J. Fluid Mech. (2019), *vol.* 861, R4, doi:10.1017/jfm.2018.980

JFM RAPIDS
journals.cambridge.org/rapids

Deep learning of mixing by two 'atoms' of stratified turbulence

Hesam Salehipour[1,2,†] **and W. R. Peltier**[1]

[1]Department of Physics, University of Toronto, Toronto, ON M5S 1A7, Canada

[2]Autodesk Research, MaRS Discovery District, 661 University Ave, Toronto,
ON M5G 1M1, Canada

(Received 14 September 2018; revised 2 November 2018; accepted 2 December 2018;
first published online 4 January 2019)

Current global ocean models rely on *ad hoc* parameterizations of diapycnal mixing, in which the efficiency of mixing is globally assumed to be fixed at 20 %, despite increasing evidence that this assumption is questionable. As an ansatz for small-scale ocean turbulence, we may focus on stratified shear flows susceptible to either Kelvin–Helmholtz (KHI) or Holmboe wave (HWI) instability. Recently, an unprecedented volume of data has been generated through direct numerical simulation (DNS) of these flows. In this paper, we describe the application of deep learning methods to the discovery of a generic parameterization of diapycnal mixing using the available DNS dataset. We furthermore demonstrate that the proposed model is far more universal compared to recently published parameterizations. We show that a neural network appropriately trained on KHI- and HWI-induced turbulence is capable of predicting mixing efficiency associated with unseen regions of the parameter space well beyond the range of the training data. Strikingly, the high-level patterns learned based on the KHI and weakly stratified HWI are 'transferable' to predict HWI-induced mixing efficiency under much more strongly stratified conditions, suggesting that through the application of appropriate networks, significant universal abstractions of density-stratified turbulent mixing have been recognized.

Key words: ocean processes, stratified turbulence, turbulent mixing

1. Introduction

A vital mechanism for ventilating the abyssal ocean is that due to vertical mixing of deep, cold and nutrient-rich waters with shallower, warm and nutrient-scarce waters (Wunsch & Ferrari 2004). Mediated by the complex interactions of the internal wave field in the ocean interior, these mixing events emerge at the smallest scales and

[†] Email address for correspondence: h.salehipour@utoronto.ca

© Cambridge University Press 2019

https://doi.org/10.1017/jfm.2018.980 Published online by Cambridge University Press

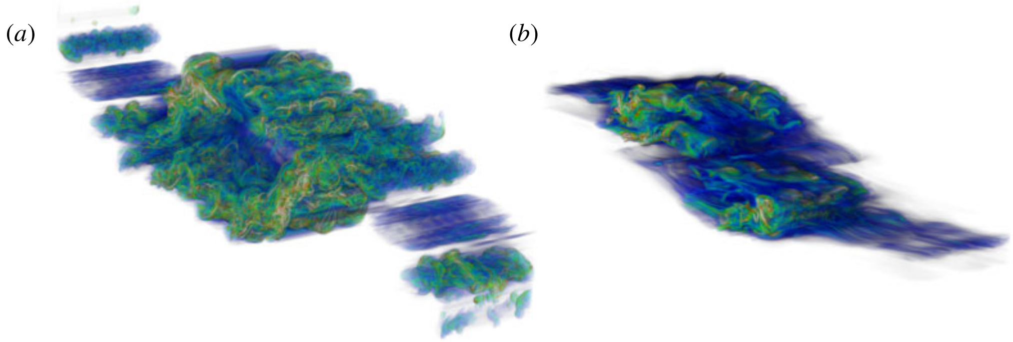

FIGURE 1. Schematic of two 'atoms' of turbulence in stratified shear flows associated with Kelvin–Helmholtz instability (*a*) and Holmboe instability (*b*).

undergo transition to turbulence that leads to an irreversible conversion of kinetic energy to potential energy. Figure 1 illustrates two flavours of these events that may develop in stratified shear flows, namely the Kelvin–Helmholtz instability (KHI) and the Holmboe wave instability (HWI) (refer to Salehipour, Caulfield & Peltier (2016*a*), Salehipour, Peltier & Caulfield (2018) for an in-depth comparison of these instability mechanisms). Despite their critical role in modulating the large-scale meridional overturning circulation of the ocean, the effect of these small-scale 'atoms' of ocean turbulence are often overly simplified by parameterizing them as involving a constant mixing rate that is always 20 % of the local dissipation rate of kinetic energy (Gregg *et al.* 2018). However, detailed numerical simulations and experimental measurements have collectively demonstrated significant departures from this fixed canonical value (see, for example, Monismith, Koseff & White 2018). Recent advances have been made in proposing alternative parameterizations of mixing efficiency based on forced and homogeneously stratified flows (see, for example, Mater & Venayagamoorthy 2014; Maffioli, Brethouwer & Lindborg 2016) or freely evolving and inhomogeneously stratified flows (see, for example, Salehipour *et al.* 2016*b*; Mashayek *et al.* 2017). Even in the latter conditions that are more realistic, the focus has been mainly on the fully turbulent flows for which the imprint of the initial 'atom' involved is minimal. For instance, the effect of ubiquitous large overturns (see, for example, figure 1) that are convectively unstable, leading to highly efficient mixing (with efficiency as high as 0.8–0.9), have been ignored in these earlier investigations.

Our main goal in this paper is to propose a data-driven approach that substantially improves previous parameterizations by encompassing all the data that is available based on direct numerical simulation (DNS) of these 'atoms'. To introduce this approach in the current study we focus on two of the distinct archetypical flavours of stratified turbulence. Section 2 presents the cornerstone of this paper, which involves a large compilation of data associated with KHI and HWI. These data are prepared in the manner described in § 3 to be further analysed in § 4 based on the application of 'deep learning' methods. We evaluate the predictions of this data-driven approach and compare them with previous methods in § 5. Our findings and discussion of future research directions are summarized in § 6.

https://doi.org/10.1017/jfm.2018.980 Published online by Cambridge University Press

