Wavelet-spectral analysis and large-eddy simulation using neural networks of droplet-laden decaying isotropic turbulence

Andreas Freund

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Antonino Ferrante, Chair

Christopher S. Bretherton

James J. Riley

Program Authorized to Offer Degree:
Applied Mathematics

University of Washington

Abstract

Wavelet-spectral analysis and large-eddy simulation
using neural networks
of droplet-laden decaying isotropic turbulence

Andreas Freund

Chair of the Supervisory Committee:
Associate Professor Antonino Ferrante
Aeronautics & Astronautics

In this work, we propose new methods for both the analysis and simulation of droplet-laden isotropic turbulence. First, we suggest using the wavelet energy spectrum to study multiphase turbulent flows to overcome the challenges of applying the Fourier energy spectrum to velocity fields with sharp velocity gradients. Also, we propose a new decomposition of the wavelet energy spectrum into three contributions corresponding to the carrier phase, droplets, and interaction between the two. We apply these new wavelet-decomposition tools in analyzing the direct numerical simulation (DNS) data of droplet-laden decaying isotropic turbulence of Dodd & Ferrante (2016, *J. Fluid Mech.* 806:356–412). Our results show that, in comparison to the spectrum of the single-phase case, the droplets (i) do not affect the carrier-phase energy spectrum at high wavenumbers ($k_m/k_{\min} \geq 128$), (ii) increase the energy spectrum at high wavenumbers ($k_m/k_{\min} \geq 256$) by increasing the interaction energy spectrum at these wavenumbers, and (iii) decrease the energy at low wavenumbers ($k_m/k_{\min} \leq 16$) by increasing the dissipation rate at these wavenumbers.

Then, we propose a model for large-eddy simulation (LES) of decaying isotropic turbulence laden with droplets with diameter of Taylor length-scale. The main challenge in creating LES models for such flow is that the presence of the droplets introduces additional subgrid-scale (SGS) closure terms to the filtered governing equations of motion of the flow.

By processing available DNS data of Dodd & Ferrante, we analyze these terms *a priori* to show that they are all significant to warrant modeling. Then, we propose a new modeling approach that we call mixed-ANN (MANN) LES because it is a mixed LES model that uses the standard Smagorinsky SGS stress model in the carrier fluid, and artificial neural networks (ANNs) to predict the SGS closure terms at the interface. Such an approach is justified because the SGS energy in the carrier flow away from the droplet interface is practically unaffected by the droplets, as we have previously shown with our wavelet analysis of the DNS data. Furthermore, we have performed the first *a posteriori* analysis of such flow for droplets of different Weber numbers, and show that our LES method closely reproduces the temporal decay of the filtered-velocity turbulence kinetic energy as well its p.d.f. of the filtered DNS, show that the modeling of the SGS terms at the interface is necessary for reproducing the results of the filtered DNS, and provide both physical- and spectral-space analysis of the LES results. Finally, the MANN LES approach could be applied to a variety of multiphase turbulent flows due to its ease of implementation, adaptability and performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Abbreviations

ANN: artificial neural network

CNN: convolutional neural network

DNS: direct numerical simulation

DWT: discrete wavelet transform

LES: large-eddy simulation

MANN: mixed artificial neural network

p.d.f.: probability density function

PLIC: piecewise-linear interface calculation

ReLU: rectified linear unit

rms: root-mean square

SGS: subgrid scale

TKE: turbulence kinetic energy

VoF: volume of fluid

Lowercase Roman

$(f_\sigma)_i$: force per unit volume due to surface tension in the i-th direction

k^F: filtered-velocity turbulence kinetic energy

k_m: wavelet wavenumber at scale m

m: wavelet scale

p: pressure

s: training-label scaling function

t: time

u_i: velocity in the i-th direction

$\bar{u}_i$: filtered velocity in the i-th direction

w_i: wavelet coefficients of velocity in the i-th direction

$\bar{w}_i$: wavelet coefficients of filtered velocity in the i-th direction

x_i: spatial coordinate in the i-th direction

Uppercase Roman

C_m: wavelet-spectrum scaling constant

C_S: Smagorinsky constant

D: droplet diameter

$\mathcal{D}$: domain

$\mathcal{D}^{(m)}$: set of indices on which the scale-m wavelet coefficients are defined

$\mathcal{D}_C^{(m)}$: carrier part of the decomposition of $\mathcal{D}^{(m)}$

$\mathcal{D}_D^{(m)}$: droplet part of the decomposition of $\mathcal{D}^{(m)}$

$\mathcal{D}_I^{(m)}$: interaction part of the decomposition of $\mathcal{D}^{(m)}$

E: Fourier energy spectrum

$\widetilde{E}$: wavelet energy spectrum

L: nondimensional domain length

N: number of grid points per dimension

P_i: sum of direction-i momentum-equation closure terms

$\widetilde{P}$: wavelet-spectral subgrid-scale closure-term sink/source rate of energy

Re: Reynolds number

S_{ij}: rate-of-strain tensor

$\bar{S}_{ij}$: filtered-velocity rate-of-strain tensor

$\widetilde{S}$: wavelet-spectral surface-tension source/sink rate of energy

$\widetilde{S}^F$: wavelet-spectral filtered surface-tension source/sink rate of energy

$\widetilde{T}$: wavelet-spectral energy-transfer rate due to advection and pressure

$\widetilde{T}^F$: wavelet-spectral filtered-velocity energy-transfer rate due to advection and pressure

$\widetilde{V}$: wavelet-spectral viscous dissipation rate

$\widetilde{V}^F$: wavelet-spectral filtered-velocity viscous dissipation rate

We: Weber number

Lowercase Greek

δ: Dirac delta function

$\bar{\varepsilon}$: viscous dissipation rate averaged over distance from interface

ε^F: filtered-velocity viscous dissipation rate

η: Kolmogorov length scale

κ: curvature of droplet surface

λ: Taylor length scale

μ: dynamic viscosity

ρ: density

σ: surface tension

τ: subgrid-scale closure term

ϕ_m: mass fraction

ϕ_v: volume fraction

χ: volume-of-fluid function

Uppercase Greek

Δ: filter width

Δt: time step

Δx: grid spacing

Π: sink of power due to subgrid-scale closure terms

Ψ^F: power due to filtered surface tension

ACKNOWLEDGMENTS

Thank you to my advisor, Professor Antonino Ferrante, for his mentorship and guidance over the last five years. His dedication to producing insightful and impactful results has pushed me to reach my full potential as a researcher. Thanks also to the rest of my committee, Professors Chris Bretherton, Jim Riley, and Alberto Aliseda, for their helpful feedback throughout my PhD, from their courses and my general exam to the editing of this thesis and my defense.

Thank you to my parents for always being supportive of my academic endeavours and believing in me when I didn't. And thank you to my brother, Alex. You always know how to make me laugh, and moving away from you was one of the hardest parts of grad school.

Thank you to the best friends I made at UW. Brian, you might be the most driven person I know, but you've always been there to help me balance work with life. Jeremy, you know how to push me out of my comfort zone, and I hope to continue our travels in the near future. Kathleen, you were probably the first friend I made in grad school, and I don't think I would've ever learned to sail without you as my crewmate. Ellie, I always look forward to our discussions of popular culture. And Emily, I forgive you for moving to New York.

Thank you to the Department of Applied Mathematics, including the students, staff, and faculty. I can't imagine a friendlier department. And thank you to my fellow students of the Computational Fluid Mechanics group in the Department of Aeronautics & Astronautics, particularly Pablo, Abhiram, and Michael. I always appreciated being able to commiserate with you on disappointing simulation results.

Finally, thank you to all of the friends I've made in Seattle. You make me incredibly grateful to have had the opportunity to come to UW for grad school.

This work was partially supported by a National Science Foundation CAREER Award, grant number ACI-1054591, received by Professor Ferrante, and facilitated through the use of advanced computational, storage, and networking infrastructure provided by the Hyak supercomputer system at the University of Washington, Seattle, as well as the use of the Extreme Science and Engineering Discovery Environment (XSEDE) [56] under XRAC grant number TG-CTS100024. XSEDE is supported by National Science Foundation grant number ACI-1053575.

DEDICATION

to my parents, Susanne and Roland

Chapter 1
INTRODUCTION

1.1 Background

Multiphase flow refers to a flow comprising two or more phases. These phases could include different states of matter (gas, liquid, or solid) or fluids with different properties. In dispersed multiphase flow, the carrier phase is connected and laden with the dispersed phase, which is disconnected. Examples of dispersed multiphase flow include solid particles in a gas, gaseous bubbles in a liquid, or liquid droplets in a different liquid. As with single-phase flows, dispersed multiphase flows can become turbulent at high speeds and exhibit interesting physical phenomena.

Dispersed multiphase turbulence, whether particle-, bubble-, or droplet-laden, has applications throughout nature and engineering. Particularly relevant to the current COVID-19 pandemic is the transmission of respiratory droplets carrying the SARS-CoV-2 virus [41], but other natural applications include the formation of rain droplets in clouds [51] and air entrainment and spray production by breaking waves [10]. Common engineering applications are spray combustion [53] and bubble cavitation on propellers [26]. See [3] for a review of turbulent dispersed multiphase flow, including experimental and computational approaches for their study.

In this work, we are primarily interested in the numerical simulation of droplet-laden turbulence. In particular, the droplets are finite-sized, so they cannot be represented by infinitesimally small point particles. Furthermore, the turbulence will be decaying, isotropic, and homogeneous. Decaying means that no energy is added to the flow after initialization, and isotropic and homogeneous mean that the flow is statistically invariant to rotation and translation, respectively, so the flow will have periodic boundary conditions. Isotropic

homogeneous turbulence is therefore an idealized situation that lets us focus on analyzing the interaction of the droplets with turbulence without worrying about the effects of the boundary conditions.

There are two main paradigms for simulating droplet-laden turbulence: direct numerical simulation (DNS) and large-eddy simulation (LES). DNS aims to numerically solve the governing equations with fine enough spatial and temporal resolution to resolve all scales of turbulence, all the way down to the Kolmogorov length scale, which is the scale at which viscous dissipation dominates. An overview of DNS of droplet-laden isotropic turbulence is given in Section 1.2. Because DNS fully resolves all scales of turbulence, it requires very large computational grids and very small time steps, making it a prohibitively expensive approach to simulating turbulence, and this cost only increases with the additional overhead of simulating the interaction of two fluids. On the other hand, LES only aims to resolve the larger scales of turbulence while modeling the effects of the unresolved scales. Therefore, LES is able to use a smaller computational grid and larger time steps, providing great computational savings. LES for droplet-laden isotropic turbulence is the primary topic of Chapter 3 and is described in more detail therein.

1.2 Direct numerical simulation of droplet-laden isotropic turbulence

In this section, we provide the governing equations for incompressible two-fluid flow (Section 1.2.1) as well as some details regarding the DNS of droplet-laden isotropic turbulence (Section 1.2.2). Lastly, we provide a more detailed description of the flow we will investigate in this work (Section 1.2.3).

In this work, we use the same numerical methods for the DNS of droplet-laden turbulence as Dodd & Ferrante [12]. In Chapter 2, we use their actual data, and in Chapter 3, we use their methods to perform the DNS ourselves. In both chapters, we use the same flow parameters as them. See also [14] for a broader review of DNS for bubble- and droplet-laden turbulent flows.

1.2.1 Governing equations for incompressible two-fluid flow

The governing equations for droplet-laden isotropic turbulence are the two-phase incompressible Navier–Stokes equations [12]:

$$\frac{\partial u_j}{\partial x_j} = 0, \tag{1.1a}$$

$$\rho\left(\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j}\right) = -\frac{\partial p}{\partial x_i} + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x_j}\left(\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right) + \frac{1}{\mathrm{We}}(f_\sigma)_i, \tag{1.1b}$$

where $i, j = 1, 2, 3$ and in which we are using Einstein's summation notation. Herein, $\boldsymbol{x}$ and t are (three-dimensional) space and time, $\boldsymbol{u}(\boldsymbol{x}, t)$ and $p(\boldsymbol{x}, t)$ are velocity and pressure, Re and We are the Reynolds and Weber numbers, $\rho(\boldsymbol{x}, t)$ and $\mu(\boldsymbol{x}, t)$ are the density and viscosity, and $\boldsymbol{f}_\sigma(\boldsymbol{x}, t)$ is the force per unit volume due to surface tension at the interface:

$$\boldsymbol{f}_\sigma(\boldsymbol{x}, t) = \kappa(\boldsymbol{x}, t)\boldsymbol{n}(\boldsymbol{x}, t)\delta(s), \tag{1.2}$$

where κ is the interface curvature, $\boldsymbol{n}$ is the unit normal to the interface (directed into the droplet interior), and $\delta(s)$ is the Dirac delta function with s a coordinate normal to the interface such that $s = 0$ at the interface, ensuring that the surface-tension force $\boldsymbol{f}_\sigma$ is only applied at the interface.

The Reynolds and Weber numbers are defined as

$$\mathrm{Re} = \frac{\tilde{\rho}_c\widetilde{U}\tilde{L}}{\tilde{\mu}_c}, \qquad \mathrm{We} = \frac{\tilde{\rho}_c\widetilde{U}^2\tilde{L}}{\tilde{\sigma}}, \tag{1.3}$$

and they measure the ratios of inertial to viscous forces and inertial to surface-tension forces, respectively. Here, the notation $\tilde{\cdot}$ refers to a reference dimensional value, so $\tilde{\rho}_c$, $\tilde{\mu}_c$, $\widetilde{U}$, $\tilde{L}$, and $\tilde{\sigma}$ are the reference carrier density, carrier viscosity, velocity, length scale, and surface tension. These reference values were used to nondimensionalize the governing equations (1.1). The density and viscosity were also nondimensionalized by $\tilde{\rho}_c$ and $\tilde{\mu}_c$, respectively, so $\rho_c = \mu_c = 1$. In the remainder of this work, we will only refer to nondimensionalized values.

1.2.2 Numerical method

To track the droplets, Dodd & Ferrante [12] use the volume-of-fluid (VoF) method [4], which comprises several components. The VoF interface is reconstructed using a piecewise-linear interface calculation (PLIC) [60]. This reconstruction depends on the computation of the interface normal, which is done using a combination of the centered-columns [50] and the Youngs [60] methods, called the mixed Youngs–centered method [2]. After the normal is computed, the interface is located using the analytical tools of [36]. The VoF function, $\chi(\boldsymbol{x}, t)$, is defined such that $\chi = 0$ in the carrier fluid, $\chi = 1$ in the droplets, and $0 < \chi < 1$ for cells containing the interface with a value of χ equal to the fraction of grid cell occupied by the droplet fluid. The VoF equation is an advection equation for χ:

$$\frac{\partial \chi}{\partial t} + u_j \frac{\partial \chi}{\partial x_j} = 0. \tag{1.4}$$

Our VoF method [4] uses a modified version of the Eulerian implicit–Eulerian algebraic–Lagrangian explicit algorithm of [49] to advect χ. Because κ and $\boldsymbol{n}$ depend on χ, so does the force of surface tension $\boldsymbol{f}_\sigma$, so we will sometimes use the notation $\boldsymbol{f}_\sigma(\chi)$. Density and viscosity are also related to χ by

$$\rho(\boldsymbol{x}, t) = \rho_d \chi(\boldsymbol{x}, t) + \rho_c(1 - \chi(\boldsymbol{x}, t)), \tag{1.5a}$$

$$\mu(\boldsymbol{x}, t) = \mu_d \chi(\boldsymbol{x}, t) + \mu_c(1 - \chi(\boldsymbol{x}, t)), \tag{1.5b}$$

where ρ_c and μ_c are the carrier-fluid density and viscosity and ρ_d and μ_d are the droplet density and viscosity.

The two-fluid incompressible Navier–Stokes equations (1.1) are solved using the FastP* pressure-correction method [11]. Pressure-correction methods solve the incompressible Navier–Stokes equations by computing an approximate velocity at each time step while ignoring the pressure-gradient term. Then a Poisson equation is solved to find the pressure, and the approximate velocity is updated to enforce the divergence-free condition (1.1a). With two-fluid flows, the Poisson equation will no longer have constant coefficients, which means that a more expensive iterative solver such as multigrid must be used to find the pressure rather

case	$\mathrm{We}_{\mathrm{rms}}$	ρ_d/ρ_c	μ_d/μ_c	ϕ_m	ϕ_v
A				0	0
B	0.1	10	10	0.5	0.05
C	1	10	10	0.5	0.05
D	5	10	10	0.5	0.05
E	1	1	10	0.005	0.05
F	1	100	10	5	0.05
G	1	10	1	0.5	0.05
H	1	10	100	0.5	0.05

Table 1.1: Cases investigated by Dodd & Ferrante [12].

than a fast Poisson solver that leverages the fast Fourier transform. However, FastP* avoids this by reformulating the Poisson equation in such a way that it has constant coefficients. We delay a full explanation of FastP* until Section 3.3.4.

1.2.3 Flow description

We use the same flow parameters as Dodd & Ferrante [12, §3.1.2]. An initial field of decaying isotropic homogeneous turbulence without gravity is randomly generated with a prescribed energy spectrum. This flow is simulated free of droplets until nondimensional time $t = 1$, at which point the Reynolds number with respect to the Taylor length-scale λ is $\mathrm{Re}_\lambda = 83$. Then, at $t = 1$, 3130 nonevaporating spherical droplets are randomly seeded with zero velocity and a constraint on the minimum distance between droplets to delay coalescence. The diameter of the droplets at injection is equal to 32 grid cells, which is 1.1λ or 20η, where η is the Kolmogorov length-scale at $t = 1$. The number of droplets is chosen such that they occupy 5% of the domain. The computational grid has nondimensional length 1 and 1024 points in each direction, so $\Delta x_{\mathrm{DNS}} = 1/1024$, and periodic boundary conditions are applied. We set the time step to be $\Delta t_{\mathrm{DNS}} = 0.1\,\Delta x_{\mathrm{DNS}} = 1/10240$, and we let the DNS run until $t = 5$.

Case A is the reference droplet-free case. In the droplet-laden cases (B–H), Dodd & Ferrante varied one of the following three parameters: the droplet Weber number based on the root mean square velocity of turbulence ($0.1 \leq \mathrm{We}_{\mathrm{rms}} \leq 5$) in cases B, C, and D; the droplet- to carrier-fluid density ratio ($1 \leq \rho_d/\rho_c \leq 100$) in cases E, C, and F; or the droplet- to carrier-fluid viscosity ratio ($1 \leq \mu_d/\mu_c \leq 100$) in cases G, C, and H. In Section 2.3, we present the wavelet-spectral analysis of all cases A–H at nondimensional time $t = 3.5$. Table 1.1 shows the eight different cases studied, and ϕ_m and ϕ_v are the mass and volume fractions of the droplets, respectively.

1.3 Objectives

This work began with the objective of obtaining a better understanding of the effect of droplets on the spectral energy content of turbulence. To do so, we decided to employ the wavelet transform and defined a new decomposition for the energy spectra of multiphase flows, which led us to some interesting results. This investigation is detailed in Chapter 2. Some of our findings from using wavelet-spectral analysis then helped inform an LES model for droplet-laden isotropic turbulence. Chapter 3 covers both the *a priori* and *a posteriori* analyses using this new LES method. Ideas for further research are shared in Chapter 4.

Chapter 2
WAVELET-SPECTRAL ANALYSIS

In this chapter, we describe our wavelet-spectral analysis of droplet-laden isotropic turbulence. This work was published in [17].

2.1 Introduction

2.1.1 Background

Spectral analysis has been an important tool in understanding the physics of dispersed multiphase flows [14]. For example, Elghobashi & Truesdell [15] simulated decaying homogeneous isotropic turbulence interacting with dispersed solid particles of diameter less than the Kolmogorov length scale. They found that the particles caused an increase in spectral turbulence kinetic energy (TKE) at high wavenumbers. Ferrante & Elghobashi [16] extended this analysis to find that the energy spectrum is affected by a spectral two-way coupling function due to fluid–particle drag interaction, which results in an increase of energy at high wavenumbers for a range of sub-Kolmogorov particles with Stokes number between 0.1 and 5. The Stokes number is the ratio of the particle relaxation time to the Kolmogorov time scale.

Because the particles of these simulations were smaller than the Kolmogorov length scale of turbulence and represented by infinitesimal points rather than finite volumes, the authors computed the energy spectra in the traditional way, that is, using the Fourier transform across the entire domain. Spectral analysis becomes more difficult for particles of Taylor length-scale size as studied by Lucci *et al.* [37]. These larger particles have a finite volume in the computational domain, so the velocity field becomes discontinuous at the particle–fluid interfaces, and these discontinuities manifest themselves as oscillations in the energy spectra [37, Fig. 35]. These oscillations are a consequence of how the energy spectrum $E(\kappa)$

is defined:

$$E(\kappa) = \sum_{\kappa \leq |\boldsymbol{\kappa}| < \kappa+1} \tfrac{1}{2} \hat{u}_j^*(\boldsymbol{\kappa}) \hat{u}_j(\boldsymbol{\kappa}), \tag{2.1}$$

where κ is the wavenumber, $\hat{u}_i(\boldsymbol{\kappa})$ is the Fourier transform of the velocity u_i, and $(\,\cdot\,)^*$ denotes complex conjugation. The energy spectrum is thus defined using the Fourier transform, which uses infinite sinusoidal waves to represent functions in the wavenumber domain. Such basis functions cannot easily capture discontinuities. Lucci *et al.* [37] therefore use an alternative definition of the energy spectrum. Instead of computing the spectrum using the Eulerian velocity field, they track a large number of fluid points and use their velocity to compute a Lagrangian TKE spectrum. While this approach is definitely valuable, it adds the computational costs of computing, writing, and storing the fluid points' velocity during the simulations.

A related area of research is that of experimental "pseudoturbulence" generated by rising bubbles [47]. Hot-wire anemometers also produce discontinuous data due to the popping of bubbles by the probe. A variety of methods has been used to overcome the artifacts introduced by the discontinuities when producing energy spectra such as interpolation, selective sampling in the Fourier transform, and fitting of Gaussian curves. However, these techniques were derived for the one-dimensional velocity time series produced by the anemometer, so they would not be easily applicable to our three-dimensional velocity fields. Most importantly, these techniques involve modifications of the velocity field, while we propose a way to generate TKE spectra without artificially modifying the velocity field at all.

2.1.2 Motivation

We are interested in the analysis of energy spectra for homogeneous isotropic turbulence laden with droplets of Taylor length-scale size using the flow parameters set in Section 1.2.3. These finite-sized droplets present similar issues as the particles of [37]. Figure 2.1 shows the Fourier energy spectra of droplet-laden turbulence studied via DNS by Dodd & Ferrante [12]. In that figure, note the general differences between the droplet-free and droplet-laden cases:

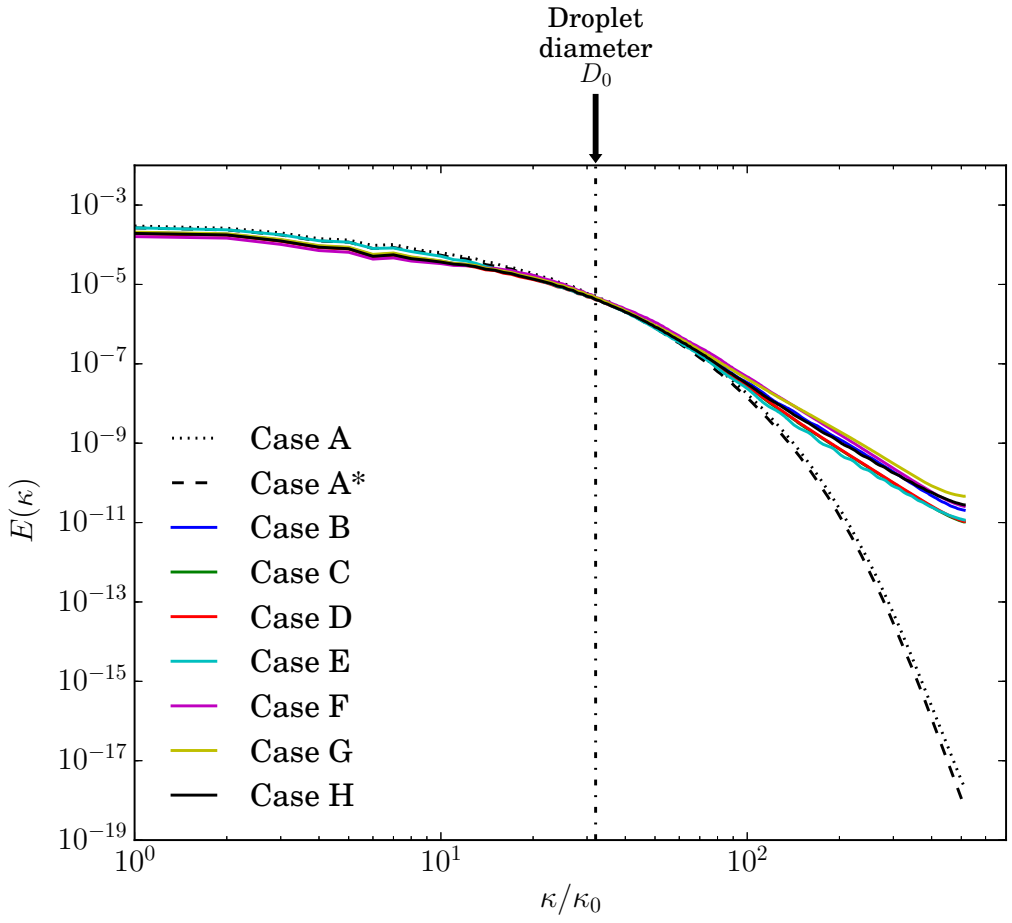

Figure 2.1: Fourier spectra of TKE for the DNS cases of Dodd & Ferrante [12]. Dashed lines represent droplet-free cases and solid lines droplet-laden cases.

introduction of the droplets causes energy at low wavenumbers to decrease, energy at high wavenumbers to increase, and the spectra to oscillate at high wavenumbers. The last feature is a result of the Fourier transform's inability to handle sharp changes at droplet interfaces.

