JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2021MS002817

Parameterization of Stochastically Entraining Convection Using Machine Learning Technique

Jihoon Shin[1] and **Jong-Jin Baik**[1]

[1]School of Earth and Environmental Sciences, Seoul National University, Seoul, South Korea

Key Points:

- A stochastic mixing model with a machine learning technique is proposed for mass flux convection schemes
- The deep neural network is found to predict entrainment and detrainment rates better than previously proposed parameterizations
- The single-column model simulations with the new mixing model produce realistic mean and variance of various convective updraft properties

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

J.-J. Baik,
jjbaik@snu.ac.kr

Citation:

Shin, J., & Baik, J.-J. (2022). Parameterization of stochastically entraining convection using machine learning technique. *Journal of Advances in Modeling Earth Systems*, *14*, e2021MS002817. https://doi.org/10.1029/2021MS002817

Received 9 SEP 2021
Accepted 13 APR 2022

Abstract A stochastic mixing model with a machine learning technique is proposed for mass flux convection schemes. The model consists of the stochastic differential equations (SDEs) for the fractional entrainment rate, fractional detrainment rate, fractional dilution rate, and vertical acceleration. Unknowns in SDEs are parameterized using a deep neural network with the inputs of cloud and environment properties. The deep neural network is found to predict entrainment and detrainment rates better than previously proposed parameterizations. The new mixing model is implemented in a unified convection scheme (UNICON) and tested in a single-column mode for two marine shallow convection cases. It is shown that the simulations with the new mixing model produce realistic mean and variance of various convective updraft properties and that the appropriate amount of stochasticity is generated. Consistently accurate simulations of updraft mass fluxes and moist conserved variables reduce model errors in the original UNICON. Additional sensitivity simulations enabling or disabling the stochasticity in mixing and initialization suggest that most of the cloud variabilities are generated from the mixing process.

Plain Language Summary The mixing between cumulus clouds and nearby environment is one of the largest sources of uncertainty in climate modeling. The air mass flux crosses into a cloud is called the entrainment and out of a cloud is called the detrainment. In this study, we develop a stochastic mixing model using a machine learning (ML) technique, which models the mixing process of convection. It is found that the ML model predicts entrainment and detrainment rates better than previously proposed parameterizations, with the inputs of cloud and environment properties. The single-column model simulations with the new mixing model produce realistic mean and variance of various shallow cumulus properties. The simulation results also suggest that most of the cloud variabilities are generated from the mixing process.

1. Introduction

In general circulation models (GCMs), non-local turbulence that is not resolved by their limited horizontal resolution is parameterized by convection schemes. A popular way to parameterize the vertical transport of heat, moisture, and momentum by convection is through the mass flux schemes. The mass flux schemes compute changes in the mass flux and properties of convection, where the key process is the mixing between cumulus and nearby environment by entrainment and detrainment. The convective entrainment and detrainment are complex turbulent mixing processes involving the phase change of hydrometeors, so constitute one of the largest sources of uncertainty in GCMs (Klocke et al., 2011; Murphy et al., 2004).

Entrainment and detrainment are also important in that they are the main sources of variabilities among convective clouds. The main challenge in modern convection parameterizations is to represent a realistic distribution of clouds in a given environment. Many stochastic convection schemes are based on assumed mass flux distribution. However, there were also attempts to understand underlying physical processes responsible for developing the cloud variabilities. The variabilities among convective clouds can be generated from the variabilities from the near-surface or cloud base, or by the stochastic mixing process (Romps & Kuang, 2010).

Entrainment and detrainment of mass are defined as the mass flux crosses into (entrainment) or out of (detrainment) cloud volume. The entrainment and detrainment rates of a cloud at a given height can be formally defined as (Siebesma, 1998)

$$E = -\oint_{\hat{\boldsymbol{n}}\cdot(\boldsymbol{u}-\boldsymbol{u}_i)<0} \rho\hat{\boldsymbol{n}}\cdot(\boldsymbol{u}-\boldsymbol{u}_i)\,dl, \tag{1}$$

$$D = \oint_{\hat{\boldsymbol{n}}\cdot(\boldsymbol{u}-\boldsymbol{u}_i)>0} \rho\hat{\boldsymbol{n}}\cdot(\boldsymbol{u}-\boldsymbol{u}_i)\,dl, \tag{2}$$

where E and D are the entrainment and detrainment rates ($\mathrm{kg\,m^{-1}\,s^{-1}}$), respectively, ρ is the density of air, $\hat{\boldsymbol{n}}$ is an outward unit vector perpendicular to the interface, $\boldsymbol{u}$ is the velocity of air at the cloud interface, and $\boldsymbol{u}_i$ is the velocity of the cloud interface. Entrainment and detrainment are often represented as the fractional entrainment and detrainment rates $\epsilon = E/M$ and $\delta = D/M$ ($\mathrm{m^{-1}}$), where $M = \rho w a$ is the convective mass flux, w is the vertical velocity, and a is the cross-sectional area of the cloud.

Due to their importance on weather and climate models, many studies have been conducted to parameterize the entrainment and detrainment in the last several decades. Many of them are in a form of deterministic formulas as a function of cloud and environment properties. Neggers et al. (2002) proposed $\epsilon \propto 1/w$ assuming a constant mixing timescale. Gregory (2001) proposed $\epsilon \propto B/w^2$ and von Salzen and McFarlane (2002) proposed $\epsilon \propto dB/dz$, where B is the buoyancy of the cloud. Lu et al. (2016) suggested a parameterization based on fitting a power-law equation of w, B, and turbulent dissipation rate on ϵ. Dawe and Austin (2013) suggested power law fits of $B\partial\overline{\theta_v}/\partial z$ on ϵ and $w\chi_c$ on δ, where $\overline{\theta_v}$ is the environmental virtual potential temperature and χ_c is the critical mixing fraction. Another notable way of parameterizing ϵ and δ is the buoyancy sorting scheme (Kain & Fritsch, 1990; Raymond & Blyth, 1986). Buoyancy sorting schemes assume a spectrum of mixed air between clouds and the environment, and then the mixtures with positive (negative) buoyancy are entrained (detrained). However, some deficiencies are reported for the original Kain-Fritsch buoyancy sorting scheme, so improvements to the Kain-Fritsch scheme were developed for practical use (Bretherton et al., 2004; de Rooy & Siebesma, 2008; Park, 2014).

Another view of mixing of convection is as a purely stochastic process (Romps & Kuang, 2010). Romps and Kuang (2010) modeled entrainment as discrete events which may be described as a stochastic Poisson process. The model is motivated by the observation that cloud-base properties are uncorrelated with upper-level cloud properties (i.e., cloud variabilities are generated by the mixing process). However, subsequent studies suggest that the stochastic mixing model also needs to include some kind of dependency on cloud properties to simulate various regimes of convection (Romps, 2016; Suselj et al., 2019). In summary, the modern view of entrainment and detrainment processes is that they are strongly dependent on cloud properties and also exhibit a considerable randomness (Dawe & Austin, 2013).

Another important issue about the mixing process is the role of the moist cloud shell which is a subsiding or negatively buoyant region around the cloud core. Traditionally entrainment and detrainment rates are diagnosed using the budget equations of conservative scalars between updrafts and environment, without considering the cloud shell. After Romps (2010), several methods have been developed to calculate entrainment and detrainment rates directly from large-eddy simulations (LESs) (Dawe & Austin, 2011b; Yeo & Romps, 2013; Z. Wang, 2020). These studies revealed that the presence of a cloud shell biases the budget calculations. In addition, Hannah (2017) showed that entrainment and dilution of scalars are not well correlated, suggesting a need for the explicit consideration of cloud heterogeneity.

In this study, we will introduce neural stochastic differential equations (SDEs) for the mixing process of convective clouds. The SDEs have been extensively used in dynamic systems with random processes and are a very useful tool to describe the Lagrangian motion of turbulent flows. The neural SDEs, also called the latent SDEs, are the SDEs that their drift and diffusion are modeled by neural network (Li et al., 2020; Tzen & Raginsky, 2019). Use of the machine learning (ML) model helps to explain the complex non-linear system that is hard to be physically modeled. Four uncertain parameters in the governing equations of mass flux schemes are modeled using neural SDEs: ϵ, δ, fractional dilution rate of scalars, and vertical acceleration. In this framework, the dependence of the mixing process on cloud properties and also the stochasticity in the mixing process can be modeled realistically. Also, the effect of cloud shells in the mixing process is considered. Section 2 provides the conceptual framework of a new mixing model. Section 3 provides settings of LESs and single-column model (SCM) simulations. Section 4 presents the results from training and testing the ML model, and Section 5 provides the results from SCM simulations. Section 6 presents the results when different data sets are used for training the ML model. A summary and conclusions are given in Section 7.

2. Conceptual Framework

2.1. Stochastic Equations for Vertical Evolution

Following Siebesma (1998), the governing equations for individual updrafts can be formulated using the conservation law of mass and a scalar variable $\hat{\phi}$ (superscript hat denotes updraft and overline denotes grid mean) with a steady-state plume approximation ($\partial\psi/\partial t = 0$, where ψ is any convection properties):

$$\frac{\partial \hat{M}}{\partial z} = \hat{M}(\epsilon - \delta), \tag{3}$$

$$\frac{\partial \left(\hat{M}\hat{\phi}\right)}{\partial z} = \hat{M}\left(\hat{\phi}_\epsilon \epsilon - \hat{\phi}_\delta \delta\right) + \hat{M}\hat{S}_\phi, \tag{4}$$

where $\hat{S}_\phi$ is the source of ϕ within an updraft and $\hat{\phi}_\epsilon\left(\hat{\phi}_\delta\right)$ is average ϕ of entraining (detraining) air. Combining Equations 3 and 4,

$$\frac{\partial \hat{\phi}}{\partial z} = -\epsilon\left(\hat{\phi} - \hat{\phi}_\epsilon\right) + \delta\left(\hat{\phi} - \hat{\phi}_\delta\right) + \hat{S}_\phi. \tag{5}$$

In most bulk plume schemes, the entraining air is assumed to have same properties as the environmental air $\left(\hat{\phi}_\epsilon = \overline{\phi}\right)$ and the detraining air is assumed to have same properties as the mean updraft air $\left(\hat{\phi}_\delta = \hat{\phi}\right)$, which reduce Equation 5 to $\partial\hat{\phi}/\partial z = -\epsilon\left(\hat{\phi} - \overline{\phi}\right) + \hat{S}_\phi$. However, previous studies found that the entraining air does not have properties of the environment because of the existence of subsiding cloud shell that includes the recently detrained air from the cloud core (Dawe & Austin, 2011a; Hannah, 2017).

Directly modeling $\hat{\phi}_\epsilon$ and $\hat{\phi}_\delta$ is hard since it might need additional prognostic equations for the cloud shell. As an alternative, we define the fractional dilution rate ϵ_ϕ as the tendency of $\hat{\phi}$ by the mixing process divided by the anomaly of $\hat{\phi}$. In a steady state, ϵ_ϕ can be formulated as

$$\epsilon_\phi = -\frac{\partial\hat{\phi}/\partial z - \hat{S}_\phi}{\hat{\phi} - \overline{\phi}} = \frac{\epsilon\left(\hat{\phi} - \hat{\phi}_\epsilon\right) - \delta\left(\hat{\phi} - \hat{\phi}_\delta\right)}{\hat{\phi} - \overline{\phi}}. \tag{6}$$

The fractional dilution rate can be understood as a diagnosed fractional entrainment rate obtained using the scalar budget equation. As explained in the introduction, the use of wrong diagnoses for the entrainment rate (not considering the effect of cloud shells) might prevent the accurate calculation of mass fluxes and scalars simultaneously. As so, we will use different mixing rates for mass flux (ϵ and δ) and scalars (ϵ_ϕ), while the identical ϵ_ϕ is used for all scalars.

The vertical velocity of updraft also can be described by Equation 5 with the source term of buoyancy, vertical pressure gradient force, and the Coriolis force (which is typically neglected). Since the vertical pressure gradient force term is hard to be parameterized, most parameterizations of vertical velocity equation partition the vertical pressure gradient force term into buoyancy and entrainment terms and use the form of

$$\hat{w}\frac{\partial\hat{w}}{\partial z} = a\hat{B} - b\epsilon\hat{w}^2, \tag{7}$$

where a and b are constants. a and b are found to be highly case-dependent (de Roode et al., 2012) and sensitive to how the convective updrafts are defined (X. Wang & Zhang, 2014). This is due to the fact that the pressure gradient force term, which is hard to be physically parameterized, is the dominant sink term in the vertical velocity budget (de Roode et al., 2012). This motivates the use of ML for the total vertical acceleration $d\hat{w}/dt = \dot{w}$, rather than modeling each term.

The strategy of our stochastic mixing model is to set SDEs using a neural network for the most uncertain parameters: ϵ, δ, ϵ_ϕ, and $\dot{w}$. Since SDE is usually used to explain the time evolution, the fractional mixing rates are expressed as mixing time scales in the unit of (s^{-1}). They can be converted from the fractional mixing rates in unit height, as

$$\epsilon^t = \hat{\iota w}\epsilon, \tag{8}$$

$$\delta^t = \hat{\iota w}\delta, \tag{9}$$

$$\epsilon^t_\phi = \hat{\iota w}\epsilon_\phi. \tag{10}$$

The final governing equations for the vertical evolution of the mass flux and scalars of individual convective updrafts are expressed as total derivatives, following the convention of Lagrangian models of turbulent flows:

$$\begin{cases} \dfrac{1}{\hat{M}}\dfrac{d\hat{M}}{dt} = \epsilon^t - \delta^t \\ \dfrac{d\hat{\phi}}{dt} = -\epsilon^t_\phi\left(\hat{\phi} - \overline{\phi}\right) + \hat{S}^t_\phi \\ \dfrac{d\hat{\iota w}}{dt} = \iota\dot{w} \end{cases} \tag{11}$$

where $\hat{S}^t_\phi$ is the source of $\hat{\phi}$ in unit time. Note that the equations are written in total derivative, but $\hat{M}$, $\hat{\phi}$, and $\hat{\iota w}$ are time-invariant in a given environment due to the steady-state assumption. Here, ϵ^t, δ^t, ϵ^t_ϕ, and $\iota\dot{w}$ are determined by the following SDEs:

$$d\log\left(\epsilon^t\right) = \mu_1\left[\log\left(\epsilon^t\right)_{\exp} - \log\left(\epsilon^t\right)\right]dt + \sigma_1 dW_1, \tag{12}$$

$$d\log\left(\delta^t\right) = \mu_2\left[\log\left(\delta^t\right)_{\exp} - \log\left(\delta^t\right)\right]dt + \sigma_2 dW_2, \tag{13}$$

$$d\log\left(\epsilon^t_\phi\right) = \mu_3\left[\log\left(\epsilon^t_\phi\right)_{\exp} - \log\left(\epsilon^t_\phi\right)\right]dt + \sigma_3 dW_3, \tag{14}$$

$$d\iota\dot{w} = \mu_4\left[\iota\dot{w}_{\exp} - \iota\dot{w}\right]dt + \sigma_4 dW_4. \tag{15}$$

For the variables of $\chi_i = \left\{\log\left(\epsilon^t\right),\ \log\left(\delta^t\right),\ \log\left(\epsilon^t_\phi\right),\ \iota\dot{w}\right\}$, the four equations can be summarized as

$$d\chi_i = \mu_i\left[\chi_{i,\exp} - \chi_i\right]dt + \sigma_i dW_i, \tag{16}$$

where σ_i is the diffusion coefficient, dW_i is the increment of Wiener process, $\chi_{i,\exp}$ is the expected value at a given state. μ_i, $\chi_{i,\exp}$, and σ_i in Equation 16 are machine learned as described in Section 2.2. From those, the SDEs are solved to get χ_i, and χ_i is then inserted into the final governing equations (Equation 11). The equations can be understood as the continuous time limit of the first-order autoregressive model (AR(1)) (Brockwell et al., 1991; Stramer et al., 1996). The drift terms of the equations indicate that χ_i approach to $\chi_{i,\exp}$, where the speed of the drift is determined by μ_i. The use of log for the mixing rates of ϵ^t, δ^t, and ϵ^t_ϕ guarantees that those mixing rates are always positive. Also, LES (Dawe & Austin, 2013) and aircraft observation (Cheng et al., 2015) studies show that the fractional entrainment and detrainment rates are well modeled by log-normal distributions.

The proposed framework allows us to account for the time coherency of the mixing rates, in contrast to the model of Romps (2010) that assumes independent discrete entrainment events. This is a more realistic approach since turbulent eddies of various spatiotemporal scales experience non-linear interactions. The time series of $\log_{10}(\epsilon^t)$ derived from the Barbados Oceanographic and Meteorological Experiment (BOMEX) LES (Figure 1) clearly show that the entrainment is an autocorrelated process. The lag-1 autocorrelations of $\log(\epsilon^t)$, $\log(\delta^t)$, $\log\left(\epsilon^t_\phi\right)$, and $\iota\dot{w}$ with $\Delta t = 60$ s calculated using our data set are found to be 0.529, 0.582, 0.376, and 0.555, respectively.

In numerical implementations, the SDEs are used as discretized form:

$$\chi^t_i - \chi^{t-1}_i = \mu_i\left[\chi_{i,\exp} - \chi^{t-1}_i\right]\Delta t + \sigma_i\sqrt{\Delta t}\xi_i, \tag{17}$$

where ξ_i is the white Gaussian noise $N(0, 1)$, Δt is the time step size, and χ^t_i and χ^{t-1}_i are χ_i at times of t and $t - \Delta t$, respectively.