2. The parent DNS dataset

We model a stratified mixing layer by assuming initial velocity and density distributions that have a hyperbolic tangent form, as

$$\overline{u}(z,0) = U_0 \tanh\left(\frac{z}{d}\right), \quad \overline{\rho}(z,0) = \rho_0 \left[1 - \tanh\left(\frac{z}{\delta}\right)\right], \tag{2.1a,b}$$

in the Boussinesq approximation such that $\rho_0 \ll \rho_r$ (note that here density represents departures from a hydrostatic state associated with ρ_r). Also, U_0 and ρ_0 denote, respectively, half the total velocity and density jumps across the shear layer (with a total depth $2d$) and the density layer (with a total depth of 2δ). As a result of this canonical setting, the dimensionless Boussinesq equations are governed by four important non-dimensional parameters, namely the (initial) Reynolds number Re; the bulk Richardson number Ri_b; the Prandtl number Pr; and the initial scale ratio R, defined together as

$$Re = \frac{U_0 d}{\nu}, \quad Ri_b = \frac{g\rho_0 d}{\rho_r U_0^2}, \quad Pr = \frac{\nu}{\kappa}, \quad R = \frac{d}{\delta}, \tag{2.2a–d}$$

in which ν is the kinematic viscosity, κ is the molecular diffusivity and g is the gravitational acceleration. Table 1 lists all the DNS analyses from which data will be employed for training and validation of the proposed artificial neural networks. These simulations have been thoroughly analysed and discussed previously in a number of recent publications on KHI (Salehipour & Peltier 2015; Salehipour, Peltier & Mashayek 2015; Salehipour *et al.* 2016*b*) and HWI (Salehipour *et al.* 2016*a*, 2018). For details of each simulation, interested readers are referred to the relevant papers.

For the supervised machine learning application to be discussed herein, we have further subdivided these datasets into training and validation sets with an approximate 80 %–20 % ratio, as indicated in table 1. Both these subsets include examples of flow evolution due to KHI and HWI. We have intentionally chosen the validation dataset to include all DNS cases with extreme values for their initial parameters, which are well beyond the range of similar parameters employed for training purposes. This enables us to investigate the extent to which our trained model is generalizable and thus robust. Note that our training dataset had a very limited number of HWI examples (compared to KHI) and that these examples are also at much smaller values of Ri_b and R.

3. Preprocessing of DNS data

The result of each three-dimensional DNS experiment associated with the evolution of either KHI or HWI is comprised of n_s snapshots in time, where each saved snapshot represents three-dimensional fields of flow quantities, namely the density ρ and velocity fields $\boldsymbol{u} = (u, v, w)$ (table 1 lists n_s for each simulation). The intensity of turbulent activity may be represented by the pointwise dissipation rate of total kinematic energy, $\epsilon(\boldsymbol{x}, t)$, defined as

$$\epsilon(\boldsymbol{x}, t) = 2\nu s_{ij} s_{ij}, \tag{3.1}$$

in which $s_{ij} = (\partial u_i/\partial x_j + \partial u_j/\partial x_i)/2$ is the total strain rate tensor. We may also reduce the above three-dimensional fields into a one-dimensional profile by performing horizontal averaging (to be denoted here by an overbar). Thus the horizontally

https://doi.org/10.1017/jfm.2018.980 Published online by Cambridge University Press

	Training dataset					Validation dataset				
	Re	Ri_b	Pr	R	n_s	Re	Ri_b	Pr	R	n_s
KHI	6 000	0.12	1	1	201	6000	0.12	16	1	298
	6 000	0.12	2	1	250	6 000	0.001	1	1	172
	6 000	0.12	4	1	200	6 000	0.22	1	1	185
	6 000	0.12	8	1	188	20 000	0.16	1	1	150
	6 000	0.005	1	1	409	30 000	0.16	1	1	169
	6 000	0.01	1	1	541					
	6 000	0.02	1	1	293					
	6 000	0.04	1	1	208					
	6 000	0.08	1	1	100					
	6 000	0.16	1	1	150					
	6 000	0.20	1	1	150					
	6 000	0.02	8	1	126					
	6 000	0.04	8	1	133					
	6 000	0.10	8	1	150					
	6 000	0.14	8	1	125					
	6 000	0.16	8	1	150					
	6 000	0.18	8	1	154					
	6 000	0.20	8	1	146					
	4 000	0.16	1	1	150					
	4 000	0.16	8	1	150					
	8 000	0.16	1	1	150					
	8 000	0.16	8	1	106					
	12 000	0.16	1	1	150					
	Re	Ri_b	Pr	R	n_s	Re	Ri_b	Pr	R	n_s
HWI	4000	0.16	8	2.83	250	6000	0.32	8	10	183
	6000	0.16	8	2.83	201	6000	0.32	8	5	267
	6000	0.16	8	5	195	6000	0.16	8	25	187
	6000	0.16	8	10	163					
	6000	0.08	8	5	214					
	6000	0.08	8	10	182					

TABLE 1. The collection of initial parameters (as defined in (2.2)) employed for conducting DNS experiments associated with either KHI or HWI. n_s indicates the number of saved snapshots for each individual simulation. The split between training and validation sets is also highlighted.

averaged dissipation rate of total kinematic energy, $\overline{\epsilon}(z, t)$, and the mean flow density, $\overline{\rho}(z, t)$, are defined as

$$\overline{\epsilon}(z, t) = \frac{1}{L_x L_y} \int \epsilon(\boldsymbol{x}, t)\, \mathrm{d}x\, \mathrm{d}y, \quad \overline{\rho}(z, t) = \frac{1}{L_x L_y} \int \rho(\boldsymbol{x}, t)\, \mathrm{d}x\, \mathrm{d}y, \qquad (3.2a,b)$$

where L_x and L_y denote the size of the computational domain in the streamwise and spanwise directions.