Figure 2.2 shows an example of how the droplets exhibit sharp changes in velocity around their interfaces with the carrier fluid. Though we are dealing with deformable droplets and not particles, even if the shear is continuous at the interface, the velocity gradient at the interface is still discontinuous [12, Fig. 30]; thus, high-frequency modes are enhanced when taking the Fourier transform. This has led us to question whether the introduction of droplets actually causes the carrier fluid to contain more energy at high wavenumbers. Also, Maxey [38] suggests providing a closer analysis of the energy spectrum in his review of [12]. Furthermore, our interest in the droplets' effect on the energy spectrum is to help improve subgrid-scale models for large-eddy simulation (LES) of multiphase turbulent flows, which will be the focus of Chapter 3.

2.1.3 Our approach

Instead of using the Fourier transform to compute the energy spectrum, we use the wavelet-transform approach as described by Meneveau [40]. We extend his wavelet formulation to multiphase incompressible turbulence. A wavelet is typically an oscillating function with compact support. The wavelet transform is used to represent a function in a basis of wavelets. The primary advantage of the wavelet transform for this application is that it preserves spatial information, making it a natural choice for a two-fluid flow. Besides restricting effects of nonsmoothness locally, the spatial information allows us to decompose the spectrum as explained in Section 2.2.5.

As the computation of nonhomogeneous and multiphase turbulent flows becomes increasingly tractable, wavelet and wavelet-like approaches have been applied in a few recent studies. For example, wavelets have been used to analyze particle-laden flows [5] and turbulent combustion [29]. A similar approach was used to compute spectra on nonperiodic domains by using spatial filtering with a customized nonwavelet basis [48].

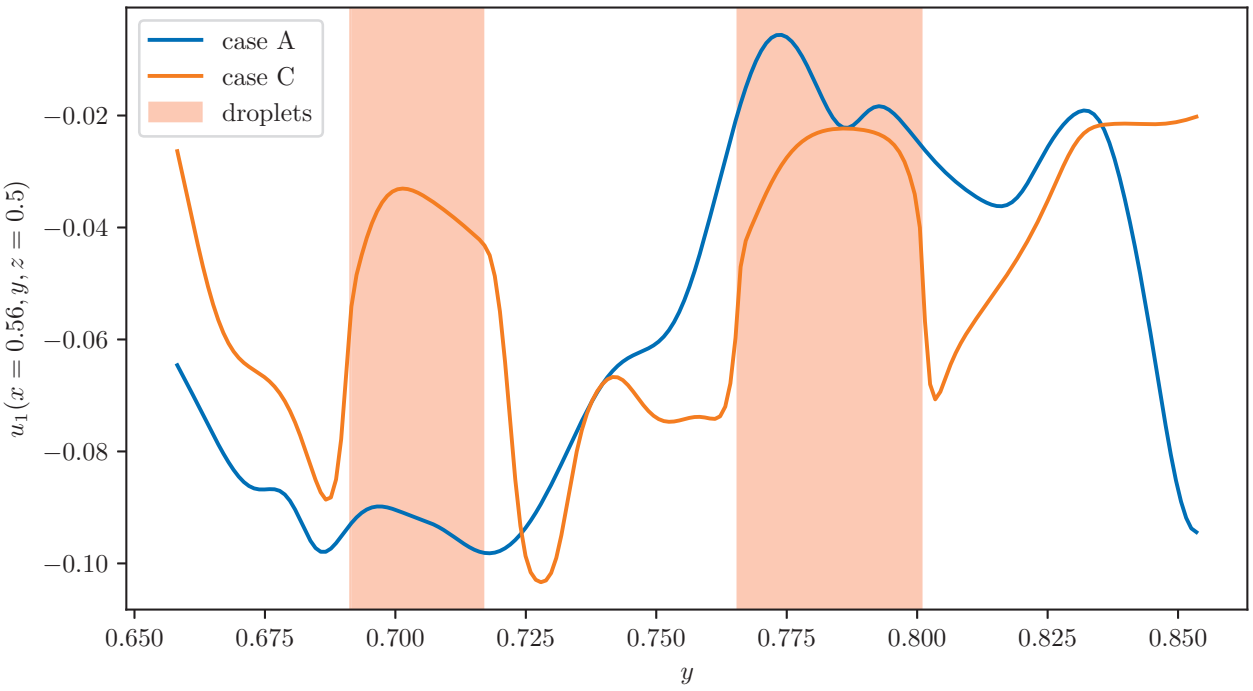

Figure 2.2: Velocity profiles along a line in the domain show sharp jumps at the interfaces for droplet-laden case C compared to droplet-free case A.

2.2 Mathematical description

2.2.1 Discrete wavelet transform

The basic idea behind the discrete wavelet transform (DWT) is to represent a function using self-similar basis functions, which are dilated and translated in the space of the data set. These basis functions are made orthonormal by using a logarithmic spacing of scales. In the following, we use a notation similar to that of Meneveau [40].

If $g[n]$ is our "mother" wavelet, where g is a discrete function sampled at $x_n = n\,\Delta x$ with an integer n and the grid spacing Δx, the basis functions take the form

$$g^{(m)}[n - 2^m l] = \frac{1}{2^{m/2}} g\left[\frac{n - 2^m l}{2^m}\right], \tag{2.2}$$

where m is the wavelet-scale index, which determines the amount of dilation, and l determines the translation away from the initial position. Associated with the wavelet functions is also a smoothing function $h[n]$ that is dilated and translated in the same way. In three dimensions, we must use products of $g^{(m)}$ and $h^{(m)}$ in order to represent our data with dilations in every possible combination of directions. In total, there are seven possible combinations: one volumetric dilation, i.e., one in all three directions at once; three planar dilations, i.e., one for each plane (x-y, y-z, and x-z); and three linear dilations, i.e., one in each direction (x, y, and z). Using the index q to distinguish between these seven combinations, we thus define a three-dimensional basis function $g^{(m,q)}[\boldsymbol{n} - 2^m \boldsymbol{l}]$ [40, (30)], where $\boldsymbol{n}$ and $\boldsymbol{l}$ are the three-dimensional counterparts to the indices n and l.

With our basis functions defined, we may represent each component of velocity as

$$u_i[\boldsymbol{n}] = \sum_{m=1}^{M} \sum_{q=1}^{7} \sum_{\boldsymbol{l} \in \mathcal{D}^{(m)}} w_i^{(m,q)}[\boldsymbol{l}]\, g^{(m,q)}[\boldsymbol{n} - 2^m \boldsymbol{l}], \quad i = 1, 2, 3, \tag{2.3}$$

where the wavelet coefficients are given by

$$w_i^{(m,q)}[\boldsymbol{l}] = \sum_{\boldsymbol{n} \in \mathcal{D}} u_i[\boldsymbol{n}]\, g^{(m,q)}[\boldsymbol{n} - 2^m \boldsymbol{l}]. \tag{2.4}$$

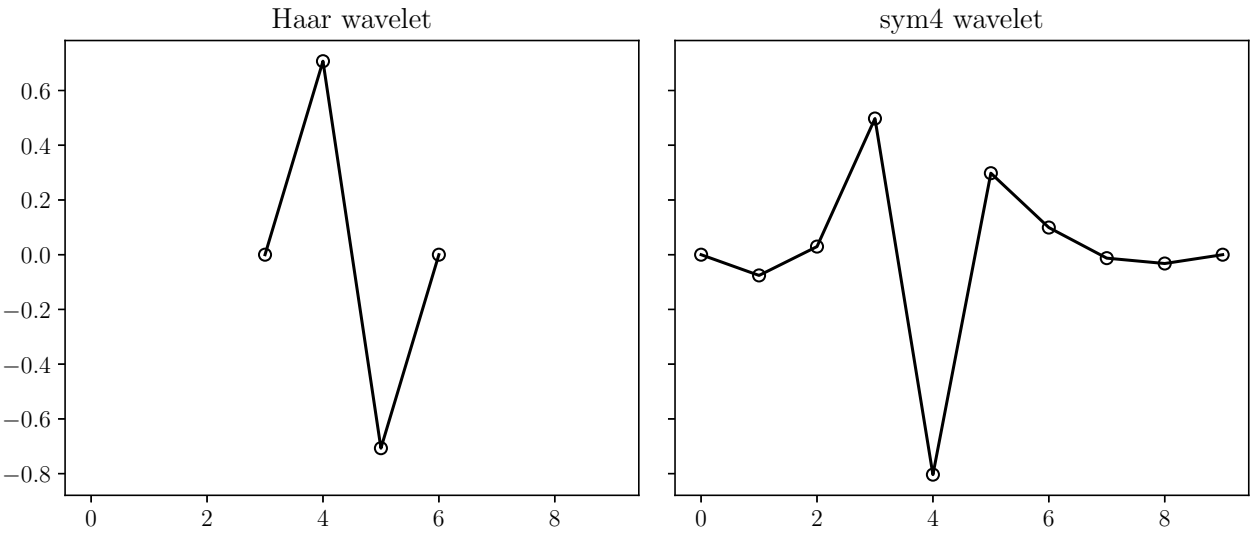

Figure 2.3: Comparison of the Haar wavelet and the sym4 wavelet used in this study.

Here $\mathcal{D} = \{1, \ldots, N\}^3$ is the set of indices on which our velocity field is defined (in our case $N = 1024$) and $\mathcal{D}^{(m)} = \{1, \ldots, N/2^m\}^3$ is the set of indices on which the scale-m wavelet coefficients are defined. $M = \log_2 N$ is the number of scales that we can use (so in our case $M = 10$). The DWT conserves energy because

$$\sum_{i=1}^{3} \sum_{\boldsymbol{n} \in \mathcal{D}} u_i[\boldsymbol{n}]^2 = \sum_{i=1}^{3} \sum_{m=1}^{M} \sum_{q=1}^{7} \sum_{\boldsymbol{l} \in \mathcal{D}^{(m)}} w_i^{(m,q)}[\boldsymbol{l}]^2. \tag{2.5}$$

2.2.2 Wavelet spectrum

The wavenumber k_m is related to the wavelet scale m by

$$k_m = \frac{2\pi}{2^m \, \Delta x}, \tag{2.6}$$

where $\Delta x = L/N$ is the size of the spatial discretization of our DNS data with L the nondimensional side length of the cubical domain and N the number of grid points per domain (here $\Delta x = 1/1024$). The largest scale corresponds to $m = M$, so using (2.6), the minimum wavelet wavenumber is $k_{\min} = k_M = 2\pi$.

The wavelet spectrum is then given by

$$\widetilde{E}(k_m) = C_m \sum_{i=1}^{3} \left\langle \frac{1}{2} \sum_{q=1}^{7} w_i^{(m,q)}[\boldsymbol{l}]^2 \right\rangle_{\boldsymbol{l}\in\mathcal{D}^{(m)}}, \tag{2.7}$$

where $\langle \cdot \rangle_{\boldsymbol{l}\in\mathcal{D}^{(m)}}$ denotes the average over $\boldsymbol{l} \in \mathcal{D}^{(m)}$, and

$$C_m = \frac{\Delta x}{2\pi(\log 2)2^{2m}} \tag{2.8}$$

is included to scale the spectrum so that $\widetilde{E}(k_m)$ represents the energy spectrum per unit wavenumber.

2.2.3 Choice of wavelet

A comparison of energy spectra produced with various wavelets is given by Perrier *et al.* [42]. The choice of wavelet is also discussed by Meneveau [40, §5.3]. We have used both the Haar and sym4 wavelets to analyze our data. Both are common options and their definitions can be found in the PyWavelets wavelet package for Python [34]. The Haar wavelet has the advantage of being simple and easy to compute. Because it requires fewer points to calculate than other wavelet basis functions (Fig. 2.3), it also allows for more spatial resolution, which would be desirable when we decompose the spectrum. However, since it uses fewer points than other functions, a single basis function encodes less information. As a result, more small-scale wavelet coefficients are needed to represent the flow field in the wavelet basis, so the Haar wavelet produces relatively flat spectra. On the other hand, the sym4 wavelet produces wavelet spectra more closely resembling the respective Fourier spectra (in that they decay at the high wavenumbers) because its basis function oscillates more in space than the Haar wavelet, more closely resembling a truncated sine function of the Fourier transform. This property is important in analyzing the wavelet spectra in comparison to the Fourier spectra, and so sym4 is the wavelet we adopted in this study.

2.2.4 Evolution of wavelet spectrum

The DWT of the momentum equation (1.1b) (after rewriting so that $\partial u_i/\partial t$ is on the left-hand side) gives

$$\frac{\partial}{\partial t} w_i^{(m,q)}[\boldsymbol{l}] = -\left\{ u_j \frac{\partial u_i}{\partial x_j} + \frac{1}{\rho}\frac{\partial p}{\partial x_i} \right\}^{(m,q)}[\boldsymbol{l}] + \left\{ \frac{1}{\rho \mathrm{Re}} \frac{\partial}{\partial x_j}\left(\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) \right\}^{(m,q)}[\boldsymbol{l}] + \left\{ \frac{1}{\rho \mathrm{We}} (f_\sigma)_i \right\}^{(m,q)}[\boldsymbol{l}]. \quad (2.9)$$

We use the convention $\{\,\cdot\,\}^{(m,q)}[\boldsymbol{l}]$ to denote the DWT, and we are able to apply the DWT term-by-term since it is a linear operator. Then, by multipying each term of (2.9) by $w_i^{(m,q)}[\boldsymbol{l}]$ and by applying the chain rule in reverse to the left-hand side of the resulting equation,

$$w_i^{(m,q)}[\boldsymbol{l}] \frac{\partial}{\partial t} w_i^{(m,q)}[\boldsymbol{l}] = \frac{\partial}{\partial t}(\tfrac{1}{2} w_i^{(m,q)}[\boldsymbol{l}]^2), \quad (2.10)$$

we get

$$\frac{\partial}{\partial t}(\tfrac{1}{2} w_i^{(m,q)}[\boldsymbol{l}]^2) = -w_i^{(m,q)}[\boldsymbol{l}] \left\{ u_j \frac{\partial u_i}{\partial x_j} + \frac{1}{\rho}\frac{\partial p}{\partial x_i} \right\}^{(m,q)}[\boldsymbol{l}] + w_i^{(m,q)}[\boldsymbol{l}] \left\{ \frac{1}{\rho \mathrm{Re}} \frac{\partial}{\partial x_j}\left(\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) \right\}^{(m,q)}[\boldsymbol{l}] + w_i^{(m,q)}[\boldsymbol{l}] \left\{ \frac{1}{\rho \mathrm{We}} (f_\sigma)_i \right\}^{(m,q)}[\boldsymbol{l}]. \quad (2.11)$$

Summing each term of (2.11) over q and i and averaging the resulting equation over all scale-m wavelet coefficients gives us

$$\frac{\partial}{\partial t} \widetilde{E}(k_m) = \widetilde{T}(k_m) - \widetilde{V}(k_m) + \widetilde{S}(k_m). \quad (2.12)$$

$\widetilde{E}(k_m)$ is the wavelet energy spectrum defined in (2.7), and the terms on the right-hand side of (2.12) are defined by

$$\widetilde{T}(k_m) = -C_m \sum_{i=1}^{3} \left\langle \sum_{q=1}^{7} w_i^{(m,q)}[\boldsymbol{l}] \left\{ u_j \frac{\partial u_i}{\partial x_j} + \frac{1}{\rho}\frac{\partial p}{\partial x_i} \right\}^{(m,q)}[\boldsymbol{l}] \right\rangle_{\boldsymbol{l} \in \mathcal{D}^{(m)}}, \quad (2.13a)$$

$$\widetilde{V}(k_m) = -C_m \sum_{i=1}^{3} \left\langle \sum_{q=1}^{7} w_i^{(m,q)}[\boldsymbol{l}] \left\{ \frac{1}{\rho \mathrm{Re}} \frac{\partial}{\partial x_j}\left(\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) \right\}^{(m,q)}[\boldsymbol{l}] \right\rangle_{\boldsymbol{l} \in \mathcal{D}^{(m)}}, \quad (2.13b)$$

$$\widetilde{S}(k_m) = \quad C_m \sum_{i=1}^{3} \left\langle \sum_{q=1}^{7} w_i^{(m,q)}[\boldsymbol{l}] \left\{ \frac{1}{\rho \mathrm{We}} (f_\sigma)_i \right\}^{(m,q)}[\boldsymbol{l}] \right\rangle_{\boldsymbol{l} \in \mathcal{D}^{(m)}}, \quad (2.13c)$$

where $\widetilde{T}(k_m)$ is the energy-transfer rate due to advection and pressure, $\widetilde{V}(k_m)$ is the viscous dissipation rate, and $\widetilde{S}(k_m)$ is the surface-tension source/sink rate of energy at wavenumber k_m.

2.2.5 Decomposition of wavelet spectrum

To decompose $\widetilde{E}$ into carrier-phase and droplet parts, we selectively average over the computational domain. We define $\widetilde{E}_C(k_m)$ to average only over spatial indices $\boldsymbol{l}$ where the scale-m wavelet at $\boldsymbol{l}$ is entirely contained in the carrier fluid. Similarly, $\widetilde{E}_D(k_m)$ is the average over $\boldsymbol{l}$ where the wavelet is entirely in a droplet. The "interaction" part $\widetilde{E}_I(k_m)$ is the average over $\boldsymbol{l}$ where the wavelet is partially contained in both carrier and droplet fluids. We define three new physical-space sets $\mathcal{D}_C^{(m)}$, $\mathcal{D}_D^{(m)}$, and $\mathcal{D}_I^{(m)}$ such that $\mathcal{D}^{(m)} = \mathcal{D}_C^{(m)} \cup \mathcal{D}_D^{(m)} \cup \mathcal{D}_I^{(m)}$. These sets are defined such that, for our choice of wavelet, $\mathcal{D}_C^{(m)}$ is all points $\boldsymbol{l} \in \mathcal{D}^{(m)}$ such that the level-m wavelet centered at $\boldsymbol{l}$ does not intersect any droplets, $\mathcal{D}_D^{(m)}$ is all points $\boldsymbol{l} \in \mathcal{D}^{(m)}$ such that the level-m wavelet centered at $\boldsymbol{l}$ does not intersect any carrier fluid, and $\mathcal{D}_I^{(m)} = \mathcal{D}^{(m)} \setminus (\mathcal{D}_C^{(m)} \cup \mathcal{D}_D^{(m)})$.

Then, according to these definitions, $\widetilde{E}$ is decomposed as

$$\widetilde{E}(k_m) = \frac{\left|\mathcal{D}_C^{(m)}\right|\widetilde{E}_C(k_m) + \left|\mathcal{D}_D^{(m)}\right|\widetilde{E}_D(k_m) + \left|\mathcal{D}_I^{(m)}\right|\widetilde{E}_I(k_m)}{|\mathcal{D}^{(m)}|}, \tag{2.14}$$

where $|\cdot|$ denotes the number of points in the set and, omitting the dependence on k_m,

$$\widetilde{E}_C = C_m \sum_{i=1}^{3} \left\langle \frac{1}{2} \sum_{q=1}^{7} w_i^{(m,q)} [\boldsymbol{l}]^2 \right\rangle_{\boldsymbol{l} \in \mathcal{D}_C^{(m)}}, \qquad \frac{\partial}{\partial t}\widetilde{E}_C = \widetilde{T}_C - \widetilde{V}_C, \tag{2.15a}$$

$$\widetilde{E}_D = C_m \sum_{i=1}^{3} \left\langle \frac{1}{2} \sum_{q=1}^{7} w_i^{(m,q)} [\boldsymbol{l}]^2 \right\rangle_{\boldsymbol{l} \in \mathcal{D}_D^{(m)}}, \qquad \frac{\partial}{\partial t}\widetilde{E}_D = \widetilde{T}_D - \widetilde{V}_D, \tag{2.15b}$$

$$\widetilde{E}_I = C_m \sum_{i=1}^{3} \left\langle \frac{1}{2} \sum_{q=1}^{7} w_i^{(m,q)} [\boldsymbol{l}]^2 \right\rangle_{\boldsymbol{l} \in \mathcal{D}_I^{(m)}}, \qquad \frac{\partial}{\partial t}\widetilde{E}_I = \widetilde{T}_I - \widetilde{V}_I + \widetilde{S}. \tag{2.15c}$$

Also, we use the same technique to decompose the terms of our evolution equation (2.12), as shown in the equations on the right of (2.15a)–(2.15c), in which $\widetilde{S}_C = \widetilde{S}_D = 0$ and $\widetilde{S} = \widetilde{S}_I$ because $\boldsymbol{f}_\sigma = 0$ everywhere in the field except at the interface. Thus, the direct effect of the

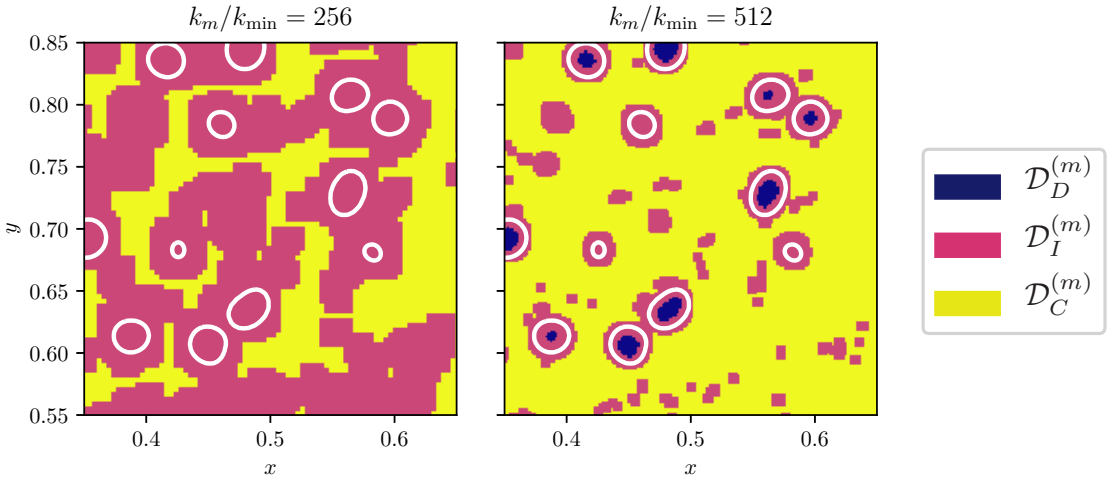

Figure 2.4: Example of decomposition $\mathcal{D}^{(m)} = \mathcal{D}_C^{(m)} \cup \mathcal{D}_D^{(m)} \cup \mathcal{D}_I^{(m)}$ for case C with $k_m/k_{\min} = 256$ and 512 ($m = 2$ and 1) in a subregion of the x-y plane for $z = 0.5$.

surface-tension force contributes directly only to the evolution of $\widetilde{E}_I(k_m)$. The process for numerically computing the decomposition is presented in greater detail in Appendix A.

Figure 2.4 shows an example of the domain decomposition in a plane for case C with the two highest wavenumbers. Note that, when increasing the wavenumber from $k_m/k_{\min} = 256$ to 512, the details of the flow in the vicinity of the droplet interface are represented in more detail since the size of the wavelet basis function used for the analysis becomes smaller. At the largest wavenumber, $k_m/k_{\min} = 512$ (for $m = 1$), the sym4 wavelet extends 8 points in each spatial direction. Because the carrier part of our decomposition cannot contain points at which the wavelet intersects a droplet, $\mathcal{D}_C^{(1)}$ can only include points farther than $8\,\Delta x$ from the droplet interfaces. This distance is equivalent to $D/4$ or 5η. A wavelet centered at any point closer than that to a droplet will always cross its interface. As the wavenumber halves, the size of the wavelet doubles, so for $k_m/k_{\min} = 256$ (for $m = 2$), $\mathcal{D}_C^{(2)}$ can only include points farther than $D/2$ or 10η from the droplet interfaces. And for $k_m/k_{\min} = 128$ (for $m = 3$), $\mathcal{D}_C^{(3)}$ can only include points farther than D or 20η from the droplets. For the cases studied here, our decomposition cannot represent any wavenumbers smaller than

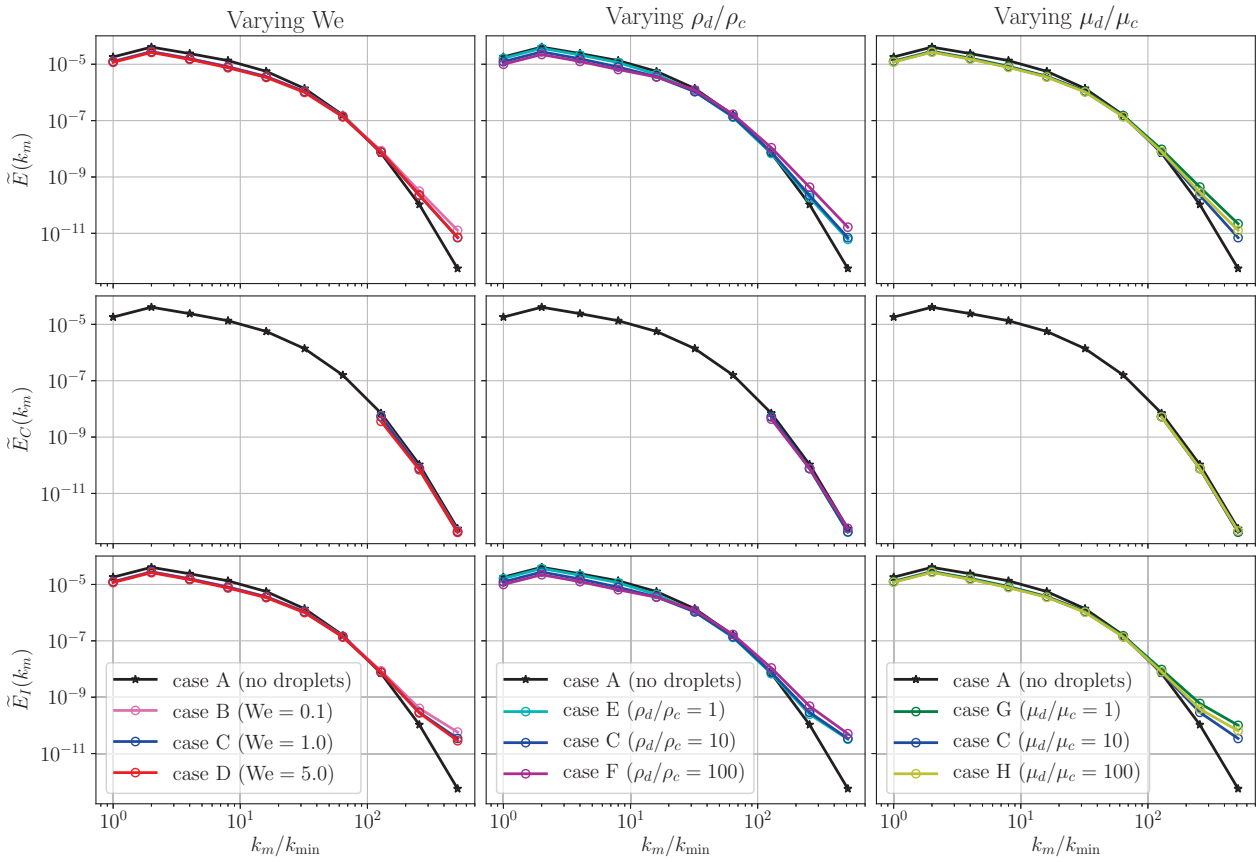

Figure 2.5: Top row: total wavelet spectra $\widetilde{E}(k_m)$ (2.7). Middle row: carrier-phase wavelet spectra $\widetilde{E}_C(k_m)$ (2.15a). Bottom row: interaction wavelet spectra $\widetilde{E}_I(k_m)$ (2.15c).