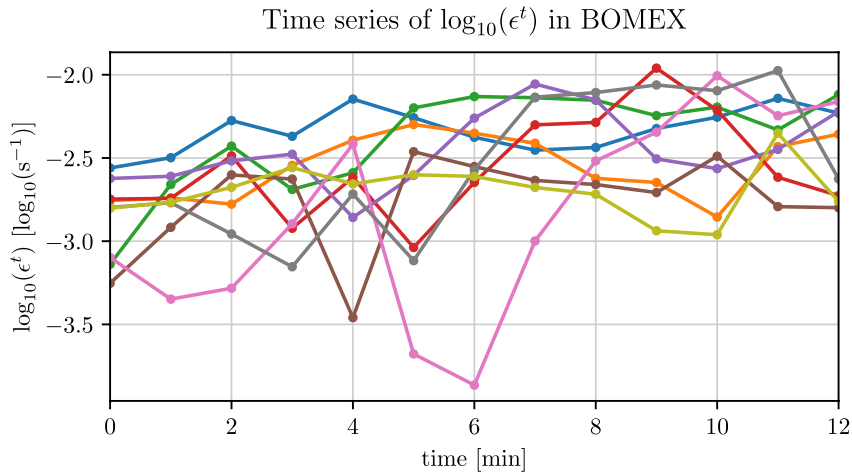

Figure 1. Time series of $\log_{10}(\epsilon^t)$ measured in the Barbados Oceanographic and Meteorological Experiment (BOMEX) LES (large-eddy simulation). Nine time series are derived from randomly selected updrafts starting from the height of 600 m and lasting more than 12 min. The updrafts are tracked in a Lagrangian way as described in Appendix A. The simulation setup for the BOMEX LES is described in Section 3.1.

2.2. Machine Learning Model Configuration

The unknown parameters in SDEs, μ_i, $\chi_{i,\exp}$, and σ_i, are modeled using a deterministic feed-forward neural network. The network accepts properties of a convective updraft and environment at a given height as inputs. The selection of the input variables will be discussed in Section 4. The feed-forward neural network has three hidden layers with 16 neurons in each layer. Scaled exponential linear unit (SELU) activation functions (Klambauer et al., 2017) follow each hidden layer. SELU helps the output distribution to retain a mean of 0 and a standard deviation of 1 which in turn avoids exploding and vanishing gradients. Before the final output layer, a dropout layer with a rate of 0.2 is added to reduce the risk of overfitting. The structure of the feed-forward network is optimized to give the best performance.

The unknown parameters in SDEs, μ_i, $\chi_{i,\exp}$, and σ_i, cannot be directly measured from the LES data set. Then how is the neural network trained for μ_i, σ_i, and $\chi_{i,\exp}$ which can not be directly obtained from the LES data set? From Equation 17, it can be derived that the distribution of χ_i^t, $P\left(\chi_i^t\right)$, is a normal distribution with mean of $\chi_i^{t-1} + \mu_i\left[\chi_{i,\exp} - \chi_i^{t-1}\right]\Delta t$ and standard deviation of $\sigma_i\sqrt{\Delta t}$. Δt corresponds to the LES output frequency of 60 s. While training the network, a probabilistic layer is embedded as a final layer so that the outputs are probability distributions of $P\left(\chi_i^t\right)$. The training network is trained such that the modeled $P\left(\chi_i^t\right)$ is the best fit to LES data samples by minimizing the loss function l which is defined as

$$l = -\frac{1}{n}\sum_{i=1}^{n}\log f\left(x_i|\theta\right), \tag{18}$$

where n is the number of training samples, f is the probability density function (PDF), x_i is training samples, and θ is parameters of the PDF. The data samples are trained altogether regardless of height and time of sampling. The configuration of the deep learning network is depicted in Figure 2.

2.3. Stochastic Initialization of Convective Updrafts at the Near-Surface

To fully represent the variabilities between convective updrafts, near-surface variabilities should also be realistic. For the stochastic initialization of convective updrafts at the near-surface, we follow the method of Shin and Park (2020). Vertical velocity and thermodynamic scalars of convective updrafts at the near-surface are randomly sampled from the multivariate Gaussian distribution with a constraint of $w > 0$, where standard deviations and inter-variable correlations are derived from the surface-layer similarity theory. The standard deviations are obtained using similarity functions suggested by Liu et al. (1998):

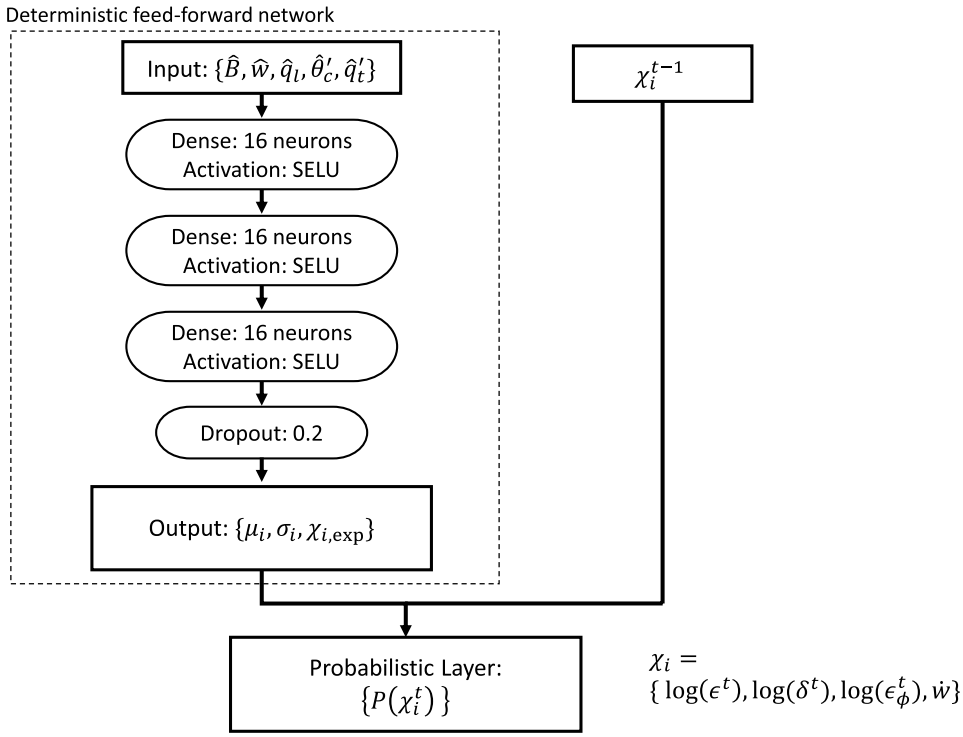

Figure 2. A diagram of how the probabilistic deep learning network is connected for training. The network is trained to minimize negative log-likelihood. After the training, the deterministic feed-forward network part is used to determine the parameters of the stochastic differential equations.

$$\sigma_w/u_* = \phi_{\sigma_w}\left(\frac{z}{L}\right) = 1.25\left(1.0 - 3\frac{z}{L}\right)^{1/3}, \tag{19}$$

$$\sigma_\theta/\theta^* = \phi_{\sigma_\theta}\left(\frac{z}{L}\right) = -2.0\left(1.0 - 8\frac{z}{L}\right)^{-1/3}, \tag{20}$$

$$\sigma_q/q^* = \phi_{\sigma_q}\left(\frac{z}{L}\right) = -2.4\left(1.0 - 8\frac{z}{L}\right)^{-1/3}, \tag{21}$$

where z is the geometric height, $u_* = \left[\left(\overline{w'u'}\right)_s^2 + \left(\overline{w'v'}\right)_s^2\right]^{1/4}$ is the frictional velocity, $\theta^* = -\left(\overline{w'\theta'}\right)_s/u_*$, and $q^* = -\left(\overline{w'q'}\right)_s/u_*$. $\left(\overline{w'\theta'}\right)_s$, $\left(\overline{w'q'}\right)_s$, $\left(\overline{w'u'}\right)_s$, and $\left(\overline{w'v'}\right)_s$ are the kinematic surface fluxes of sensible heat, water vapor, zonal momentum, and meridional momentum, respectively, by non-organized symmetric turbulent eddies and $L = -u_*^3\bar{\theta}_v/\left[gk\left(\overline{w'\theta'_v}\right)_s\right]$ is the Monin-Obukhov length scale with the von Karman constant k, the reference virtual potential temperature $\overline{\theta_v}$, and the buoyancy flux at the surface $\left(\overline{w'\theta'_v}\right)_s$. The correlations can then be calculated as

$$r_{w\theta} = \left(\overline{w'\theta'}\right)_s/\left(\sigma_w\sigma_\theta\right) = -1/\left(\phi_{\sigma_w}\phi_{\sigma_\theta}\right), \tag{22}$$

$$r_{wq} = \left(\overline{w'q'}\right)_s/\left(\sigma_w\sigma_q\right) = -1/\left(\phi_{\sigma_w}\phi_{\sigma_q}\right), \tag{23}$$

$$|r_{\theta q}| = |r_{wq}|/|r_{w\theta}| = |\phi_{\sigma_\theta}/\phi_{\sigma_q}| = 0.83. \tag{24}$$

The number density PDF of updraft radius R at the surface, P_n, is parameterized as

$$P_n(\hat{x})/N = a_1\hat{x}^{-2-\hat{x}^{1.7}}, \quad \int_0^\infty (P_n(\hat{x})/N)\,d\hat{x} = 1, \tag{25}$$

where $\hat{x} = R/R_b$ is the dimensionless updraft plume radius normalized by the scale break radius R_b, N is the total updraft number density in unit of (# m^{-2}), and a_1 is a normalization constant. In this study, R_b is set to 170 m, which represents the typical size of shallow convection. We assume that correlations between updraft radius and other updraft variables are zero at the surface. For more detailed derivation and physical explanation, refer to Shin and Park (2020).

3. Experimental Setting

3.1. Large-Eddy Simulations

The University of California, Los Angeles large-eddy simulation (UCLA-LES) model (Stevens et al., 1999, 2005) is used to simulate two shallow convection cases. The UCLA-LES solves a set of anelastic equations with the turbulence model of Smagorinsky-Lilly. Cloud microphysical processes are parameterized based on the two-moment warm rain scheme of Seifert and Beheng (2001) with modifications detailed in Stevens and Seifert (2008).

The first case is the BOMEX (Holland & Rasmusson, 1973) following the settings of Siebesma et al. (2003). In this simulation, radiation and the production of precipitation are turned off. The domain size is $6.4 \times 6.4 \times 3.0$ km, and the grid size is $25 \times 25 \times 25$ m. The model is run for 6 hr, and outputs from time intervals of 1 min during the last 2 hr (a total of 120 instantaneous snapshots) are analyzed.

The second case is the Rain in Cumulus over the Ocean (RICO) field campaign following the settings of vanZanten et al. (2011). The domain size is $12.8 \times 12.8 \times 4.0$ km, and the grid size is $40 \times 40 \times 40$ m. The model is run for 24 hr, and outputs from time intervals of 1 min during the last 4 hr (a total of 240 instantaneous snapshots) are analyzed. For the RICO case, the production of precipitation is allowed and the number density of cloud droplets is set to a fixed value of 70 cm^{-3}.

Individual clouds are detected using the cloud tracking algorithm of Dawe and Austin (2012) which tracks clouds by considering the spatiotemporal connectivity of cloudy grid cells. The algorithm categorizes cloudy grid cells into three types. The "core" region is defined as the grid boxes with positive condensate, vertical velocity, and buoyancy, and the "condensed" region is defined as the grid boxes with condensate. The "plume" region is defined as the grid boxes having positive vertical velocity and containing radioactively decaying passive tracer emitted continuously from the surface with a concentration higher than one spatial standard deviation at each height (Couvreux et al., 2010). Additionally, all condensed points are also flagged as plume points, so that the condensed region is always a subset of the plume. The decaying time scale of the passive tracer is set to 15 min in the BOMEX case and 30 min in the RICO case.

It is important to determine the cloud type that corresponds to convective updrafts in a convection scheme when developing the convection scheme using LES data. Typically, convection schemes are designed to represent the properties of the cloud core region. In our study, we focus on the analysis of plume regions with positive vertical velocity (hereinafter referred to as convective updrafts). This is for two reasons, first to include dry convection in the sub-cloud layer, and second to allow forced convection (negatively buoyant but have positive vertical velocity). We add a constraint that vertical velocity should be greater than zero, to exclude cloud shell or convective downdraft.

The entrainment and detrainment rates of convective updrafts are measured using the method of Yeo and Romps (2013). The entrainment and detrainment rates are calculated by counting the number of Lagrangian particles that go into or out of the cloud interface in finite time and height intervals. In our LES simulations, Lagrangian particles are imposed using the online Lagrangian Particle Tracking Module (LPTM; Heus, van Dijk et al. (2008)). A total number of 2,031,297 particles for the BOMEX case and 1,605,133 particles for the RICO case are imposed, and the properties of individual Lagrangian particles are recorded every 30 s. The entrainment and detrainment rates are calculated for each convective updraft using a time interval of 30 s and a height interval of 50 m.

Finally, the fractional dilution rate ϵ_ϕ^t and vertical acceleration $\dot{w}$ of individual convective updrafts are calculated. Also, our framework needs mixing rates at current time step $\left(\chi_i^t\right)$ and previous time step $\left(\chi_i^{t-1}\right)$. The method of calculating these variables is explained in Appendix A. For every model layer and sampling time, input and output variables for training the neural network are obtained. We exclude the samples at the time steps when a convective updraft merges with other updrafts or is split into multiple updrafts. We also exclude the samples at

the lowest model layer. This results in a total of 48,845 samples for the BOMEX case and 238,606 samples for the RICO case.

3.2. Single-Column Model

The SCM used in our study is the single-column version of Community Atmospheric Model version 5 with the unified convection scheme (UNICON; Park (2014)) implemented, identical to the one used by Park et al. (2019). The model has 80 vertical layers and uses the leap-frog time stepping method with a time step of 300 s. SCM is driven by same forcing specified in LES simulations, each for the BOMEX and RICO cases. UNICON computes the vertical evolution of conservative scalars $\phi = \{\theta_c, q_t, u, v, \zeta\}$ within convective updrafts and downdrafts. Here, $\theta_c \equiv \theta - (L_v/C_p/\pi)q_l - (L_s/C_p/\pi)q_i$ is the condensate potential temperature, where L_v and L_s are the latent heats of vaporization and sublimation, respectively, C_p is the specific heat at constant pressure, π is the Exner function; $q_t \equiv q_v + q_l + q_i$ is the total water specific humidity, where q_v, q_l, and q_i are the specific humidity of water vapor, liquid, and ice, respectively; u is the zonal wind speed; v is the meridional wind speed; ζ is the mass or number concentration of aerosols and chemical species.

Our mixing model is implemented in the UNICON scheme and substitutes the existing mixing model which is a modified version of the buoyancy sorting algorithm. The vertical velocity equation is also replaced. The trained neural network is converted to a Fortran code using the Fortran-Keras Bridge library (Ott et al., 2020). Original UNICON simulates convective downdrafts, associated mesoscale organized flow, and its effect on convection. In our SCM experiments, we disable convective downdrafts and associated mesoscale organization flow since the distribution of near-surface thermodynamic variables in the presence of mesoscale organization has not yet been studied. The standard RICO case simulated by UCLA-LES does not show notable mesoscale organization for the first 24 hr (Seifert et al., 2015; Seifert & Heus, 2013). However, we note that transport due to convective downdrafts in the RICO case is relatively small compared to updrafts but not negligible (Heus, Pols, et al., 2008; Suselj et al., 2019).

The conversion rate of cloud water to rain, the autoconversion rate, in the original UNICON is based on Kessler (1969), where it is linearly proportional to cloud water content when the cloud water content is larger than a threshold value. The threshold value and the autoconversion efficiency are specifically tuned to produce a reasonable amount of precipitation in global simulations. However, we found that the original autoconversion scheme produces an excessive amount of precipitation and severely distorts the distribution of cloud properties for the RICO case. Thus we utilize the autoconversion scheme of Khairoutdinov and Kogan (2000) in the form of

$$\left(\frac{\partial q_r}{\partial t}\right)_{\text{auto}} = c\hat{q}_c^a \hat{N}_c^b, \tag{26}$$

where q_r is the rain water specific humidity, $\hat{q}_c = \hat{q}_l + \hat{q}_i$ is the specific humidity of in-cumulus condensate, $\hat{N}_c = \hat{N}_l + \hat{N}_i$ is the number density of cloud droplets ($\hat{N}_l$: the number density of liquid cloud droplets, $\hat{N}_i$: the number density of ice cloud droplets), and a, b, and c are the fitting parameters. The values of the fitting parameters are set equal to Kogan (2013), which are specifically fitted to the RICO case. The values are $a = 4.22$, $b = -3.01$, and $c = 7.98 \times 10^{10}$ when $\hat{q}_c$ is in unit of kg kg^{-1} and $\hat{N}_c$ is in unit of cm^{-3}. The autoconversion scheme cannot be implemented in UNICON in general cases, since the microphysics in UNICON is a simple single-moment scheme. Thankfully, since the RICO case assumes a constant cloud droplet number density of 70 cm^{-3}, the parameterized autoconversion rate becomes a function only of $\hat{q}_c$. We do not change the other rain processes like accretion and evaporation since the simulated rain rate of the RICO case is found to be sensitive only to the autoconversion process.

Finally, a free parameter in UNICON is tuned to simulate reasonable mass flux. The only parameter changed is the convective updraft fractional area at the surface $\hat{A}_s$, changed from the original value of 0.040 to 0.025. The value of the parameter was determined by selecting the one with the lowest root-mean-square error (RMSE) of $\overline{\theta_c}$ and $\overline{q_t}$ among several SCM simulations with different $\hat{A}_s$. The SCM simulation with the new mixing model takes ~15% more time compared to the original UNICON, mainly due to the computation of the neural network. The computation time can be reduced by using computationally cheaper activation function or by reducing the size of

the network. Accommodating the neural network adds about 220 KB of memory footprint per process (including loading related libraries), which is negligible compared to the total memory consumption of the model code.