The (generally) time-dependent mixing efficiency, $\mathscr{E}$, may be computed precisely by invoking the concept of irreversible diapycnal mixing (originally introduced by Winters *et al.* 1995), which relies on a special kind of reduction operator, namely a three-dimensional sorting of the density field into a notional state that is strictly stably

https://doi.org/10.1017/jfm.2018.980 Published online by Cambridge University Press

stratified (Peltier & Caulfield 2003), and is defined as

$$\mathscr{E}(t) = \frac{\mathscr{M}(t)}{\mathscr{M}(t) + \langle \overline{\epsilon}(z, t) \rangle}, \tag{3.3}$$

where $\langle\,\rangle$ denotes vertical averaging. For a precise definition of $\mathscr{M}(t)$ refer to (2.18) of Salehipour *et al.* (2016*a*) and the cited discussions therein. The required parallel implementation of the sorting procedure is described in Salehipour *et al.* (2015) (see, for example, their figure 1). Such an elaborate technique for calculating $\mathscr{E}$ is only viable in numerical simulations such as those employed in this work because in practice oceanographers only measure one-dimensional profiles in depth and are therefore unable to perform the same analysis. Indeed, there is a similar subtlety in defining the ‘background’ buoyancy frequency, $N^2(z, t)$, as described in Salehipour & Peltier (2015) (see their discussion leading to (2.23)) and more recently in Arthur *et al.* (2017). In order to distinguish between irreversible mixing and reversible stirring, $N^2(z, t)$ must be defined based on the same notional state obtained by the three-dimensional sorting procedure. For consistency with common practice in oceanography, in this paper we may define N^2 using the mean flow density introduced in (3.2) such that $N^2(z, t) = -(g/\rho_r)\,\mathrm{d}\overline{\rho}/\mathrm{d}z$.

We seek a mapping between the instantaneous vertical profiles of $\overline{\epsilon}(z, t_0)$ and $N^2(z, t_0)$ (i.e. at a given time t_0) and the precisely computed values of mixing efficiency, $\mathscr{E}(t_0)$. Once the network is trained, this mapping would essentially reveal a reduction operator that is conceivably very different from a straightforward vertical averaging; one which also incorporates the structural pattern and length scales that implicitly exist and are thus ‘hidden’ in these profiles. Thus the inputs to our artificial neural network are tuples of $(\mathcal{X}_1, \mathcal{X}_2)$ defined respectively as

$$\mathcal{X}_1(z, t_0) \equiv \frac{\overline{\epsilon}(z, t_0)}{\kappa \displaystyle\int N^2(z, t_0)\,\mathrm{d}z}, \quad \mathcal{X}_2(z, t_0) \equiv \frac{N^2(z, t_0)}{\displaystyle\int N^2(z, t_0)\,\mathrm{d}z}. \tag{3.4a,b}$$

Furthermore, the true ‘labels’ in our supervised learning setting are the instantaneous values of mixing efficiency, namely

$$\mathcal{Y}(t_0) \equiv \mathscr{E}(t_0). \tag{3.5}$$

It is important to highlight that $(\mathcal{X}_1, \mathcal{X}_2)$ appear in a normalized form to render $\overline{\epsilon}(z, t_0)$ and $N^2(z, t_0)$ comparable in terms of their dimensionality and physical relevance, and furthermore to extend the applicability of the trained network to oceanographic profiles. In (3.4), $\mathcal{X}_1$ represents the vertical profile of kinetic energy dissipation rate relative to the molecular diffusion rate in the absence of mean flow shear. The vertical profiles are assumed to have a fixed length of $i = 512$ points in which the ‘dead’ regions of the simulation (near top and bottom boundaries) have been excluded by focusing on the largest segments of the profiles where $|\mathrm{d}\overline{\rho}(z)/\mathrm{d}z| \geqslant 10^{-3}$. This approach is analogous to identifying ‘patches’ of turbulence from DNS calculations (Smyth, Moum & Caldwell 2001).

4. Deep convolutional neural networks

Deep learning methods involve a multilayer stacking of simple modules that perform linear or nonlinear input–output mappings whose weights and biases are

https://doi.org/10.1017/jfm.2018.980 Published online by Cambridge University Press

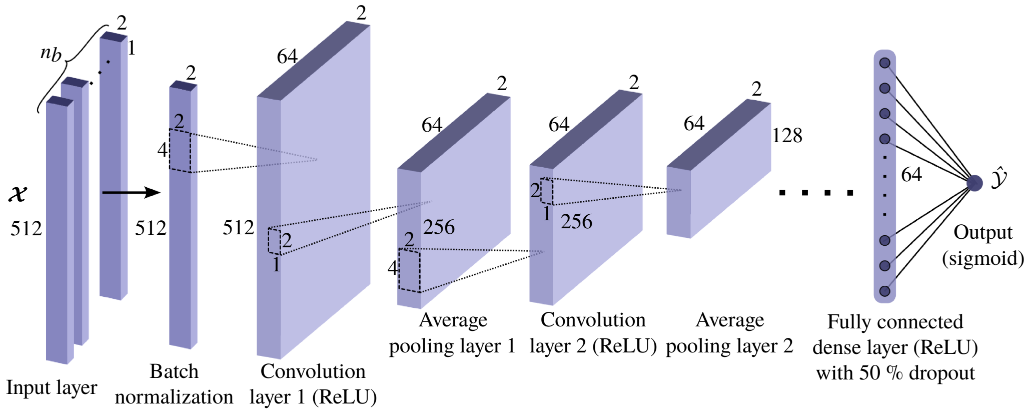

FIGURE 2. Illustration of the type of convolutional neural network investigated in this paper with increasing number of stacked layers. Refer to the text for definition of the various layers. For any given example data point, the tuples of $\boldsymbol{\mathcal{X}} = (\mathcal{X}_1, \mathcal{X}_2)$ are of size (512×2), representing vertical profiles of normalized $\epsilon(z)$ and $N^2(z)$ as defined in (3.4).

subject to 'training' through an optimization procedure (LeCun, Bengio & Hinton 2015). These techniques became widely popularized after Krizhevsky, Sutskever & Hinton (2012), from University of Toronto, employed a 'deep convolutional neural network' to classify a dataset of 1.2 million images and won first place in the 2012 ImageNet competition. A convolutional neural network (CNN) is a special type of neural network architecture that relies on the convolution operator in lieu of general matrix multiplication in at least one layer of its configuration (Goodfellow, Bengio & Courville 2016). In two dimensions, this operator may be defined as

$$\widetilde{\boldsymbol{\mathcal{X}}}(i,j) = (\boldsymbol{\mathcal{X}} * \mathscr{K})(i,j) = \sum_{m} \sum_{n} \boldsymbol{\mathcal{X}}(m,n)\mathscr{K}(i-m, j-n), \qquad (4.1)$$

where the input field $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{i\times j}$ and the convolution kernel $\mathscr{K} \in \mathbb{R}^{m\times n}$ is represented by characteristic filter lengths of size $m \leqslant i$ and $n \leqslant j$. A convolved (i.e. filtered) field is constructed by traversing the kernel $\mathscr{K}$ over the dimensions of $\boldsymbol{\mathcal{X}}$. To keep the filtered field, $\widetilde{\boldsymbol{\mathcal{X}}}$, the same size as $\boldsymbol{\mathcal{X}}$, zero-padding is often employed.