$k_m/k_{\min} = 128$ ($m = 3$) for $\mathcal{D}_C^{(m)}$ because for $m = 4$ the wavelets become large enough that they cross the droplet interface.

2.3 Results

Figure 2.5 shows the total wavelet spectra $\widetilde{E}(k_m)$, the carrier wavelet spectra $\widetilde{E}_C(k_m)$, and the interaction wavelet spectra $\widetilde{E}_I(k_m)$ for all cases A–H. Note that $\widetilde{E}_C(k_m) = \widetilde{E}(k_m)$ for case A and that we plot $\widetilde{E}(k_m)$ of case A for reference in the $\widetilde{E}_I(k_m)$ plot. When comparing the spectra of the droplet-laden cases B–H to the droplet-free case A, we notice: (i) $\widetilde{E}_C(k_m)$

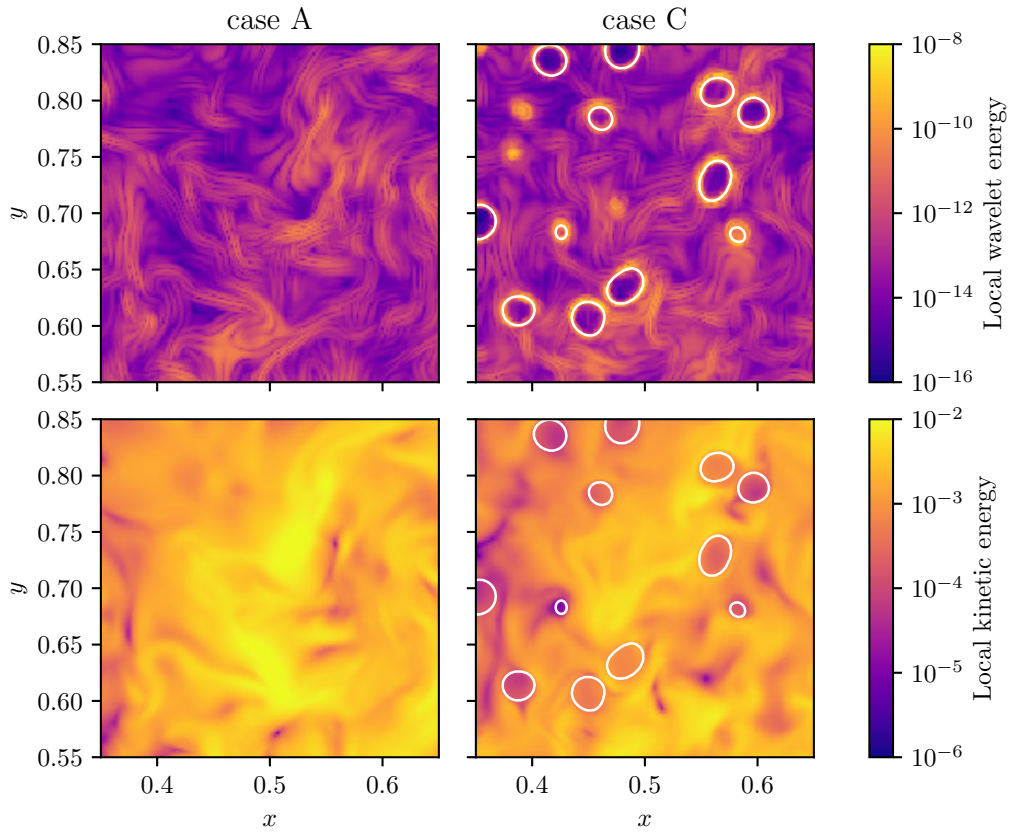

Figure 2.6: Comparison of local wavelet energy at $k_m/k_{\min} = 512$ and local kinetic energy for cases A and C in a subregion of the x-y plane at $z = 0.5$.

is nearly unchanged at $k_m/k_{\min} \geq 128$, (ii) $\widetilde{E}(k_m)$ is increased at $k_m/k_{\min} = 256$ and 512, and (iii) $\widetilde{E}(k_m)$ is reduced at $k_m/k_{\min} \leq 16$.

(i) $\widetilde{E}_C(k_m)$ is nearly the same in all cases for $k_m/k_{\min} \geq 128$. According to (2.15a), $\widetilde{E}_C(k_m)$ depends on $\mathcal{D}_C^{(m)}$, which in turn depends on the scale m. Because $\mathcal{D}_C^{(m)}$ can only contain points at which wavelets at scale m will not extend into any droplets, the closest distance any point in $\mathcal{D}_C^{(m)}$ can be to a droplet is $D = 20\eta$, $D/2 = 10\eta$, and $D/4 = 5\eta$ for $k_m/k_{\min} = 128$, 256, and 512, respectively. Therefore, since $\widetilde{E}_C(k_m)$ is nearly the same in all cases, this means that the presence of droplets does not affect the wavelet energy at large wavenumbers $k_m/k_{\min} \geq 128$ (i.e., the turbulence scales $\leq 5\eta$) in the flow regions at distances greater than

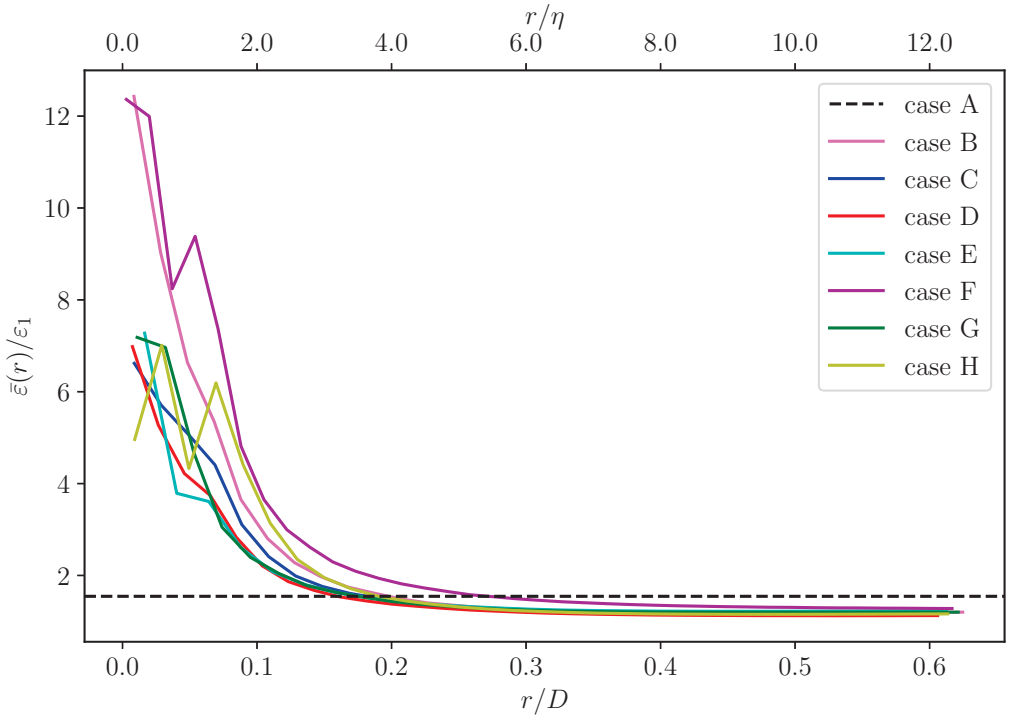

Figure 2.7: Viscous dissipation rate $\bar{\varepsilon}(r)$ averaged over points at distance r from the nearest droplet interface, normalized by the dissipation rate ε_1 at $t = 1$. The dashed line represents the normalized dissipation rate $\varepsilon/\varepsilon_1$ at $t = 3.5$ for the droplet-free case.

$D = 20\eta$ from the droplets. For a qualitative display of this phenomenon, Fig. 2.6 shows the local wavelet energy at $k_m/k_{\min} = 512$ for a subregion of the x-y plane for the droplet-free case A and droplet-laden case C. The flow regions away from the droplets have nearly the same wavelet-energy magnitude. To explain why the carrier energy at these wavenumbers is unaffected, we analyze the terms of the evolution equation of $\widetilde{E}_C(k_m)$ (2.15a), namely $\widetilde{T}_C(k_m)$ and $\widetilde{V}_C(k_m)$. Figure 2.8 shows the carrier-phase energy-transfer rate $\widetilde{T}_C(k_m)$, and Fig. 2.9 shows the carrier-phase viscous dissipation rate $\widetilde{V}_C(k_m)$. As for $\widetilde{E}_C(k_m)$, $\widetilde{T}_C(k_m)$ and $\widetilde{V}_C(k_m)$ are nearly the same in all cases at $k_m/k_{\min} \geq 128$. (Note that, for case F, there is a larger number of smaller droplets due to droplet breakup than in the other cases; consequently, $\mathcal{D}_C^{(m)}$ for $k_m/k_{\min} = 128$ has only 164 points and the statistics for this case are thus unreliable and not reported.) Thus, $\widetilde{E}_C(k_m)$ evolves nearly at the same rate with and without droplets according to (2.15a) because $\widetilde{V}_C(k_m)$ and $\widetilde{T}_C(k_m)$ are nearly unaffected by the droplets at high wavenumbers. This means that the droplets are not affecting the energy-transfer rate or viscous dissipation rate at high wavenumbers $k_m/k_{\min} \geq 128$ at distances $\geq 20\eta$ from their surface, and thus, according to (2.15a), they are not affecting the energy $\widetilde{E}_C(k_m)$ either. To see what is happening in physical space locally as a function of the distance r from the droplet interface close to the droplet (for $r \leq D/4$), in Fig. 2.7, we plot the viscous dissipation rate $\bar{\varepsilon}(r) = 2\mu\overline{S_{ij}S_{ij}}/\mathrm{Re}$, where

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.16}$$

is the rate-of-strain tensor and the overbar denotes the average over the points at distance r from the droplet interface, normalized by the dissipation rate ε_1 at $t = 1$. Droplets increase $\bar{\varepsilon}(r)$ near the interface ($r \leq D/4 = 5\eta$) due to the increased velocity gradients near the interface relative to the droplet-free case, and away from the droplet interface ($r \geq D/4 = 5\eta$) the dissipation settles to a uniform value slightly lower than that of single-phase flow due to the overall reduction of TKE. This confirms that away from the droplet interface the region of greatest change here appears to be within about $D/4 = 5\eta$ of the droplet (or 4η if we use the Kolmogorov length scale at the time our statistics are computed). Therefore, we conclude that the thickness of the dissipation layer around the droplets is indeed $D/4$

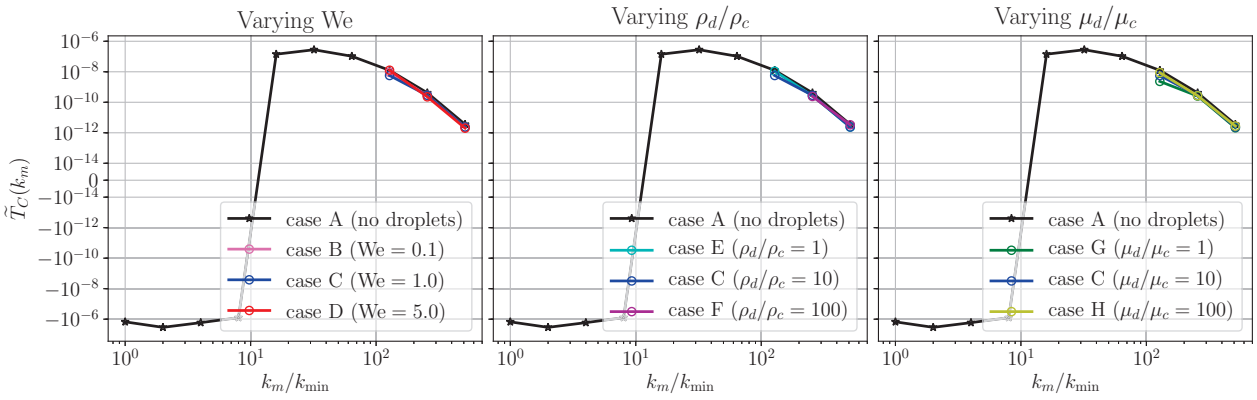

Figure 2.8: Carrier-phase energy-transfer rate $\widetilde{T}_C(k_m)$ (2.13a).

as we have seen through our wavelet analysis for the highest wavenumber $k_m/k_{\min} = 512$ (besides case F, whose increased density ratio increases the width of the dissipation layer). This result is also confirmed by the physical-space statistics of the topological study of the same dataset of [13].

(ii) $\widetilde{E}(k_m)$ is higher for the droplet-laden cases than in case A at $k_m/k_{\min} \geq 256$ (Fig. 2.5, top row). According to (2.14), $\widetilde{E}(k_m)$ is a weighted average of $\widetilde{E}_C(k_m)$, $\widetilde{E}_D(k_m)$, and $\widetilde{E}_I(k_m)$. $\widetilde{E}_C(k_m)$ is nearly the same for all cases (Fig. 2.5, middle row), so it does not contribute to the observed increase at high wavenumbers of $\widetilde{E}(k_m)$. $\widetilde{E}_D(k_m)$ does not contribute much to $\widetilde{E}(k_m)$ because $\mathcal{D}_D^{(m)}$ is always less than 2% of the domain. $\widetilde{E}_I(k_m)$, instead, is also increased at high wavenumbers by the droplets (Fig. 2.5, bottom row). Remember that $\widetilde{E}_I(k_m)$ only includes points at which wavelets cross the interface, and the wavelet transform will detect any sharp changes. All of the droplet-laden cases have some degree of velocity nonsmoothness at the interface like in Fig. 2.2. Decreasing We makes the droplets less deformable, increasing ρ_d/ρ_c gives the droplets more inertia, and increasing μ_d/μ_c increases the difference between $\partial u_c/\partial x_i$ and $\partial u_d/\partial x_i$ across the interface [12, §3.4.4]. These factors cause the velocity gradients to peak at the droplet interface and in turn $\widetilde{E}_I(k_m)$ to contain more energy at high wavenumbers, and this increase is reflected in $\widetilde{E}(k_m)$, which is slightly

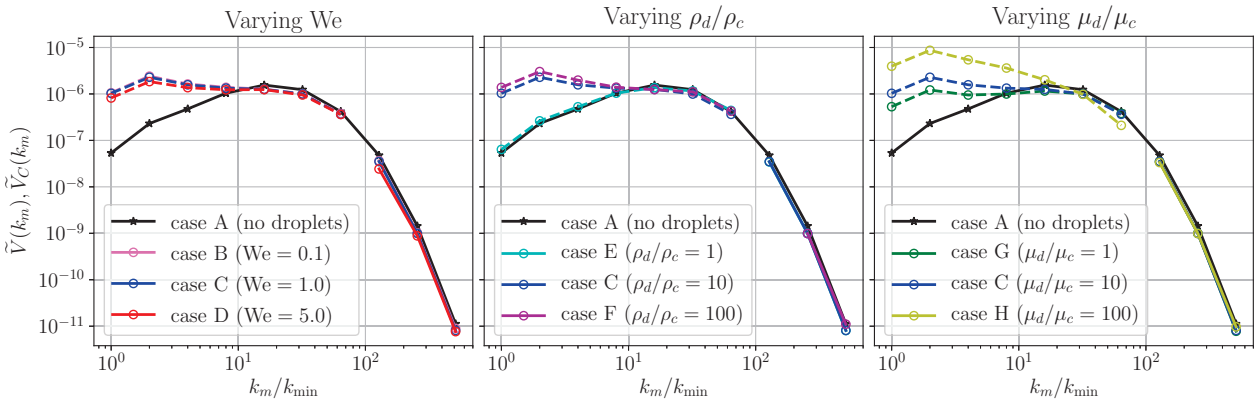

Figure 2.9: Dashed lines: total viscous dissipation rate $\widetilde{V}(k_m)$ (2.13b) (shown only for $k_m/k_{\min} \leq 64$). Solid lines: carrier-phase viscous dissipation rate $\widetilde{V}_C(k_m)$ (2.13b).

lower than $\widetilde{E}_I(k_m)$ due to the averaging with $\widetilde{E}_C(k_m)$. This is the same phenomenon that causes an oscillating increase in energy at high wavenumbers for the Fourier energy spectrum (Fig. 2.1). However, thanks to our novel wavelet-spectrum decomposition, we are able to isolate this effect to the portion of the domain containing the interface.

(iii) $\widetilde{E}(k_m)$ is lowered for the droplet-laden cases with $\rho_d/\rho_c > 1$ at $k_m/k_{\min} \leq 16$ with respect to that of case A (Fig. 2.5, top row). This effect can also be seen to a lesser extent in the Fourier energy spectrum in Fig. 2.1. The only parameter that changes the amount of this decrease is the density ratio: case F has the highest density ratio and shows the greatest decrease; case E is the only droplet-laden case with density ratio 1, and it shows nearly no decrease at these wavenumbers. This pattern suggests that energy is decreased at low wavenumbers due to the increased inertia of higher-density droplets. According to (2.12), the evolution of $\widetilde{E}(k_m)$ depends on $\widetilde{V}(k_m)$, $\widetilde{T}(k_m)$, and $\widetilde{S}(k_m)$. However, we focus our attention on $\widetilde{V}(k_m)$ because $\widetilde{S}(k_m)$ is one order of magnitude smaller than $\widetilde{V}(k_m)$ and $\widetilde{T}(k_m)$, while $\widetilde{T}(k_m)$ is not as significantly affected by the droplets as $\widetilde{V}(k_m)$. Droplets with higher density than the surrounding fluid increase the dissipation rate $\widetilde{V}(k_m)$ at low wavenumbers, as shown in Fig. 2.9, by increasing the velocity gradients occurring in the flow. Because of

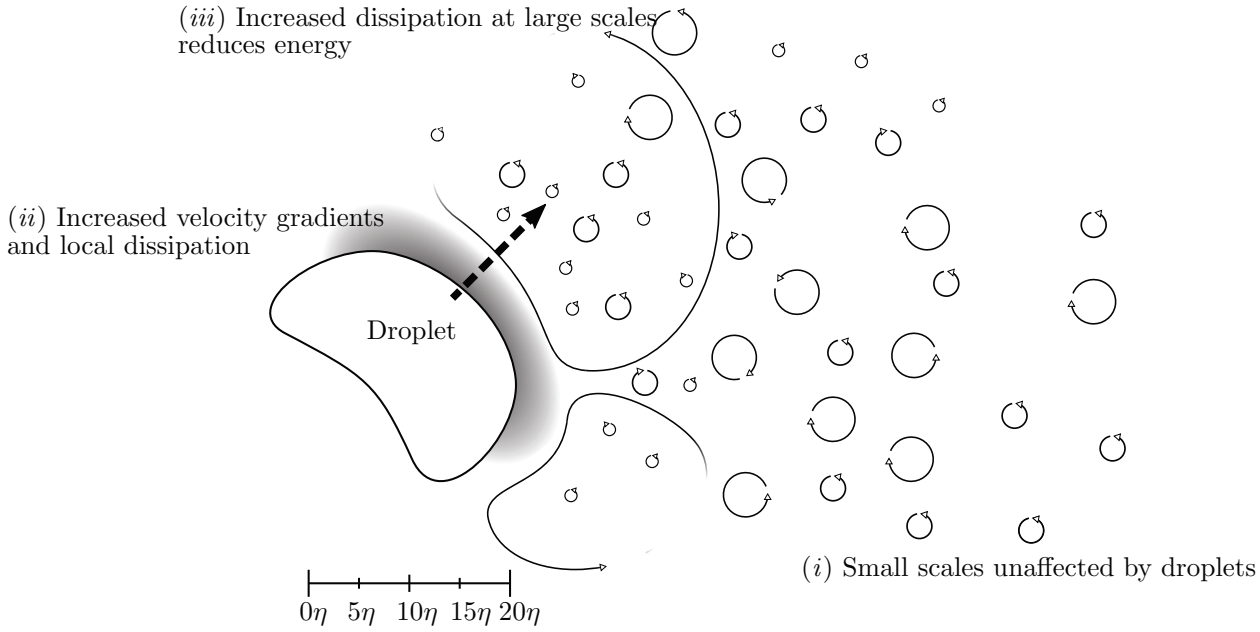

Figure 2.10: Schematic of results (i)–(iii).

their higher inertia, they will disturb the large-scale eddies more by being more resistant to the carrier flow motion with respect to no droplets (case A) or buoyant droplets (case E). In summary, while droplets with density ratio 1 (case E), by not changing $\widetilde{V}(k_m)$ at low wavenumbers with respect to case A, do not change $\widetilde{E}(k_m)$ at low wavenumbers according to (2.12), droplets with increasing density ratio from 10 to 100, by increasing the dissipation rate $\widetilde{V}(k_m)$ at the low wavenumbers $k_m/k_{\min} \leq 16$ with respect to case A, reduce $\widetilde{E}(k_m)$ at these wavenumbers according to (2.12).

2.4 Conclusion

We have presented the wavelet energy-spectra equations, proposed a new decomposition for two-fluid incompressible isotropic turbulence, and applied it to analyze the DNS dataset of droplet-laden decaying isotropic turbulence of Dodd & Ferrante [12]. The proposed domain decomposition allows separating the effects of droplets on the spectra of carrier flow,

droplet flow, and flow regions crossing the interface. Our results show that, at an average distance from the droplets larger than 5η or $D/4$ and at high wavenumbers ($k_m/k_{\min} \geq 128$), the carrier-phase spectra are nearly unaffected by the droplets for the given parameter set (in particular for a volume fraction of 5%). Also, droplets increase the energy at high wavenumbers ($k_m/k_{\min} \geq 256$) near the interface due to the local larger velocity gradients. Furthermore, the droplets increase the dissipation rate at low wavenumbers ($k_m/k_{\min} \leq 16$) and thereby decrease the TKE spectra at those wavenumbers. A visual summary of these results is given in the schematic of Fig. 2.10.

Note that our results do not easily lend themselves to a power-law analysis. For one, the wavelet spectra do not necessarily have the same slopes as their Fourier counterparts [42]. Secondly, our Reynolds number is not high enough ($\mathrm{Re}_\lambda = 83$) to see an inertial subrange of wavenumbers (see the dashed lines of Fig. 2.1), which is instead present at $\mathrm{Re}_\lambda \gtrsim 257$ [24, Fig. 3].

These results show that wavelets are an important tool for understanding the physics of multiphase turbulence. In particular, we will show in Chapter 3 that this opens new avenues to formulate LES models for multiphase flows. Because we decomposed droplet-laden DNS results and saw that spectral effects of the droplets were restricted to nearby regions at distances from the droplets smaller than $D/4$, especially that $\widetilde{V}_C(k_m)$ and $\widetilde{E}_C(k_m)$ were unaffected by the droplets at high wavenumbers. This indicates that an LES model in a flow configuration similar to the one described here could still model the subgrid scales at distances larger than $D/4$ or 5η from the droplets as single-phase flow. The wavelet decomposition could also have useful applications to related multiphase experiments. For example, the time-series velocity data gained from hot-film anemometry in bubbly flows typically contains velocity spikes due to penetration of the bubbles by the probe, similar to the high velocity gradients at the interfaces of our droplets. The wavelet decomposition method could be used to generate carrier-phase wavelet energy spectra in the frequency domain.

Chapter 3

LARGE-EDDY SIMULATION USING NEURAL NETWORKS

In this chapter, we will show how the findings from our wavelet-spectral analysis of Chapter 2 can be used, with the help of neural networks, to create a new LES method for droplet-laden turbulence. This work will soon be submitted [18].