4. Training and Testing of the Machine Learning Model

4.1. Training of the Machine Learning Model

The neural network described in Section 2.2 is trained and tested with the combined samples from the BOMEX and RICO cases. The total 287,451 samples are randomly partitioned into the training set, validation set, and test set, with ratios of 64%, 16%, and 20%, respectively. The validation set is used to estimate the model skill during the training of the model, and the test set is used to evaluate the performance of the model after the training. The neural network of three hidden layers with 16 neurons per layer is iteratively updated through the stochastic gradient descent with a batch size (number of samples used in a single update of model weights) of 32 and a learning rate of 0.001 with the Adam optimizer (Kingma & Ba, 2015).

As the training progresses, the loss function evaluated from the training set (training loss) keeps decreasing, while the loss function evaluated from the validation set (validation loss) stops decreasing and starts to increase at some point due to overfitting. In order to prevent overfitting, the neural network is trained until the validation loss does not decrease for the following 50 epochs. The training takes 250–500 epochs depending on the random state of the stochastic gradient descent. The neural network is trained and tested using TensorFlow library (Abadi et al., 2015) and TensorFlow Probability library (Dillon et al., 2017). The hyperparameters that affect the model performance, which are batch size, learning rate, number of hidden layers, and number of neurons per layer, are tuned with Keras Tuner library (O'Malley et al., 2019) to get the optimal performance. The setup for the Karas Tuner is described in Text S1 in Supporting Information S1.

4.2. Selecting Input Variables for the Machine Learning Model

We selected input variables for the ML model that are physically meaningful for predicting mixing rates. This is also an important procedure to reduce the size of the network and the computation time. First, the candidates of input variables are chosen from previous studies. The candidates are buoyancy $\hat{B} = \left(\hat{\theta}_v - \overline{\theta_v}\right)/\overline{\theta_v}$, vertical velocity $\hat{w}$, specific humidity of liquid water $\hat{q}_l$, anomaly of condensate potential temperature $\hat{\theta}_c' = \hat{\theta}_c - \overline{\theta_c}$, anomaly of the total water specific humidity $\hat{q}_t' = \hat{q}_t - \overline{q_t}$, vertical gradient of environmental virtual potential temperature $\partial\overline{\theta_v}/\partial z$, updraft radius $\hat{R} = \sqrt{\hat{a}/\pi}$, and environmental vertical wind shear $V_{\text{shear}} = \sqrt{\left(\partial\overline{u}/\partial z\right)^2 + \left(\partial\overline{v}/\partial z\right)^2}$.

The thermodynamic variables are all in the form of the anomaly with respect to the mean environment (if $\overline{q_l} = 0$). In fact, the most fundamental variables describing the thermodynamic states of updrafts and environment are $\hat{\theta}_c$, $\overline{\theta_c}$, $\hat{q}_t$, and $\overline{q_t}$. However, if the ML model is trained with these variables, it cannot be used in a climate different from the one in which it was trained (e.g., warm climate). This is a well-known issue of ML based physics parameterization (Rasp et al., 2018). Therefore, we select anomalous variables to limit the sample space. It is also a physically rational choice since turbulent mixing is proportional to the difference in properties of the two fluids. Previous studies indicate that relative humidity in environment is an important factor affecting entrainment and detrainment rates (Bechtold et al., 2008; Lu et al., 2018; Stirling & Stratton, 2012; Zhao et al., 2018; Zhu et al., 2021). However, relative humidity is not considered as a potential input variable here since the anomaly of the total water specific humidity $\hat{q}_t'$ which has a similar physical meaning as relative humidity was found to be a better proxy.

To select the input variables, we calculate permutation importance (PI; Altmann et al. (2010)) to quantify the relative importance of the variables. The PI is defined as the decrease in a model score when the values of a single variable are randomly shuffled. We calculate the increase in mean squared error when input variables are permuted, for the neural networks predicting $\log(\epsilon)$, $\log(\delta)$, $\log(\epsilon^t)$, $\log(\delta^t)$, $\log\left(\epsilon_\phi^t\right)$, and $\dot{w}$. To increase statistical significance, 20 random permutations for each input variable are done. For this analysis, the training of the neural network is only done for the data samples with $\hat{q}_l > 0$ since the cloud tracking in the sub-cloud layer is inaccurate (Dawe & Austin, 2012).

Table 1
Permutation Importance of the Candidates of Input Variables and Base Mean Squared Error (MSE) When Predicting log(ϵ), log(δ), log(ϵ^t), log(δ^t), log $\left(\epsilon_\phi^t\right)$, and $\dot{w}$

	$\hat{\boldsymbol{B}}$	$\hat{\boldsymbol{w}}$	$\hat{q}_l$	$\hat{\theta}_c'$	$\hat{q}_t'$	$\partial\overline{\theta_v}/\partial z$	$\hat{R}$	V_{shear}	MSE
log(ϵ)	28.2 ± 0.7	18.0 ± 0.3	45.4 ± 1.0	44.1 ± 0.9	54.6 ± 1.0	175.7 ± 2.1	21.5 ± 0.5	8.6 ± 0.3	0.633
log(δ)	21.6 ± 0.8	854.9 ± 5.2	114.7 ± 1.6	12.8 ± 0.6	39.4 ± 0.7	87.4 ± 0.9	5.5 ± 0.2	15.4 ± 0.4	0.324
log(ϵ^t)	27.8 ± 0.6	39.9 ± 1.0	14.0 ± 0.3	22.7 ± 0.8	67.5 ± 1.3	140.7 ± 1.7	19.2 ± 0.5	7.9 ± 0.3	0.619
log(δ^t)	21.4 ± 0.8	301.4 ± 2.3	108.5 ± 1.1	11.5 ± 0.4	19.8 ± 0.7	78.7 ± 1.0	6.0 ± 0.2	13.8 ± 0.3	0.582
log $\left(\epsilon_\phi^t\right)$	10.5 ± 0.3	46.0 ± 0.5	5.2 ± 0.2	2.8 ± 0.1	14.5 ± 0.4	26.8 ± 0.5	15.2 ± 0.3	3.3 ± 0.2	0.804
$\dot{w}$	24.3 ± 0.5	103.2 ± 1.7	26.1 ± 0.6	60.1 ± 1.1	90.6 ± 1.4	127.8 ± 1.6	15.8 ± 0.7	7.6 ± 0.4	0.463

Note. Permutation importance is written as percentage (%) of increased MSE when the variable is permuted with respect to base MSE. Note that all the variables are normalized before training. Mean and standard deviation for 20 permutations are shown. The finally selected variables are denoted in bold.

Table 1 shows the results of the PI analysis. Dawe and Austin (2013) found that the fractional entrainment rate is best predicted by $\hat{B}$ and $\partial\overline{\theta_v}/\partial z$. In our analysis, $\partial\overline{\theta_v}/\partial z$ shows the largest PI values for predicting log(ϵ) and log(ϵ^t). However, the buoyancy $\hat{B}$ has relatively low PI values for predicting log(ϵ) and log(ϵ^t). This is thought to be because our definition of convective updraft includes regions with positive and negative buoyancy, while Dawe and Austin (2013) used the core region for the analysis. For log(δ) and log(δ^t), the vertical velocity $\hat{w}$ shows the largest PI values, which is one of the best predictors pointed out by Dawe and Austin (2013) as well.

The log of the fractional dilution rate log $\left(\epsilon_\phi^t\right)$ does not have predictors with significantly high PI values. This indicates that the fractional dilution rate cannot be predicted well by a single variable. For the vertical acceleration $\dot{w}$, $\partial\overline{\theta_v}/\partial z$ shows the largest PI, followed by $\hat{w}$ and $\hat{q}_t'$. In contrast, $\hat{B}$ shows a relatively low PI value, which seems to be an unexpected result because buoyancy is a main source of the vertical velocity budget. X. Wang and Zhang (2014) reported that the buoyancy term becomes small in the vertical momentum budget when convective updraft is defined to include negative buoyancy region.

Among the candidates of input variables, the vertical wind shear V_{shear} has the lowest PI values and the updraft radius $\hat{R}$ has the second lowest. Moreover, there is a possibility that PIs of other variables excluding these two variables are underestimated. As displayed in Figure 3, the variables other than $\hat{R}$ and V_{shear} are highly correlated. When variables are correlated, their PIs tend to be underestimated. This is because the permutation of one variable has little effect on model performance since the same information can be obtained from the correlated variables. The low correlations of V_{shear} and $\hat{R}$ with other variables confirm that the low PI values of these variables actually represent the low importance of these variables. The vertical wind shear is one of the main factors controlling the cloud-top entrainment of stratocumulus (Mellado, 2017), but it seems to have a low impact on cumulus-type convection. The low dependency of ϵ and δ on shallow cumulus radius is also pointed out by Dawe and Austin (2013). For this reason, we choose the final input variables excluding these two: $\hat{B}$, $\hat{w}$, $\hat{q}_l$, $\hat{\theta}_c'$, $\hat{q}_t'$, and $\partial\overline{\theta_v}/\partial z$.

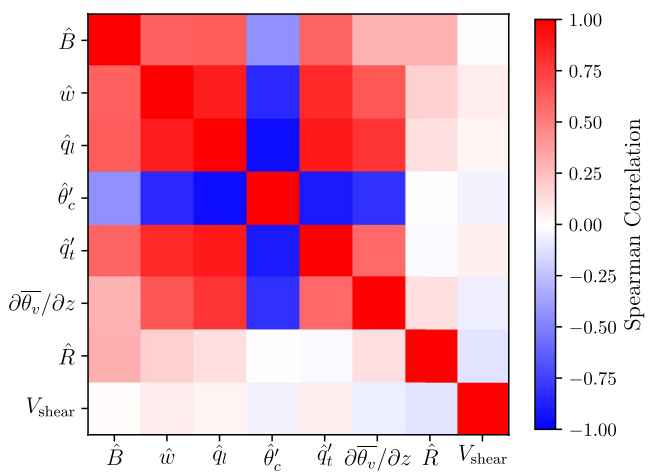

Figure 3. A plot of cross-correlation matrix for the candidates of input variables. Spearman correlation is used to see a monotonic relationship between variables.

4.3. Performance of the Machine Learning Model

In this subsection, the performance of the ML model for predicting the mixing rates is tested. We compare our model with various parameterizations of entrainment and detrainment rates proposed by previous studies. In addition, the multiple linear regression model with same inputs as the ML model for predicting dependent variables of log(ϵ) and log(δ) is examined (log is used to ensure that ϵ and δ are positive). The list of tested parameterizations is given in Table 2. The fitting and evaluation of the parameterizations are done with separate subsets of the data set, where the fitting is done with the training set plus validation set, and the evaluation is done with the test set (see Section 4.1 for how the data set is partitioned). The fitting parameters

Table 2
A List of Tested Entrainment and Detrainment Parameterizations

Entrainment model	Figure	Detrainment model	Figure	References
$\epsilon = a/\hat{R}$	Figure 4a	–	–	Turner (1963)
$\epsilon = a/\hat{w}$	Figure 4b	–	–	Neggers et al. (2002)
$\epsilon = a\hat{B}/\hat{w}^2$	Figure 4c	–	–	Gregory (2001)
$\epsilon = a\hat{B}^b\hat{w}^c$	Figure 4d	–	–	Lu et al. (2016)
$\epsilon = a\hat{B}^b\left(\partial\overline{\theta_v}/\partial z\right)^c$	Figure 4e	$\delta = a\hat{w}^b\hat{\chi}_c^c$	Figure 5a	Dawe and Austin (2013)
Buoyancy sorting	Figure 4f	Buoyancy sorting	Figure 5b	Kain and Fritsch (1990)
Linear regression	Figure 4g	Linear regression	Figure 5c	This study
Machine learning	Figure 4h	Machine learning	Figure 5d	This study

Note. a, b, and c are fitting parameters. The linear regression model for the entrainment or detrainment is $Y = a_0 + a_1\hat{B} + a_2\hat{w} + a_3\hat{q}_l + a_4\hat{\theta}_c' + a_5\hat{q}_t' + a_6\partial\overline{\theta_v}/\partial z$, where $Y = \{\log(\epsilon), \log(\delta)\}$, and a_i $(i = 0, \ldots, 6)$ are fitting parameters.

in the parameterizations are newly fitted to our data set without using the default values. The entrainment model of Lu et al. (2016) was suggested in two versions, with and without turbulent dissipation rate ($\epsilon = a\hat{B}^b\hat{w}^c\hat{\varepsilon}^d$ or $\epsilon = a\hat{B}^b\hat{w}^c$ where ε is the turbulent dissipation rate). Here, we use the version without turbulent dissipation rate since the turbulent dissipation rate is hard to be calculated in one-dimensional convection parameterization. The buoyancy sorting is original version of Kain and Fritsch (1990), where $\epsilon = \epsilon_0\hat{\chi}_c^2$ and $\delta = \epsilon_0(1 - \hat{\chi}_c)^2$ with $\epsilon_0 = 0.02\ \mathrm{m}^{-1}$ which is a typical value for shallow convection. Except for the ML model, other parameterizations are fitted only on data samples with $\hat{q}_l > 0$ and $\hat{B} > 0$ since some parameterizations require a positive buoyancy condition. We note that this analysis is not a fair comparison for the performance of the parameterizations since each parameterization has its own method of computing entrainment/detrainment rate and defining cloud region. Although many of these parameterizations are designed to predict bulk entrainment and detrainment rates, this analysis tests the performance of predicting directly measured entrainment and detrainment rates. The purpose of this analysis is to examine how well the ML model can explain the dependency of the mixing rates on cloud and environment properties.

Figure 4 shows the joint PDF of LES measured ϵ versus predicted ϵ by various parameterizations. As discussed in the previous section, $\hat{R}$ is not a good predictor for ϵ ($R^2 = 0.055$; Figure 4a). A simple $\hat{w}^{-1}$ relation shows $R^2 = 0.287$, which is found to perform reasonably well as a single variable parameterization (Figure 4b). The parameterization of Dawe and Austin (2013) does not show good performance in our analysis (Figure 4e), implying that the parameterization is only applicable for the core region. The buoyancy sorting scheme shows almost no skill for predicting ϵ (Figure 4f). The original Kain and Fritsch (1990) scheme is found to have some deficiency, for example, produces too small ϵ in low relative humidity environment (Kain, 2004). de Rooy et al. (2013) demonstrated that the simple function with height performs better than the original Kain-Fritsch scheme when predicting ϵ in the BOMEX case. The best parameterization except the ML model is the parameterization of Lu et al. (2016) (Figure 4d) which is slightly better than the multiple linear regression model. The ML model shows $R^2 = 0.655$ and outperforms the second-best parameterization of Lu et al. (2016) nearly by double the variance explained.

For the fractional detrainment rate δ (Figure 5), the ML model outperforms other parameterizations with $R^2 = 0.665$. Here, the parameterization of Dawe and Austin (2013) exhibits reasonable predictive performance ($R^2 = 0.423$; Figure 5a), unlike the entrainment parameterization. This predictive performance is largely due to the dependence of δ on the vertical velocity, where simple $\hat{w}^{-1}$ parameterization shows $R^2 = 0.404$ (not shown). The buoyancy sorting scheme shows low prediction skill and produces too large δ value (Figure 5b). Bretherton et al. (2004) reported that the original Kain-Fritsch scheme can produce excessive detrainment since all the negative buoyant mixtures are detrained from the updraft. A more realistic approach is not to detrain the negative buoyant mixture with a positive vertical velocity that can travel a certain length scale (Bretherton et al., 2004; Park, 2014). It seems that the fractional detrainment rate can be explained well by simple regression formula with a small number of variables compared to the fractional entrainment rate.

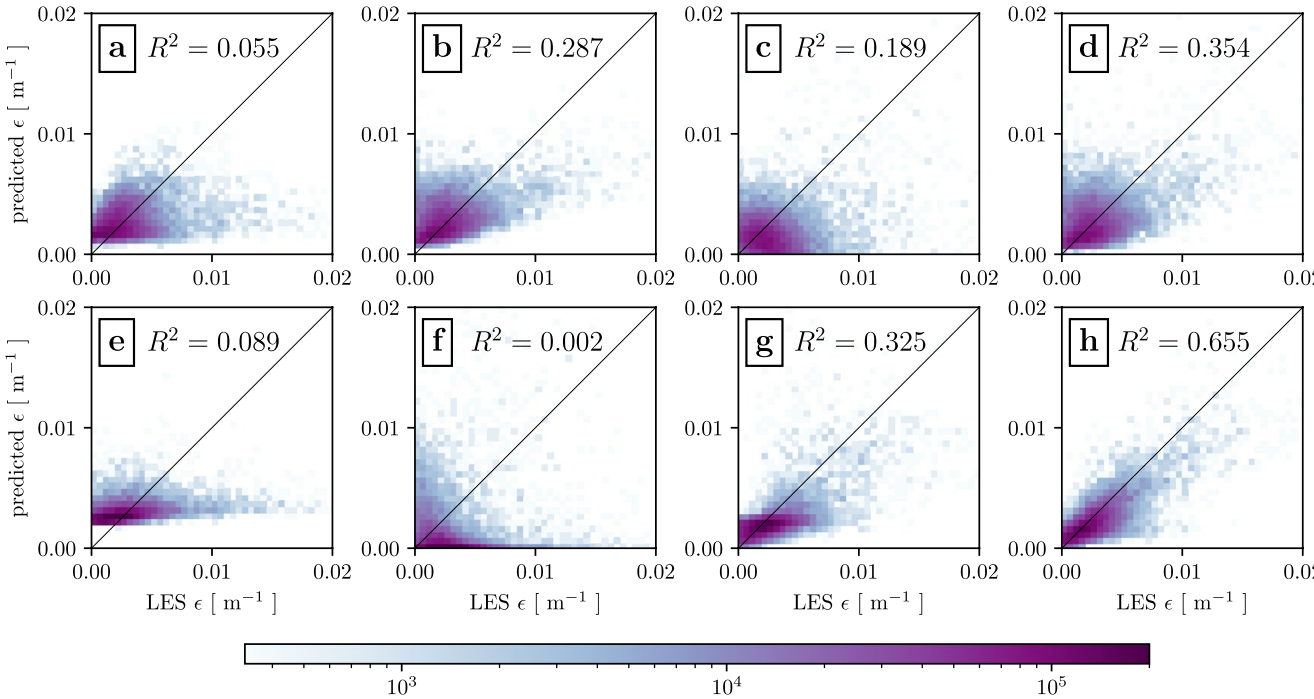

Figure 4. Joint probability density functions of large-eddy simulation measured ϵ versus predicted ϵ by various parameterizations for individual clouds. (a) $\epsilon = a/\hat{R}$, (b) $\epsilon = a/\hat{w}$, (c) $\epsilon = a\hat{B}/\hat{w}^2$, (d) $\epsilon = a\hat{B}^b\hat{w}^c$, (e) $\epsilon = a\hat{B}^b\left(\partial\overline{\theta_v}/\partial z\right)^c$, (f) buoyancy sorting scheme, (g) multiple linear regression model $\left(\log(\epsilon) = a_0 + a_1\hat{B} + a_2\hat{w} + a_3\hat{q}_l + a_4\hat{\theta}_c' + a_5\hat{q}_t' + a_6\partial\overline{\theta_v}/\partial z\right)$, and (h) machine learning model. a, b, c, and a_i ($i = 0, \ldots, 6$) are fitting parameters.