Figure 2 illustrates the schematic configuration of the selected neural network architectures to be employed in this paper, which may consist of one to seven convolution layers labelled as CNN1 to CNN7. Each configuration receives the input $\boldsymbol{\mathcal{X}} = (\mathcal{X}_1, \mathcal{X}_2) \in \mathbb{R}^{512\times 2}$, as defined in (3.4), and passes it to a 'batch normalization' layer (Ioffe & Szegedy 2015) that, for any given 'batch' of the training dataset (with size n_b), normalizes $\mathcal{X}_1$ and $\mathcal{X}_2$ individually by subtracting the batch mean and dividing by its variance. The batch-normalized data are then subsequently fed into a series of convolution layers each having 64 filters with a kernel $\mathscr{K} \in \mathbb{R}^{4\times 2}$. Each convolution kernel undergoes a nonlinear activation function of type $f(x) = \max(0, x)$, also known as a Rectified Linear Unit (ReLU). The output of each convolution layer is followed by an 'average pooling' operator which effectively reduces the size of the profile by half through averaging any two adjacent data in the profile (i.e. averaging window of size (2×1)). The reduced outputs are then reshaped appropriately to be fed into a dense (or fully connected) layer with 64 neurons that also employs

https://doi.org/10.1017/jfm.2018.980 Published online by Cambridge University Press

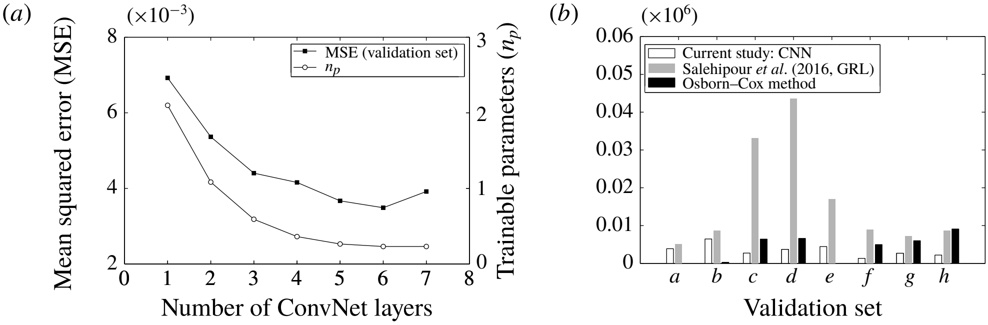

FIGURE 3. (*a*) Comparing the predictions of CNN1–CNN7 on the validation set. (*b*) The results of CNN6 (this study) are compared quantitatively with other methods in terms of mean squared error for various sets of validation data, labelled (*a*–*h*) as per figure 5.

ReLU as its nonlinearity function. To avoid overfitting and to improve the model predictions, we regularize the network by a method known as 'dropout' (Hinton *et al.* 2012), which randomly turns off 50 % of the neurons, thereby eliminating their contribution in the 'backpropagation' procedure (a method to apply the chain rule to derive gradients of the loss function with respect to trainable parameters in the network) of the optimization process. As a result the network is forced to learn robust features that emerge more frequently in the random subsets during training. Finally we use a single neuron to represent the network output, $\hat{\mathcal{Y}}$. We have chosen a sigmoid activation function for the output layer because efficiency values must be between 0 and 1. We have used the Adam optimizer (Kingma & Ba 2014) for performing stochastic gradient descent to minimize the loss function, defined as the mean squared error $\sum_{n_b}(\mathcal{Y}-\hat{\mathcal{Y}})^2/n_b$, where the batch size is set as $n_b = 100$.

Notice that the number of trainable parameters, n_p, in the network decreases from CNN1 to CNN6 and increases slightly from CNN6 to CNN7 (see figure 3*a*). Taking CNN1 for instance, the network has copious neurons that link the first convolution layer (after pooling) to a fully connected dense layer, leading to $n_p > 2 \times 10^6$. As the network becomes deeper, increasingly more structure is built into the network due to the locality of the convolution operators that may be contrasted to the global connectivity of the dense layers. By definition (4.1), the convolution layer shares its kernel parameters ($m \times n$ weights for a two-dimensional kernel) across a given 'tiling' of its input field. Notice that for deeper convolution layers whose inputs are derivative of the previous pooling operator with q filters, there are $m \times n \times q$ trainable weights and one trainable bias that are shared by each tiling of the convolution layer. For instance in CNN2, $n_p \sim 64 \times (4 \times 2 + 1)_{conv1} + 64 \times (4 \times 2 \times 64 + 1)_{conv2} + (128 \times 2 \times 64 \times 64)_{dense}$. Refer to Goodfellow *et al.* (2016, Chap. 9) for further details on parameter sharing in convolutional networks. Figure 3(*a*) also evaluates the effect of increasing the depth of the network in so far as the validation data is concerned. Clearly CNN6 outperforms others, which might be explained by the observed saturation of network training capacity also shown in this figure.

Obviously there are many parameters (or hyper-parameters) that we have assumed to be fixed within the above networks. Moreover, there are many other types of deep neural networks (DNN) (Goodfellow *et al.* 2016) that could be exploited – an alternatively good candidate being the recurrent neural network, which enables handling input sequences of vertical profiles with varying dimensions. We only note

https://doi.org/10.1017/jfm.2018.980 Published online by Cambridge University Press

in passing that we have also investigated a deep feed-forward neural network (that consists of deep stacking of dense layers only) in this work, but the CNN results were substantially more accurate. We wish to emphasize that our focus in this paper has not been to find the optimal configuration (or architecture) for producing the least possible error on the validation set. This paper rather intends to make the first step in introducing the idea of employing deep learning methods for the purpose of parameterizing subgrid scale processes using DNS datasets. We have open-sourced our code and postprocessed dataset in the hope of encouraging the community to further enhance such a data-driven approach to parameterization. Indeed, we believe the ultimate success of these efforts will rely upon a cohesive community-driven collaboration. Fortunately, these data-driven ideas are being embraced most recently in the broader context of Earth system modelling (Schneider *et al.* 2017).