3.1 Introduction

3.1.1 Motivation

As in Chapter 2, we are interested in droplet-laden isotropic turbulence, particularly the DNS study of Dodd & Ferrante [12], who investigated the flow physics of Taylor length-scale size droplet-laden decaying isotropic turbulence. They derived the turbulence kinetic energy (TKE) budget equations for the entire two-fluid flow, as well as for the carrier- and droplet-fluid of the flow. The two-fluid TKE equation includes the power-of-surface-tension term, and the carrier- and droplet-fluid TKE equations include power-of-viscous-stress and power-of-pressure terms, none of which are present in the single-phase TKE equation. Using these equations, they showed that the droplets increase the decay rate of TKE mainly due to an increased dissipation rate of TKE resulting from the high velocity gradients at the interface of the droplets, and explained the role of the power of surface tension in the dynamics of TKE based on the rate of change of droplet surface area. In addition to explaining the physical mechanisms of droplet-laden isotropic turbulence, Dodd & Ferrante [12] also analyzed the effects on the dynamics of TKE of changing the Weber number, density ratio, and viscosity ratio between the droplets and the carrier fluid.

We then extended the physical-space analysis of [12] to the spectral analysis of the turbulence kinetic energy of droplet-laden isotropic turbulence by using the wavelet transform

(Chapter 2). We decomposed the wavelet spectrum into carrier, droplet, and interaction parts and, in doing so, the results showed that, in comparison to the spectrum of the single-phase case, the droplets (i) do not affect the carrier-phase energy spectrum at high wavenumbers ($k_m/k_{\min} \geq 128$), (ii) increase the energy spectrum at high wavenumbers ($k_m/k_{\min} \geq 256$) by increasing the interaction energy spectrum at these wavenumbers, and (iii) decrease the energy at low wavenumbers ($k_m/k_{\min} \leq 16$) by increasing the dissipation rate at these wavenumbers, where $k_{\min} = 2\pi$ is the minimum domain-spanning wavenumber. See Fig. 2.10 for a schematic summarizing these results. These three observations motivate the herein proposed modeling methodology. Result (i) suggests that we can treat the SGS in the carrier-phase as in a single-phase flow, which is what we propose in our model; result (ii) suggests that modeling of the SGS terms for the interfacial region is important, which is what we show in both our *a priori* and *a posteriori* analysis herein; result (iii) suggests that the *a posteriori* LES will have to increase the dissipation rate at low wavenumbers in order to decrease the energy at low wavenumbers, which will be shown in our results.

As suggested by result (ii), our proposed LES model will treat the SGS stress in the carrier flow away from the interface (at distances larger than $D/4$ or 5η, where D is the initial droplet diameter and η is the Kolmogorov length-scale) as for single-phase isotropic turbulence. To this end, for the carrier-flow SGS stress, we use a standard single-phase LES model, i.e., the Smagorinsky model [54], and, at the droplet interface, we use artificial neural networks (ANNs) to model all the SGS closure terms. We call this approach a mixed artificial neural network (MANN) LES model summarized in the schematic of Fig. 3.1.

This MANN LES approach provides three important advantages. First, it is easy to implement in existing multiphase Navier–Stokes solvers. Second, this approach is versatile since by decoupling the problems of modeling the carrier-flow and interfacial closure terms, existing single-phase SGS stress models can be used in the bulk of the domain without having to modify the ANNs. Also, this approach can be applied to other multiphase flows besides droplet-laden ones, provided that the small-scale energy of the carrier fluid away from the interface can be approximated by that of single-phase flow as in our case.

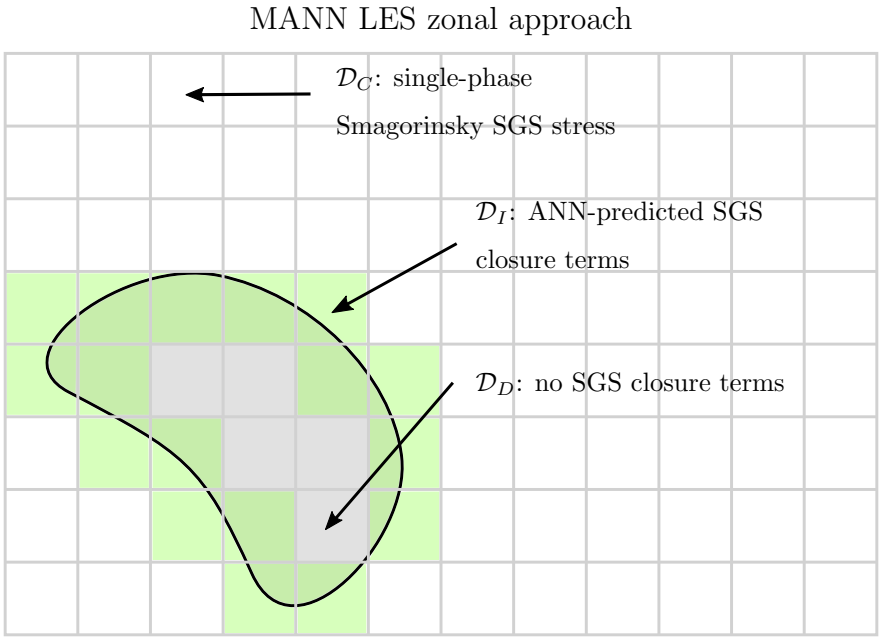

Figure 3.1: Schematic summarizing the MANN LES approach.

3.1.2 Previous work

LES for multiphase flows has been a topic of interest for some time [9] though almost all of the existing literature focuses on *a priori* analysis [55] and the feasibility of applying existing SGS modeling techniques [57, 31]. Some work has also focused on the modeling of SGS surface tension and recreating proper volume-of-fluid fields on coarse grids [35]. Common flows in these studies are single droplets/bubbles in turbulence [55, 31], liquid-jet atomization [9, 31, 28], and phase inversion [57]. Our interest is focused on LES modeling of droplet-laden decaying isotropic turbulence in which the volume fraction is sufficiently high to provide coupling effects and alter the dynamics of turbulence kinetic energy, and the droplets are of Taylor length-scale size as in [12]. Previous LES studies of such flows provide exclusively the *a priori* analysis [25, 27] and propose a potential SGS model [27] without providing the *a posteriori* analysis. To the best of our knowledge, the present work is the first to propose an LES model, and to report the *a posteriori* analysis of such flow.

We have been inspired by recent advances in turbulence modeling using ANNs [32, 46]. In particular, Beck, Flad & Munz [6] were able to train an ANN to model SGS stress for LES of isotropic turbulence. Sirignano, MacArt & Freund [52] developed a method for embedding an ANN into the Navier–Stokes equations to further improve machine-learned LES models by incorporating physical constraints during training. Raissi, Perdikaris & Karniadakis [44, 45] have also worked on physics-informed neural networks. Yang, Zafar, Wang & Xiao [59] applied a similar approach to wall-bounded LES. Given the promising results in the aforementioned studies, we believe that multiphase LES is an area in which ANNs can be useful. At the same time, while we have chosen to use ANNs for predicting closure terms at the interface, we avoid the additional computational overhead of making ANN predictions in the bulk of the flow when an existing single-phase SGS stress model will suffice for the carrier fluid.

3.1.3 Outline

Section 3.2 provides the *a priori* analysis of the SGS closure terms of droplet-laden decaying isotropic turbulence from the DNS of [8], including the LES equations (Section 3.2.3) and the *a priori* results (Section 3.2.4). In Section 3.3 we present the *a posteriori* analysis, including an introduction to ANNs (Section 3.3.3), an explanation of our numerical method (Section 3.3.4), and the results of the proposed MANN LES model (Section 3.3.5). Section 3.4 provides the concluding remarks.

3.2* A priori *analysis

The goal of our *a priori* analysis is to determine which SGS closure terms are statistically important, by comparing their relative orders of magnitude, to require modeling for accurate LES of droplet-laden isotropic turbulence. First, we need to define the governing equations and derive such closure terms. As described in Section 3.1.2, e.g., [25, 27, 28, 31, 55] have already provided the filtered Navier–Stokes equations of two-fluid incompressible flows, such as droplet-laden flows. However, since our formulation differs from those of the two *a priori*

studies of the same flow investigated here [25, 27], we clarify our formulation in the following sections.

3.2.1 DNS data

In our *a priori* analysis, we have chosen to focus on the flow parameters of case C of [12] because its droplets' physical parameters are the middle case of the parameter ranges studied. For this case, the droplet-to-carrier fluid density and viscosity ratios are $\rho_d/\rho_c = \mu_d/\mu_c = 10$, and the Weber number with respect to the root-mean-square velocity at $t = 1$ is $\mathrm{We}_{\mathrm{rms}} = 1$. This case will be used for our *a priori* analysis in Section 3.2.4 and to train our ANNs in Section 3.3.3. In our *a posteriori* analysis, we will see how our method works with case C, but we will also evaluate our method using cases B and D of [12], for which $\mathrm{We}_{\mathrm{rms}} = 0.1$ and 5, respectively, with all other parameters the same as in case C (Table 1.1). Whereas we used the DNS data of [12] in Chapter 2, we now perform the DNS ourselves to obtain additional data. See Section 1.2 for the overview of the DNS method of [12] and all of the relevant flow parameters.

3.2.2 Filtering

Before introducing the filtered equations for two-fluid incompressible flows in Section 3.2.3 for LES, we need to define how we will be filtering the governing equations (1.1) and (1.4). For a given function $f(\boldsymbol{x})$, we denote its filtered counterpart by $\bar{f}(\boldsymbol{x})$ and define it as the convolution of f with a filter g:

$$\bar{f}(\boldsymbol{x}) = \iiint f(\boldsymbol{x}')g(\boldsymbol{x} - \boldsymbol{x}')\, d\boldsymbol{x}'. \tag{3.1}$$

Our choice of a filter g is important because rather than just informing our LES model, it will directly impact the generation of training data for our ANNs as explained below in Section 3.3.3.

When deciding what filter to apply to (1.1) and (1.4), we considered three common choices: the uniform filter, the spectral filter, and the Gaussian filter [43]. The uniform filter

(also known as the box filter) is the simplest of the three and consists of a spatial average of a given width, so it is sharp in the physical domain. The spectral filter is similar to the uniform filter, but is instead sharp in the spectral domain. Therefore, it exactly removes wavenumber modes higher than a certain cut-off wavenumber. The Gaussian filter lies somewhere in the middle, removing some of the small scales in both the physical and spectral domains.

We chose to employ the uniform filter, defined as

$$g(\boldsymbol{x} - \boldsymbol{x}') = g(x_1 - x_1')g(x_2 - x_2')g(x_3 - x_3') \tag{3.2}$$

with

$$g(x_i - x_i') = \begin{cases} 1/\Delta & \text{if } |x_i - x_i'| \leq \Delta/2, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{for } i = 1, 2, 3, \tag{3.3}$$

for a filter size Δ. We adopted the uniform filter for its simplicity and the fact that it has been adopted before in practice, e.g., in the ANN-based method of [52]. Furthermore, and more importantly, the uniform filter has compact support, i.e., it is only nonzero on a bounded set of points, which in this case is exactly as wide as the filter width. The Gaussian or spectral filters are not compactly supported, so when applied to a flow variable, the discontinuities present at the interface will affect the whole domain. As explained in Section 3.3.1, it is important to isolate the points in the filtered domain at the interface between droplets and carrier fluid, and the uniform filter allows that because only points within one filter width of the interface will be affected by the interface.

The filter size is another important parameter. We have chosen $\Delta = 8\,\Delta x_{\text{DNS}}$, where Δx_{DNS} is the size of a DNS grid cell. Therefore, our coarse grid will be 128^3, and the droplets will now have an initial diameter of $4\,\Delta x_{\text{LES}}$ in the LES rather than $32\,\Delta x_{\text{DNS}}$ of the DNS. We found this degree of coarsening large enough to require accurate SGS modeling while not being so great that the droplets are no longer resolved at all. If we make Δ smaller, we remove fewer SGS effects, so LES performs nearly as well as DNS and using SGS models is no longer necessary. On the other hand, if we make Δ much larger, the diameter of the droplets will be less than the size of a grid cell, meaning that the droplets would not be

resolved at all by the computational grid, so LES would need to track and take into account SGS droplets, which is an interesting but different problem.

3.2.3 LES equations for incompressible two-fluid flows

Filtering Eqs. (1.1) and (1.4) using (3.1) gives

$$\frac{\partial \bar{u}_j}{\partial x_j} = 0, \tag{3.4a}$$

$$\bar{\rho}\left(\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j}\right) = -\frac{\partial \bar{p}}{\partial x_i} + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x_j}\left(\bar{\mu}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right)\right) + \frac{1}{\mathrm{We}} f_\sigma(\bar{\chi})_i + \tau_i^{\mathrm{conv}} + \tau_i^{\mathrm{accel}} + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x_j}\tau_{ij}^{\mathrm{visc}} + \frac{1}{\mathrm{We}}\tau_i^{\mathrm{surf}}, \tag{3.4b}$$

$$\frac{\partial \bar{\chi}}{\partial t} + \bar{u}_j \frac{\partial \bar{\chi}}{\partial x_j} = \tau^{\mathrm{adv}}, \tag{3.4c}$$

where

$$\tau_i^{\mathrm{conv}} = \bar{\rho}\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} - \overline{\rho u_j \frac{\partial u_i}{\partial x_j}}, \tag{3.5a}$$

$$\tau_i^{\mathrm{accel}} = \bar{\rho}\frac{\partial \bar{u}_i}{\partial t} - \overline{\rho \frac{\partial u_i}{\partial t}}, \tag{3.5b}$$

$$\tau_{ij}^{\mathrm{visc}} = \overline{\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)} - \bar{\mu}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right), \tag{3.5c}$$

$$\tau_i^{\mathrm{surf}} = \overline{f_\sigma(\chi)_i} - f_\sigma(\bar{\chi})_i, \tag{3.5d}$$

$$\tau^{\mathrm{adv}} = \bar{u}_j \frac{\partial \bar{\chi}}{\partial x_j} - \overline{u_j \frac{\partial \chi}{\partial x_j}}. \tag{3.5e}$$

All the closure terms defined in (3.5) are functions of $\boldsymbol{x}$ and t. Note that the filtering operation commutes with all linear operations, including differentiation. The closure terms arise when applying the filter to nonlinear terms. In the single-phase case, $\tau_i^{\mathrm{accel}} = 0$ and $\tau_{ij}^{\mathrm{visc}} = 0$ because ρ and μ are constant, and τ_i^{surf} and τ^{adv} don't exist because there is no surface tension or VoF function. Therefore, the filtered single-phase momentum equation only includes τ_i^{conv}.

Finally, we rewrite (3.4b) with $\bar{\rho}$ on the right-hand side so that it is in the form used by

our Navier–Stokes solver:

$$\frac{\partial \bar{u}_i}{\partial t} = -\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} + \frac{1}{\bar{\rho}}\left(-\frac{\partial \bar{p}}{\partial x_i} + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x_j}\left(\bar{\mu}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right)\right) + \frac{1}{\mathrm{We}} f_\sigma(\bar{\chi})_i\right) + \hat{\tau}_i^{\mathrm{conv}} + \hat{\tau}_i^{\mathrm{accel}} + \hat{\tau}_i^{\mathrm{visc}} + \hat{\tau}_i^{\mathrm{surf}}, \tag{3.6}$$

where

$$\hat{\tau}_i^{\mathrm{conv}} = \frac{1}{\bar{\rho}}\tau_i^{\mathrm{conv}} = \frac{1}{\bar{\rho}}\left(\bar{\rho}\bar{u}_j\frac{\partial \bar{u}_i}{\partial x_j} - \overline{\rho u_j \frac{\partial u_i}{\partial x_j}}\right), \tag{3.7a}$$

$$\hat{\tau}_i^{\mathrm{accel}} = \frac{1}{\bar{\rho}}\tau_i^{\mathrm{accel}} = \frac{1}{\bar{\rho}}\left(\bar{\rho}\frac{\partial \bar{u}_i}{\partial t} - \overline{\rho \frac{\partial u_i}{\partial t}}\right), \tag{3.7b}$$

$$\hat{\tau}_i^{\mathrm{visc}} = \frac{1}{\bar{\rho}\,\mathrm{Re}}\frac{\partial}{\partial x_j}\tau_{ij}^{\mathrm{visc}} = \frac{1}{\bar{\rho}\,\mathrm{Re}}\frac{\partial}{\partial x_j}\left(\overline{\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)} - \bar{\mu}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right)\right), \tag{3.7c}$$

$$\hat{\tau}_i^{\mathrm{surf}} = \frac{1}{\bar{\rho}\,\mathrm{We}}\tau_i^{\mathrm{surf}} = \frac{1}{\bar{\rho}\,\mathrm{We}}(\overline{f_\sigma(\chi)}_i - f_\sigma(\bar{\chi})_i). \tag{3.7d}$$

The $\hat{\tau}$ notation of the closure terms defined in (3.7) will be useful in the rest of our *a priori* analysis and when defining our numerical method in Section 3.3.4, but the τ definitions of (3.5) will be what are actually computed by the ANNs. For the *a posteriori* analysis, after the ANNs predict the closure terms in the τ form (3.5), we compute the $\hat{\tau}$ according to the first column of equalities in (3.7), which are, then, used in (3.6) by our solver. Note that τ_i^{accel}, $\tau_{ij}^{\mathrm{visc}}$, τ_i^{surf}, and τ^{adv} (and therefore also $\hat{\tau}_i^{\mathrm{accel}}$, $\hat{\tau}_{ij}^{\mathrm{visc}}$, and $\hat{\tau}_i^{\mathrm{surf}}$) are zero at $\boldsymbol{x}$ where $\bar{\chi}(\boldsymbol{x}) = 0$ or 1, which, when using the uniform filter, will be true everywhere except at the cells containing the interface. In the cases of τ_i^{accel}, $\tau_{ij}^{\mathrm{visc}}$ and τ^{adv}, this is the case because $\bar{\rho}$, $\bar{\mu}$ and $\bar{\chi}$ are only nonconstant where the filter has overlapped the interface, and, in the case of τ_i^{surf}, this is the case because the surface-tension force (1.2) is only nonzero at the interface. However, τ_i^{conv} and $\hat{\tau}_i^{\mathrm{conv}}$ must be computed across the entire domain. We can rewrite (3.5a) as

$$\tau_i^{\mathrm{conv}} = \begin{cases} \bar{\rho}\bar{u}_j\dfrac{\partial \bar{u}_i}{\partial x_j} - \overline{\rho u_j \dfrac{\partial u_i}{\partial x_j}} & \text{if } 0 < \bar{\chi}(\boldsymbol{x}) < 1, \\ \bar{\rho}\left(\bar{u}_j\dfrac{\partial \bar{u}_i}{\partial x_j} - \overline{u_j \dfrac{\partial u_i}{\partial x_j}}\right) & \text{otherwise,} \end{cases} \tag{3.8}$$

and (3.7a) as

$$\hat{\tau}_i^{\text{conv}} = \begin{cases} \dfrac{1}{\bar{\rho}}\left(\bar{\rho}\bar{u}_j\dfrac{\partial \bar{u}_i}{\partial x_j} - \overline{\rho u_j \dfrac{\partial u_i}{\partial x_j}}\right) & \text{if } 0 < \bar{\chi}(\boldsymbol{x}) < 1, \\ \bar{u}_j\dfrac{\partial \bar{u}_i}{\partial x_j} - \overline{u_j\dfrac{\partial u_i}{\partial x_j}} & \text{otherwise.} \end{cases} \tag{3.9}$$

In other words, τ_i^{conv} is equivalent to the standard single-phase SGS stress term away from the interface. As seen in Chapter 2, the small-scale spectral energy content of droplet-laden isotropic turbulence is unaffected in the carrier phase compared to single-phase flows. Therefore, we choose to use a standard LES model for the SGS stress in τ_i^{conv} away from the interface. Our mixed LES model then uses ANNs to predict $\hat{\tau}_i^{\text{conv}}$, $\hat{\tau}_i^{\text{accel}}$, $\hat{\tau}_i^{\text{visc}}$, $\hat{\tau}_i^{\text{surf}}$, and τ^{adv} at the interface and the Smagorinsky SGS stress model to predict $\hat{\tau}_i^{\text{conv}}$ in the carrier fluid. We explain this choice in more detail in Section 3.3.2.

In the two existing *a priori* analyses of this flow [25, 27], the LES equations differ from ours. Jain and Lele [25] as the filtering operation use Favre averaging, which is a density-weighted average applied to the Navier–Stokes equations (1.1). The Favre average of a flow variable $f(\boldsymbol{x})$ is defined as $\overline{\rho(\boldsymbol{x})f(\boldsymbol{x})}/\bar{\rho}(\boldsymbol{x})$, and this results in different closure terms. Our primary motivation for opting out of Favre averaging is that it would introduce a closure term to the mass-conservation equation, which we do not have in Eq. (3.4a). On the other hand, Jofre *et al.* [27] use the so-called "large-eddy and interface simulation (LEIS)" approach of [33] for their *a priori* study. They do not use Favre averaging, but they argue that the Hinze length scale, which corresponds to the droplet size at which surface tension balances break-up forces, is at least one order of magnitude times larger than the Kolmogorov length scale, so by using a filter size between the Hinze and Kolmogorov scales, the interface will be sufficiently resolved to capture droplet deformations and breakup. With the large-scale effects of the droplets captured, they argue that they may then neglect SGS interface effects. Therefore, in their LEIS formulation, they only filter the two fluids and not the interface, meaning that the interface preserves jump conditions and the only closure term is the single-phase SGS stress term, which they propose to treat analogously to the LES modeling of turbulent boundary layers modified to allow for slip velocity boundary conditions at the

interface. Although our filter size is between the Kolmogorov and Hinze scales, meaning that the LES grid is able to approximately capture and advect the deformable droplets, our *a priori* analysis, obtained by filtering the interface, shows that the SGS interface effects are significant, so we do filter as well the interface in our formulation and account for its SGS terms in our *a posteriori* analysis. In summary, while, in their *a priori* study, Jofre *et al.* argue that, for such a filter size, it is sufficient to model only the SGS stress of the carrier fluid, our *a priori* analysis (Section 3.2.4) shows that the interfacial closure terms are of the same order of magnitude of the SGS stress of the carrier fluid, and, thus, they cannot be neglected in our zonal MANN LES approach, and our *a posteriori* analysis (Section 3.3.5) shows the MANN LES approach works in capturing the evolution of the TKE of decaying isotropic turbulence.

3.2.4 Results

Our *a priori* analysis uses the DNS data of case C, which we generated as described in Section 3.2.1, to calculate the SGS closure terms, Eqs. (3.5e) and (3.7), in their discretized forms using second-order central finite differences. By analyzing the orders of magnitude for the statistics of the closure terms, we will know which are important to model.

Because, except τ_i^{conv}, all other closure terms are zero anywhere the filter has not overlapped the interface, as explained above in Section 3.2.3, we compute them only in those cells that contain the interface, namely over the domain region

$$\mathcal{D}_I = \{\boldsymbol{x} \mid \epsilon < \bar{\chi}(\boldsymbol{x}) < 1 - \epsilon\}, \tag{3.10}$$

where ϵ is a prescribed tolerance (to avoid grid points containing only a tiny amount of the interface). We have chosen $\epsilon = 10^{-3}$, which leads to the set $\mathcal{D}_I$ containing about 10% of the 128^3 filtered grid. It will later be useful to have also defined

$$\mathcal{D}_C = \{\boldsymbol{x} \mid \bar{\chi}(\boldsymbol{x}) \leq \epsilon\}, \tag{3.11a}$$

$$\mathcal{D}_D = \{\boldsymbol{x} \mid \bar{\chi}(\boldsymbol{x}) \geq 1 - \epsilon\}, \tag{3.11b}$$

	$\hat{\tau}_1^{\text{conv}}$	$\hat{\tau}_1^{\text{accel}}$	$\hat{\tau}_1^{\text{visc}}$	$\hat{\tau}_1^{\text{surf}}$	$\bar{u}_j(\partial\bar{u}_1/\partial x_j)$
mean	-1.62×10^{-5}	4.94×10^{-5}	-8.38×10^{-6}	1.09×10^{-4}	4.55×10^{-6}
mean mag.	8.74×10^{-3}	8.28×10^{-3}	5.50×10^{-3}	2.79×10^{-2}	2.74×10^{-2}
std. dev.	1.52×10^{-2}	1.46×10^{-2}	8.28×10^{-3}	4.92×10^{-2}	4.52×10^{-2}
minimum	-2.39×10^{-1}	-2.41×10^{-1}	-9.54×10^{-2}	-1.48	-7.58×10^{-1}
25%	-4.40×10^{-3}	-3.91×10^{-3}	-3.43×10^{-3}	-1.48×10^{-2}	-1.48×10^{-2}
50%	-1.11×10^{-5}	3.57×10^{-6}	-3.32×10^{-6}	4.75×10^{-5}	-1.29×10^{-4}
75%	4.35×10^{-3}	4.03×10^{-3}	3.39×10^{-3}	1.51×10^{-2}	1.47×10^{-2}
maximum	2.28×10^{-1}	2.35×10^{-1}	9.81×10^{-2}	1.73	7.11×10^{-1}

Table 3.1: Statistics at $t = 3$ for the momentum-equation (3.6) closure terms for $i = 1$ and the nonlinear term $\bar{u}_j(\partial\bar{u}_1/\partial x_j)$.

which are the subsets of the domain we consider to be in the carrier fluid and droplets, respectively. These definitions (3.10) and (3.11) are similar to those of $\mathcal{D}_I^{(m)}$, $\mathcal{D}_C^{(m)}$, and $\mathcal{D}_D^{(m)}$ in Section 2.2.5 in the sense that they split the domain into three regions corresponding to the interface, carrier fluid, and droplet regions of the computational domain.