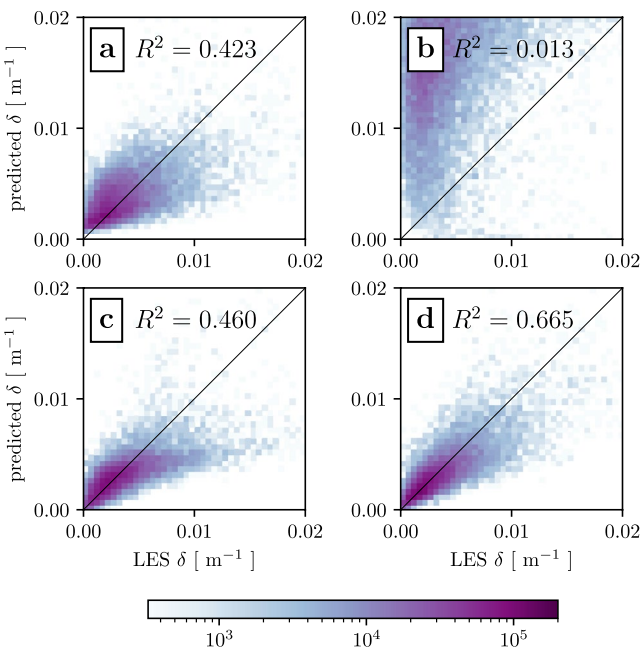

Figure 5. Joint probability density functions of large-eddy simulation measured δ versus predicted δ by various parameterizations for individual clouds. (a) $\delta = a\hat{w}^b\hat{\chi}_c^c$, (b) buoyancy sorting scheme, (c) multiple linear regression model $\left(\log(\delta) = a_0 + a_1\hat{B} + a_2\hat{w} + a_3\hat{q}_l + a_4\hat{\theta}_c' + a_5\hat{q}_t' + a_6\partial\overline{\theta_v}/\partial z\right)$, and (d) machine learning model. a, b, c, and a_i ($i = 0, \ldots, 6$) are fitting parameters.

In addition to ϵ and δ, the skill for predicting $\dot{w} = dw/dt$ is tested (Figure 6). The most commonly used parameterization, Equation 7, surprisingly exhibits almost the same performance as the ML model. However, the parameterization can produce expected performance only when the entrainment rate is accurately predicted. One interesting fact we found is that the entrainment drag term of the vertical velocity equation should utilize the fractional dilution rate ϵ_ϕ, rather than the fractional mass entertainment rate ϵ. If ϵ is used, the prediction skill decreases as $R^2 = 0.014$ (not shown). The vertical momentum is affected by the presence of cloud shell like other scalar variables, so the use of the fractional dilution rate is more appropriate. The ML model predicts $\dot{w}$ with lower prediction skill compared to ϵ and δ. This result may indicate that large stochasticity acts on vertical acceleration due to the uncertainty caused by the vertical pressure gradient, or the ML model does not work well for predicting $\dot{w}$.

The superior performance of neural network is due to its ability to approximate the arbitrary continuous function from multiple inputs (Cybenko, 1989). Figure 7 shows the relationship between ϵ and six selected input variables for the ML model. $\hat{B}$, $\hat{w}$, $\hat{q}_l$, and $\hat{q}_t'$ are negatively correlated with ϵ, while $\hat{\theta}_c'$ is positively correlated with ϵ. $\partial\overline{\theta_v}/\partial z$ does not show a noticeable relationship with ϵ, which displays the largest PI value (Table 1). This suggests that other variables obscure the true strength of the dependence of ϵ on $\partial\overline{\theta_v}/\partial z$. The relationship between input variables and δ is a little more complex than that between input variables and ϵ (Figure 8). δ shows non-monotonic responses to $\hat{q}_l$, $\hat{\theta}_c'$, and $\hat{q}_t'$. Here, a strong inverse relationship between $\hat{w}$ and δ is apparent, where $\hat{w}$ explains largest variabilities on δ among other variables. The ML model successfully reproduces the dependency of ϵ and δ on the six input

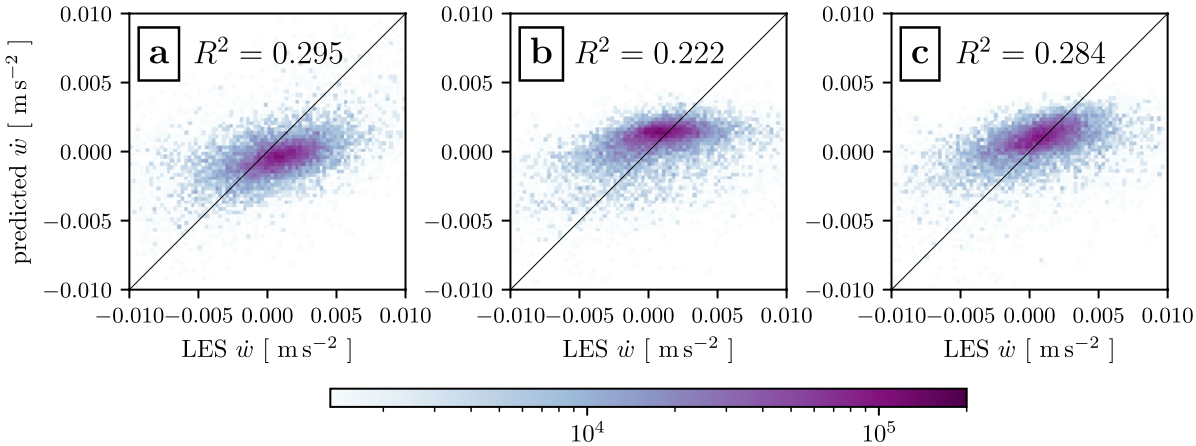

Figure 6. Joint probability density functions of large-eddy simulation measured $\dot{w}$ versus predicted $\dot{w}$ by various parameterizations for individual clouds. (a) $\dot{w} = a\hat{B} - b\epsilon_\phi \hat{w}^2$, (b) multiple linear regression model, and (c) machine learning model. a and b are fitting parameters.

variables throughout the sample space (red lines in Figures 7 and 8). However, the ML model tends to slightly underestimate ϵ.

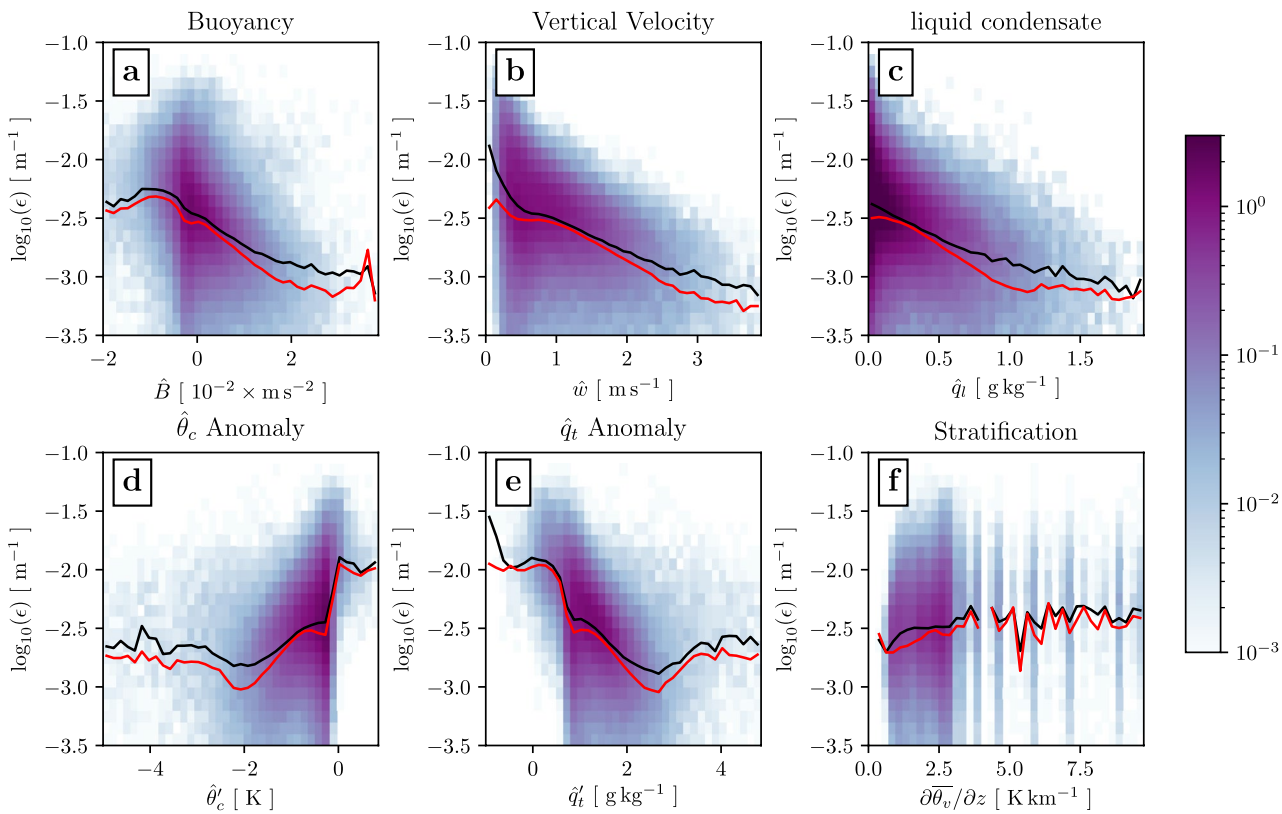

Figure 7. Joint probability density functions of large-eddy simulation (LES) measured $\log_{10}(\epsilon)$ versus selected six input variables for the machine learning (ML) model for individual clouds. (a) Buoyancy, (b) vertical velocity, (c) liquid condensate, (d) $\hat{\theta}_c$ anomaly, (e) $\hat{q}_t$ anomaly, and (f) vertical gradient of environmental virtual potential temperature. Black lines indicate the mean of LES measured $\log_{10}(\epsilon)$ as a function of the x-axis variable, and red lines indicate the mean $\log_{10}(\epsilon)$ predicted by the ML model as a function of the x-axis variable.

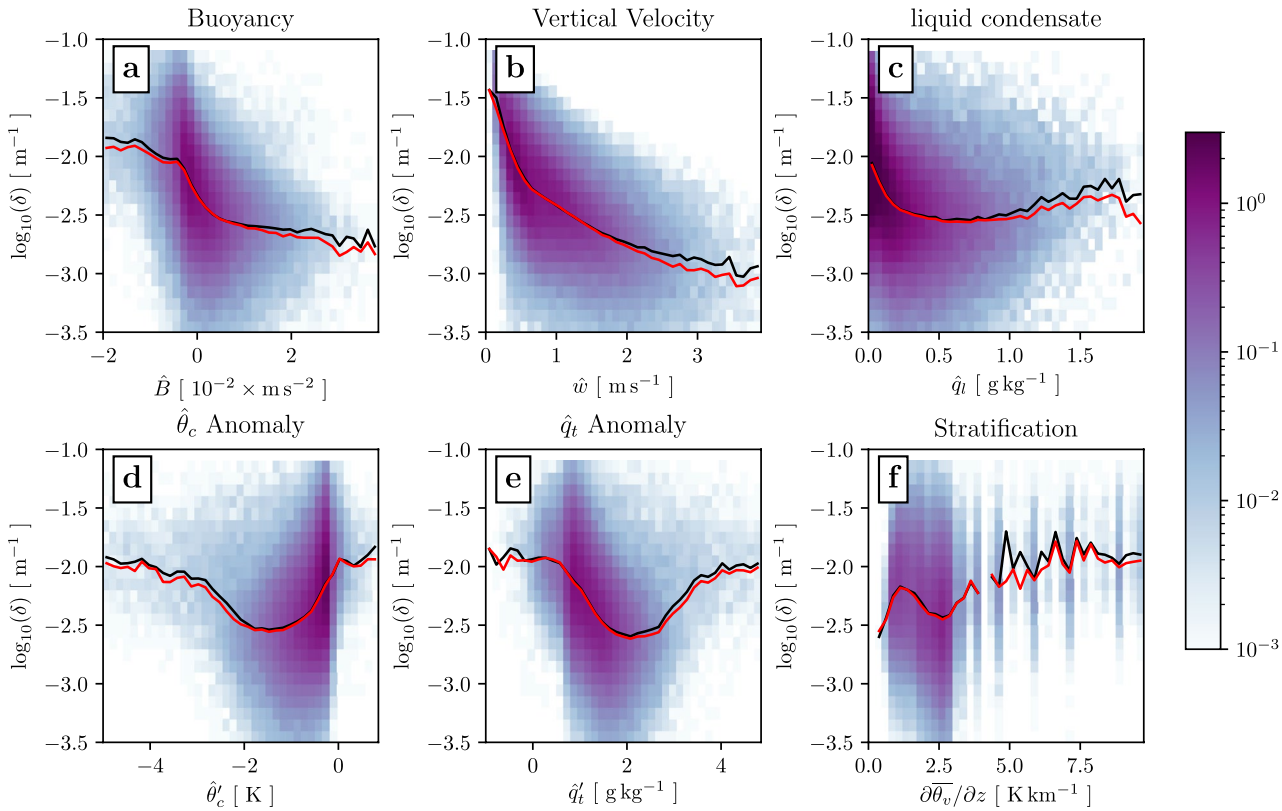

Figure 8. Joint probability density functions of large-eddy simulation (LES) measured $\log_{10}(\delta)$ versus selected six input variables for the machine learning (ML) model for individual clouds. (a) Buoyancy, (b) vertical velocity, (c) liquid condensate, (d) $\hat{\theta}_c$ anomaly, (e) $\hat{q}_t$ anomaly, and (f) vertical gradient of environmental virtual potential temperature. Black lines indicate the mean of LES measured $\log_{10}(\delta)$ as a function of the x-axis variable, and red lines indicate the mean $\log_{10}(\delta)$ predicted by the ML model as a function of the x-axis variable.

5. Single-Column Model Simulation Results

The new mixing model is tested using the SCM simulations of the BOMEX and RICO cases. We simulate different configurations of SCM listed in Table 3 in order to figure out the source of cloud variabilities. ML-FullVar is the default configuration with the ML based mixing model, where both the stochastic mixing and stochastic initialization are enabled. ML-MixVar is the same configuration as ML-FullVar, but the stochastic initialization is disabled by setting the initial condition of updrafts as the mean values of the surface PDF. ML-InitVar is the same configuration as ML-FullVar but with the stochastic mixing disabled. The stochastic mixing can be disabled by setting the mixing rates as its expected value, $\chi_{i,\mathrm{exp}}$. In addition, we test the mixing model of Park (2014) with the stochastic initialization on and off. Note that P14-NoVar is slightly different from the original UNICON, where

Table 3
A List of Single-Column Model Simulation Configurations

Configuration name	Mixing model	Stochastic mixing	Stochastic initialization
ML-FullVar	Machine learning	On	On
ML-MixVar	Machine learning	On	Off
ML-InitVar	Machine learning	Off	On
P14-InitVar	Park (2014)	Off	On
P14-NoVar	Park (2014)	Off	Off
UNICON	Park (2014)	Off	Off

Note. P14-NoVar is slightly different from the original unified convection scheme, where the fractional updraft area at the surface $\hat{A}_s$ is changed from 0.040 to 0.025, convective downdrafts by mixing are not allowed, and the auto-conversion scheme is replaced by Kogan (2013).