5. Results

It is expected that the network learns the inherent (and intricate) patterns amongst the spatial structures of $N^2(z)$ and $\overline{\epsilon}(z)$ and maps it properly to the true DNS-based values of mixing efficiency. Figure 4(*a*i–*a*iv) illustrates the evolution of the local structures within these profiles for one KHI and one HWI case among the unseen validation set; more in-depth discussion on these structures are provided in Salehipour *et al.* (2016*a*) (see, for example, their figure 12). The other panels in this figure illustrate the corresponding outputs of the first convolution layer that are filtered by various kernels whose weights and biases have been learned during the training procedure based on the CNN6 network (or CNN for brevity). For brevity and clarity, only the five most descriptive filtered outputs (out of 64), which have been hand-picked, are shown here and denoted respectively as filters (*b*, *c*, *d*, *e*, *f*) as per their labels in figure 4.

It appears that filter *b* reproduces the structure of its input profiles merely at a different amplitude through, for example, a simple linear scaling. Filters *c* and *d*, on the other hand, seem to produce an interesting attenuation of less important (insofar as mixing efficiency is concerned) segments of the profile. These segments include regions with negligible turbulent dissipation (see figure 4(*c*i,*c*ii,*d*i,*d*ii) or regions with $N^2(z, t) \approx 0$ (see figure 4(*c*iii,*c*iv,*d*iii,*d*iv). In contrast, filters (*e*) and (*f*) have been trained to detect and isolate features of the input profiles that contribute more prominently to irreversible mixing. Figure 4(*c*ii,*d*ii) illustrates zero output fields for HWI, implying that our basic approach to isolate quiescent regions of the profile (discussed in § 3) needs hardly any improvement for the HWI case, unlike that for the KHI case. It is crucial to note that the identification of relevant regions with distinct dynamical effects on mixing has emerged inevitably through the training procedure of our deep neural network, and is surprisingly reminiscent of (at least qualitatively) the identification into quiescent (by filter *c* and *d*), intermittent (for example by filter *f*) and turbulent patches (for example by filter *e*) proposed by Portwood *et al.* (2016) in the context of homogeneous stratified turbulence. We therefore believe a similar approach based on a convolutional neural network could be ideally suited to classify a turbulent field into these distinct regions.

As demonstrated in figure 4(*a*i–*a*iv), HWI and KHI have categorically different localization of $N^2(z)$ and $\overline{\epsilon}(z)$. It is nonetheless very interesting that a single convolution kernel, that has been trained with a disproportionately higher number of KHI examples, results in extracted features that are meaningful (and not distorted) for HWI, regardless of this difference in localization of these vertical profiles. In

https://doi.org/10.1017/jfm.2018.980 Published online by Cambridge University Press

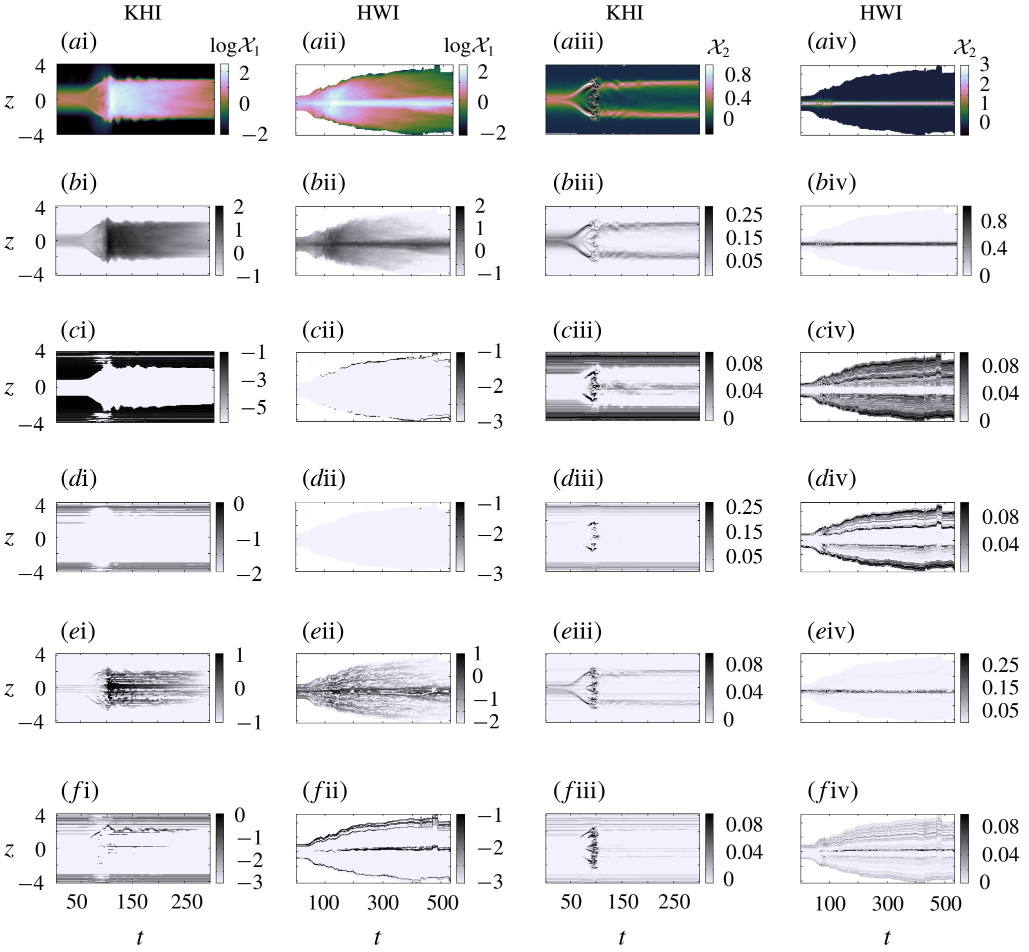

FIGURE 4. (*a*i–*a*iv) Spatiotemporal structure of normalized $N^2(z, t)$ and $\overline{\epsilon}(z, t)$ defined as $\mathcal{X}_1$ and $\mathcal{X}_2$ in (3.4) (at a given instance t_0) for a representative KHI case (labelled as case (*c*) in figure 5) and a representative HWI case (labelled as case (*g*) in figure 5). The following rows include outputs of the first convolution layer in CNN6 that demonstrate filtered versions of their corresponding field (either $\mathcal{X}_1$ in log-scale or $\mathcal{X}_2$). In the text, these filters are referred to by the alphabetical part of their labels. For instance filter (*b*) produces panels (*b*i–*b*ii) associated with $\mathcal{X}_1$ for the KHI and HWI cases, respectively.

