and $G(x_i)$

$$l_{VGG}(G) = s_\phi^{-1} \sum_{i=1}^{N_t} \|\phi(y_i) - \phi(G(x_i))\|_2^2 .$$

The generator network is expected to fool the discriminator network, so (13) contains l_{Gen} called **generative loss**. The l_{Gen} is defined based on the cross-entropy loss of the discriminator network

$$\sum_{i=1}^{N_t} \log\left[1 - D(G(x_i))\right], \tag{14}$$

where $D(G(x_i))$ means the binary classification result of the reconstructed HR image by the generator network G. In practice, we define

$$l_{Gen}(G) = \sum_{i=1}^{N_t} -\log D(G(x_i))$$

for better gradient behavior.

3 Experimental Results of Adaptive Basis from SRGAN

Let $D_0^1 = 10^{-2}$, $D_0^2 = 10^{-3}$. We solved for both W^1 and W^2 via spectral method with $N = 50$ and $N_t = 1500$, then train SRGAN with input data $\mathcal{F}^{-1}\left(W^1\right)$ and target data $\mathcal{F}^{-1}\left(W^2\right)$. The training of SRGAN includes two stages: (1) We train the generator G for 50 epochs to get a pretrained model; (2) We train the entire SRGAN for 200 epochs. Adam and SGD optimizers are applied to training of G and D respectively. We set batch size to be 32 and learning rate to be 10^{-4} for both optimizers. The training was carried out on a desktop with Nvidia graphics cards GTX 1080 Ti. We set $D_0^* = D_0^2$ in the following experiments.

Figure 4 shows two time slices of the input $\mathcal{F}^{-1}\left(W^2\right)$ (top) with $G\left(\mathcal{F}^{-1}\left(W^2\right)\right)$ (bottom) at $D = D_0^2$. Columnwise, it can be seen that thinner layers are created by the network G. Due to identical dimension constraint of the input and output images, the up-scaling layers in SRGAN [8] have been removed. This adaptation lowers the fidelity of the generated images, as we see in each column of Fig. 4. However, the SRGAN generated snapshots are only used to construct reduced basis.

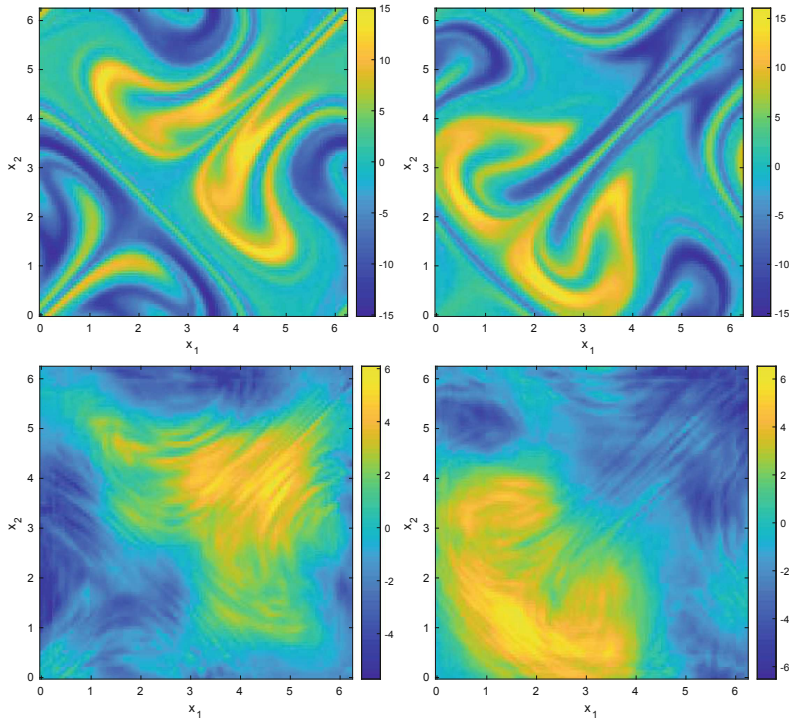

Fig. 4. Input (top) and SRGAN output (bottom) at $D_0^2 = 10^{-3}$.

The fact that sharp layers are generated by SRGAN training is more important for our task of computing residual diffusivity.

Set $D_0^1 = 10^{-2}$ and $D_0^2 = 10^{-3}$, the comparison of predictions of D_{11}^E by SVD and SRGAN assisted SVD is shown in Table 1. The number of adaptive basis is $m = 100$ for both methods.

Table 1. Comparison of $\hat{D}_{11,N}^{E,a}$ for flow (7) with $D_0^1 = 10^{-2}$, $D_0^2 = 10^{-3}$.