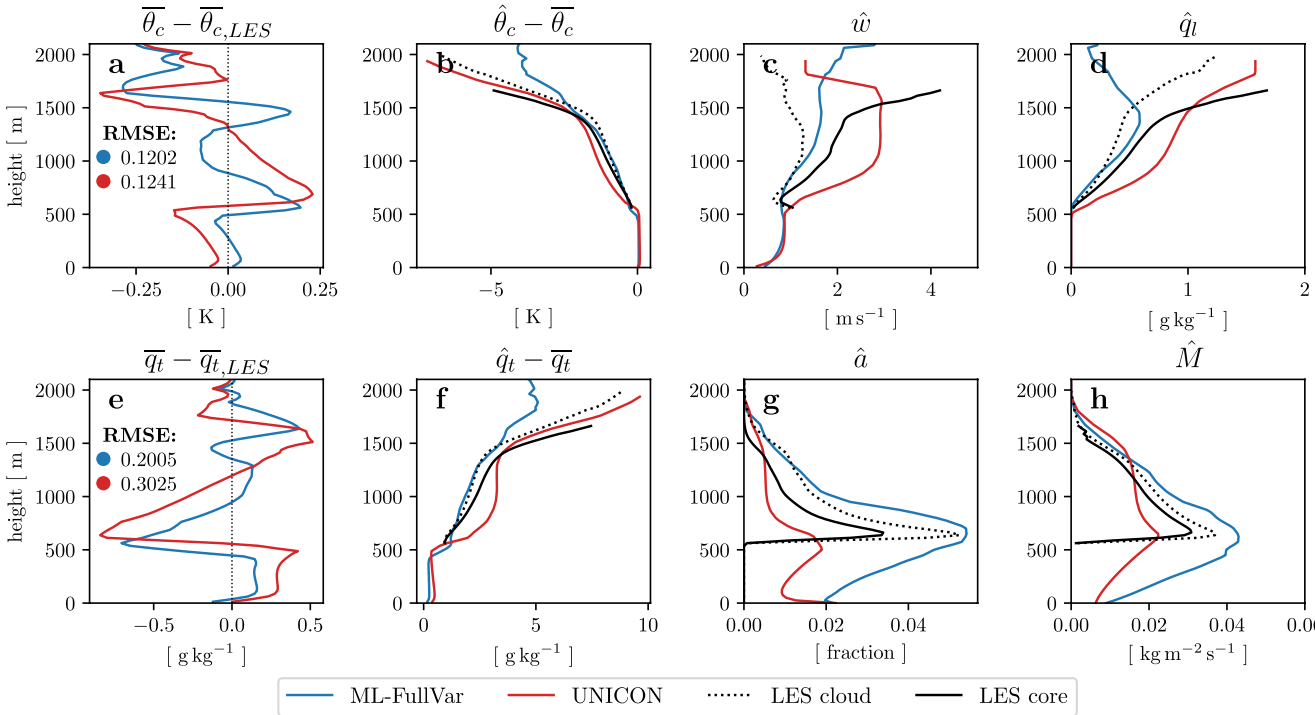

Figure 9. Vertical profiles of (a and e) error of environmental mean moist conserved variables with respect to large-eddy simulation (LES), (b and f) difference of the moist conserved variables from updrafts with respect to the environmental mean moist conserved variables, (c) updraft vertical velocity, (d) updraft liquid water, (g) updraft fractional area, and (h) updraft fractional mass flux averaged over t = 4–6 hr simulated by ML-FullVar and the original unified convection scheme for the Barbados Oceanographic and Meteorological Experiment case. In panels (a and e), root-mean-square errors of $\overline{\theta_c}$ and $\overline{q_t}$ within LES vertical domain are shown.

the fractional updraft area at the surface $\hat{A}_s$ is changed from 0.040 to 0.025, convective downdrafts by mixing are not allowed, and the auto-conversion scheme is replaced by Kogan (2013).

5.1. Mean Vertical Profiles

Figure 9 shows the simulated vertical profiles of various updraft properties for the BOMEX case. Here, we compare ML-FullVar with the original UNICON to examine the performance of the new mixing model with respect to the existing scheme. UNICON exhibits cold and moist biases in the sub-cloud layer, and warm and dry biases in the cloud layer below 1,300 m (Figures 9a and 9e). ML-FullVar reduces these biases, especially for $\overline{q_t}$. The RMSE of $\overline{\theta_c}$ and $\overline{q_t}$ are reduced by 3% and 34% in ML-FullVar, respectively, compared to UNICON. Based on the fact that the physical tendency of the mean conservative scalar $\overline{\phi}$ due to the multiple updrafts can be expressed as $\left(\partial\overline{\phi}/\partial t\right)_{\text{conv}} = -\partial\left[\sum_i \hat{M}^i\left(\hat{\phi}^i - \overline{\phi}\right)\right]/\partial z$ (i denotes individual updrafts), the reduction of the error can be contributed by more realistic simulation of updraft mass flux and moist conservative scalars within updrafts. ML-FullVar simulates a smooth mass flux profile similar to LES, while UNICON simulates mass flux with a rapid slope change near the height of 1,500 m (Figure 9h). The rapid decrease of mass flux near the inversion height in UNICON occurs since the bulk plume scheme lacks variation in convection top heights and terminates at a certain height (Shin & Park, 2020). ML-FullVar also simulates realistic $\hat{\theta}_c$ and $\hat{q}_t$ in the cloud layer below 1,500 m, where simulated profiles are not much deviated from LES cloud and core profiles (Figures 9b and 9f). In contrast, UNICON shows a rapid increase of $\hat{q}_t - \overline{q_t}$ (decrease of $\hat{\theta}_c - \overline{\theta_c}$) in the lower cloud layer, which results in relative θ_c flux divergence (q_t flux convergence) in the lower cloud layer and the excessive dry biases (warm biases). UNICON also suffers from rapid increases of $\hat{w}$ and $\hat{q}_l$ in the lower cloud layer and shows too large values throughout the cloud layer compared to LES. The reason for the rapid increases will be discussed later. For the BOMEX case, the updraft fractional area and the mass flux in ML-FullVar are likely to represent the LES cloud rather than the LES core. However, they can be easily controlled by the surface updraft fractional area $\hat{A}_s$ which is apparently the most important tuning parameter.

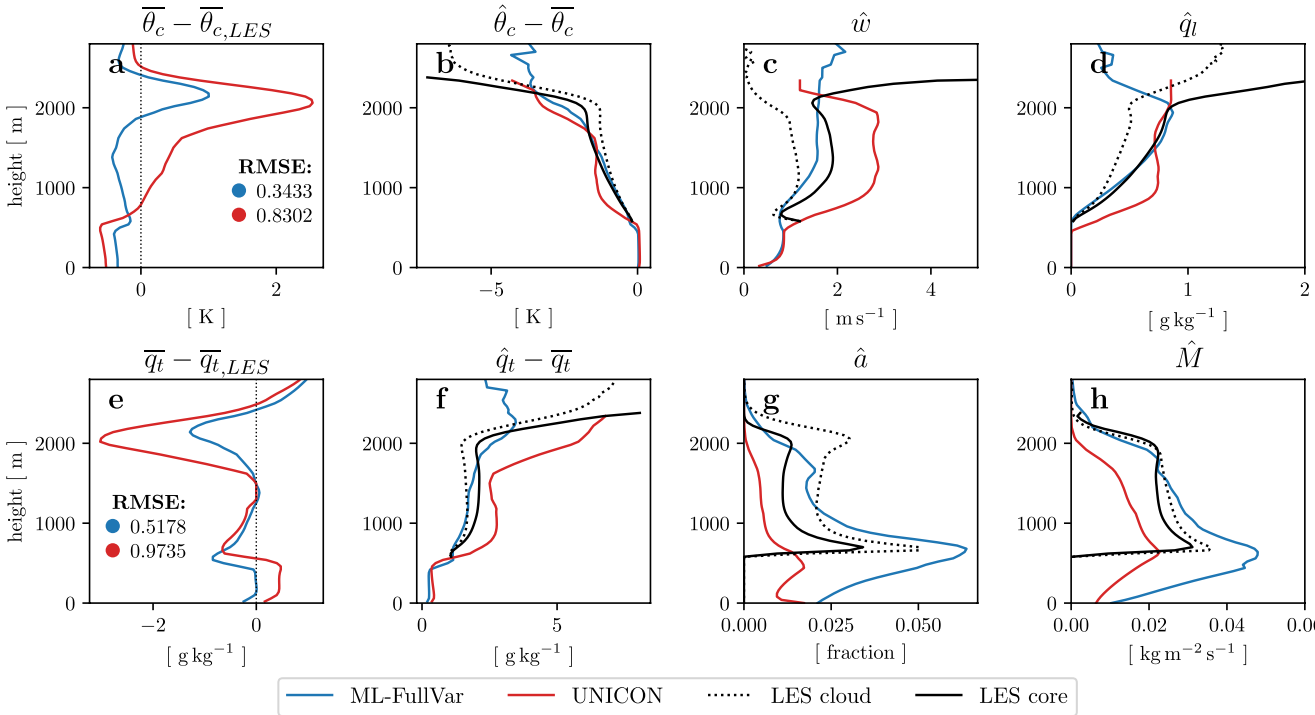

Figure 10. Vertical profiles of (a and e) error of environmental mean moist conserved variables with respect to large-eddy simulation (LES), (b and f) difference of the moist conserved variables from updrafts with respect to the environmental mean moist conserved variables, (c) updraft vertical velocity, (d) updraft liquid water, (g) updraft fractional area, and (h) updraft fractional mass flux averaged over t = 20–24 hr simulated by ML-FullVar and the original unified convection scheme for the Rain in Cumulus over the Ocean case. In panels (a and e), root-mean-square errors of $\overline{\theta_c}$ and $\overline{q_t}$ within LES vertical domain are shown.

The simulated vertical profiles for the RICO case lead to similar discussions as for the BOMEX case (Figure 10). For the RICO case, UNICON exhibits excessive warm and dry biases in the cloud layer above 1,700 m up to > 2 K and < −2 g kg^{-1}, respectively. ML-FullVar greatly reduces the biases, where 59% of RMSE in $\overline{\theta_c}$ and 47% of RMSE in $\overline{q_t}$ are reduced. The large biases in the upper cloud layer in the UNICON simulation are mainly due to the fact that UNICON simulates too small updraft mass flux (Figure 10h). The lack of mass flux convergence in the upper cloud layer leads to excessive warm and dry biases. Here again, UNICON exhibits the rapid increases of $\hat{w}$, $\hat{q}_l$, and $\hat{q}_t - \overline{q_t}$ (decrease of $\hat{\theta}_c - \overline{\theta_c}$) in the lower cloud layer, while ML-FullVar shows the smoother and realistic profiles. Notably, the simulated profiles in ML-FullVar are more likely to follow LES core profiles, although the ML model is trained for the non-core region. However, simulated profiles of $\hat{\theta}_c - \overline{\theta_c}$ and $\hat{q}_t - \overline{q_t}$ above the inversion height of ~2,000 m are largely deviated from the LES core or cloud profiles. For LES, the constraints of $q_l > 0$ and $B > 0$ lead to sampling of highly undiluted air parcels above the inversion height, characterized as a large magnitude of anomalies. In contrast, negatively buoyant and unsaturated updrafts are included in ML-FullVar, so the magnitude of anomalies is much smaller.

The vertical profiles simulated by all SCM configurations listed in Table 3 are shown as Figures S1 and S2 in Supporting Information S1 for the BOMEX and RICO cases, respectively. In addition, Table S1 in Supporting Information S1 provides the RMSEs of $\overline{\theta_c}$ and $\overline{q_t}$ simulated by the SCM configurations. All configurations without the ML mixing model (P14-InitVar, P14-NoVar, and UNICON) suffer from the rapid increases of $\hat{w}$, $\hat{q}_l$, and $\hat{q}_t - \overline{q_t}$ (decrease of $\hat{\theta}_c - \overline{\theta_c}$) in the lower cloud layer, while the configurations with the ML mixing model do not. Comparing the performances of the ML configurations, the mass flux of ML-InitVar rapidly decreases in the upper cloud layer, especially in the RICO case (Figure S2h in Supporting Information S1). It appears that the updrafts simulated by ML-InitVar do not extend beyond a certain height since ML-InitVar does not produce updrafts with high vertical velocity $\left(\hat{w} > 2\,\mathrm{m\,s^{-1}}\right)$ stochastically (see fourth column in Figure 12). The realistic simulations of convective mass flux and updraft thermodynamic variables by ML-FullVar and ML-MixVar contribute to generally smaller RMSEs compared to other configurations (Table S1 in Supporting Information S1). In summary, the

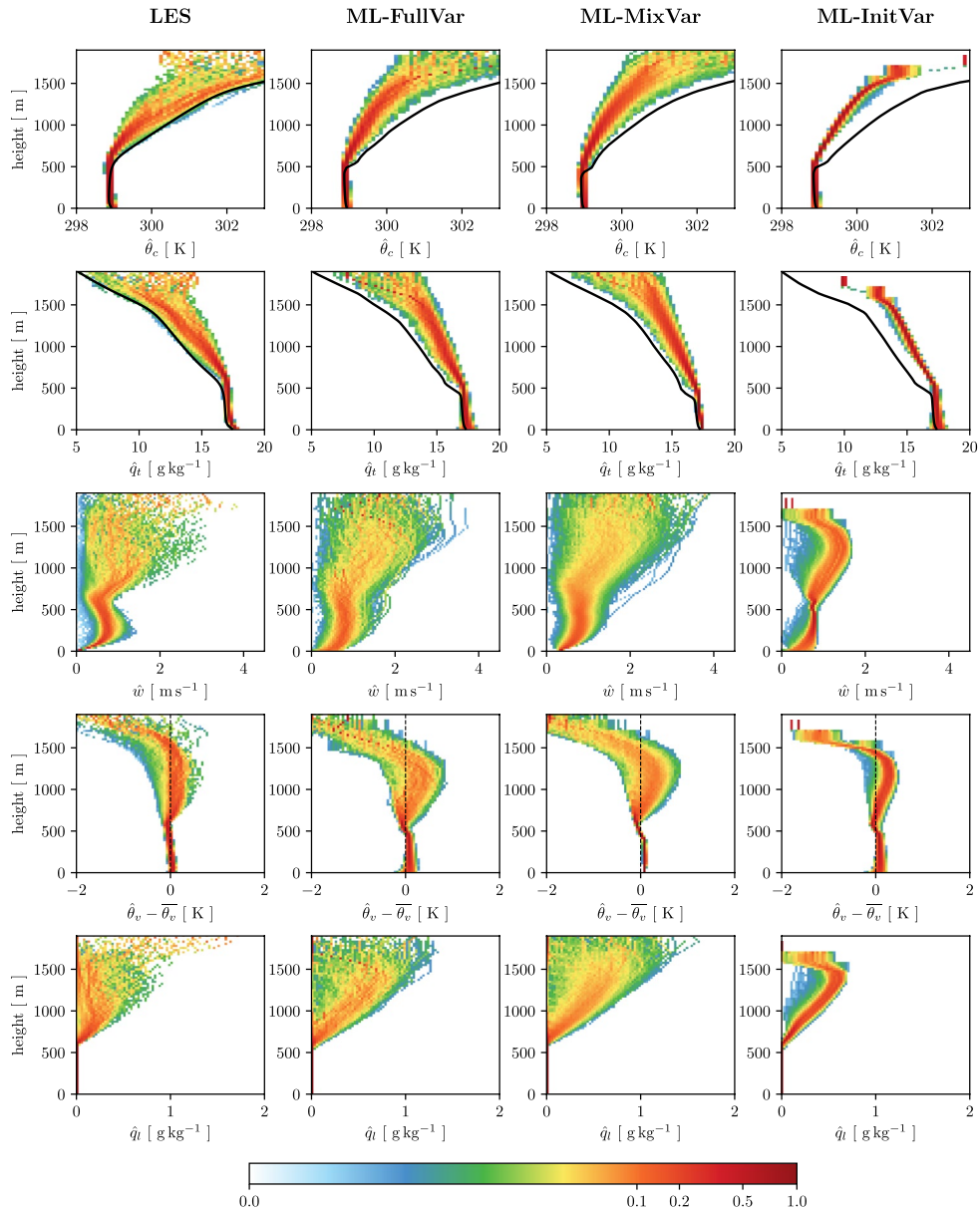

Figure 11. Normalized mass flux distributions as functions of various variables simulated by large-eddy simulation, ML-FullVar, ML-MixVar, and ML-InitVar for the Barbados Oceanographic and Meteorological Experiment case. Each row represents $\hat{\theta}_c$, $\hat{q}_t$, $\hat{w}$, $\hat{\theta}_v - \overline{\theta_v}$, and $\hat{q}_l$, respectively. Solid lines denote mean environmental profiles.

improvement seen in ML-FullVar compared to the original UNICON is the combination of better prediction of the mixing rates by the ML model and the use of stochastic mixing.