other words, the extracted features represent repeated patterns that are not tied to a specific position in the input field $\boldsymbol{\mathcal{X}}$. This might be explained by recalling that the CNN architecture has the important property of parameter sharing that is inherent in the convolution operator. This property implies a strong prior knowledge that essentially assumes the nearby and local values of co-located $\epsilon(z)$ and $N^2(z)$ may have self-similar patterns that are relevant in approximating the induced mixing efficiency. Moreover, the pooling operator encourages the network to learn features that are translationally invariant. As a result, the CNN network is able to detect similar patterns even when the characteristic structure of the normalized $\epsilon(z)$ and $N^2(z)$ are localized very differently; an issue that becomes particularly relevant to the two 'atoms' of stratified turbulence investigated herein.

https://doi.org/10.1017/jfm.2018.980 Published online by Cambridge University Press

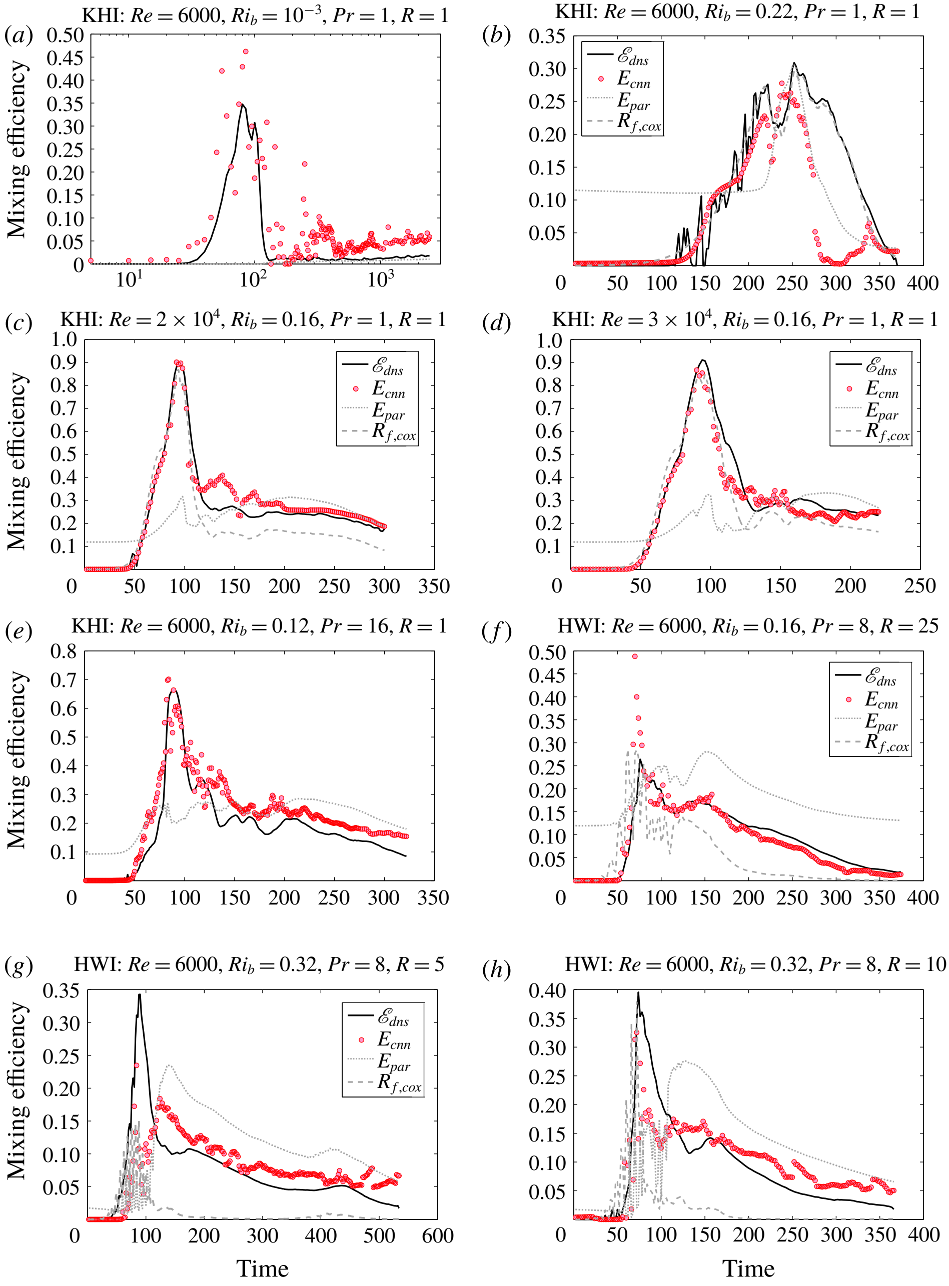

FIGURE 5. Comparing the precise calculation of mixing efficiency, $\mathscr{E}_{dns}$ (3.3) with those predicted by a convolutional neural network, denoted by E_{cnn} (this study), and the most recent multiparameter parameterization (Salehipour *et al.* 2016*b*), denoted by E_{par}, and that estimated by the Osborn–Cox method, $R_{f,cox}$ (defined in appendix A, not available in panels (*a*,*e*)). Each panel illustrates temporal evolution of mixing efficiency due to either KHI or HWI under the various specified initial conditions for the validation set listed in table 1.

https://doi.org/10.1017/jfm.2018.980 Published online by Cambridge University Press