Table 3.1 shows statistics for the x-direction closure terms (3.7) in (3.6) ($\hat{\tau}_1^{\text{conv}}$, $\hat{\tau}_1^{\text{accel}}$, $\hat{\tau}_1^{\text{visc}}$, and $\hat{\tau}_1^{\text{surf}}$) computed at the nondimensional time $t = 3$. These statistics include the mean; mean magnitude; standard deviation; minimum; first, second, and third quartiles; and maximum. We only report the closure terms for $i = 1$ since the flow is isotropic, i.e., statistically invariant to rotation, and, therefore, the closure terms for $i = 2$ and 3 are nearly identical. In the last column, we show the same statistics for the nonlinear term of the filtered momentum equation (3.6), $\bar{u}_j(\partial\bar{u}_1/\partial x_j)$, for comparison. Table 3.2 shows the same statistics as Table 3.1, but for the closure term τ^{adv} (3.5e) of the VoF advection equation (3.4c) along with the term $\bar{u}_j(\partial\bar{\chi}/\partial x_j)$ for comparison.

Table 3.1 shows that the statistical values of $\hat{\tau}_1^{\text{conv}}$, $\hat{\tau}_1^{\text{accel}}$, $\hat{\tau}_1^{\text{visc}}$, and $\hat{\tau}_1^{\text{surf}}$ are all comparable to that of the nonlinear term and therefore none of them can be neglected in the LES SGS

	τ^{adv}	$\bar{u}_j(\partial\bar{\chi}/\partial x_j)$
mean	8.34×10^{-4}	-3.52×10^{-4}
mean mag.	2.37×10^{-1}	9.91×10^{-1}
std. dev.	3.64×10^{-1}	1.40
minimum	-2.66	-9.80
25%	-1.66×10^{-1}	-6.92×10^{-1}
50%	-1.34×10^{-2}	-3.00×10^{-2}
75%	1.08×10^{-1}	6.50×10^{-1}
maximum	3.63	9.11

Table 3.2: Statistics at $t = 3$ for τ^{adv} and $\bar{u}_j(\partial\bar{\chi}/\partial x_j)$ of the VoF advection equation (3.4c).

	$\hat{\tau}_1^{\text{conv}}/P_1$	$\hat{\tau}_1^{\text{accel}}/P_1$	$\hat{\tau}_1^{\text{visc}}/P_1$	$\hat{\tau}_1^{\text{surf}}/P_1$
mean	-15.8%	26.8%	20.4%	68.6%

Table 3.3: Means of the relative contributions of each SGS closure term (3.7) of the momentum equation (3.6) to the closure-term sum P_1 at $t = 3$.

models. Of these four terms, $\hat{\tau}_1^{\text{surf}}$, with the highest standard deviation and largest range between minimum and maximum, has the widest distribution, and, also, about the same mean magnitude as that of the nonlinear term (about one order of magnitude larger than the remaining terms), so we can expect that $\hat{\tau}_i^{\text{surf}}$ will play an important role in our model. We see in Table 3.2 that the distribution of τ^{adv} is also comparable to that of $\bar{u}_j(\partial\bar{\chi}/\partial x_j)$, so we must model it as well, leaving us with a total of five closure terms to be modeled: $\hat{\tau}_i^{\text{conv}}$ (3.7a), $\hat{\tau}_i^{\text{accel}}$ (3.7b), $\hat{\tau}_i^{\text{visc}}$ (3.7c), $\hat{\tau}_i^{\text{surf}}$ (3.7d), and τ^{adv} (3.5e).

Lastly, Table 3.3 shows the mean relative contribution of the SGS closure terms (3.7) of the momentum equation (3.6) to the sum of the closure terms:

$$P_i = \hat{\tau}_i^{\text{conv}} + \hat{\tau}_i^{\text{accel}} + \hat{\tau}_i^{\text{visc}} + \hat{\tau}_i^{\text{surf}}. \tag{3.12}$$

Table 3.3 shows that, on average, $\hat{\tau}_1^{\text{surf}}$ accounts for almost 70% of P_1, again indicating that it will be an important term to model accurately.

3.3 A posteriori *analysis*

3.3.1 Overview of our approach

In Chapter 2, by using wavelet-spectral analysis of the DNS data of droplet-laden isotropic turbulence [12], we showed that, in comparison to the spectrum of the single-phase case, the droplets do not affect the carrier-phase energy spectrum at high wavenumbers (see Fig. 2.5). We did so by decomposing the wavelet spectrum into carrier, droplet, and interaction parts. The carrier spectrum at a given wavenumber includes only the contributions of wavelets corresponding to that wavenumber that are contained entirely in the carrier fluid, and the droplet spectrum is defined similarly. The interaction spectrum includes contributions from wavelets that overlap the interface. The carrier spectrum was unchanged relative to the single-phase spectrum at wavenumbers $k_m/k_{\text{min}} \geq 128$ where $k_{\text{min}} = 2\pi$ is the smallest wavenumber since the nondimensional width of the computational domain is 1 in each direction. Furthermore, the interaction energy spectrum at wavenumbers $k_m/k_{\text{min}} \geq 256$ increased relative to the single-phase spectrum, causing the total energy spectrum at these wavenumbers to also increase.

For the choice of parameters in Chapter 2, the smallest wavelets could only be as close as $5\eta = 8\,\Delta x_{\text{DNS}}$ to a droplet where η is the Kolmogorov length-scale. Within this distance there is a "dissipation layer" in which elevated velocity gradients produce an increase in the viscous dissipation rate. In the present study, we keep all of the parameters the same as in Chapter 2 with the exception of a grid that has been coarsened in each direction by a factor of 8. Therefore, the wavelet-spectral analysis shows that, with the current grid size and using the uniform filter, effects of the droplets are restricted to the grid cells containing the interface. From that knowledge, our strategy in modeling the closure terms is to focus our attention on the interface, where we adopt a machine-learning approach for training ANNs to be used in

the *a posteriori* LES to predict all the closure terms in the interfacial region of the domain $\mathcal{D}_I$. Accurate prediction of the SGS closure terms in this region is important to properly model the SGS effects contributing to the increase at high wavenumbers of the interaction energy spectrum. In the bulk of the carrier fluid, $\mathcal{D}_C$, where the droplets have not affected SGS energy content, we adopt a standard SGS stress model for single-phase flow. For the few grid cells containing only droplet fluid (less than 2% of the 128^3 grid), $\boldsymbol{x} \in \mathcal{D}_D$, we don't include any SGS stresses. Figure 3.1 shows for a schematic of the three regions $\mathcal{D}_I$, $\mathcal{D}_C$, and $\mathcal{D}_D$ of the computational domain and how each region is modeled by the MANN LES approach.

3.3.2 Subgrid-scale stress

We begin by modeling $\hat{\tau}_i^{\text{conv}}$ of (3.9) in the carrier fluid, which is equal to the SGS stress closure term for the LES of single-phase isotropic turbulence. Because this is already a well studied problem, we will use a standard model to estimate this term.

In the interest of simplicity, we choose to use the Smagorinsky SGS model [54]:

$$\hat{\tau}_i^{\text{conv}} = 2(C_S\Delta)^2 \frac{\partial}{\partial x_j}\left(\sqrt{2\bar{S}_{ij}\bar{S}_{ij}}\bar{S}_{ij}\right) \quad \text{for } \boldsymbol{x} \in \mathcal{D}_C, \tag{3.13}$$

where C_S is a constant that roughly scales with the filter size Δ and

$$\bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) \tag{3.14}$$

is the rate-of-strain tensor of the filtered velocity. We use $C_S = 0.17$, which we confirmed to be a reasonable choice for our filter size in a droplet-free test case with the same Reynolds number and grid size. Although the constant-coefficient Smagorinsky model is often outperformed by more sophisticated SGS stress models, it is still frequently used as a base case against which to compare new methods such as the ANN-powered single-phase LES models of [6, 52]. Considering that our Reynolds number and filter size are relatively small, the Smagorinsky model will suffice for predicting the SGS stress in the carrier fluid.

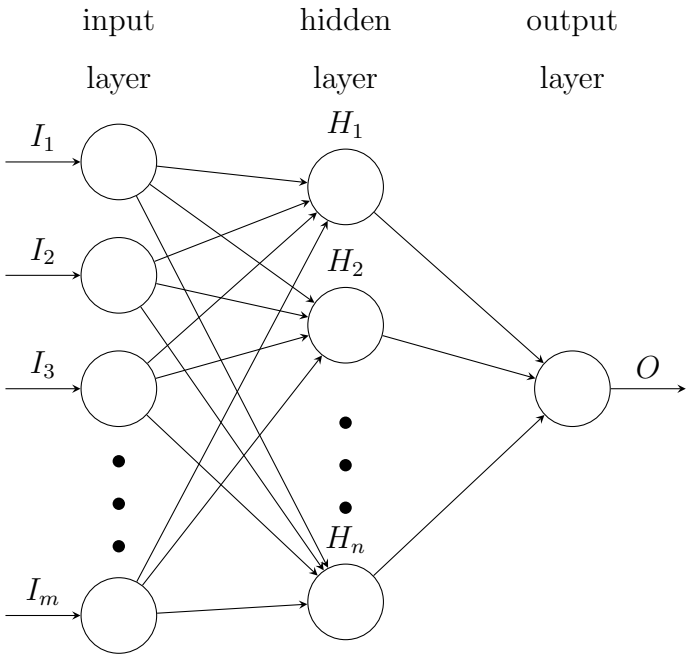

Figure 3.2: Schematic of the structure of a densely connected ANN with a single hidden layer.

3.3.3 Artificial neural networks

Introduction

The basic idea behind ANNs is to mimic the structure of the brain's interconnected neurons to solve problems in a way similar to how a brain might [39]. We use ANNs with densely connected layers, whose basic structure is shown in Fig. 3.2; see, e.g., [19, Chapter 10] for an overview of dense ANNs. The ANN consists of an input layer, a number of hidden layers, and an output layer. The inputs $I_1, \ldots, I_m$ to the input layer are called features, and the goal of the ANN is to predict the proper output given a set of features, which in our case will be values of the flow variables at various points. The features get passed to the neurons $H_1, \ldots, H_n$ in the first hidden layer. The neurons in any hidden layer take as inputs some values $x_1, \ldots, x_n$ fed to them by the neurons of the previous layer. The output of a given

neuron H_i in a hidden layer is $f(b+w_j x_j)$, where the w_j are weight coefficients, b is a constant called the bias, and f is an activation function. Finally the output layer produces the final answer $O = f(b + w_j x_j)$, where x_j are the outputs of the final hidden layer. Through a process called backpropagation, when the ANN gets trained, gradients of the output with respect to the training variables (the weights and bias at every neuron) are calculated, allowing the network to be optimized in producing the correct output, which is called the label associated with a given set of features. We use regression ANNs because we use them to predict numerical values, so the activation function at the output layer is $f(x) = x$.

In setting up the ANN, we have a number of choices regarding so-called hyperparameters that govern the structure of the network and how it is trained. One of these parameters is the number of hidden layers and how many neurons are contained in each hidden layer, called the layer width (n in Fig. 3.2). Some problems such as image recognition benefit from deep learning, in which sometimes hundreds of hidden layers are used. Training such deep networks takes a lot of time and training data, but luckily "shallower" networks often suffice. We discuss our choice of network architecture in greater detail below.

Two more hyperparameters are the optimization algorithm and the activation function f in the hidden layers. For the former, we choose the Adam optimizer [30], and for the latter, we use the rectified linear unit (ReLU)

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.15}$$

The Adam optimizer and the ReLU activation function have been successfully used before for LES applications [6] and ReLU can be quickly computed, which is an important property when training ANNs. Both Adam and ReLU are widely used and fairly standard choices. Lastly, we need to provide a metric over which to optimize the ANN, i.e., a measure of how well predictions match labels, which is traditionally called the loss. Our loss function will be the mean squared error, i.e., the ℓ^2 error, of all training labels and corresponding ANN predictions, which is also used in [6].

	τ_1^{conv}	τ_2^{conv}	τ_3^{conv}	τ_1^{accel}	τ_2^{accel}	τ_3^{accel}	τ_{1j}^{visc}	τ_{2j}^{visc}	τ_{3j}^{visc}	τ_1^{surf}	τ_2^{surf}	τ_3^{surf}	τ^{adv}
$\bar{u}_1$	✓	✓	✓	✓			✓	✓	✓				✓
$\bar{u}_2$	✓	✓	✓		✓		✓	✓	✓				✓
$\bar{u}_3$	✓	✓	✓			✓	✓	✓	✓				✓
$\bar{\chi}$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 3.4: Variables used as features for training the ANN for each closure term τ.

The loss function is also used to evaluate performance of the ANN during and after training. Before training, we set aside 20% of the training dataset as a test set used to test the performance of the ANN after training. We want the loss of the test set to be as close to 0 as possible. Of the remaining training set, another 20% is taken to be a validation set. During training, each time the entire training set is fed through the ANN and used to optimize the training variables, the loss of the validation set is computed. One complete pass of the training set is called an epoch. After some number of epochs, the validation loss will hopefully stop changing appreciably, at which point we consider the ANN to have converged.

Training of the ANNs

For training the neural network, we use the DNS data of droplet-laden isotropic turbulence of case C of [12] at times $t = 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3$. We purposely trained the ANNs only up to time $t = 3$, such that subsequent times could be used for testing *a posteriori* LES and testing the ability of the trained networks to predict times for $t > 3$. After filtering the DNS velocity and VoF fields (using the uniform filter of width $8\,\Delta x_{\text{DNS}}$ as described in Section 3.2.2), we compute the LES closure terms, τ_i^{conv}, τ_i^{accel}, τ_{ij}^{visc}, τ_i^{surf}, and τ^{adv} as defined in (3.5) on the set $\mathcal{D}_I$ defined in (3.10), using finite-difference approximations on the 1024^3 grid. The computed values of the closure terms are then used as our training labels of the ANN. For features, at a given point, we include all neighboring points (including those at the corners) for some combination of $\bar{u}_i$, $i = 1, 2, 3$, and $\bar{\chi}$, as specified in Table 3.4, giving

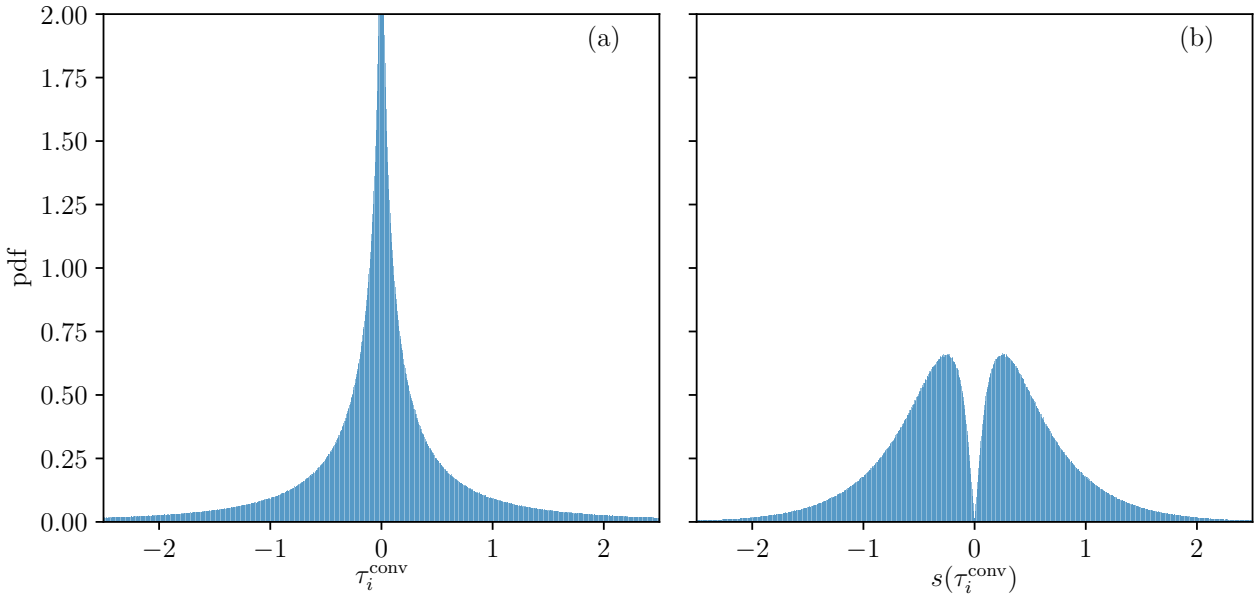

Figure 3.3: Probability density functions (p.d.f.'s) of training data for τ_i^{conv} (normalized by its standard deviation) (a) before, and (b) after scaling the data using Eq. (3.16).

us a maximum total of 108 features. Note that here "neighboring" refers to the 128^3 grid because that will be the only information available during the LES performed on the 128^3 grid. This means that the labels are computed using the fine grid while the features are computed using the coarse grid, such that the ANN's job will be to interpolate the SGS closure terms with a limited amount of information. The variables that we include in the features are the variables from which the closure terms depend on, i.e., $\bar{u}_i$ and $\bar{\chi}$, except for τ_i^{accel}, which depends on $\partial\bar{u}_i/\partial t$ instead of $\bar{u}_i$.

Finally, we process the data in an effort to improve convergence of the ANNs. First, we normalize the labels so that they have a standard deviation of one. Then, we further rescale the labels as follows:

$$s(\tau) = \text{sign}(\tau)\sqrt{|\tau|} \quad \text{for all closure terms } \tau, \tag{3.16}$$

such that their distribution is closer to Gaussian. The definition of $s(\tau)$ is also called a signed square root. Before the rescaling operation of Eq. (3.16), the distributions of the closure

	τ_i^{conv}	τ_i^{accel}	τ_{ij}^{visc}	τ_i^{surf}	τ^{adv}
inputs	108	54	108	27	108
hidden layers	4	4	4	4	4
layer width	128	128	128	256	128
outputs	1	1	3	1	1

Table 3.5: ANN parameters for each closure term.

terms are heavily skewed toward 0 and also have fairly long tails, as seen in Fig. 3.3(a) for τ_i^{conv}. This original distribution is undesirable because it means that, during the training phase, the ANN will see many labels very close to 0 and not as many at the extremes, which is important for an accurate prediction of the ANN. After our rescaling operation, as seen in Fig. 3.3(b), the p.d.f. of τ_i^{conv} no longer has the high peak at 0 nor the very long tails. Although even the rescaled distribution is not perfectly Gaussian, due to the dip at zero, we found that this procedure was sufficient to drastically improve the prediction results of our ANNs. Similar trends for the p.d.f.'s of all other closure terms to the ones shown for τ_i^{conv} in Fig. 3.3 were found and herein not shown.

Table 3.5 shows the parameters of our ANNs. Besides the number of inputs, which is determined by the number of features, and the number of outputs, determined by the dimensionality of the closure term, we need to choose the number of hidden layers and the number of neurons per hidden layer, which is called the layer width, not to be confused with the filter width. We use uniform layer widths for each hidden layer, and we found that 4 layers were sufficient for each closure term. We do however use fairly wide layers with 128 parameters with the exception of τ_i^{surf}, for which we found that 256 gave better results. We suspect that this is due to the added complexity of τ_i^{surf}, which depends on a second derivative due to the curvature, rather than a first derivative as with the other terms. And since Table 3.3 shows that $\hat{\tau}_1^{\text{surf}}$ is on average almost 70% of the total sum of the SGS closure terms P_1, it is especially important that we predict τ_i^{surf} accurately. At the input layer, we

	loss	correlation
τ_i^{conv}	9.87×10^{-2}	0.903
τ_i^{accel}	1.80×10^{-2}	0.980
τ_{ij}^{visc}	6.59×10^{-2}	0.940
τ_i^{surf}	1.54×10^{-1}	0.877
τ^{adv}	5.39×10^{-2}	0.961

Table 3.6: Loss and correlation coefficients of the test sets after training the five closure terms.

also incorporate batch normalization [23] to automatically make sure our training features are well conditioned (as we did for the labels manually). Note that we only use a total of 5 ANNs rather than 13 as shown in Table 3.4 because we have just one ANN corresponding to $i = 1$ and then rotate the dimensions of the features to obtain the $i = 2$ and 3 predictions.

We also tried incorporating some more advanced ANN techniques. One of the techniques, that we explored, was adding dropout between layers [22] as a regularization technique to avoid overfitting the ANNs to our training data. During training, dropout temporarily drops connections between neurons, effectively shrinking the layer width by, say, 20% at random, to encourage the ANN from becoming reliant on only a few of the connections between neurons. Another idea that we explored was to use residual networks [20]. They allow the user to more quickly train deeper ANNs by adding the output of an intermediate layer to the output of a deeper layer, bypassing some of the hidden layers. This allows information to propagate more quickly to deeper layers during training. However, the use of dropout and residual layers for deeper ANNs didn't produce any significant reduction in test-dataset loss; thus, we decided not to use them.

We train our regression ANNs using the Keras front end to TensorFlow [7, 1]. TensorFlow is a popular software library developed by Google for a wide range of machine-learning applications, particularly neural networks. Keras is a high-level Python library built on

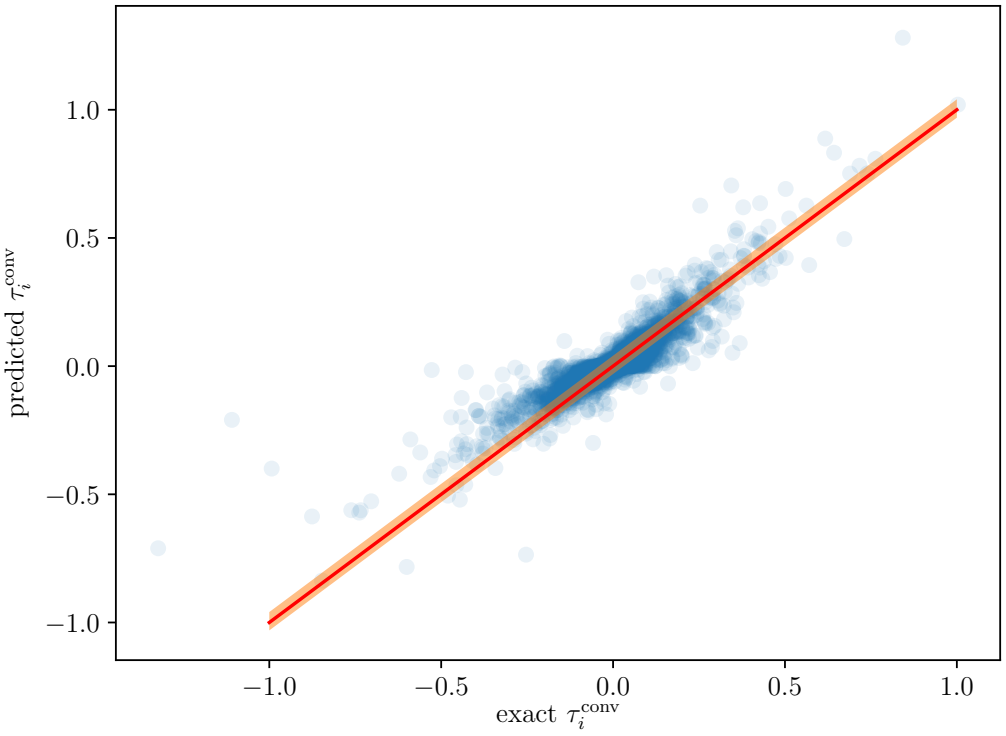

Figure 3.4: Exact values of τ_i^{conv} versus the predicted values of τ_i^{conv} by the ANN. The shaded orange region shows one standard deviation above and below the mean error in the ANN predictions.

TensorFlow. However, because Keras is a Python framework, calling it to make predictions from our Fortran code would create a bottleneck for the computation. To avoid this, we use the frugally-deep library [21], which allows us to convert our trained Keras ANNs into a format such that we can use C++ to make the predictions, and, in turn, call this C++ library by our Fortran code.

As explained in Section 3.3.3, to confirm the accuracy of our trained ANNs, we set aside a random 20% of the training data to be used as a test dataset. Keeping a test set hidden from the ANN during training is important to avoid overfitting. We need to evaluate the performance of the ANN on data it hasn't seen before. Table 3.6 shows the losses and the Pearson correlation coefficients between the test set labels and predictions for each of the 5

closure terms as a measure of how well our predictions match the true values of the closure terms. Losses are close to 0 while correlations are close to 1, meaning that our training was successful. Furthermore, due to the large size of our training dataset, we only had to train our ANNs over 4 epochs. Note that these convergence statistics were computed immediately after training and before applying the inverse of the scaling function in (3.16). Fig. 3.4 shows a scatter plot of the exact values of τ_i^{conv} in the test set compared to the values predicted by the ANN. Here, the inverse of the scaling function in (3.16) has been applied. Perfectly predicted values lie along the red line. The error between the exact values and ANN predictions has a standard deviation of 3.58×10^{-2} and mean 4.96×10^{-3}. The region within one standard deviation of the mean error is represented by the shaded orange region in Fig. 3.4. Due to the high concentration of points near zero, only the outliers are clearly visible, but about 85% of the points lie in the shaded orange region.

3.3.4 Numerical method

For the *a posteriori* LES, we solve the filtered Navier-Stokes equations (3.4a) and (3.6) using the pressure-correction method for two-fluid incompressible flows, FastP* [11], coupled with the VoF method [4] while modeling the SGS convective term ($\hat{\tau}_i^{\text{conv}}$ for $\boldsymbol{x} \in \mathcal{D}_C$) with the Smagorinsky model, Eq. (3.13) [54], and the other SGS terms due to the droplets ($\hat{\tau}_i^{\text{conv}}$, $\hat{\tau}_i^{\text{accel}}$, $\hat{\tau}_i^{\text{visc}}$, $\hat{\tau}_i^{\text{surf}}$, and τ^{adv} for $\boldsymbol{x} \in \mathcal{D}_I$) using our ANN's closure-term predictions. We call this methodology mixed artificial neural network (MANN) LES because of combining the use of standard Smagorinsky, as for single-phase flow for the convective term in the carrier phase, and of ANNs for the closure terms at the interface.