D_0		5×10^{-4}	4×10^{-4}	3×10^{-4}	2×10^{-4}	10^{-4}
$\hat{D}_{11,60}^{E}$		1.3847	1.3940	1.4105	1.4395	1.4951
$\hat{D}_{11,50}^{E,a}$	SVD	1.5258	1.5597	1.5969	1.6381	1.6854
	SRGAN	1.2429	1.2663	1.3056	1.3786	1.5293
Relative error	SVD	10.2%	11.9%	13.2%	13.8%	12.7%
	SRGAN	10.2%	**9.2%**	**7.4%**	**4.2%**	**2.3%**

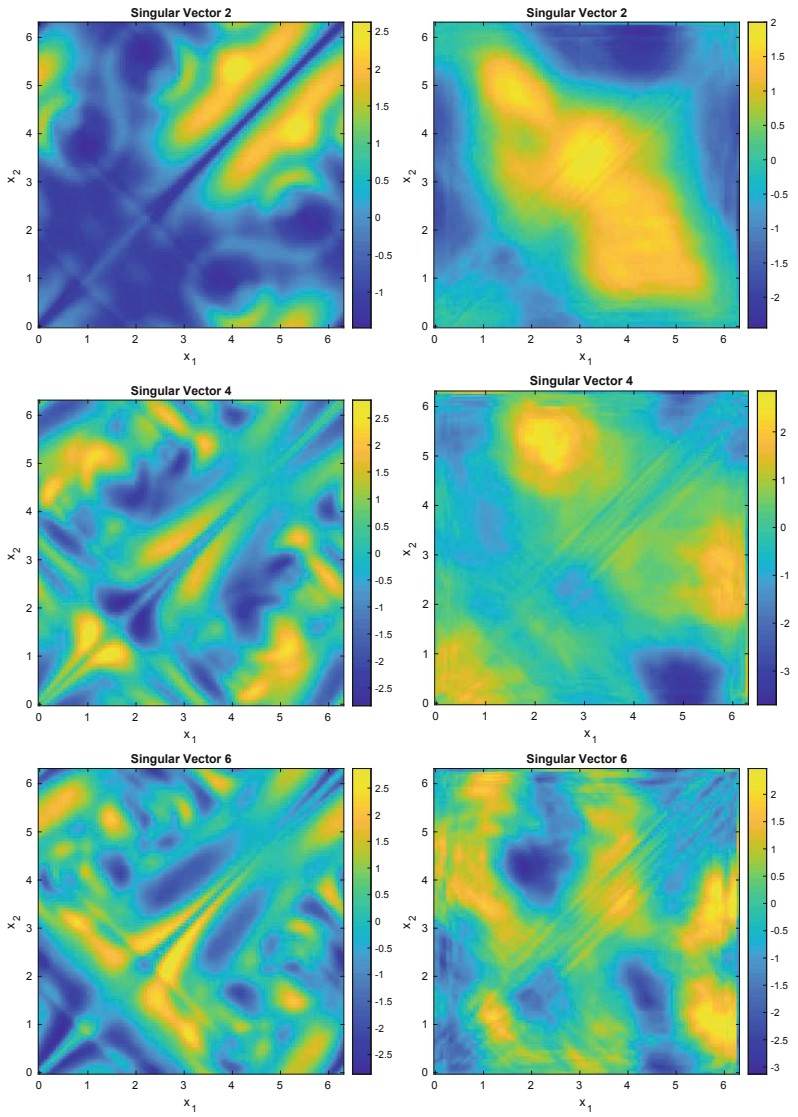

Fig. 5. Singular vectors of W^2 (left col.), $\mathcal{F}\left(G\left(\mathcal{F}^{-1}\left(W^2\right)\right)\right)$ (right col.).

When D_0^1 is closer to D_0^2, SRGAN assisted SVD may have even better predictions at smaller D_0. In Table 2, $D_0^1 = 5 \times 10^{-3}$, $D_0^2 = 10^{-3}$ and $N = 50$ and we predict the $\hat{D}_{11,60}^E$ at $D_0 = 3 \times 10^{-4}$, 2×10^{-4} and 10^{-4}. For $D_0^1 = 5 \times 10^{-3}$, $D_0^2 = 10^{-3}$, singular vectors of $\mathcal{F}\left(G\left(\mathcal{F}^{-1}\left(W^2\right)\right)\right)$ also have thinner structures than that of W^2, as shown in right column and left column of Fig. 5 respectively. Table 3 summarizes predictions for $D_0 = 2 \times 10^{-5}$ and 10^{-5} from $D_0^1 = 10^{-3}$, $D_0^2 = 10^{-4}$ and $N = 60$.

Table 2. Comparison of $\hat{D}^{E,a}_{11,N}$ for flow (7) with $D^1_0 = 5 \times 10^{-3}$, $D^2_0 = 10^{-3}$.