Next we assess to what extent the learned mapping function can be generalized to the unseen validation dataset. Figure 5 demonstrates the prediction results of our deep learning approach based on the CNN network (see figure 2), denoted by E_{cnn} as well as their true DNS-based values (denoted by $\mathscr{E}_{dns}$). For further comparison, this figure also includes estimates of mixing efficiency due to (i) the Osborn–Cox method, denoted by $R_{f,cox}$ (refer to appendix A for its definition), and (ii) the multiparameter parameterization of Salehipour *et al.* (2016*b*), denoted by E_{par}. The latter relies on two dimensionless parameters, the buoyancy Reynolds number $Re_b(t) = \langle\overline{\epsilon}(z, t)\rangle/(\nu\langle N^2(z, t)\rangle)$ and a bulk Richardson number $Ri(t) = \langle N^2(z, t)\rangle/\langle(\mathrm{d}\overline{\boldsymbol{u}}(z, t)/\mathrm{d}z)^2\rangle$. Figure 3(*b*) provides the mean squared error associated with these various estimates applied to each validation set, labelled as in figure 5. As mentioned earlier in § 2, we have intentionally chosen the validation dataset to include simulations with extreme initial parameters (to avoid trivial 'interpolation' between the training dataset). The associated results in figure 5 therefore consist of (*a*) very weakly and (*b*) very strongly stratified KHI, (*c*,*d*) KHI with extremely high initial values of Re, (*e*) KHI at high Pr, (*f*) HWI with a density layer that is very much sharper than its shear layer with $R = 25$ and (*g*,*h*) very strongly stratified HWI.

Our CNN-based predictions are markedly superior to those predicted by the Osborn–Cox model or indeed by any published parameterization of mixing efficiency, including our own most recent suggestion (Salehipour *et al.* 2016*b*). The predictions of E_{cnn} are exceptionally accurate at higher Reynolds (cases *c*,*d*) and Prandtl (case *e*) numbers, considering that the ensuing turbulence is significantly more energetic than those employed for training purposes. Perhaps most surprising is the reasonable accuracy of E_{cnn} for HWI-induced turbulence under strong stratification with $Ri_b = 0.32$ (cases *g*,*h*). As discussed in depth in Salehipour *et al.* (2018), unlike KHI, which is quite sensitive to its initial conditions, HWI reveals the striking characteristics that (regardless of its initial conditions) it self-organizes towards a critical state with a particular distribution of mean density and velocity (i.e. a critical state associated with a high probability density function of the gradient Richardson number, $Ri_g(z) = N^2(z)/(\mathrm{d}\overline{u}/\mathrm{d}z)^2$ near 1/4). Furthermore, the mechanics involved in this self-organization are entirely different for a given Ri_b, or even depending on the thickness ratio R. Remarkably, however, the universal common features discovered by the network reveal plausible transferability to HWI at significantly higher Ri_b (cases *g*,*h*) or R (case *f*), as if the network has learned the pathways available for self-organization! While the predictions of E_{cnn} in (case *a*) for KHI under extremely weak stratifications might have the largest variance compared to $\mathscr{E}_{dns}$, it is nonetheless very interesting that the increasing trend of $\mathscr{E}_{dns}$ with time is correctly predicted by CNN. The mixing efficiencies under such weak stratifications (for example, the training case with $Ri_b = 5 \times 10^-3$) are so small that they do not impact adversely the mean squared error loss function employed during training. This may explain the higher variance of E_{cnn} observed in case (*a*) despite its low MSE as shown in figure 3(*b*). For strongly stratified KHI (case *b*), CNN predicts accurately the evolution of mixing efficiency towards its maximum, but suggests a more rapid decay than that inferred from $\mathscr{E}_{dns}$. The underlying reason for this relative inaccuracy of E_{cnn} during a short period is not known to us.

Although $R_{f,cox}$ relies on additional information regarding the scalar dissipation field that is not required as an input by our deep neural network, its predictions are not consistently accurate. Most worrisome is perhaps for strongly stratified HWI (see cases *g*,*h*), where $R_{f,cox}$ predicts essentially negligible mixing, a prediction that is simply erroneous. For KHI cases $R_{f,cox}$ estimates are reasonable, albeit being less

https://doi.org/10.1017/jfm.2018.980 Published online by Cambridge University Press

accurate than E_{cnn} (see figure 3*b*) with the exception of case (*b*), where Osborn–Cox estimates are almost perfect. The parameterization of Salehipour *et al.* (2016*b*) has been constructed entirely based on the fully turbulent flows that are only subject to KHI. As a result and as expected, E_{par} systematically overestimates the efficiency for HWI cases and fails to capture the high efficiencies attained during the convectively unstable roll-up of primary instabilities of either KHI or HWI type.

6. Summary

Using properly normalized vertical structures of $N^2(z)$ and $\overline{\epsilon}(z)$, we have proposed a data-driven approach based on deep convolutional neural networks (CNN) that can accurately predict the value of mixing efficiency for the entire life cycle of KHI and HWI (i.e. two 'atoms' of turbulence in stratified flows) beyond the range of initial conditions that have been employed for training the network. The large overturns that are convectively unstable are no longer ignored in such an approach. We have also shown that the results of the CNN model for KHI and HWI are more reliable and accurate than those based on the Osborn–Cox method.

Deep neural networks have a compositional hierarchy in which low-level features are composed to form higher-level features (for example, for image recognition, the first layers of a CNN detect basic abstract features such as edges, then deeper layers combine edges to form motifs, and subsequent layers assemble parts from motifs). We believe the proposed CNN model has similarly discovered such an 'abstract' level of stratified turbulence with characteristics that are so universal that even with a small portion of data associated with HWI, the generic behaviour of its induced mixing efficiency can be predicted robustly for wildly different initial conditions.

What makes such a data-driven approach especially appealing is its capability to become increasingly more accurate, robust and generic. This is foreseen to be achieved by (i) experimenting with many other types of DNN architectures, (ii) tuning the hyper-parameters (of which there are many) and, perhaps most importantly, (iii) further enriching the training dataset by adding additional examples of KHI and HWI, as they become available, or perhaps more excitingly by including more 'atoms' of ocean turbulence such as those induced by, for example, double-diffusion, Taylor and Rayleigh–Taylor instabilities. Another exciting future direction would involve using observed profiles (either from laboratory or real environments) to estimate mixing efficiency based on the proposed model, especially due to the relative inaccuracies of the Osborn–Cox method.

Acknowledgements

H.S. acknowledges the SOSCIP TalentEdge postdoctoral fellowship and the support from Autodesk Research. H.S. is especially grateful to Dr F. Iorio. The source codes, based on Google's TensorFlow library, are publicly available at the github repository https://github.com/hsalehipour/dl-Osborn. The research of W.R.P. is supported by NSERC Discovery Grant A9627 and SOSCIP.