In a pressure-correction method, at each time step, an approximate velocity, u_i^*, is computed using the Navier–Stokes momentum equation without the pressure-gradient term. The approximate velocity field is then used to compute the pressure by solving a Poisson equation for pressure, which is, then, used to compute the divergence-free velocity field at the next time step. Next, we explain how this works with FastP* with our filtered equations ((3.4a) and (3.6)) for LES.

Let $(\,\cdot\,)^n$ denote the value of a variable at time step n. At the beginning of time step n, the closure term τ^{adv} is predicted using the ANN with input $\bar{\chi}^n$. To ensure that mass is conserved, we then compute the mean of τ^{adv} across the entire domain and subtract it from τ^{adv} so that it has zero mean. Using (3.4c), $\bar{\chi}^n$ is then updated to $\bar{\chi}^{n+1}$ and used to compute $\bar{\rho}^{n+1}$ and $\bar{\mu}^{n+1}$ according to (1.5). Next, we compute the momentum-equation closure terms τ_i^{conv}, τ_i^{accel}, τ_i^{visc}, and τ_i^{surf} using our ANNs as described in Section 3.3.3 and, also, compute

$$R_i^n = -\bar{u}_j^n \frac{\partial \bar{u}_i^n}{\partial x_j} + \frac{1}{\bar{\rho}^{n+1}} \left(\frac{1}{\text{Re}} \frac{\partial}{\partial x_j} \left(\bar{\mu}^{n+1} \left(\frac{\partial \bar{u}_i^n}{\partial x_j} + \frac{\partial \bar{u}_j^n}{\partial x_i} \right) \right) + \frac{1}{\text{We}} f_\sigma(\bar{\chi}^{n+1})_i \right), \tag{3.17}$$

which is equal to the right side of (3.6) without the pressure-gradient or closure terms. Then, we use the first column of equalities in (3.7) to find $\hat{\tau}_i^{\text{conv}}$, $\hat{\tau}_i^{\text{accel}}$, $\hat{\tau}_i^{\text{visc}}$, and $\hat{\tau}_i^{\text{surf}}$, sum them to get P_i^n as defined in (3.12), and compute $R_i^n + P_i^n$, which is an approximation to $\partial \bar{u}_i / \partial t$.

Before computing the approximate velocity u_i^*, we limit P_i^n to improve the performance of our ANNs' predictions and to avoid the accumulation of errors. This limiter keeps P_i^n from becoming too large; otherwise errors will grow unboundedly and, eventually, cause the numerical solution to overflow. This is due to the feedback loop created by the solver and the ANNs. When the ANNs introduce any error, it will affect the velocity, which, then, is fed back to the ANNs, thus, producing even less accurate closure terms. The term τ_i^{accel} is especially sensitive to this effect because it depends on the time derivative of velocity, which will blow up more rapidly than the velocity itself.

The limiter methodology begins by computing what the approximate velocity would be with $P_i^n = 0$, that is, if there were no contribution from the closure terms. This is done by using the second-order Adams–Bashforth method as

$$U_i^* = \bar{u}_i^n + 1.5 R_i^n \, \Delta t_{\text{LES}} \, - 0.5 (R_i^{n-1} + P_i^{n-1}) \, \Delta t_{\text{LES}}. \tag{3.18}$$

The limiter, then, uses U_i^* as a guideline for how large the approximate velocity u_i^* should be. Once P_i^n has been updated by the limiter, we compute the approximate velocity u_i^* as

$$u_i^* = \bar{u}_i^n + 1.5 (R_i^n + P_i^n) \, \Delta t_{\text{LES}} \, - 0.5 (R_i^{n-1} + P_i^{n-1}) \, \Delta t_{\text{LES}}. \tag{3.19}$$

The limiter enforces that

$$0 \le u_i^* \le (1+L)U_i^* \quad \text{if } U_i^* > 0, \tag{3.20a}$$

$$(1+L)U_i^* \le u_i^* \le 0 \quad \text{otherwise.} \tag{3.20b}$$

In other words, u_i^* must be the same sign as U_i^* and cannot cause $|U_i^*|$ to increase by more than a factor of $1+L$. Plugging (3.19) into (3.20), and solving for P_i^n gives us

$$\frac{-\bar{u}_i^n + 0.5\,\Delta t_{\text{LES}}\,(R_i^{n-1} + P_i^{n-1})}{1.5\,\Delta t_{\text{LES}}} - R_i^n \le P_i^n \le \frac{LU_i^*}{1.5\,\Delta t_{\text{LES}}} \quad \text{if } U_i^* > 0, \tag{3.21a}$$

$$\frac{LU_i^*}{1.5\,\Delta t_{\text{LES}}} \le P_i^n \le \frac{-\bar{u}_i^n + 0.5\,\Delta t_{\text{LES}}\,(R_i^{n-1} + P_i^{n-1})}{1.5\,\Delta t_{\text{LES}}} - R_i^n \quad \text{otherwise.} \tag{3.21b}$$

In our limiting procedure, these constraints are enforced by setting

$$P_i^n \leftarrow \begin{cases} \max\left\{\min\left\{P_i^n, \dfrac{LU_i^*}{1.5\,\Delta t_{\text{LES}}}\right\}, \dfrac{-\bar{u}_i^n + 0.5\,\Delta t_{\text{LES}}\,(R_i^{n-1} + P_i^{n-1})}{1.5\,\Delta t_{\text{LES}}} - R_i^n\right\} & \text{if } U_i^* > 0, \\ \min\left\{\max\left\{P_i^n, \dfrac{LU_i^*}{1.5\,\Delta t_{\text{LES}}}\right\}, \dfrac{-\bar{u}_i^n + 0.5\,\Delta t_{\text{LES}}\,(R_i^{n-1} + P_i^{n-1})}{1.5\,\Delta t_{\text{LES}}} - R_i^n\right\} & \text{otherwise.} \end{cases} \tag{3.22}$$

Note that we determined a suitable L *a posteriori* and found that $L = 5 \times 10^{-4}$ struck the balance between keeping errors from growing while not restricting the ANNs too much. As we will see in Section 3.3.5, using the Smagorinsky LES model without modeling any of the interfacial closure terms, the filtered-velocity TKE will decay too slowly, which suggests that on average the ANN predictions P_i should cause $|u_i^*| < |U_i^*|$. However, this isn't the case everywhere in the flow field, so in Fig. 3.5 we show the distributions of the exact values of P_1 for which the limiter is accurate (at $t = 3$ for case C). Points at which $U_1^* > 0$ are blue and those at which $U_1^* \le 0$ orange. The values shown in the p.d.f.'s are the differences between P_1 (before limiting) and the chosen limit (as defined by (3.21)). Fig. 3.5(a) shows the lower bound, enforcing that u_1^* won't change sign, and Fig. 3.5(b) shows the upper bound, enforcing that $|u_1^*|$ won't grow larger than $(1+L)|U_1^*|$. If the limiter is accurate and the exact P_1 does fall between these bounds, then the sign of this difference should be equal to the sign of U_1^*. In Fig. 3.5(a), we see virtually no crossover at 0, indicating that

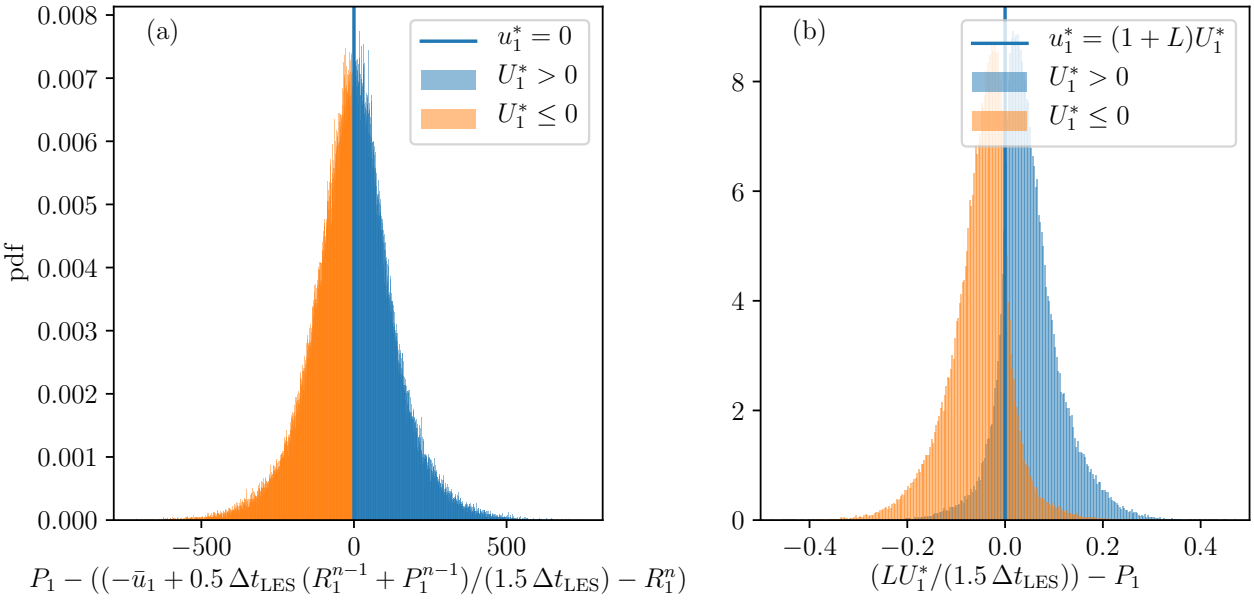

Figure 3.5: Probability density functions (p.d.f.'s) of the differences between P_1 and the bounds given in (3.21) for case C at $t = 3$ with $L = 5 \times 10^{-4}$. (a) The bound enforcing $u_i^* \geq 0$ for $U_i^* > 0$ and $u_i^* \leq 0$ for $U_i^* \leq 0$. (b) The bound enforcing $u_i^* \leq (1 + L)U_i^*$ for $U_i^* > 0$ and $u_i^* \geq (1 + L)U_i^*$ for $U_i^* \leq 0$.

the lower limit is quite conservative. Fig. 3.5(b) shows a larger overlap, meaning that some values of P_1 are greater than the upper bound set by the limiter. However, we found that an aggressive upper bound is necessary for keeping the ANN predictions from accumulating excessive errors. In practice, we found that fewer than 0.5% of the predicted values are cut off by the lower limit while about 15% are cut off by the upper limit, matching what we see in Fig. 3.5. We provide some more insight into finding an appropriate L for different flow parameters in Section 3.3.5. Once P_i^n has been updated with its limited value according to (3.22), the approximate velocity u_i^* can, then, be computed according to (3.19).

The next step in FastP* is to solve for the pressure. Normally a pressure-correction method for two-fluid (gas-liquid or liquid-liquid) flows would need to solve a variable-coefficient Poisson equation for pressure using an iterative method such as multigrid. However, FastP* [11] is able to solve the Poisson equation for pressure by splitting the coefficients $1/\bar{\rho}^{n+1}$ into constant and variable parts using a linear extrapolation to approximate the pressure at the next time step t^{n+1} as $p^* = 2\bar{p}^n - \bar{p}^{n-1}$. Enforcing the divergence-free condition $\partial \bar{u}_j^{n+1}/\partial x_j = 0$, we are left with a constant-coefficient Poisson equation:

$$\frac{\partial^2 \bar{p}^{n+1}}{\partial x_j^2} = \frac{\partial}{\partial x_j}\left(\left(1 - \frac{\rho_c}{\bar{\rho}^{n+1}}\right)\frac{\partial p^*}{\partial x_j}\right) + \frac{\rho_c}{\Delta t_{\text{LES}}}\frac{\partial u_j^*}{\partial x_j}. \tag{3.23}$$

This can be solved for $\bar{p}^{n+1}$ using a fast Poisson solver based on the fast Fourier transform, which is much less expensive than multigrid [11]. After solving (3.23) for $\bar{p}^{n+1}$, the divergence-free velocity $\bar{u}_i^{n+1}$ is computed as

$$\bar{u}_i^{n+1} = u_i^* - \Delta t_{\text{LES}}\left(\frac{1}{\rho_c}\frac{\partial \bar{p}^{n+1}}{\partial x_i} + \left(\frac{1}{\bar{\rho}^{n+1}} - \frac{1}{\rho_c}\right)\frac{\partial p^*}{\partial x_i}\right). \tag{3.24}$$

It should be mentioned that we have verified that FastP* gave nearly identical results to the pressure-correction method using multigrid for solving the standard variable density Poisson equation for pressure also in the case of *a posteriori* LES.

3.3.5 Results

Table 3.7 shows the four methods that we have used for our *a posteriori* analysis. The first method is filtering the results of the fully resolved DNS that were obtained on the 1024^3

method	grid size	carrier SGS model	interface SGS model
filtered DNS	1024^3	none	none
no LES model	128^3	none	none
Smagorinsky LES	128^3	Smagorinsky	none
MANN LES	128^3	Smagorinsky	ANN

Table 3.7: Methods used in Section 3.3.5.

grid, which represent our reference solution that we wish to match, as closely as possible, with the LES. We saved the three-dimensional velocity field of the DNS at time intervals of 0.25, and these velocity fields were then filtered according to Section 3.2.2. The second method uses no LES model on the 128^3 grid, which can be thought of as an underresolved DNS. By using no explicit LES model but a coarser grid, the increased numerical diffusivity will effectively act as an artificial SGS stress model. Then, the third method is the LES using only the classical Smagorinsky SGS stress model (with constant $C_S = 0.17$) across the entire domain and no additional treatment for the droplets, i.e., the interfacial closure terms are set equal to zero. Finally, the fourth method is our mixed LES model (MANN LES) that uses Smagorinsky for the SGS stress in the carrier-fluid part of the domain ($\mathcal{D}_C$), ANN predictions for the SGS closure terms at the interface cells ($\mathcal{D}_I$), and no closure terms inside the droplets ($\mathcal{D}_D$), as explained in Section 3.3.1.

To generate the initial conditions for our LES runs, we start with a DNS run from $t = 0$ to 1, at which point the droplets are introduced in the flow field. Then, we filter the velocity and VoF fields on the 1024^3 grid for $t = 1$, and, then, subsample the filtered results to the coarse 128^3 grid. The resulting fields are then used as the initial conditions for the LES runs for $t = 1$. After that, we perform the LES runs from $t = 1$ to $t = 5$.

Table 1.1 summarizes the parameters of the cases of Dodd & Ferrante [12], including the three that we will use to test our method. All three cases B, C, and D have the same density and viscosity ratios: $\rho_d/\rho_c = \mu_d/\mu_c = 10$; however, the Weber number varies with

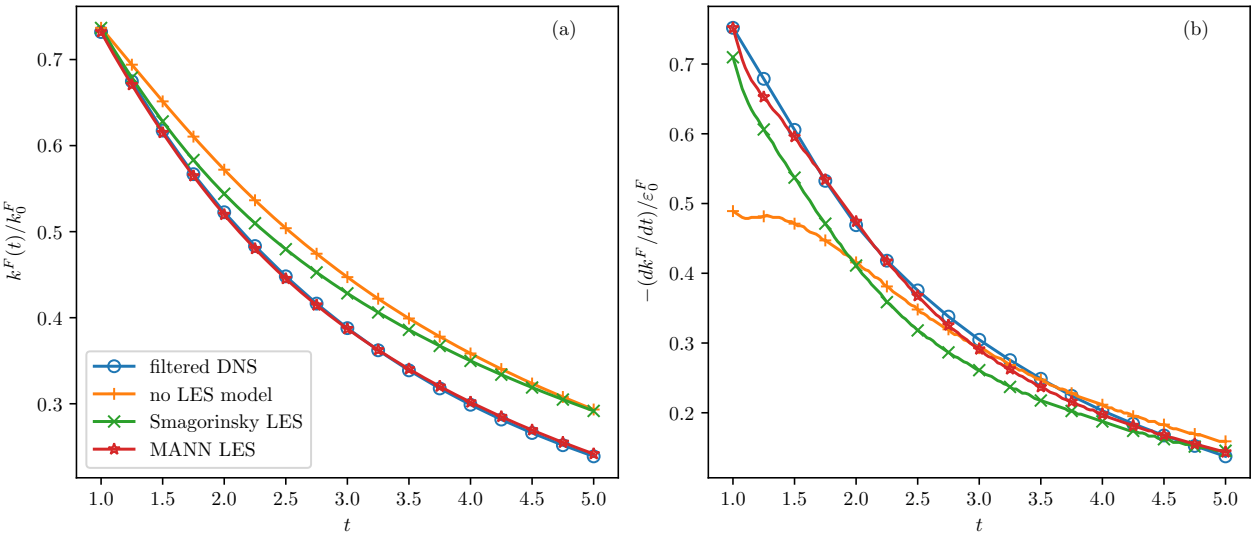

Figure 3.6: (a) Filtered-velocity TKE, $k^F(t) = \langle \frac{1}{2}\bar{u}_i\bar{u}_i \rangle$, normalized by k_0^F; (b) decay rate of filtered-velocity TKE, $-dk^F/dt$, normalized by ε_0^F, for case C.

values of 0.1, 1, and 5, respectively. We begin by testing case C in because this is the set of parameters used in training the ANNs. Then we report the results for cases B and D to test if the ANNs are able to generalize to cases with Weber number ($\mathrm{We}_{\mathrm{rms}} = 0.1$ and 5) different from the one used to train the ANN ($\mathrm{We}_{\mathrm{rms}} = 1$ of case C).

Case C ($\mathrm{We}_{\mathrm{rms}} = 1$)

In Fig. 3.6(a), we show the temporal development of the filtered-velocity TKE

$$k^F(t) = \langle \tfrac{1}{2}\bar{u}_i\bar{u}_i \rangle, \tag{3.25}$$

where $\langle \cdot \rangle$ denotes an average over the entire computational domain at a given time t. Note that $k^F(t)$ has been normalized by the initial filtered-velocity TKE, $k_0^F = k^F(0) = 3.74\times10^{-3}$. Figure 3.6(b) shows the decay rate of filtered-velocity TKE, $-dk^F/dt$, normalized by the initial filtered-velocity dissipation rate, $\varepsilon_0^F = \varepsilon^F(0) = 1.33\times10^{-2}$. Dodd & Ferrante [12] showed that, compared to single-phase isotropic turbulence, droplet-laden isotropic turbulence experiences an increased decay rate of TKE due to increased local dissipation near the interface.

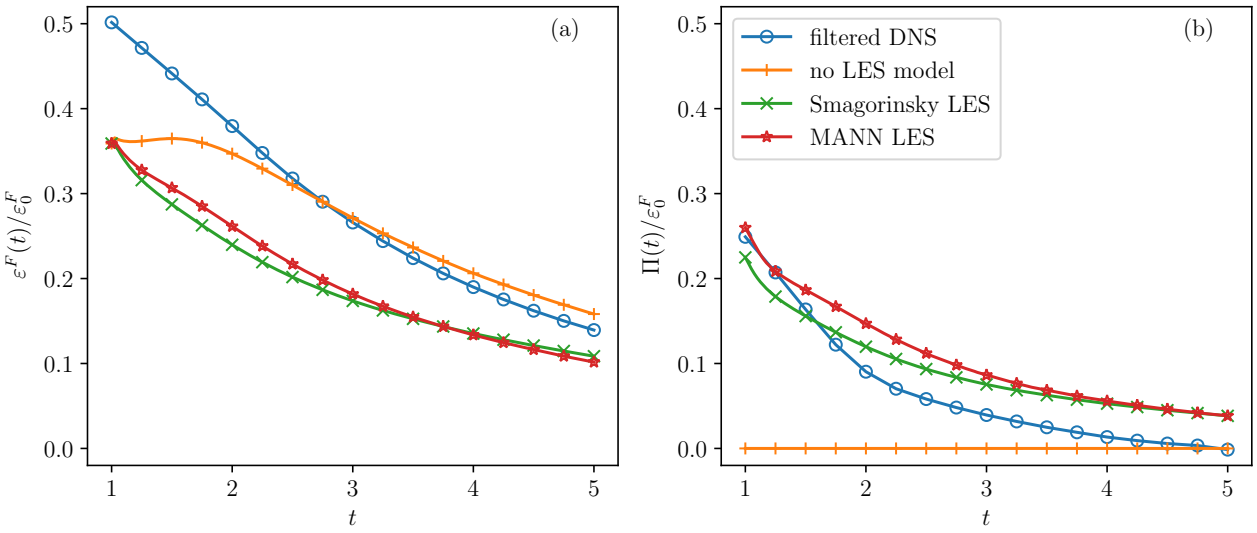

Figure 3.7: (a) Filtered-velocity dissipation rate, $\varepsilon^F(t)$; (b) sink of power due to SGS closure terms, $\Pi(t)$, of the TKE balance equation (3.26) for case C.

In Fig. 2.7, we found that this increased dissipation occurs within $D/4$ of the interface, where D is the initial droplet diameter. This corresponds to the size of one grid cell Δx_{LES} in our LES cases. Because the increase in local dissipation is then unresolved by the 128^3 grid except in cells containing the interface, we expect that the absence of an LES model will result in a diminished decay rate of TKE. Indeed, the TKE of the no-LES-model case in Fig. 3.6(a) does not decay as quickly as that of the filtered DNS, with a relative error at $t = 5$ of 22.8%. This result confirms that an LES model is needed to accurately simulate this flow on the 128^3 grid. The addition of the Smagorinsky SGS stress model increases the decay rate of TKE with respect to that of no model initially (Fig. 3.6(b)); however, this model only works by predicting the SGS eddy viscosity of the single-phase flow, which is a different phenomenon from the increased dissipation at the interface. Figure 3.6(b) shows that by $t = 2$, the decay rate drops below that of no LES model, and by $t = 5$, the Smagorinsky TKE has diverged from the filtered DNS TKE as much as the no-model case did, with a relative error of 22.1% (Fig. 3.6(a)). This result confirms that including the interfacial closure terms

using an accurate model is crucial for capturing the proper near-interface dissipation and subsequent decay of TKE. Figure 3.6(a) shows that our MANN LES model achieves this, matching closely the filtered DNS results until $t = 3$, which is the time period over which the ANNs were trained, after which, it begins to depart only very slightly from the filtered DNS result. By $t = 5$, the relative error of the MANN LES model compared to the filtered DNS is only 1.68%.

To understand why the decay rate of TKE differs between the four methods, we look at the filtered TKE balance equation

$$\frac{dk^F}{dt} = -\varepsilon^F(t) + \Psi_\sigma^F(t) - \Pi(t), \tag{3.26}$$

where

$$\varepsilon^F(t) = \frac{1}{\mathrm{Re}}\left\langle \frac{2\bar{\mu}}{\bar{\rho}}\bar{S}_{ij}\bar{S}_{ij} \right\rangle, \tag{3.27a}$$

$$\Psi_\sigma^F(t) = \frac{1}{\mathrm{We}}\left\langle \frac{1}{\bar{\rho}}\bar{u}_j f_\sigma(\bar{\chi})_j \right\rangle, \tag{3.27b}$$

$$\Pi(t) = -\langle \bar{u}_j P_j \rangle. \tag{3.27c}$$

These three terms represent the filtered-velocity dissipation rate (with $\bar{S}_{ij}$ defined in (3.14)), power due to filtered surface tension, and sink of power due to the SGS closure terms, respectively. Equation (3.26) can be derived by multiplying the momentum equation (3.6) by $\bar{u}_i$, and, then, by averaging the resulting equation over the entire computational domain. Figure 3.7 shows $\varepsilon^F(t)$ and $\Pi(t)$, normalized by ε_0^F. We found that $\Psi_\sigma^F(t)$ is two orders of magnitude smaller than these terms, so it has a relatively small effect on the decay rate of TKE, and is, thus, not shown. At the beginning of the simulation of the four cases, the filtered DNS (1024^3 grid) has the greatest filtered-velocitydissipation rate, while the three LES methods (128^3 grid) begin with a dissipation rate that is about 30% lower. This result is expected because the LES grid (128^3), by not resolving the small scales of motion below the filter size, is not able to accurately capture the correct velocity gradient tensor, S_{ij}, and, thus, the dissipation ε^F. The LES needs the SGS terms, P_i, which produce $\Pi(t)$ in the

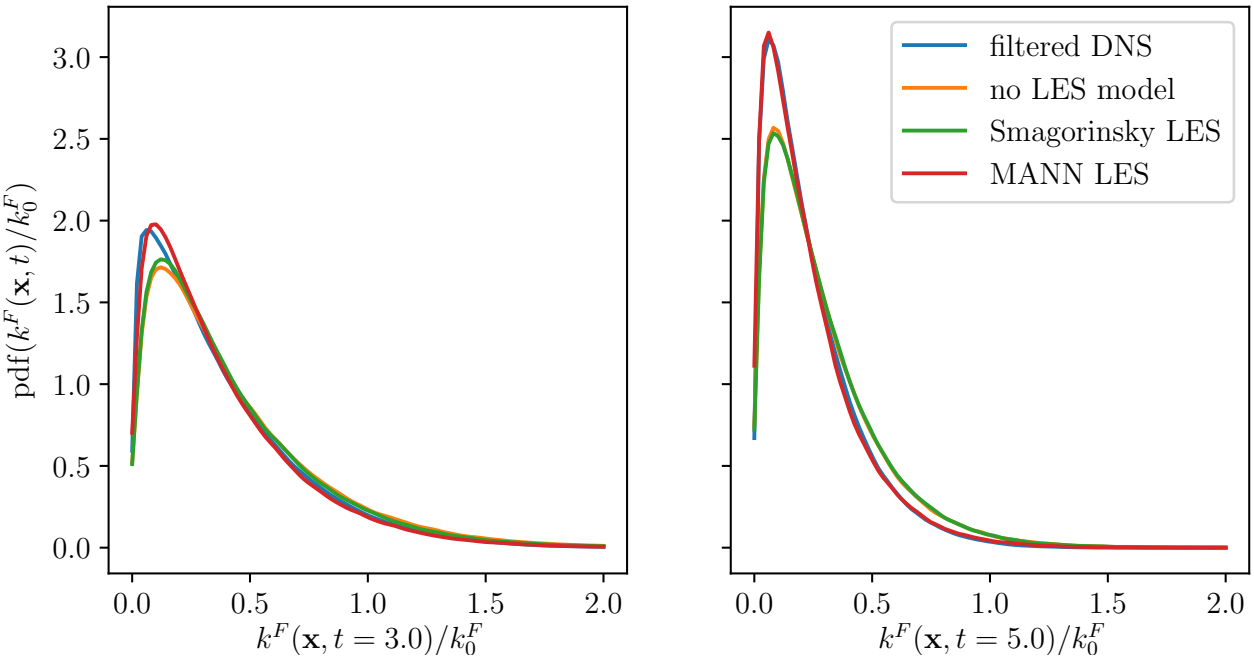

Figure 3.8: Probably density function (p.d.f.) of filtered-velocity kinetic energy, $k^F(\boldsymbol{x},t) = \frac{1}{2}\bar{u}_i\bar{u}_i$, throughout the domain at $t = 3$ and 5, normalized by k_0^F, for case C.