5.2. Cloud Variabilities

To see how cloud variabilities in the stochastic mixing model are evolved vertically, we draw the normalized mass flux distribution of various variables (Figures 11 and 12). The normalized mass flux is defined as $\hat{M}\Delta\phi / \sum \hat{M}\Delta\phi$ at each height, where $\Delta\phi$ is a bin of variable ϕ, slightly modifying the method of Romps (2016). Starting with the BOMEX case, Figure 11 demonstrates that ML-FullVar and ML-MixVar well reproduce the cloud variance in LES overall. The results of the two simulations are remarkably similar, which implies that the stochastic mixing process is the main source of cloud variabilities. ML-InitVar produces some variance but much smaller compared to LES. The deterministic parameterizations of the entrainment process have proposed a theory that

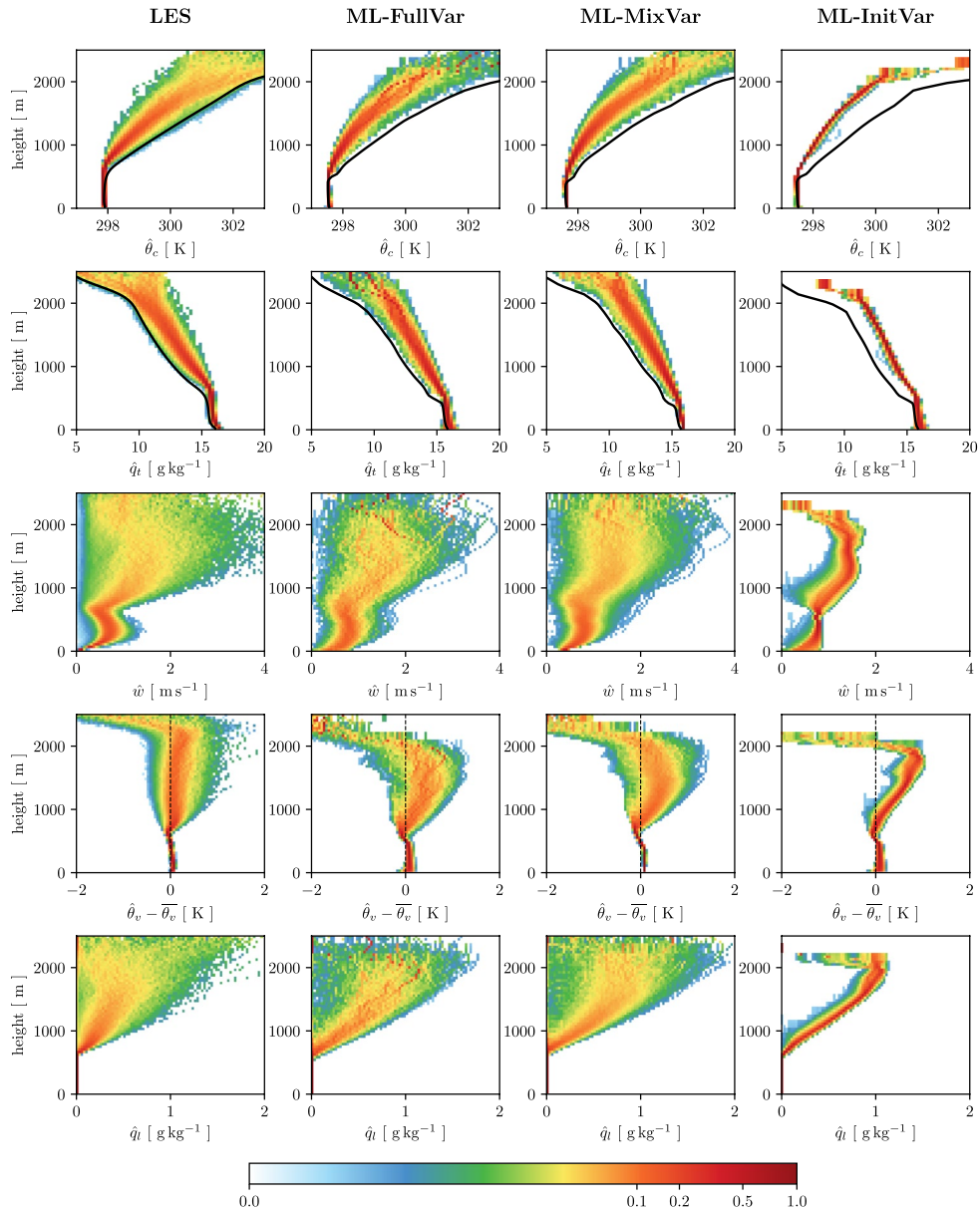

Figure 12. Normalized mass flux distributions as functions of various variables simulated by large-eddy simulation, ML-FullVar, ML-MixVar, and ML-InitVar for the Rain in Cumulus over the Ocean case. Each row represents $\hat{\theta}_c$, $\hat{q}_t$, $\hat{w}$, $\hat{\theta}_v - \overline{\theta_v}$, and $\hat{q}_l$, respectively. Solid lines denote mean environmental profiles.

cloud variabilities can be generated by the amplification of cloud-base variabilities (e.g., Neggers et al., 2002). However, as shown in this analysis, cloud variabilities can be represented correctly only when the randomness in the mixing process is considered.

The variances of two moist conserved variables (first and second rows of Figure 11) remain small in the sub-cloud layer. After updrafts penetrated the planetary boundary layer (PBL), the variances start to increase with respect to height. Compared to LES, ML-FullVar and ML-MixVar tend to simulate less diluted updrafts, resulting in means of the moist conserved variables biased away from the environment profiles. The mean and variance of the vertical velocity $\hat{w}$ increase in the lower sub-cloud layer, and the mean $\hat{w}$ decreases in the upper sub-cloud layer (third row). The mean and variance of $\hat{w}$ increase above the PBL and are distributed for a wide range of 0–3 ms^{-1}, due to the large stochasticity within vertical acceleration (Figure 6). A sensitivity simulation with the standard vertical velocity equation (Equation 7) displays a much smaller variance of $\hat{w}$ (not shown). The buoyancy of the updrafts

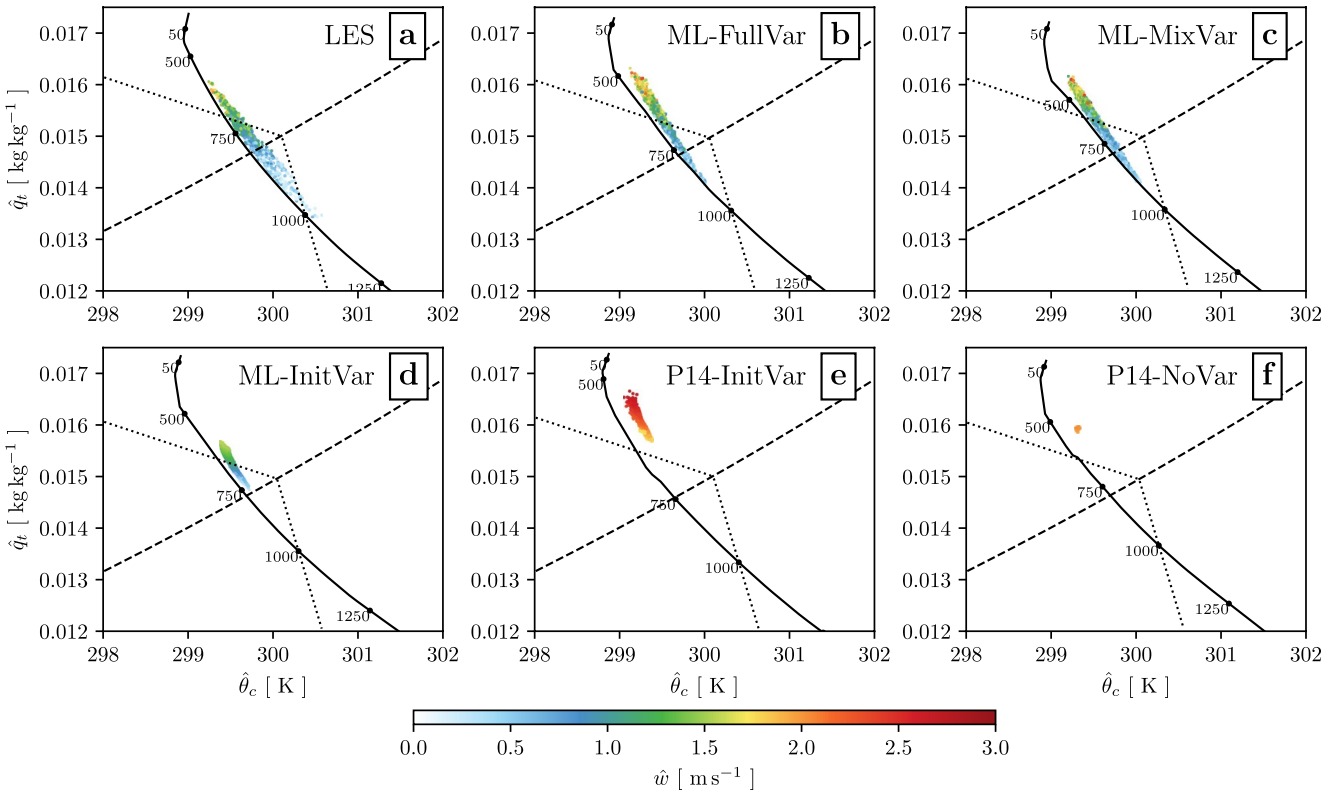

Figure 13. The Paluch diagrams of large-eddy simulation and single-column model simulations at 1,000 m for the Barbados Oceanographic and Meteorological Experiment case. Each point corresponds to a single convective updraft colored by vertical velocity. The solid line corresponds to the mean environmental profile, the dashed line denotes the saturation, and the dotted line denotes the neutral buoyancy. The circles on the mean environmental profile are labeled with the corresponding heights in meters.

is mostly positive near the surface but becomes negative near the PBL top height (fourth row), which is related to convective inhibition. After the inhibition layer, most of the updraft mass flux become positively buoyant, while some are negatively buoyant. Above 1,500 m, almost every updraft becomes negatively buoyant. ML-FullVar and ML-MixVar overestimate the variance of buoyancy in the cloud layer compared to LES. Finally, the variance of liquid water content $\hat{q}_l$ increases within the cloud layer and ranges from 0 to 1.5 g kg^{-1}. ML-FullVar and ML-MixVar produce larger $\hat{q}_l$ than LES, but as shown in Figure 9d , they tend to represent the core properties rather than the non-core region.

The normalized mass flux distributions in the RICO case show similar results (Figure 12). The RICO case shows wider spectra of updraft properties than the BOMEX case, with vertical velocity up to 4 ms^{-1} and liquid water content up to 2 g kg^{-1}. ML-FullVar and ML-MixVar simulate these variabilities quite well.

Figure 13 shows the Paluch diagrams (scatter plots of two moist conserved variables) at 1,000 m simulated by LES and the five SCM configurations in the BOMEX case. The scatters located upper-left are the updrafts that are less diluted and have properties of near surface, and the scatters located lower-right are the updrafts that are highly diluted and have properties of the environment at 1,000 m. The LES simulates updrafts with various thermodynamic states, where many convective updrafts are negatively buoyant and even unsaturated (Figure 13a). However, these negatively buoyant updrafts account for a lower fraction of the total mass flux compared to the positively buoyant updrafts as seen from Figure 11, due to their low vertical velocity. The positively buoyant updrafts have vertical velocity up to 2 m s^{-1}. ML-FullVar and ML-MixVar produce the spectrum of updrafts similar to LES. They also simulate the negatively buoyant and unsaturated updrafts, but with a much smaller number of unsaturated updrafts. ML-InitVar simulates a narrower spectrum of updrafts compared to LES and does not produce any unsaturated updrafts at this level. P14-InitVar produces a narrow spectrum of less diluted updrafts, with excessively high vertical velocities. While ML-InitVar and P14-InitVar use different mixing models, they produce a similar amount of variabilities by the stochastic initialization of updrafts. The configuration with

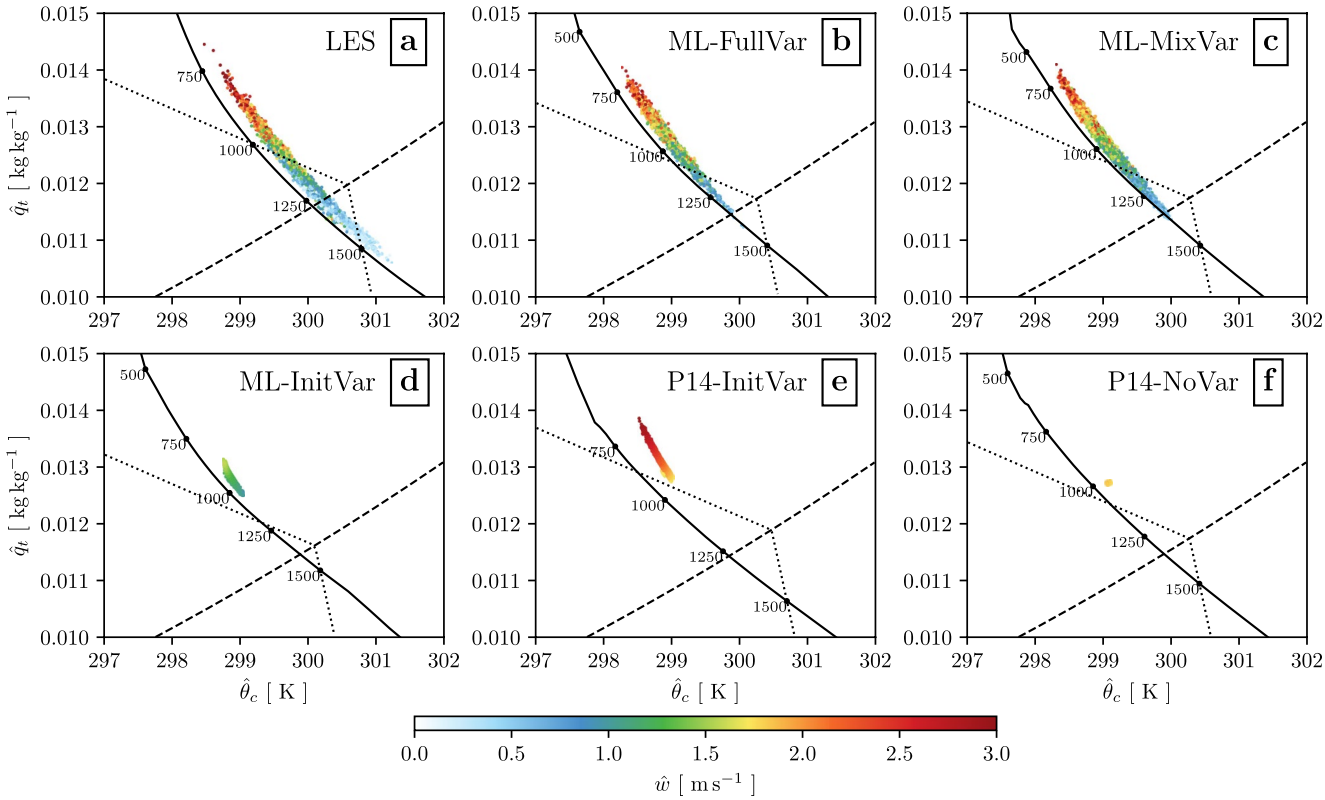

Figure 14. The Paluch diagrams of large-eddy simulation and single-column model simulations at 1,500 m for the Rain in Cumulus over the Ocean case. Each point corresponds to a single convective updraft colored by vertical velocity. The solid line corresponds to the mean environmental profile, the dashed line denotes the saturation, and the dotted line denotes the neutral buoyancy. The circles on the mean environmental profile are labeled with the corresponding heights in meters.

zero stochasticity, P14-NoVar, shows virtually no variabilities as expected. It appears that the mixing model of Park (2014) underestimates the dilution of updrafts in the lower cloud layer, which results in the steep increases of vertical velocity and liquid water content.

The Paluch diagrams of the RICO case at 1,500 m show wider spectra of updrafts compared to the BOMEX case (Figure 14). Here again, ML-FullVar and ML-MixVar well reproduce the updraft spectrum of LES, but the number of unsaturated updrafts is greatly reduced. Our definition of the convective updrafts in LES allows negatively buoyant or unsaturated convection that is decoupled from the mixing layer. Using the classification of Stull (1985), these are passive clouds that are remnants of the old decaying clouds or are formed due to gravity waves. It seems that our mixing model only simulates the active and forced convection which penetrate the convective inhibition layer, even though we trained the ML model for all types of convection. While the impact of passive clouds is small compared to active clouds, passive clouds can be handled explicitly like the stochastic convection scheme of Sakradzija et al. (2015). At least for the active and forced convection, the new mixing model well represents the development of variabilities of various variables from the surface. Not shown in this paper, ML-FullVar and ML-MixVar are capable of realistically simulating the joint PDFs and correlations of other updraft properties as well (e.g., $\hat{w}$ vs. $\hat{q}_t$).

Finally, as a measure of the stochasticity in the mixing process, we calculate the correlation profiles between updraft properties at cloud base and any height (Figures 15 and 16). Here, the cloud base is defined as the minimum height at which the total cloud fraction has a local maximum. If the mixing process is a purely deterministic function of cloud properties, then cloud properties will be highly correlated with cloud-base properties. In contrast, if the mixing process is a purely stochastic process, then the upper-level cloud properties will lose correlation with cloud-base properties rapidly. The correlation profiles of the BOMEX LES show exponential decreases of the correlations in the cloud layer for $\hat{\theta}_c$, $\hat{q}_t$, and $\hat{w}$ (Figure 15). The correlations of $\hat{\theta}_c$ and $\hat{q}_t$ are smaller than 0.5 at 800 m and approach 0 above 1,200 m. In the sub-cloud layer, the correlations are uniform with the values around 0.7 and 0.3 for $\hat{\theta}_c$ and $\hat{q}_t$, respectively. The correlation of $\hat{w}$ shows a relatively slow decrease and

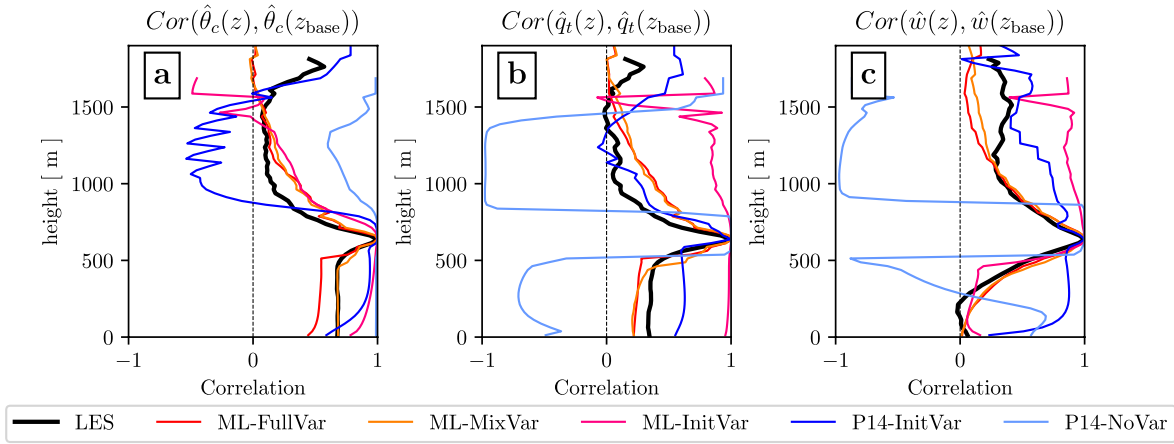

Figure 15. Correlation profiles between simulated updraft properties at cloud-base height z_{base} and at any height z for (a) $\hat{\theta}$, (b) $\hat{q}_t$, and (c) $\hat{w}$ in the Barbados Oceanographic and Meteorological Experiment case.

saturates at the value around 0.3 in the cloud layer. In the sub-cloud layer, $\hat{w}$ loses correlation exponentially as getting farther from the cloud base and approaches correlation of 0 near the surface. The results of the BOMEX LES are consistent with those of Dawe and Austin (2012).