D_0		3×10^{-4}	2×10^{-4}	10^{-4}
$\hat{D}^{E}_{11,60}$		1.4105	1.4395	1.4951
$\hat{D}^{E,a}_{11,50}$	SVD	1.5969	1.6381	1.6854
	SRGAN	1.3111	1.3862	1.5015
Relative error	SVD	13.2%	13.8%	12.7%
	SRGAN	**7.0%**	**3.7%**	**0.4%**

Table 3. Comparison of $\hat{D}^{E,a}_{11,N}$ for flow (7) with $D^1_0 = 10^{-3}$, $D^2_0 = 10^{-4}$.

D_0		2×10^{-5}	10^{-5}
$\hat{D}^{E}_{11,60}$		1.6052	1.6301
$\hat{D}^{E,a}_{11,60}$	SVD	1.5107	1.5243
	SRGAN	1.6234	1.7120
Relative error	SVD	5.9%	6.5%
	SRGAN	**1.1%**	**5.0%**

4 Conclusions

Based on snapshots at two molecular diffusivity values, we trained an adapted super-resolution deep neural network (SRGAN) to model the internal layer sharpening effect of advection-diffusion equation as a nonlinear mapping. The mapping improves the quality of standard POD basis for low cost computation of residual diffusivity in chaotic flows. Though no other DNN model is known to assist POD in our setting, we shall explore how to improve the fidelity of the generated images in the current model.

Acknowledgements. The work was supported in part by NSF grants IIS-1632935, and DMS-1924548.

References

1. Bensoussan, A., Lions, J.-L., Papanicolaou, G.: Asymptotic Analysis for Periodic Structures. AMS Chelsea Publishing, Providence (2011)
2. Biferale, L., Cristini, A., Vergassola, M., Vulpiani, A.: Eddy diffusivities in scalar transport. Phys. Fluids **7**(11), 2725–2734 (1995)
3. Camassa, R., Wiggins, S.: Chaotic advection in a Rayleigh-Bénard flow. Phys. Rev. A **43**(2), 774–797 (1990)
4. Childress, S., Gilbert, A.: Stretch, twist, fold: the fast dynamo. Lecture Notes in Physics Monographs, No. 37. Springer (1995)
5. Fannjiang, A., Papanicolaou, G.: Convection enhanced diffusion for periodic flows. SIAM J. Appl. Math. **54**(2), 333–408 (1994)

6. Heinze, S.: Diffusion-advection in cellular flows with large Peclet numbers. Arch. Ration. Mech. Anal. **168**(4), 329–342 (2003)
7. Holmes, P., Lumley, J., Berkooz, G.: Turbulence, Coherent Structures, Dynamical Systems and Symmetry. Cambridge University Press, Cambridge (1998)
8. Ledig, C., Theis, L., Huszar, F., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR, pp. 105–114 (2017)
9. Lumley, J.: The structures of inhomogeneous turbulent flows. In: Atmospheric Turbulence and Radio Wave Propagation, pp. 166–178 (1967)
10. Lyu, J., Xin, J., Yu, Y.: Computing residual diffusivity by adaptive basis learning via spectral method. Numer. Math.: Theory Methods Appl. **10**(2), 351–372 (2017)
11. Majda, A., Kramer, P.: Simplified models for turbulent diffusion: theory, numerical modelling, and physical phenomena. Phys. Rep. **314**, 237–574 (1999)
12. Nasrollahi, L., Metz, S., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: ICLR (2016)
13. Novikov, A., Ryzhik, L.: Boundary layers and KPP fronts in a cellular flow. Arch. Ration. Mech. Anal. **184**(1), 23–48 (2007)
14. Quarteroni, A., Rozza, G. (eds.): Reduced Order Methods for Modeling and Computational Reduction. MS&A, vol. 9. Springer (2014)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
16. Sirovich, L.: Turbulence and the dynamics of coherent structures. Part I: coherent structures. Q. Appl. Math. **45**, 561–571 (1987)
17. Taylor, G.: Diffusion by continuous movements. Proc. London Math. Soc. **2**, 196–211 (1921)
18. Wang, Z., Xin, J., Zhang, Z.: Computing effective diffusivity of chaotic and stochastic flows using structure-preserving schemes. SIAM J. Numer. Anal. **56**(4), 2322–2344 (2018)
19. Xin, J., Yu, Y.: Sharp asymptotic growth laws of turbulent flame speeds in cellular flows by inviscid Hamilton-Jacobi models. Annales de l'Institut Henri Poincaré, Analyse Nonlineaire **30**(6), 1049–1068 (2013)
20. Xin, J., Yu, Y.: Front quenching in G-equation model induced by straining of cellular flow. Arch. Ration. Mech. Anal. **214**, 1–34 (2014)
21. Zu, P., Chen, L., Xin, J.: A computational study of residual KPP front speeds in time-periodic cellular flows in the small diffusion limit. Phys. D **311–312**, 37–44 (2015)