TKE equation in order to add the dissipation coming from the unresolved SGS. Also, the coarser grid of the LES methods introduces more numerical diffusivity into the simulation than that of the filtered DNS. As a result, in the case of no LES model, $\varepsilon^F(t)$ remains near its initial value until $t = 1.8$, as shown in Fig. 3.7(a). The analysis of Dairay *et al.* [8] explains the same phenomenon for single-phase LES by showing that, when using no LES model, energy accumulates at small scales near the beginning of the simulation due to the lack of a model for SGS dissipation. Also, in our two-fluid LES with no LES model, $\Pi(t)$ is zero because no SGS closure terms are included ($P_i = 0$), as shown in Fig. 3.7(b). The overall result is that, at times $1 \leq t \leq 2$, for no LES model, TKE decays at a slower rate than that of the filtered DNS, as shown by the plot of dk^F/dt in Fig. 3.6(b). The dissipation rate for Smagorinsky and MANN LES at $t = 1$ is the same as that for no LES model, but $\Pi(t)$ now begins at about 70% of $\varepsilon^F(t)$ rather than remaining zero. And since MANN LES

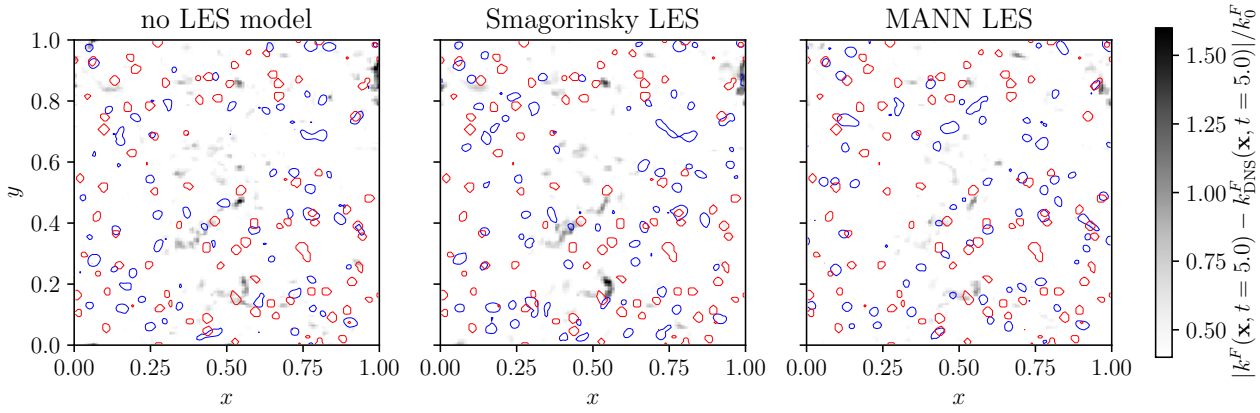

Figure 3.9: Absolute error in local filtered-velocity kinetic energy, normalized by k_0^F, for the three LES cases in comparison to the filtered DNS results at $t = 5$ and $z = 0.5$, for case C. The droplets are also shown as VoF contours ($\bar{\chi} = 0.5$) with fully resolved DNS results in red and the LES cases in blue.

models the interfacial SGS closure terms that Smagorinsky LES neglects, its $\Pi(t)$ is further elevated compared to Smagorinsky LES. Because Smagorinsky and MANN LES model SGS dissipation, they do not experience the same energy accumulation and overcompensation of dissipation that occurs with no LES model, so their $\varepsilon^F(t)$ decay immediately. The overall result is that MANN LES most closely matches the decay rate of TKE for filtered DNS at $t = 1$. As the simulation progresses, $\Pi(t)$ for MANN LES remains higher than that for Smagorinsky LES, and contributions from the interfacial SGS closure terms also increase $\varepsilon^F(t)$ slightly compared to Smagorinsky LES, so MANN LES continues to match the decay rate of TKE for filtered DNS while the TKE decay rate of Smagorinsky LES decreases more quickly. Figure 3.6(b) shows that by $t = 5$ the decay rate of $k^F(t)$ for all three LES methods converges to that of the filtered DNS, but because the decay rates for no LES model and Smagorinsky LES were not accurately captured earlier in the simulation, around times $1 \leq t \leq 3$, $k^F(t)$ remains about 20% too high by $t = 5$. On the other hand, MANN LES matches dk^F/dt throughout the simulation and therefore also $k^F(t)$.

	DNS	no LES model	Smagorinsky LES	MANN LES
local TKE error		0.174	0.169 (-2.65%)	0.150 (-14.0%)
num. droplets, $t = 4$	2933	2704 (-7.81%)	2505 (-14.6%)	2624 (-10.5%)
num. droplets, $t = 5$	2704	2480 (-8.28%)	2218 (-18.0%)	2252 (-16.7%)

Table 3.8: Average error of the local filtered-velocity TKE, $\langle |k^F(\boldsymbol{x},t) - k^F_{\mathrm{DNS}}(\boldsymbol{x},t)|/k^F_0 \rangle$, at $t = 5$, and number of droplets at $t = 4$ and 5, for case C. In parentheses is the percentage difference relative to the no-LES-model case (for the local TKE error) or percentage difference relative to the filtered DNS case (for the number of droplets).

Figure 3.8 shows the p.d.f.'s of the local filtered-velocity turbulence kinetic energy for all the grid points (128^3) of the computational domain at $t = 3$ and 5:

$$k^F(\boldsymbol{x},t) = \tfrac{1}{2}\bar{u}_i\bar{u}_i. \tag{3.28}$$

This is the local filtered-velocity TKE prior to the averaging $\langle\,\cdot\,\rangle$ operation, $k^F(t) = \langle k^F(\boldsymbol{x},t)\rangle$, hence, the dependence on $\boldsymbol{x}$. We can see that the no-LES-model and Smagorinsky cases have similar distributions of kinetic energy at both times, but they do not match the distribution of the filtered DNS, which has a higher peak around $k^F(\boldsymbol{x},t)/k^F_0 = 0.2$. On the other hand, the MANN LES model matches the distribution given by the filtered DNS. This shows that, not only does the MANN LES method produce the correct average TKE as shown in Fig. 3.6(a), but it also produces the correct distribution of TKE throughout the domain.

Figure 3.9 shows $|k^F(\boldsymbol{x},t) - k^F_{\mathrm{DNS}}(\boldsymbol{x},t)|/k^F_0$, the difference between local filtered-velocity TKE of the three LES cases and that of the filtered DNS case at a slice ($z = 0.5$) of the domain at the end of the simulation for $t = 5$. Although all three LES cases still contain regions with nonzero error, in the rightmost plot of Fig. 3.9, we see that the local filtered-velocity TKE of the MANN LES model matches the filtered DNS results better than the other two cases. In particular, the high error in the regions near $(x, y) = (0.55, 0.2)$, $(0.4, 0.4)$, $(0.1, 0.9)$, and $(1.0, 0.8)$ of the first two LES methods has been reduced by the MANN LES

case. To quantify this improvement, we compute the average error in local filtered-velocity TKE, $\langle |k^F(\boldsymbol{x},t) - k^F_{\mathrm{DNS}}(\boldsymbol{x},t)|/k^F_0 \rangle$, at $t = 5$, which is given in Table 3.8. While using the Smagorinsky model alone reduces this error by 2.65% compared to the no-LES-model case, the MANN LES reduces it by 14.0%, showing that the addition of the ANNs has a substantial effect, especially considering that the SGS closure terms predicted with the ANNs are nonzero only in about 10% of the domain cells at any given time. Note that the reduction in error for the average local TKE error is not as significant as for the total filtered-velocity TKE because the average local TKE error is an average of the error in TKE at all points. Even though MANN LES achieves the proper distribution of TKE throughout the domain, at any individual point in the domain, the error in TKE may be significant because the flow has not evolved in the same way in the coarse-grid cases as it did in the DNS.

Figure 3.9 shows also the droplets (contours of VoF function $\bar{\chi} = 0.5$) in blue for the LES cases and in red for the fully resolved DNS. The VoF field is not accurately advected in all three LES cases, and we can also see that in these cases the droplets are fewer but larger than in the DNS. This is confirmed by Table 3.8, which shows the number of droplets for all four cases at $t = 4$ and 5. The VoF method of [4] handles coalescence implicitly via the VoF function. When two droplets move toward one another and eventually occupy the same grid cell, they are considered to have coalesced. Of course, on the coarser 128^3 grid, droplets are no longer well resolved and can join more easily, so by $t = 5$, all three LES cases have up to 18% fewer droplets than the filtered DNS, which explains why droplet tracking is not very accurate even though the distribution of TKE for MANN LES is.

Lastly, Figs. 3.10(a,c,e) show the wavelet spectra of the filtered-velocity TKE, $\widetilde{E}^F(k_m)$, for the methods of Table 3.7 at $t = 1.5$, 3, and 5, respectively, where k_m is the wavenumber corresponding to the wavelet of scale m. If the grid size is N, then $m = 1, \ldots, \log_2 N$, and $k_m = 2\pi/(2^m\,\Delta x)$. To compute $\widetilde{E}^F(k_m)$, we first apply the discrete wavelet transform to $\bar{u}_i$, which gives us the associated wavelet coefficients $\bar{w}_i^{(m,q)}[\boldsymbol{l}]$ (see (2.4)). Remember that q is an index that represents the seven possible directions of dilation in the wavelet representation of $\bar{u}_i$, and $\boldsymbol{l}$ is an index in the set $\mathcal{D}^{(m)} = \{1, \ldots, N/2^m\}^3$, the locations at which the scale-m

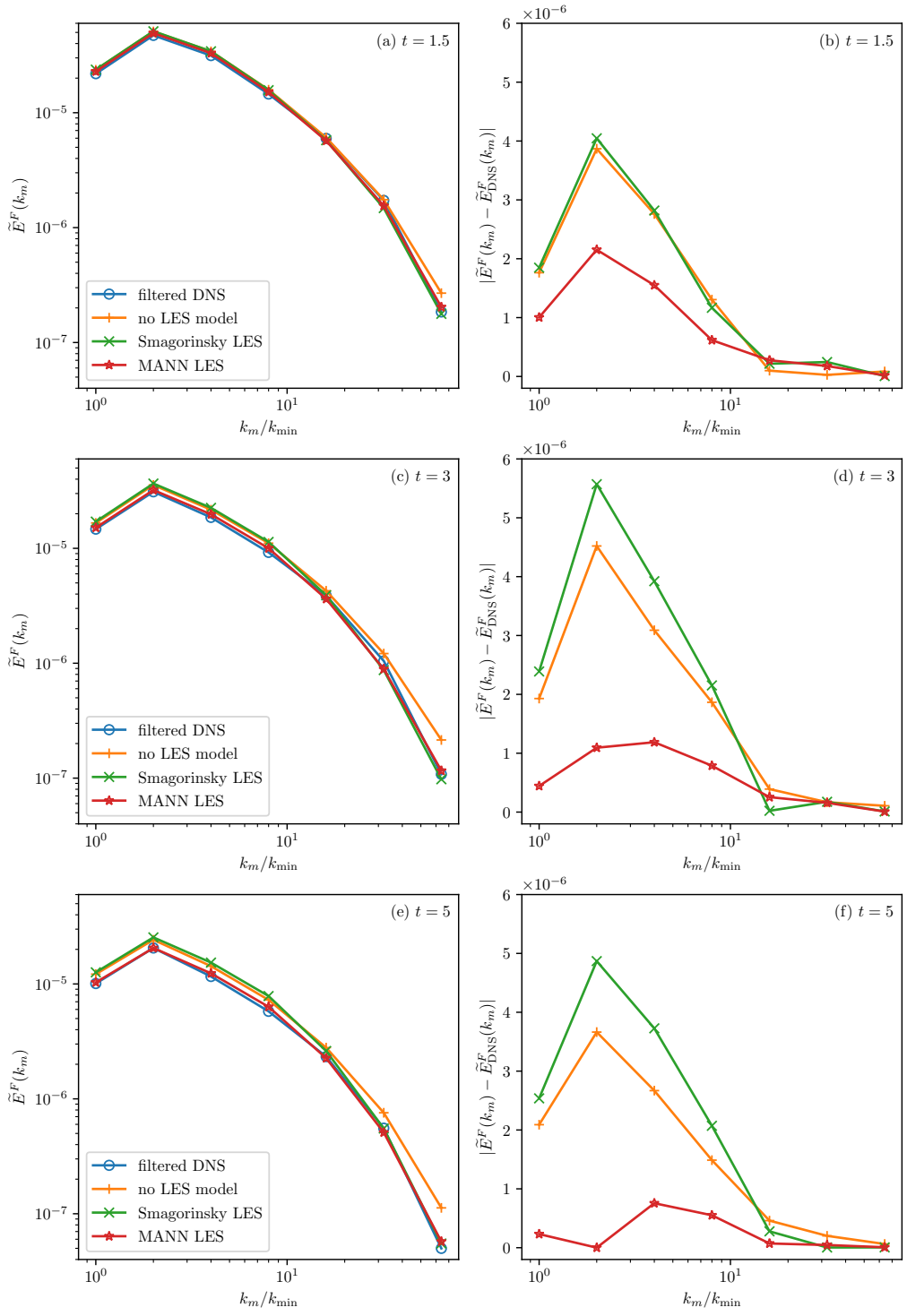

Figure 3.10: (a,c,e) Wavelet turbulence kinetic energy spectra, and (b,d,f) absolute errors in wavelet TKE spectra compared to the filtered DNS for case C at $t = 1.5$, 3, and 5.

wavelet coefficients are defined. Then, $\widetilde{E}^F(k_m)$ is defined as

$$\widetilde{E}^F(k_m) = C_m \sum_{i=1}^{3} \left\langle \frac{1}{2} \sum_{q=1}^{7} \bar{w}_i^{(m,q)}[\boldsymbol{l}]^2 \right\rangle_{\boldsymbol{l}\in\mathcal{D}^{(m)}}, \tag{3.29}$$

with C_m defined as in (2.8). For details on the wavelet spectra and the domain decompositions for droplet-laden flows, see Chapter 2. Since the droplet diameter is only $4\,\Delta x_{\mathrm{LES}}$ for the 128^3 grid of the LES cases, decomposing the wavelet spectra as we did for analyzing DNS data in Chapter 2, for which the droplet diameter was $32\,\Delta x_{\mathrm{DNS}}$, would result in carrier wavelet spectra $\widetilde{E}_C^F(k_m)$ defined only at one wavenumber, $k_m/k_{\min} = 64$. Thus, we only show the total wavelet spectra $\widetilde{E}^F(k_m)$ computed over the entire computational domain. Figures 3.10(b,d,f) also show the absolute error in the wavelet spectra compared to the filtered DNS, $|\widetilde{E}^F(k_m) - \widetilde{E}^F_{\mathrm{DNS}}(k_m)|$. Note that we have omitted wavenumbers $k_m/k_{\min} = 128$, 256, and 512 from the filtered DNS plot in Fig. 3.10 since the maximum LES wavenumber is 64; thus, that is the maximum wavenumber that we can compare the LES results against the filtered DNS spectra. Figure 3.10 shows that, as time progresses, $\widetilde{E}^F(k_m)$ of filtered DNS decays at all wavenumbers, which the LES methods match with varying degrees of success. At $t = 1.5$, in Fig. 3.10(a), all three LES methods match the spectrum of filtered DNS fairly well, but for the case of no-LES-model the energy at $k_m/k_{\min} = 64$ exceeds that of the filtered DNS. This confirms that without accounting for any SGS stresses, and, thus, SGS dissipation, energy accumulates at the highest wavenumber. Furthermore, Fig. 3.10(b) shows that MANN LES has the lowest error of the three LES methods at $t = 1.5$. Advancing in time to $t = 3$ (Fig. 3.10(c)) and $t = 5$ (Fig. 3.10(e)), the no-LES-model case continues to accumulate energy at $k_m/k_{\min} > 16$. The SGS stress of Smagorinsky LES alleviates this effect, so it matches the filtered DNS well at $k_m/k_{\min} > 16$, but it experiences an excess of energy at $k_m/k_{\min} < 16$, as does the simulation of no LES model. On the other hand, MANN LES is able to match more closely the energy of filtered DNS at all wavenumbers throughout the simulation. Thus, at $t = 5$, the $\widetilde{E}^F(k_m)$ spectrum of the MANN LES model is very close to that of the filtered DNS (Fig. 3.10(e)) with a maximum error of 7.4×10^{-7}, compared to 3.7×10^{-6} and 4.9×10^{-6} for no LES model and Smagorinsky model, respectively, as shown

in Fig. 3.10(f). Figure 3.10(f) also shows that the MANN LES considerably reduces the error of the spectrum from the filtered DNS at low wavenumbers ($k/k_m < 16$) by more than a factor of 100 compared to no LES model and the Smagorinsky LES model at $k/k_m = 2$.

To help explain how MANN LES improves the wavelet TKE spectrum with respect to the other two LES cases, we analyze the evolution equation of the wavelet energy spectrum. By applying the discrete wavelet transform to the filtered equation (3.6), and, then, by multiplying the resulting equation by $\bar{w}_i^{(m,q)}$ and averaging, we derived the evolution equation of the filtered-velocity TKE wavelet spectrum:

$$\frac{\partial}{\partial t}\widetilde{E}^F(k_m) = \widetilde{T}^F(k_m) - \widetilde{V}^F(k_m) + \widetilde{S}^F(k_m) - \widetilde{P}(k_m), \tag{3.30}$$

where

$$\widetilde{T}^F(k_m) = -C_m \sum_{i=1}^{3} \left\langle \sum_{q=1}^{7} \bar{w}_i^{(m,q)}[\boldsymbol{l}] \left\{ \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} + \frac{1}{\bar{\rho}} \frac{\partial \bar{p}}{\partial x_i} \right\}^{(m,q)}[\boldsymbol{l}] \right\rangle_{\boldsymbol{l} \in \mathcal{D}^{(m)}}, \tag{3.31a}$$

$$\widetilde{V}^F(k_m) = -C_m \sum_{i=1}^{3} \left\langle \sum_{q=1}^{7} \bar{w}_i^{(m,q)}[\boldsymbol{l}] \left\{ \frac{1}{\bar{\rho}\,\mathrm{Re}} \frac{\partial}{\partial x_j} \left(\bar{\mu} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \right) \right\}^{(m,q)}[\boldsymbol{l}] \right\rangle_{\boldsymbol{l} \in \mathcal{D}^{(m)}}, \tag{3.31b}$$

$$\widetilde{S}^F(k_m) = \quad C_m \sum_{i=1}^{3} \left\langle \sum_{q=1}^{7} \bar{w}_i^{(m,q)}[\boldsymbol{l}] \left\{ \frac{1}{\bar{\rho}\,\mathrm{We}} f_\sigma(\bar{\chi})_i \right\}^{(m,q)}[\boldsymbol{l}] \right\rangle_{\boldsymbol{l} \in \mathcal{D}^{(m)}}, \tag{3.31c}$$

$$\widetilde{P}(k_m) = -C_m \sum_{i=1}^{3} \left\langle \sum_{q=1}^{7} \bar{w}_i^{(m,q)}[\boldsymbol{l}] \{P_i\}^{(m,q)}[\boldsymbol{l}] \right\rangle_{\boldsymbol{l} \in \mathcal{D}^{(m)}}. \tag{3.31d}$$

The terms in Eqs. (3.31a)–(3.31d) represent the energy-transfer rate due to convection and pressure, the viscous dissipation rate, the surface-tension source/sink rate of energy, and the SGS closure-term sink/source rate of energy, respectively. Here $\{\cdot\}^{(m,q)}[\boldsymbol{l}]$ denotes the coefficients of the discrete wavelet transform of the term in braces, e.g., $\bar{w}_i^{(m,q)}[\boldsymbol{l}] = \{\bar{u}_i\}^{(m,q)}[\boldsymbol{l}]$. In Section 2.2.4, we derived an analogous form to this equation from the governing equation (1.1b). The wavelet-spectrum equation (3.30) includes the new term $\widetilde{P}(k_m)$ with respect to (2.12) that accounts for the addition of the SGS closure terms P_i in the LES momentum equation (3.6). We will use (3.30) to see why the MANN LES method is able to match the filtered DNS better at low wavenumbers than the other two LES cases. Because $P_i = 0$

for the no-LES-model case, also $\widetilde{P}(k_m) = 0$ for any k_m. For Smagorinsky and MANN LES, $\widetilde{P}(k_m)$ acts as a sink of energy at all wavenumbers. We saw in Fig. 3.7(b) that $\Pi(t)$ is higher for MANN LES than for Smagorinsky LES, so $\widetilde{P}(k_m)$ is on average also slightly higher for MANN LES than for Smagorinsky LES. However, this increase in $\widetilde{P}(k_m)$ occurs primarily at the highest wavenumbers from the addition of the interfacial SGS terms computed via ANNs. To fully understand why MANN LES achieves the best agreement with $\widetilde{E}(k_m)$ of the filtered DNS, particularly at low wavenumbers, we must also look at the other three terms of (3.31). The energy-transfer rate due to advection and pressure, $\widetilde{T}^F(k_m)$, is a significant sink of energy at $k_m/k_{\text{min}} < 16$, contributing to the cascade of energy from large to small scales, but the differences in the three LES cases are small relative to the filtered DNS at these wavenumbers. The surface-tension source/sink rate of energy, $\widetilde{S}^F(k_m)$, is one order of magnitude smaller than the other terms and, thus, does not provide a significant contribution to $\partial\widetilde{E}^F/\partial t$.

That leaves us to investigate $\widetilde{V}^F(k_m)$ to explain why MANN LES is able to better match the reduced energy at low wavenumbers of the filtered DNS than the cases of no LES model or Smagorinsky. In Chapter 2, we showed that droplets with higher density than the surrounding carrier-fluid increase the viscous dissipation rate $\widetilde{V}(k_m)$ at low wavenumbers, as shown in Fig. 2.9, by increasing the velocity gradients across large scales occurring in the flow. Because of their higher inertia, they will disturb the large-scale eddies more by being more resistant to the carrier flow motion with respect to no droplets or buoyant droplets ($\rho_d/\rho_c = 1$). For the present LES methods investigated herein, Figs. 3.11(a,c,e) show $\widetilde{V}^F(k_m)$ at $t = 1.5$, 3, and 5, respectively, and Figs. 3.11(b,d,f) show its absolute error with respect to the filtered DNS. In the discussion of the TKE, $k^F(t)$, in Fig. 3.6, we showed that the no-LES-model and Smagorinsky cases are unable to reproduce the decay rate of TKE. By examining the terms of the balance equation (3.26) for filtered-velocity TKE in Fig. 3.7, we showed that the no-LES-model case introduces increased numerical diffusivity compared to Smagorinsky and MANN LES but none of the SGS dissipation or interface effects. At $t = 1.5$, Fig. 3.11(a) shows that no LES model has an excess of dissipation over that of the DNS case at $k_m/k_{\text{min}} = 64$. By $t = 3$ (Fig. 3.11(c)), $\widetilde{V}^F(k_m)$ for no LES model is elevated at both

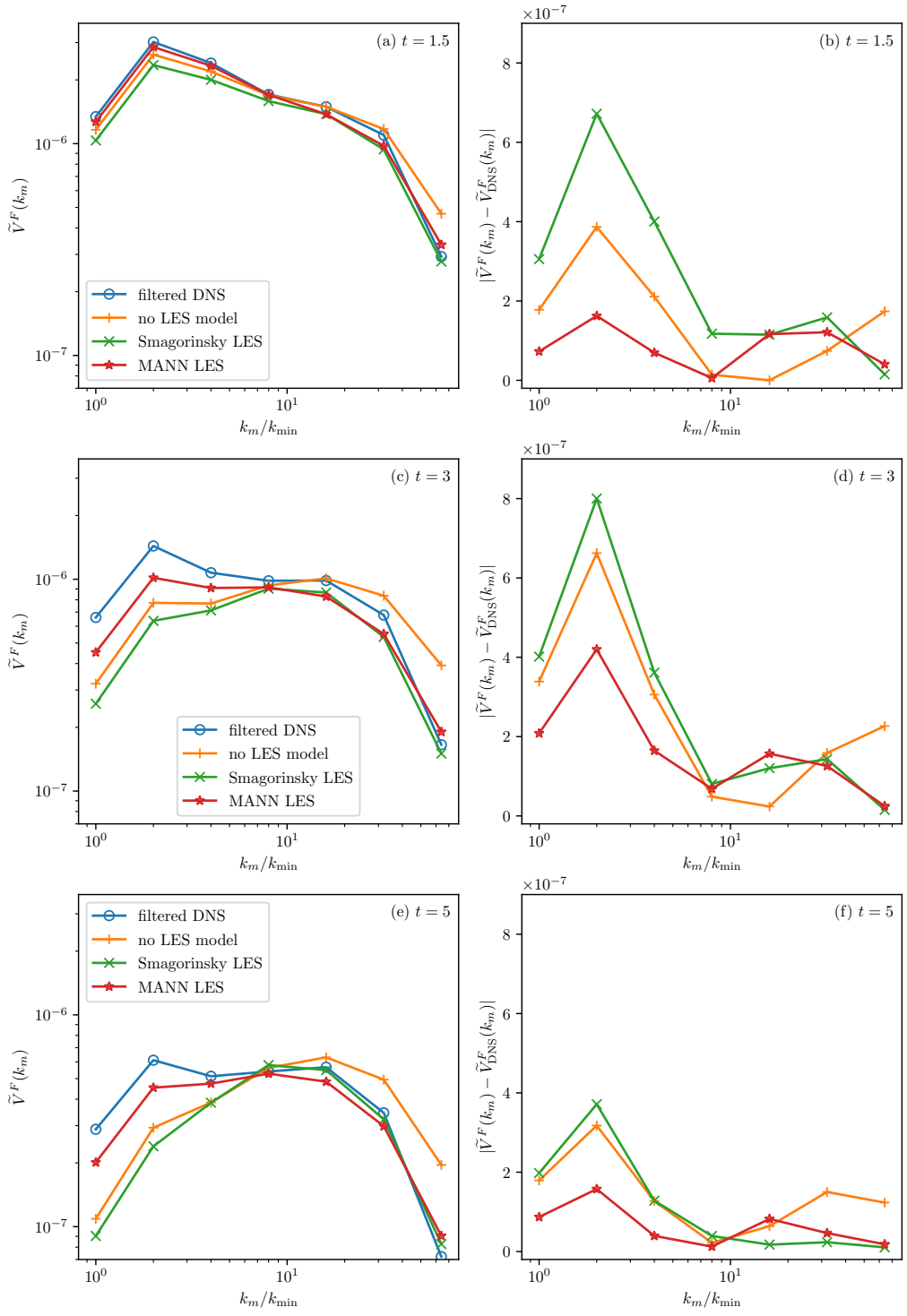

Figure 3.11: (a,c,e) Wavelet-spectral viscous dissipation rates, and (b,d,f) absolute errors compared to the filtered DNS for case C at $t = 1.5$, 3, and 5.