The correlation profiles of ML-FullVar and ML-MixVar are remarkably similar to LES, with exponential decreases of correlation in the cloud layer. ML-FullVar and ML-MixVar also produce realistic correlation profiles in the sub-cloud layer. One notable deficiency of ML-FullVar and ML-MixVar is that they simulate relatively slow decreases of correlations in the cloud layer for $\hat{\theta}_c$ and $\hat{q}_t$. The correlation profiles of the other SCM configurations are deviated largely from LES profiles and do not display a systematic trend. The largest correlation values are from P14-NoVar, which shows correlations around 1 or −1. The correlations can be changed from 1 to −1 or vice-versa when the ordering of the variable is reversed while updrafts are ascending. The high correlation values represent the lack of stochasticity in P14-NoVar. ML-InitVar and P14-InitVar show smaller correlations compared to P14-NoVar, but the profiles do not resemble the LES profiles.

The results of the RICO case are similar to those of the BOMEX case (Figure 16). In the RICO case, correlations of $\hat{\theta}_c$ and $\hat{q}_t$ decrease rapidly below 1,000 m and are saturated to the values of 0.1–0.2 in LES. ML-FullVar and ML-MixVar show much slower decreases of the correlations compared to LES.

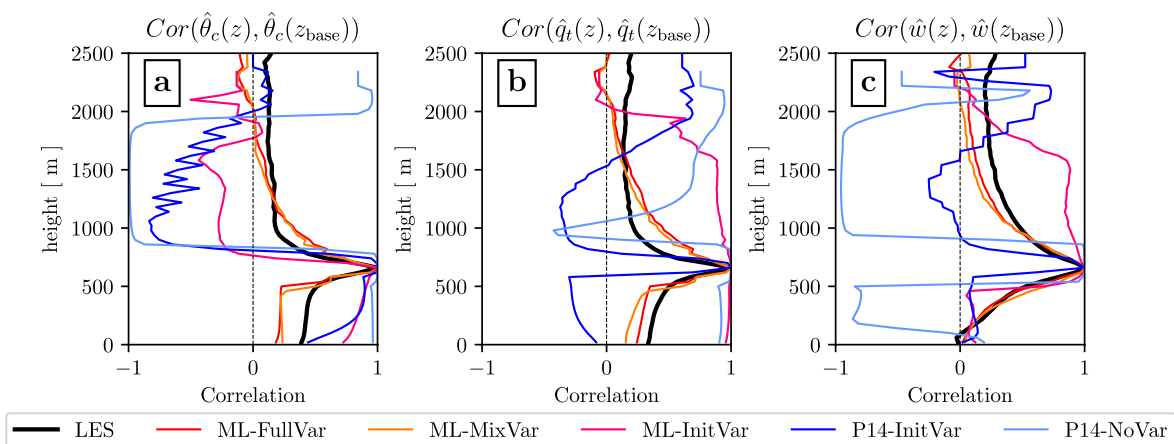

Figure 16. Correlation profiles between simulated updraft properties at cloud-base height z_{base} and at any height z for (a) $\hat{\theta}$, (b) $\hat{q}_t$, and (c) $\hat{w}$ in the Rain in Cumulus over the Ocean case.

Table 4
Table of R^2 Between ϵ (δ in Parentheses) Predicted by the Machine Learning Model and Large-Eddy Simulation Measured ϵ (δ) When Different Combinations of the Data Sets Are Used for Training and Testing

Training \ Testing	BOMEX + RICO	BOMEX	RICO
BOMEX + RICO	0.655 (0.665)	0.679 (0.637)	0.637 (0.661)
BOMEX	0.583 (0.418)	0.699 (0.706)	0.588 (0.442)
RICO	0.629 (0.648)	0.662 (0.568)	0.655 (0.653)

6. Machine Learning Model Dependency on Data Set

In order for the proposed method to be used in full GCM simulations, the ML model should simulate realistic convection properties under diverse conditions. It is ideal to train the ML model with the convection statistics from various large-scale conditions, ranging from shallow to deep convection regimes. In this study, however, the ML model is trained on only two marine shallow convection cases. Since the training and the validation are done with the same BOMEX and RICO cases, there is a possibility that the ML model is overfitted to the BOMEX and RICO cases and may not be applicable for other cases. To determine the impact of this issue, we train the ML model on one of the BOMEX and RICO cases and see if the performance degrades when tested on the other case. Here, we consider three data sets for the training and testing: BOMEX + RICO, BOMEX, and RICO.

Table 4 summarizes the offline test performance of the ML model when different combinations of the data sets are used for training and testing. In general, the ML model works best when the same data set is used for training and testing. The ML model trained on the BOMEX + RICO data set can predict ϵ and δ of all three data sets reasonably well, with R^2 slightly lower than when the same data set is used for training and testing. The noticeable decreases of R^2 are found when predicting the RICO or BOMEX + RICO data set with the ML model trained on the BOMEX data set. This is because the RICO data set has samples from stronger updrafts compared to the BOMEX data set. The RICO data set includes samples with $\hat{B} > 2 \times 10^{-2}\,\mathrm{m\,s^{-2}}$, $\hat{w} > 3\,\mathrm{m\,s^{-1}}$, and $\hat{q}_l > 1\,\mathrm{g\,kg^{-1}}$, which rarely exist in the BOMEX data set (Figure 17).

The SCM experiments with the ML models trained on the different data sets are also tested. Here, ML-FullVar configurations with the ML model trained on BOMEX + RICO, BOMEX, and RICO data sets are referred to as ML[BOMEX + RICO], ML[BOMEX], and ML[RICO], respectively. Figures S3 and S4 in Supporting Information S1 show the vertical profiles simulated by these three SCM configurations and the original UNICON for the BOMEX and RICO cases, respectively. The profiles of the thermodynamic variables simulated by ML[BOMEX + RICO], ML[BOMEX], and ML[RICO] are similar, but the profiles of the mass flux show

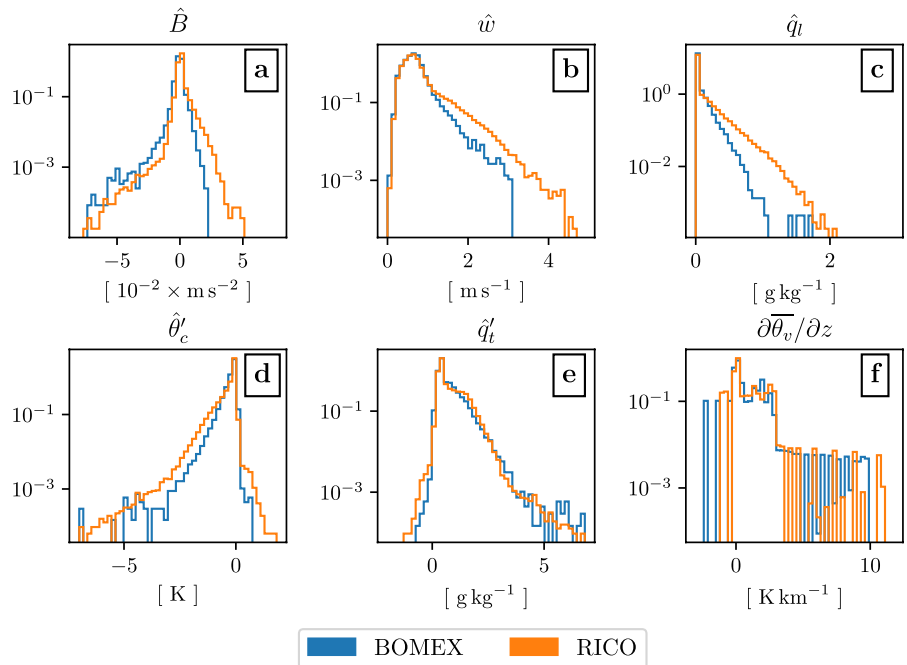

Figure 17. Histograms of (a) buoyancy, (b) vertical velocity, (c) liquid condensate, (d) $\hat{\theta}_c$ anomaly, (e) $\hat{q}_t$ anomaly, and (f) vertical gradient of environmental virtual potential temperature for the Barbados Oceanographic and Meteorological Experiment and Rain in Cumulus over the Ocean data sets.

Table 5
Root-Mean-Square Errors (RMSEs) of $\overline{\theta_c}$ (K) and $\overline{q_t}$ ($g\,kg^{-1}$) Simulated by ML[BOMEX + RICO], ML[BOMEX], ML[RICO], and the Original Unified Convection Scheme for the BOMEX and RICO Cases

Metrics	BOMEX		RICO	
Model	$\overline{\theta_c}$ RMSE	$\overline{q_t}$ RMSE	$\overline{\theta_c}$ RMSE	$\overline{q_t}$ RMSE
ML[BOMEX + RICO]	0.1202	0.2005	0.3433	0.5178
ML[BOMEX]	0.1206	0.2283	0.3805	0.5702
ML[RICO]	0.1737	0.3855	0.3961	0.5398
UNICON	0.1241	0.3025	0.8302	0.9735

Note. RMSEs are calculated within the large-eddy simulation vertical domain.

substantial differences. In the BOMEX SCM experiments, the mass flux profiles of ML[RICO] show a bulge in the cloud layer, resulting in excessive cold and moist biases at 1,000–1,700 m. We found that the ML model trained on RICO does not perform well under negative buoyancy conditions since the RICO data set has a much smaller number of samples with negative buoyancy compared to the BOMEX data set (Figure 17a). In the RICO SCM experiments, ML[BOMEX] has a relatively smaller mass flux compared to ML[BOMEX + RICO] and ML[RICO]. Finally, RMSEs of $\overline{\theta_c}$ and $\overline{q_t}$ from the SCM experiments are summarized in Table 5. Among the tested configurations, ML[BOMEX + RICO] has the smallest RMSEs for the BOMEX and RICO cases. It is notable that ML[BOMEX + RICO] shows slightly better performance compared to the case when the training and testing case is the same. In summary, the offline and SCM tests above suggest that the performance of the ML model is guaranteed when the data range of testing is a subset of the data range of training (data denote the inputs and outputs of the ML model). Considering using the ML model for cases other than BOMEX and RICO, the ML model can be used if updraft statistics for that case are within the ranges of BOMEX and RICO.

7. Summary and Conclusions

We propose a stochastic mixing model with an ML technique for the mass flux convection schemes. The strategy of the model is to set the SDEs for the following four mixing rates presented in the governing equations of mass flux scheme: fractional entrainment rate ϵ, fractional detrainment rate δ, fractional dilution rate ϵ_ϕ, and vertical acceleration $\dot{w}$. The fractional dilution rate is defined to calculate the dilution of scalars by mixing process and differs from the fractional mass entrainment rate due to the effect of cloud shell. The unknown parameters of the SDEs are modeled using the deep neural network which takes cloud and environment properties as inputs, and the network is trained using the LES data set. To generate the data samples, LES simulations of the BOMEX and RICO shallow convection cases are conducted. Then, the mixing rates for each tracked cloud are calculated from the simulation outputs.

The input variables for the ML model are selected from eight candidates that are expected to be associated with the mixing process. In order to do this, the PI analysis is done to figure out the relative importance of variables when the ML model predicts the mixing rates. The analysis showed that $\partial\overline{\theta_v}/\partial z$ is the most important variable for predicting the fractional entrainment rate and $\hat{w}$ for the fractional detrainment rate. The variables with the least PI among the candidates are the updraft radius $\hat{R}$ and the vertical wind shear V_{shear}. Thus, the following six variables are selected for the input variables of ML model excluding these two: $\hat{B}$, $\hat{w}$, $\hat{q}_l$, $\hat{\theta}_c'$, $\hat{q}_t'$, and $\partial\overline{\theta_v}/\partial z$.

The performance of the ML model for predicting the fractional entrainment rate, fractional detrainment rate, and vertical acceleration is compared with those of previously proposed parameterizations. The ML model predicts ϵ with $R^2 = 0.655$, while the second-best parameterization which utilizes a power-law fitting of buoyancy and vertical velocity shows $R^2 = 0.354$. The performance of predicting δ using the ML model is $R^2 = 0.665$. In addition, the ML model was found to represent the relationship between the input variable and ϵ or δ shown in LES well. However, the ML model predicts vertical acceleration with a relatively low predictive skill of $R^2 = 0.284$, implying that the large stochasticity is associated with the updraft vertical velocity.

Next, the stochastic mixing model is implemented in the UNICON scheme and tested using SCM simulations of the BOMEX and RICO cases. The stochastic initialization of updrafts is formulated following Shin and Park (2020). The SCM experiment with the new stochastic mixing model showed a reduction of RMSEs of environmental θ_c and q_t profiles by 3% and 34% in the BOMEX case, and 59% and 47% in the RICO case, respectively, compared to the original UNICON. Also, the profiles of simulated mean updraft variables consistently matched with LES profiles, while the original UNICON showed rapid increases in vertical velocity and liquid water content in the lower cloud layer. The following configurations of SCM are tested to figure out which process is responsible for cloud variabilities: the default simulation (ML-FullVar), the simulation without the stochastic initialization (ML-MixVar), and the simulation without the stochastic mixing (ML-InitVar). In general, ML-FullVar and ML-MixVar are capable of realistically simulating the variabilities of various updrafts properties

including buoyancy, liquid water content, and vertical velocity, while the ML-InitVar produced a limited amount of variabilities. The simulation results of ML-FullVar and ML-MixVar are remarkably similar, implying that stochastic mixing is the main source of cloud variabilities. The Paluch diagrams from ML-FullVar and ML-MixVar suggest that our model can simulate realistic specta of active and forced convection but not the decaying clouds that are decoupled from the mixing layer. ML-FullVar and ML-MixVar simulate exponential decreases of correlations between updraft properties at cloud-base height and other height in the cloud layer, which is also presented in LES, suggesting that a reasonable amount of stochasticity is produced by the mixing model. Finally, the additional offline and SCM tests with the ML model trained on the different data sets (BOMEX + RICO, BOMEX, RICO) suggests that the performance of the ML model is guaranteed when the data range of simulating case is a subset of the data range of training.

Our framework can be extended to deep convection, but there are several aspects to be considered. First, specific humidity of cloud ice is needed to be included as model input. The updraft radius is currently excluded from the model input, but there is some evidence that the updraft radius plays an important role in the development of deep convection (Khairoutdinov et al., 2009). In addition, the production of convective downdraft during the mixing process should be considered.

In recent years, there have been several attempts to replace whole sub-grid physics with ML based parameterization (Rasp et al., 2018; Yuval & O'Gorman, 2020). The method promises great performance, but since ML works as a black box, underlying physics is inexplicable. Also, the method is not well generalized in unseen climates when appropriate physical constraints are not applied (Beucler et al., 2021; Rasp et al., 2018). This study can be regarded as an attempt to applying ML only for the process that physically based formulation is difficult, which is the mixing of convection. The proposed method can be more resilient to unseen climates since training space is much smaller compared to full ML physics parameterizations. In addition, mass and energy are conserved since adiabatic processes are still calculated in analytical ways (e.g., phase change of water and radiative heating). It is expected that the neural SDE framework used in this study can be applied to other stochastic physics parameterizations.

Appendix A: Computation of Lagrangian Tendencies for the Tracked Updrafts

The fractional dilution rate ϵ_ϕ^t and the vertical acceleration $\dot{w}$ are calculated using Lagrangian tracking of each convective updraft. We calculate these rates as mean tendency during $\Delta t = 60$ s which is the LES output sampling frequency. Let's consider an updraft parcel at height z and time t. The vertical position of the parcel is updated in time with forward differencing with a small sub-step time interval of $\Delta t_{\text{sub}} = 1$ s, and then the parcel position at $t + \Delta t$ can be obtained after 60 sub-steps. The updraft vertical velocity between time interval $[t, t + \Delta t]$ is calculated using a linear interpolation between model vertical grid points and sampling times. Similarly, the parcel position at $t - \Delta t$ is calculated using the backward differencing method. Hereinafter, any updraft property x at time $t - \Delta t$, t, and $t + \Delta t$ are denoted as x^{t-1}, x^t, and x^{t+1}, respectively.

The fractional dilution rate is estimated using the decaying passive tracer which is already used to define the plume region in LES. The fractional dilution rates ϵ_ϕ at time t and $t - \Delta t$ are calculated as

$$\epsilon_\phi^t = -\frac{1}{\hat{s}^{\,t+\frac{1}{2}} - \overline{s}^{\,t+\frac{1}{2}}} \left(\frac{\hat{s}^{\,t+1} - \hat{s}^{\,t}}{\Delta t} + \frac{\hat{s}^{\,t+\frac{1}{2}}}{\tau} \right), \tag{A1}$$

$$\epsilon_\phi^{t-1} = -\frac{1}{\hat{s}^{\,t-\frac{1}{2}} - \overline{s}^{\,t-\frac{1}{2}}} \left(\frac{\hat{s}^{\,t} - \hat{s}^{\,t-1}}{\Delta t} + \frac{\hat{s}^{\,t-\frac{1}{2}}}{\tau} \right), \tag{A2}$$

where $\hat{s}$ and $\overline{s}$ are the mean concentration of the decaying passive tracer within updraft and environment, respectively, and τ is the decaying time scale of the tracer. The superscript $t + \frac{1}{2}$ denotes the average of values at t and $t + \Delta t$ $\left(\text{e.g., } \hat{s}^{\,t+\frac{1}{2}} = \frac{1}{2}\left(\hat{s}^{\,t+1} + \hat{s}^{\,t}\right)\right)$, and similar for $t - \frac{1}{2}$. The term associated with τ is added to compensate the decaying tendency of the tracer. The vertical acceleration $\dot{w}$ are calculated as

$$\dot{w}^{\,t} = \frac{\hat{w}^{\,t+1} - \hat{w}^{\,t}}{\Delta t}, \tag{A3}$$

$$\dot{w}^{t-1} = \frac{\hat{w}^{t} - \hat{w}^{t-1}}{\Delta t}. \quad \text{(A4)}$$

Data Availability Statement

The python scripts to train the machine learning model are available from https://doi.org/10.5281/zenodo.6324561. TensorFlow library can be downloaded from https://www.tensorflow.org/.