Appendix A

An alternative measure of mixing efficiency is the flux Richardson number, R_f, which assumes that the buoyancy flux, $\mathbb{B}$, is an appropriate quantity to describe diapycnal mixing $\mathscr{M}$:

$$R_f(t) = \frac{\mathbb{B}(t)}{\mathbb{B}(t) + \langle\overline{\epsilon}(z, t)\rangle}. \tag{A 1}$$

https://doi.org/10.1017/jfm.2018.980 Published online by Cambridge University Press

A widely used method for estimating mixing efficiency from observational profiles (see, for example, Monismith *et al.* 2018) is that following Osborn & Cox (1972). In this method $\mathbb{B}$ is estimated using the scalar dissipation rate $\chi = 2\kappa\langle\overline{|\boldsymbol{\nabla}\rho'|^2}\rangle$ as

$$\mathbb{B}_{cox}(t) = \frac{\chi}{2}\langle N^2\rangle\left(\frac{\mathrm{d}\overline{\rho}}{\mathrm{d}z}\right)^{-2}, \tag{A 2}$$

where turbulent fluctuations of the density field are defined as $\rho'(\boldsymbol{x}, t) = \rho(\boldsymbol{x}, t) - \overline{\rho}(z, t)$. Therefore $R_{f,cox}$ (using the 'Cox' method), plotted in figure 5 based on the original DNS data, is computed by inserting $\mathbb{B}_{cox}$ into (A 1).

References

Arthur, R. S., Venayagamoorthy, S. K., Koseff, J. R. & Fringer, O. B. 2017 How we compute N matters to estimates of mixing in stratified flows. *J. Fluid Mech.* **831**, R2.

Goodfellow, I., Bengio, Y. & Courville, A. 2016 *Deep Learning*. MIT Press.

Gregg, M. C., D'Asaro, E. A., Riley, J. J. & Kunze, E. 2018 Mixing efficiency in the ocean. *Annu. Rev. Mar. Sci.* **10**, 443–473.

Hinton, G. E, Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. R. 2012 Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580.

Ioffe, S. & Szegedy, C. 2015 Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167.

Kingma, D. P. & Ba, J. 2014 Adam: a method for stochastic optimization. arXiv:1412.6980.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. 2012 Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105.

LeCun, Y., Bengio, Y. & Hinton, G. I. 2015 Deep learning. *Nature* **521** (7553), 436–444.

Maffioli, A., Brethouwer, G. & Lindborg, E. 2016 Mixing efficiency in stratified turbulence. *J. Fluid Mech.* **794**, R3.

Mashayek, A., Salehipour, H., Bouffard, D., Caulfield, C. P., Ferrari, R., Nikurashin, M., Peltier, W. R. & Smyth, W. D. 2017 Efficiency of turbulent mixing in the abyssal ocean circulation. *Geophys. Res. Lett.* **44** (12), 6296–6306.

Mater, B. D. & Venayagamoorthy, S. K. 2014 The quest for an unambiguous parameterization of mixing efficiency in stably stratified geophysical flows. *Geophys. Res. Lett.* **41** (13), 4646–4653.

Monismith, S. G., Koseff, J. R. & White, B. L. 2018 Mixing efficiency in the presence of stratification: when is it constant? *Geophys. Res. Lett.* **45** (11), 5627–5634.

Osborn, T. R. & Cox, C. S. 1972 Oceanic fine structure. *Geophys. Astrophys. Fluid Dyn.* **3** (1), 321–345.

Peltier, W. R. & Caulfield, C. P. 2003 Mixing efficiency in stratified shear flows. *Annu. Rev. Fluid Mech.* **35**, 135–167.

Portwood, G. D., de Bruyn Kops, S. M., Taylor, J. R., Salehipour, H. & Caulfield, C. P. 2016 Robust identification of dynamically distinct regions in stratified turbulence. *J. Fluid Mech.* **807**, R2.

Salehipour, H., Caulfield, C. P. & Peltier, W. R. 2016*a* Turbulent mixing due to the Holmboe wave instability at high Reynolds number. *J. Fluid Mech.* **803**, 591–621.

Salehipour, H. & Peltier, W. R. 2015 Diapycnal diffusivity, turbulent Prandtl number and mixing efficiency in Boussinesq stratified turbulence. *J. Fluid Mech.* **775**, 464–500.

Salehipour, H., Peltier, W. R. & Caulfield, C. P. 2018 Self-organized criticality of turbulence in strongly stratified mixing layers. *J. Fluid Mech.* **856**, 228–256.

Salehipour, H., Peltier, W. R. & Mashayek, A. 2015 Turbulent diapycnal mixing in stratified shear flows: the influence of Prandtl number on mixing efficiency and transition at high Reynolds number. *J. Fluid Mech.* **773**, 178–223.

https://doi.org/10.1017/jfm.2018.980 Published online by Cambridge University Press

SALEHIPOUR, H., PELTIER, W. R., WHALEN, C. B. & MACKINNON, J. A. 2016*b* A new characterization of the turbulent diapycnal diffusivities of mass and momentum in the ocean. *Geophys. Res. Lett.* **43** (7), 3370–3379.

SCHNEIDER, T., LAN, S., STUART, A. & TEIXEIRA, J. 2017 Earth System Modeling 2.0: a blueprint for models that learn from observations and targeted high-resolution simulations. *Geophys. Res. Lett.* **44**, 12396–12417.

SMYTH, W. D., MOUM, J. & CALDWELL, D. 2001 The efficiency of mixing in turbulent patches: inferences from direct simulations and microstructure observations. *J. Phys. Oceanogr.* **31**, 1969–1992.

WINTERS, K. B., LOMBARD, P. N., RILEY, J. J. & D'ASARO, E. A. 1995 Available potential energy and mixing in density-stratified fluids. *J. Fluid Mech.* **289**, 115–128.

WUNSCH, C. & FERRARI, R. 2004 Vertical mixing, energy, and the general circulation of the oceans. *Annu. Rev. Fluid Mech.* **36**, 281–314.

https://doi.org/10.1017/jfm.2018.980 Published online by Cambridge University Press