Acknowledgments
The authors are grateful to three anonymous reviewers for providing valuable comments on this work. This work was supported by the National Research Foundation of Korea (NRF) under grants 2019R1A6A1A10073437 and 2021R1A2C1007044.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*. Retrieved from https://www.tensorflow.org/

Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: A corrected feature importance measure. *Bioinformatics*, *26*(10), 1340–1347. https://doi.org/10.1093/bioinformatics/btq134

Bechtold, P., Köhler, M., Jung, T., Doblas-Reyes, F., Leutbecher, M., Rodwell, M. J., et al. (2008). Advances in simulating atmospheric variability with the ECMWF model: From synoptic to decadal time-scales. *Quarterly Journal of the Royal Meteorological Society*, *134*(634), 1337–1351. https://doi.org/10.1002/qj.289

Beucler, T., Pritchard, M., Rasp, S., Ott, J., Baldi, P., & Gentine, P. (2021). Enforcing analytic constraints in neural networks emulating physical systems. *Physical Review Letters*, *126*(9), 098302. https://doi.org/10.1103/physrevlett.126.098302

Bretherton, C. S., McCaa, J. R., & Grenier, H. (2004). A new parameterization for shallow cumulus convection and its application to marine subtropical cloud-topped boundary layers. Part I: Description and 1D results. *Monthly Weather Review*, *132*(4), 864–882. https://doi.org/10.1175/1520-0493(2004)132<0864:anpfsc>2.0.co;2

Brockwell, P. J., Hyndman, R. J., & Grunwald, G. K. (1991). Continuous time threshold autoregressive models. *Statistica Sinica*, *1*(2), 401–410.

Cheng, M., Lu, C., & Liu, Y. (2015). Variation in entrainment rate and relationship with cloud microphysical properties on the scale of 5 m. *Science Bulletin*, *60*(7), 707–717. https://doi.org/10.1007/s11434-015-0737-8

Couvreux, F., Hourdin, F., & Rio, C. (2010). Resolved versus parametrized boundary-layer plumes. Part I: A parametrization-oriented conditional sampling in large-eddy simulations. *Boundary-Layer Meteorology*, *134*(3), 441–458. https://doi.org/10.1007/s10546-009-9456-5

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, *2*(4), 303–314. https://doi.org/10.1007/bf02551274

Dawe, J. T., & Austin, P. H. (2011a). The influence of the cloud shell on tracer budget measurements of LES cloud entrainment. *Journal of the Atmospheric Sciences*, *68*(12), 2909–2920. https://doi.org/10.1175/2011jas3658.1

Dawe, J. T., & Austin, P. H. (2011b). Interpolation of LES cloud surfaces for use in direct calculations of entrainment and detrainment. *Monthly Weather Review*, *139*(2), 444–456. https://doi.org/10.1175/2010mwr3473.1

Dawe, J. T., & Austin, P. H. (2012). Statistical analysis of an LES shallow cumulus cloud ensemble using a cloud tracking algorithm. *Atmospheric Chemistry and Physics*, *12*(2), 1101–1119. https://doi.org/10.5194/acp-12-1101-2012

Dawe, J. T., & Austin, P. H. (2013). Direct entrainment and detrainment rate distributions of individual shallow cumulus clouds in an LES. *Atmospheric Chemistry and Physics Discussions*, *13*(2), 5365–5410. https://doi.org/10.5194/acp-13-7795-2013

de Roode, S. R., Siebesma, A. P., Jonker, H. J., & de Voogd, Y. (2012). Parameterization of the vertical velocity equation for shallow cumulus clouds. *Monthly Weather Review*, *140*(8), 2424–2436. https://doi.org/10.1175/mwr-d-11-00277.1

de Rooy, W. C., Bechtold, P., Fröhlich, K., Hohenegger, C., Jonker, H., Mironov, D., et al. (2013). Entrainment and detrainment in cumulus convection: An overview. *Quarterly Journal of the Royal Meteorological Society*, *139*(670), 1–19. https://doi.org/10.1002/qj.1959

de Rooy, W. C., & Siebesma, A. P. (2008). A simple parameterization for detrainment in shallow cumulus. *Monthly Weather Review*, *136*(2), 560–576. https://doi.org/10.1175/2007mwr2201.1

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., et al. (2017). TensorFlow distributions. *arXiv preprint arXiv:1711.10604*.

Gregory, D. (2001). Estimation of entrainment rate in simple models of convective clouds. *Quarterly Journal of the Royal Meteorological Society*, *127*(571), 53–72. https://doi.org/10.1002/qj.49712757104

Hannah, W. M. (2017). Entrainment versus dilution in tropical deep convection. *Journal of the Atmospheric Sciences*, *74*(11), 3725–3747. https://doi.org/10.1175/jas-d-16-0169.1

Heus, T., Pols, F., Jonker, H., van den Akker, H., & Lenschow, D. (2008). *Analysis of the downward transport in RICO observations*. Paper presented at 18th Symposium on Boundary Layers and Turbulence, American Meteorological Society, Stockholm, Sweden.

Heus, T., van Dijk, G., Jonker, H. J., & van den Akker, H. E. (2008). Mixing in shallow cumulus clouds studied by Lagrangian particle tracking. *Journal of the Atmospheric Sciences*, *65*(8), 2581–2597. https://doi.org/10.1175/2008jas2572.1

Holland, J. Z., & Rasmusson, E. M. (1973). Measurements of the atmospheric mass, energy, and momentum budgets over a 500-kilometer square of tropical ocean. *Monthly Weather Review*, *101*(1), 44–55. https://doi.org/10.1175/1520-0493(1973)101<0044:motame>2.3.co;2

Kain, J. S. (2004). The Kain–Fritsch convective parameterization: An update. *Journal of Applied Meteorology*, *43*(1), 170–181. https://doi.org/10.1175/1520-0450(2004)043<0170:tkcpau>2.0.co;2

Kain, J. S., & Fritsch, J. M. (1990). A one-dimensional entraining/detraining plume model and its application in convective parameterization. *Journal of the Atmospheric Sciences*, *47*(23), 2784–2802. https://doi.org/10.1175/1520-0469(1990)047<2784:aodepm>2.0.co;2

Kessler, E. (1969). *On the distribution and continuity of water substance in atmospheric circulations* (pp. 1–84). Springer.

Khairoutdinov, M. F., & Kogan, Y. (2000). A new cloud physics parameterization in a large-eddy simulation model of marine stratocumulus. *Monthly Weather Review*, *128*(1), 229–243. https://doi.org/10.1175/1520-0493(2000)128<0229:ancppi>2.0.co;2

Khairoutdinov, M. F., Krueger, S. K., Moeng, C.-H., Bogenschutz, P. A., & Randall, D. A. (2009). Large-eddy simulation of maritime deep tropical convection. *Journal of Advances in Modeling Earth Systems*, *1*(4), 15. https://doi.org/10.3894/james.2009.1.15

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 972–981).

Klocke, D., Pincus, R., & Quaas, J. (2011). On constraining estimates of climate sensitivity with present-day observations through model weighting. *Journal of Climate*, *24*(23), 6092–6099. https://doi.org/10.1175/2011jcli4193.1

Kogan, Y. (2013). A cumulus cloud microphysics parameterization for cloud-resolving models. *Journal of the Atmospheric Sciences*, *70*(5), 1423–1436. https://doi.org/10.1175/jas-d-12-0183.1

Li, X., Wong, T.-K. L., Chen, R. T., & Duvenaud, D. (2020). Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics* (pp. 3870–3882).

Liu, X., Tsukamot, O., Oikawa, T., & Ohtaki, E. (1998). A study of correlations of scalar quantities in the atmospheric surface layer. *Boundary-Layer Meteorology*, *87*(3), 499–508. https://doi.org/10.1023/a:1000947709324

Lu, C., Liu, Y., Zhang, G. J., Wu, X., Endo, S., Cao, L., et al. (2016). Improving parameterization of entrainment rate for shallow convection with aircraft measurements and large-eddy simulation. *Journal of the Atmospheric Sciences*, *73*(2), 761–773. https://doi.org/10.1175/jas-d-15-0050.1

Lu, C., Sun, C., Liu, Y., Zhang, G. J., Lin, Y., Gao, W., et al. (2018). Observational relationship between entrainment rate and environmental relative humidity and implications for convection parameterization. *Geophysical Research Letters*, *45*(24), 495–504. https://doi.org/10.1029/2018gl080264

Mellado, J. P. (2017). Cloud-top entrainment in stratocumulus clouds. *Annual Review of Fluid Mechanics*, *49*(1), 145–169. https://doi.org/10.1146/annurev-fluid-010816-060231

Murphy, J. M., Sexton, D. M., Barnett, D. N., Jones, G. S., Webb, M. J., Collins, M., & Stainforth, D. A. (2004). Quantification of modelling uncertainties in a large ensemble of climate change simulations. *Nature*, *430*(7001), 768–772. https://doi.org/10.1038/nature02771

Neggers, R. A. J., Siebesma, A. P., & Jonker, H. J. J. (2002). A multiparcel model for shallow cumulus convection. *Journal of the Atmospheric Sciences*, *59*(10), 1655–1668. https://doi.org/10.1175/1520-0469(2002)059<1655:ammfsc>2.0.co;2

O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., et al. (2019). *Keras Tuner*. https://github.com/keras-team/keras-tuner

Ott, J., Pritchard, M., Best, N., Linstead, E., Curcic, M., & Baldi, P. (2020). A Fortran-Keras deep learning bridge for scientific computing. *Scientific Programming, 2020*, 8888811. https://doi.org/10.1155/2020/8888811

Park, S. (2014). A unified convection scheme (UNICON). Part I: Formulation. *Journal of the Atmospheric Sciences*, *71*(11), 3902–3930. https://doi.org/10.1175/jas-d-13-0233.1

Park, S., Shin, J., Kim, S., Oh, E., & Kim, Y. (2019). Global climate simulated by the Seoul National University atmosphere model version 0 with a unified convection scheme (SAM0-UNICON). *Journal of Climate*, *32*(18), 2917–2949. https://doi.org/10.1175/jcli-d-18-0796.1

Rasp, S., Pritchard, M. S., & Gentine, P. (2018). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, *115*(39), 9684–9689. https://doi.org/10.1073/pnas.1810286115

Raymond, D. J., & Blyth, A. M. (1986). A stochastic mixing model for nonprecipitating cumulus clouds. *Journal of the Atmospheric Sciences*, *43*(22), 2708–2718. https://doi.org/10.1175/1520-0469(1986)043<2708:asmmfn>2.0.co;2

Romps, D. M. (2010). A direct measure of entrainment. *Journal of the Atmospheric Sciences*, *67*(6), 1908–1927. https://doi.org/10.1175/2010jas3371.1

Romps, D. M. (2016). The stochastic parcel model: A deterministic parameterization of stochastically entraining convection. *Journal of Advances in Modeling Earth Systems*, *8*(1), 319–344. https://doi.org/10.1002/2015ms000537

Romps, D. M., & Kuang, Z. (2010). Nature versus nurture in shallow convection. *Journal of the Atmospheric Sciences*, *67*(5), 1655–1666. https://doi.org/10.1175/2009jas3307.1

Sakradzija, M., Seifert, A., & Heus, T. (2015). Fluctuations in a quasi-stationary shallow cumulus cloud ensemble. *Nonlinear Processes in Geophysics*, *22*(1), 65–85. https://doi.org/10.5194/npg-22-65-2015

Seifert, A., & Beheng, K. D. (2001). A double-moment parameterization for simulating autoconversion, accretion and selfcollection. *Atmospheric Research*, *59*, 265–281. https://doi.org/10.1016/s0169-8095(01)00126-0

Seifert, A., & Heus, T. (2013). Large-eddy simulation of organized precipitating trade wind cumulus clouds. *Atmospheric Chemistry and Physics*, *13*(11), 5631–5645. https://doi.org/10.5194/acp-13-5631-2013

Seifert, A., Heus, T., Pincus, R., & Stevens, B. (2015). Large-eddy simulation of the transient and near-equilibrium behavior of precipitating shallow convection. *Journal of Advances in Modeling Earth Systems*, *7*(4), 1918–1937. https://doi.org/10.1002/2015ms000489

Shin, J., & Park, S. (2020). A stochastic unified convection scheme (UNICON). Part I: Formulation and single-column simulation for shallow convection. *Journal of the Atmospheric Sciences*, *77*(2), 583–610. https://doi.org/10.1175/jas-d-19-0117.1

Siebesma, A. P. (1998). Shallow cumulus convection. In *Buoyant Convection in Geophysical Flows* (pp. 441–486). Springer.

Siebesma, A. P., Bretherton, C. S., Brown, A., Chlond, A., Cuxart, J., Duynkerke, P. G., et al. (2003). A large eddy simulation intercomparison study of shallow cumulus convection. *Journal of the Atmospheric Sciences*, *60*(10), 1201–1219. https://doi.org/10.1175/1520-0469(2003)60<1201:alesis>2.0.co;2

Stevens, B., Moeng, C.-H., Ackerman, A. S., Bretherton, C. S., Chlond, A., de Roode, S., et al. (2005). Evaluation of large-eddy simulations via observations of nocturnal marine stratocumulus. *Monthly Weather Review*, *133*(6), 1443–1462. https://doi.org/10.1175/mwr2930.1

Stevens, B., Moeng, C.-H., & Sullivan, P. P. (1999). Large-eddy simulations of radiatively driven convection: Sensitivities to the representation of small scales. *Journal of the Atmospheric Sciences*, *56*(23), 3963–3984. https://doi.org/10.1175/1520-0469(1999)056<3963:lesord>2.0.co;2

Stevens, B., & Seifert, A. (2008). Understanding macrophysical outcomes of microphysical choices in simulations of shallow cumulus convection. *Journal of the Meteorological Society of Japan*, *86*, 143–162. https://doi.org/10.2151/jmsj.86a.143

Stirling, A., & Stratton, R. (2012). Entrainment processes in the diurnal cycle of deep convection over land. *Quarterly Journal of the Royal Meteorological Society*, *138*(666), 1135–1149. https://doi.org/10.1002/qj.1868

Stramer, O., Brockwell, P., & Tweedie, R. (1996). Continuous-time threshold AR (1) processes. *Advances in Applied Probability*, *28*(3), 728–746. https://doi.org/10.2307/1428178

Stull, R. B. (1985). A fair-weather cumulus cloud classification scheme for mixed-layer studies. *Journal of Applied Meteorology and Climatology*, *24*(1), 49–56. https://doi.org/10.1175/1520-0450(1985)024<0049:afwccc>2.0.co;2

Suselj, K., Kurowski, M. J., & Teixeira, J. (2019). A unified eddy-diffusivity/mass-flux approach for modeling atmospheric convection. *Journal of the Atmospheric Sciences*, *76*(8), 2505–2537. https://doi.org/10.1175/jas-d-18-0239.1

Turner, J. (1963). The motion of buoyant elements in turbulent surroundings. *Journal of Fluid Mechanics*, *16*(1), 1–16. https://doi.org/10.1017/s0022112063000549

Tzen, B., & Raginsky, M. (2019). Neural stochastic differential equations: Deep latent Gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*.

vanZanten, M. C., Stevens, B., Nuijens, L., Siebesma, A. P., Ackerman, A. S., Burnet, F., et al. (2011). Controls on precipitation and cloudiness in simulations of trade-wind cumulus as observed during RICO. *Journal of Advances in Modeling Earth Systems*, *3*(2), M06001. https://doi.org/10.1029/2011ms000056

von Salzen, K., & McFarlane, N. A. (2002). Parameterization of the bulk effects of lateral and cloud-top entrainment in transient shallow cumulus clouds. *Journal of the Atmospheric Sciences*, *59*(8), 1405–1430. https://doi.org/10.1175/1520-0469(2002)059<1405:potbeo>2.0.co;2

Wang, X., & Zhang, M. (2014). Vertical velocity in shallow convection for different plume types. *Journal of Advances in Modeling Earth Systems*, *6*(2), 478–489. https://doi.org/10.1002/2014ms000318

Wang, Z. (2020). A method for a direct measure of entrainment and detrainment. *Monthly Weather Review*, *148*(8), 3329–3340. https://doi.org/10.1175/mwr-d-20-0046.1

Yeo, K., & Romps, D. M. (2013). Measurement of convective entrainment using Lagrangian particles. *Journal of the Atmospheric Sciences*, *70*(1), 266–277. https://doi.org/10.1175/jas-d-12-0144.1

Yuval, J., & O'Gorman, P. A. (2020). Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nature Communications*, *11*(1), 1–10. https://doi.org/10.1038/s41467-020-17142-3

Zhao, M., Golaz, J.-C., Held, I. M., Guo, H., Balaji, V., Benson, R., et al. (2018). The GFDL global atmosphere and land model AM4.0/LM4.0: 2. Model description, sensitivity studies, and tuning strategies. *Journal of Advances in Modeling Earth Systems*, *10*(3), 735–769. https://doi.org/10.1002/2017ms001209

Zhu, L., Lu, C., Yan, S., Liu, Y., Zhang, G. J., Mei, F., et al. (2021). A new approach for simultaneous estimation of entrainment and detrainment rates in non-precipitating shallow cumulus. *Geophysical Research Letters*, *48*(15), e2021GL093817. https://doi.org/10.1029/2021gl093817

Reference From the Supporting Information

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, *18*(1), 6765–6816.