$k_m/k_{\min} = 32$ and 64, and by $t = 5$ (Fig. 3.11(e)), at $k_m/k_{\min} > 8$. The Smagorinsky SGS stress model used in both Smagorinsky LES and in the carrier fluid of MANN LES caused $\varepsilon^F(t)$ to remain lower for those cases in Fig. 3.7(a). As a result, $\widetilde{V}^F(k_m)$ for Smagorinsky and MANN LES is below that of filtered DNS at $k_m/k_{\min} > 8$ at $t = 1.5$, 3, and 5. The exception is MANN LES at $k_m/k_{\min} = 64$, which matches the filtered DNS at these three times. Figure 3.7 also showed that the addition of the interfacial SGS closure terms with MANN LES caused an increase in $\varepsilon^F(t)$ compared to Smagorinsky LES between $t = 1$ and 3. Now Fig. 3.11 shows that $\widetilde{V}^F(k_m)$ exhibits a similar effect. At $t = 1.5$, $\widetilde{V}^F(k_m)$ for all three LES methods is below that of the filtered DNS at $k_m/k_{\min} < 16$, but MANN LES is the closest, and Smagorinsky LES actually is farther from the filtered DNS than no LES model. By $t = 3$ and 5, this trend continues, with MANN LES offering a substantial improvement in $\widetilde{V}^F(t)$ at low wavenumbers over Smagorinsky LES. Thus, of the three LES methods, MANN LES best matches the increased spectral dissipation rate at low wavenumbers of the filtered DNS. In summary, $\widetilde{P}(k_m)$ is a sink of energy for both Smagorinksy and MANN LES for all wavenumbers, but due to the ANN-predicted interfacial SGS closure terms of MANN LES, its $\widetilde{P}(k_m)$ acts as a greater sink of energy. As a result, $\Pi(t)$ is higher for MANN LES than for Smagorinsky and contributes to the improved decay rate of filtered-velocity TKE. Similarly, $\widetilde{V}(k_m)$ is a sink of energy for all four methods, but the ANN-predicted interfacial SGS closure terms of MANN LES have a significant effect on modifying $\widetilde{V}^F(k_m)$ at low wavenumbers to match the DNS, resulting in a slightly increased $\varepsilon^F(t)$ compared to Smagorinsky LES. This shows that the addition of the interfacial closure terms allows the droplets to disrupt large-scale eddies more similarly to the DNS, and, in turn, reduce the energy spectrum at those wavenumbers, $\widetilde{E}^F(k_m)$.

To summarize the results for case C, the analysis of the filtered-velocity TKE ($k^F(t)$ and dk^F/dt, Fig. 3.6), the p.d.f. of the local filtered-velocity TKE ($k^F(\boldsymbol{x}, t)$, Fig. 3.8), and wavelet-spectral energy ($\widetilde{E}^F(k_m)$, Fig. 3.10) and dissipation rate ($\widetilde{V}^F(k_m)$, Fig. 3.11) of the three LES methods of Table 3.7 for case C show that our MANN LES model provides the closest results to the filtered DNS, significantly improving the results of no LES model or

only Smagorinsky LES, by modeling via ANNs the SGS effects for the interfacial region of the flow. This shows that the ANNs were able to accurately predict the interfacial closure terms for the same flow parameters as used in training, not only over the period of time used to generate the training data ($t = 1$ to 3) but also at later times ($t = 3$ to 5).

Cases B ($\mathrm{We_{rms}} = 0.1$) *and D* ($\mathrm{We_{rms}} = 5$)

Next, we want to apply the MANN LES to cases B and D ($\mathrm{We_{rms}} = 0.1$ and 5, respectively) at different Weber number than in case C ($\mathrm{We_{rms}} = 1$), without retraining the ANNs with the DNS data of these cases, but using the same ANNs trained with the DNS of case C. Before using our MANN LES model to simulate these cases, we need to choose a suitable limit L to the growth of the velocity due to the ANN predictions according to (3.20). Table 3.3 shows the average relative contributions of the four momentum-equation SGS closure terms with respect to the exact P_1 for case C at $t = 3$. The largest contribution comes from $\hat{\tau}_1^{\mathrm{surf}}$, which accounts for almost 70% of P_1. Thus, knowing that $\hat{\tau}_i^{\mathrm{surf}}$ dominates P_i and noting the dependence of $\hat{\tau}_i^{\mathrm{surf}}$ on 1/We, we estimate that P_i scales roughly with 1/We. Therefore, for case B with $\mathrm{We_{rms}} = 0.1$, we set $L = 5 \times 10^{-3}$, and for case D with $\mathrm{We_{rms}} = 5$, we set $L = 1 \times 10^{-4}$. This choice of L produced reasonably accurate results for the MANN LES model as shown below.

The filtered-velocity TKE, $k^F(t) = \langle \frac{1}{2} \bar{u}_i \bar{u}_i \rangle$, for cases B and D is shown in Figs. 3.12(a,c), and its decay rate dk^F/dt is shown in Figs. 3.12(b,d). Analogously to what we observed for case C, at $1 \leq t \leq 3$, the no-LES-model case produces a decay rate that is too low with respect to that of the filtered DNS because the increase in SGS dissipation everywhere in the field is not accounted for. Thus, for case B, at $t = 5$, the relative error for the TKE of no LES model compared to that of the filtered DNS is 20.8% (Fig. 3.12(a)). The Smagorinsky SGS model improves the TKE decay rate only initially but quickly diverges from that of the filtered DNS, resulting in a relative error in $k^F(t)$ of 18.9% at $t = 5$ (Fig. 3.12(a)) because it is not accounting for the SGS terms at the droplet interface. The MANN LES model considerably improves the decay rate of $k^F(t)$ with respect to the other two LES methods

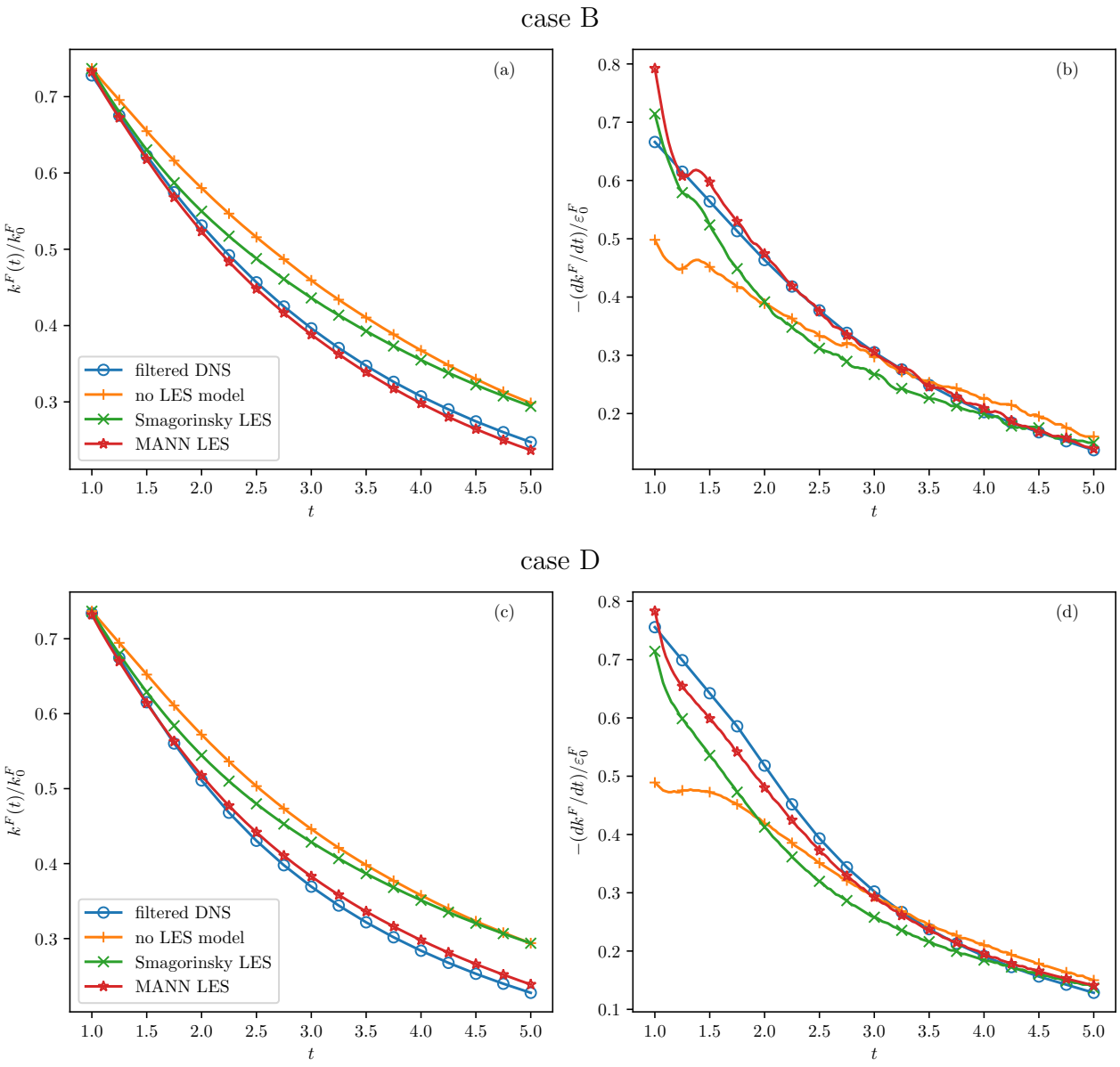

Figure 3.12: (a,c) Filtered-velocity TKE, $k^F(t) = \langle \frac{1}{2}\bar{u}_i\bar{u}_i \rangle$, normalized by k_0^F; (b,d) decay rate of filtered-velocity TKE, $-dk^F/dt$, normalized by ε_0^F, for cases B and D.

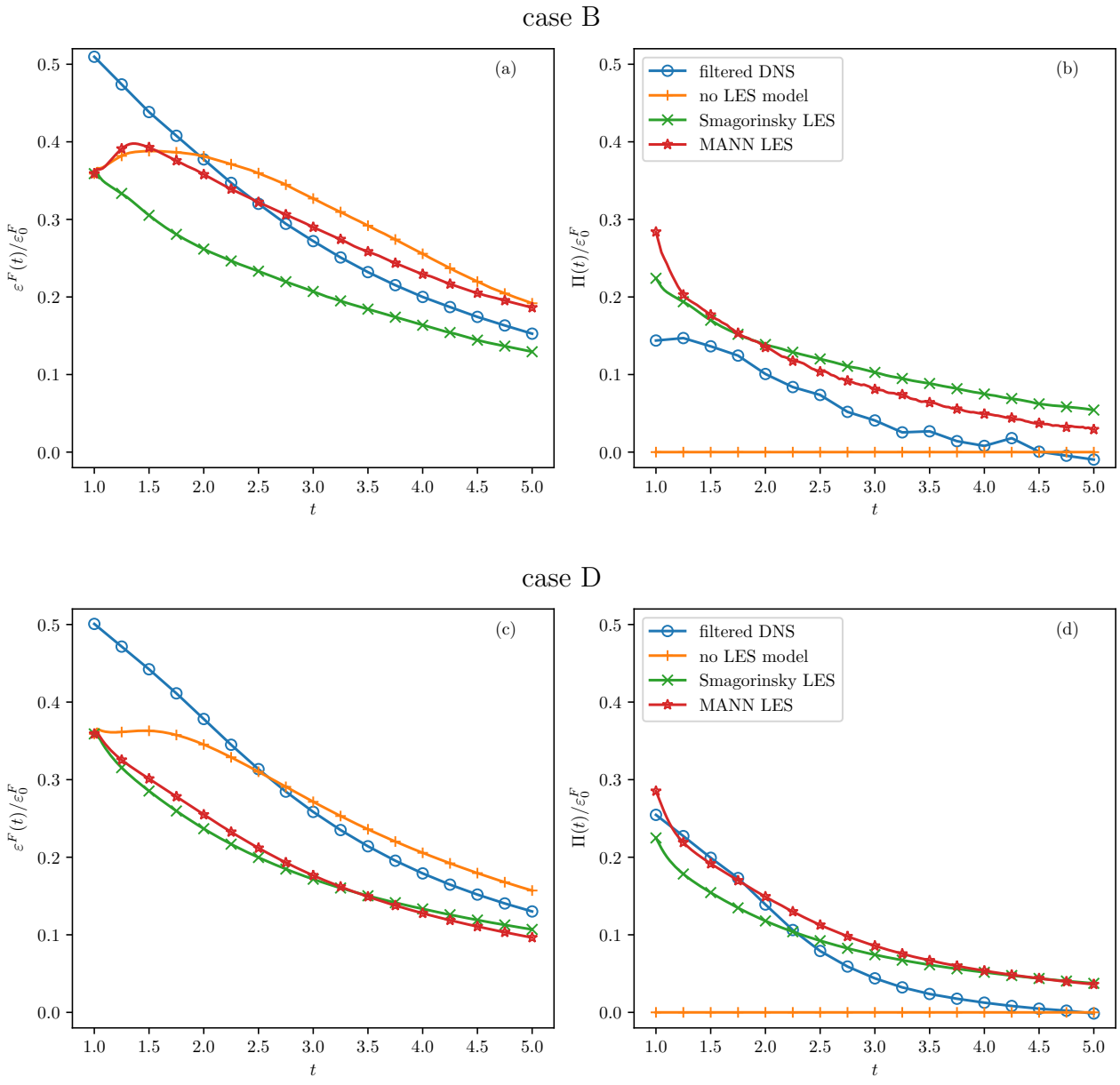

Figure 3.13: (a,c) Filtered-velocity dissipation rate, $\varepsilon^F(t)$; (b,d) sink of power due to SGS closure terms, $\Pi(t)$, of the TKE balance equation (3.26) for cases B and D.

and gives a $k^F(t)$ much closer to that of the filtered DNS, such that by $t = 5$, its relative error is only 4.12% for case B. However, the MANN LES overpredicts the decay rate at $1 \leq t \leq 2$ (Fig. 3.12(b)) such that the TKE is below that of the filtered DNS. For case D, the relative error for our model is 6.12% compared to 29.1% for no model and 29.0% for Smagorinsky (Fig. 3.12(c)), and as with case C, the decay rate of TKE is slightly underpredicted with MANN LES (Fig. 3.12(d)). Our method does not perform quite as well as with case C, when the relative error was only 1.68%, but this was expected given that only case C was used for training the ANNs.

Figures 3.13(a,c) show the terms $\varepsilon^F(t)$ and $\Pi(t)$ of the TKE balance equation (3.26) for case B, and Figs. 3.13(b,d) show these terms for case D. Again $\Psi^F(t)$ is significantly smaller than these terms and is not shown. In both cases B and D, the no-LES-model case behaves similarly as in case C. At $t = 1$, for the three LES methods, $\varepsilon^F(t)$ begins at about 30% of that of the filtered DNS, and for no LES model, the dissipation rate is increased due to numerical diffusivity, but $\Pi(t)$ is zero, so the overall decay rate of TKE is too low beginning at $t = 1$. The results for Smagorinsky LES are also similar to those of case C. The dissipation rate decays normally, and $\Pi(t)$ is significant relative to $\varepsilon^F(t)$, but it is not high enough to overcome the decrease in $\varepsilon^F(t)$, so the overall decay rate of TKE is higher than that of no LES model but still lower than that of filtered DNS. However, MANN LES performs relatively poorly in case B. Figure 3.12(a) showed that $k^F(t)$ is too low for case B with MANN LES so the decay rate of TKE is higher than for the filtered DNS (Fig. 3.12(b)). Figure 3.13(b) shows that, in case B, $\Pi(t)$ for the filtered DNS is about 40% lower at $t = 1$ than in cases C or D, so the SGS closure terms do not contribute as much to the decay of TKE. But at $t = 1$, $\Pi(t)$ for MANN LES is still about as high in case B as in cases C and D, so the decay of TKE due to SGS closure terms is significantly overpredicted. This effect also causes $\varepsilon^F(t)$ to increase from $t = 1$ to 1.3 for MANN LES (Fig. 3.13(a)), which it does not do in either of the other cases. The result is that the decay rate of TKE is too high for MANN LES beginning at the start of the simulation at $t = 1$. By $t = 1.25$, $\Pi(t)$ for MANN LES falls to about the same level as Smagorinsky LES. However, $\varepsilon^F(t)$, while it does begin to decrease,

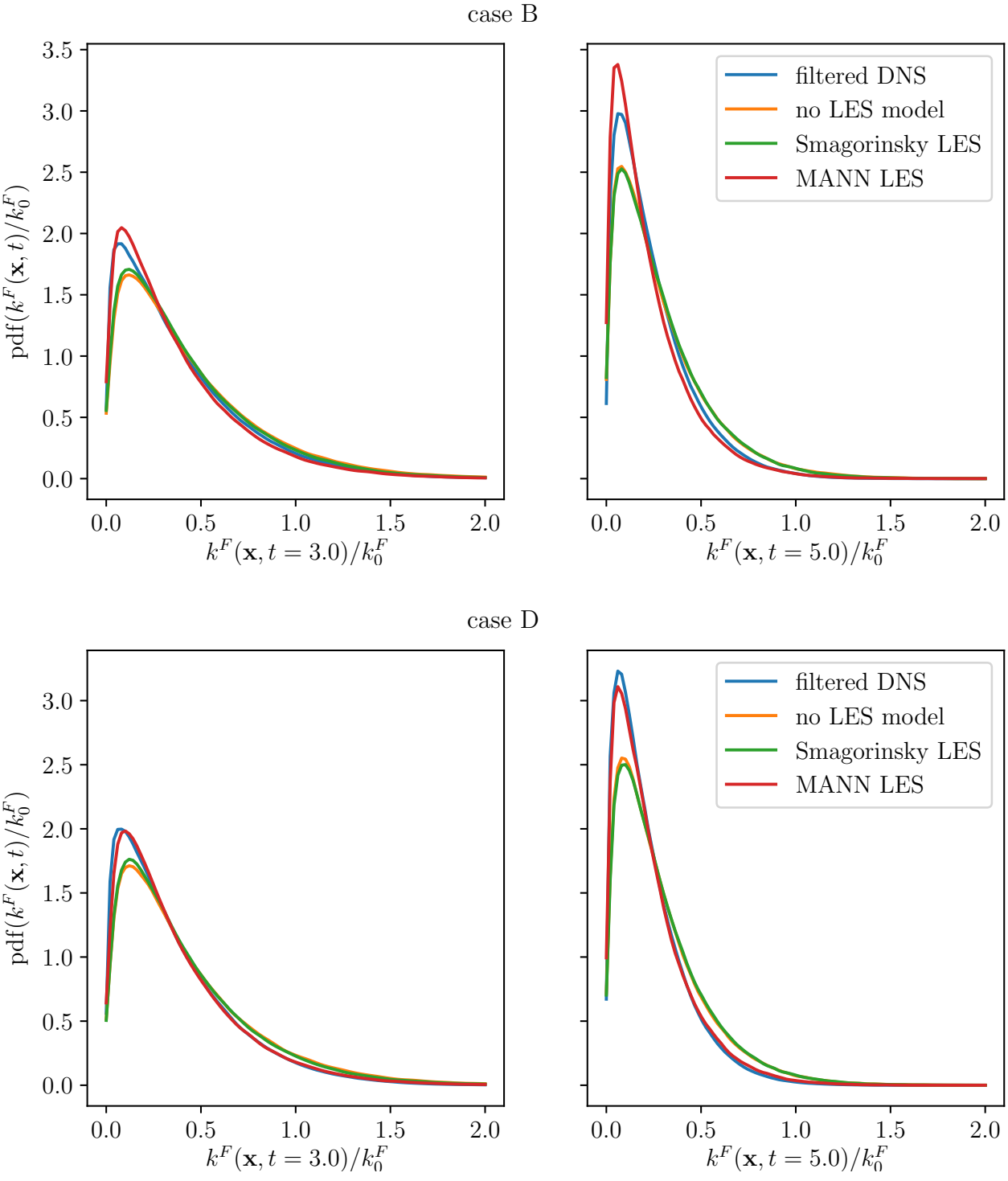

Figure 3.14: Probably density function (p.d.f.) of filtered-velocity kinetic energy, $k^F(\boldsymbol{x},t) = \frac{1}{2}\bar{u}_i\bar{u}_i$, throughout the domain at $t = 3$ and 5, normalized by k_0^F, for cases B and D.

		DNS	no LES model	Smagorinsky LES	MANN LES
case B	local TKE error		0.179	0.172 (−3.94%)	0.162 (−14.0%)
	num. droplets, $t = 4$	2907	1849 (−36.4%)	1842 (−36.6%)	2284 (−21.4%)
	num. droplets, $t = 5$	2669	1527 (−42.8%)	1504 (−43.6%)	1962 (−26.5%)
case D	local TKE error		0.173	0.170 (−1.88%)	0.144 (−16.8%)
	num. droplets, $t = 4$	3596	6629 (+84.3%)	5838 (+62.3%)	5486 (+52.6%)
	num. droplets, $t = 5$	3527	6899 (+95.6%)	6156 (+74.5%)	5421 (+53.7%)

Table 3.9: Average error in local filtered-velocity TKE, $\langle |k^F(\boldsymbol{x},t) - k^F_{\mathrm{DNS}}(\boldsymbol{x},t)| \rangle / k^F_0$, at $t = 5$, and number of droplets at $t = 4$ and 5, for cases B and D. In parentheses is change relative to the no-LES-model case (for the local TKE error) or change relative to the filtered DNS case (for the number of droplets).

stays significantly elevated for the remainder of the simulation. Therefore, the decay rate of TKE for MANN LES remains too high until $t = 3$ when the decay rates begin to converge for all four methods. In summary, an overprediction in $\Pi(t)$ at $t = 1$ causes $k^F(t)$ of MANN LES to be below that of the filtered DNS throughout the simulation. In case D, MANN LES matches the filtered DNS in $\Pi(t)$ from $t = 1$ to 2 (Fig. 3.13(d)) and only causes a slight increase in $\varepsilon^F(t)$ as in case C (Fig. 3.13(c)). After $t = 2$, $\Pi(t)$ of MANN LES is not enough to overcome the reduced $\varepsilon^F(t)$, so the decay rate of TKE begins to prematurely decrease, causing $k^F(t)$ to be slightly higher than that of filtered DNS, as shown in Fig. 3.12(c).

Next, the distributions of local filtered-velocity TKE, $k^F(\boldsymbol{x},t) = \frac{1}{2}\bar{u}_i\bar{u}_i$, given by the four methods, are shown in Fig. 3.14. The mixed model for case B doesn't perform quite as well as for case C and case D. At $t = 5$, the p.d.f. for MANN LES in case B peaks at around $k^F(\boldsymbol{x}, t = 5)/k^F_0 = 0.2$ but at a higher level than the filtered DNS, meaning that the ANNs cause a decrease in energy at too many points. This was explained by Fig. 3.13(a) as the product of a large increase in $\varepsilon^F(t)$ and the resulting overpredicted decay rate of TKE. Therefore, the average TKE, $k^F(t)$, is too low at $t = 5$, and the distribution in local TKE, $k^F(\boldsymbol{x},t)$, is

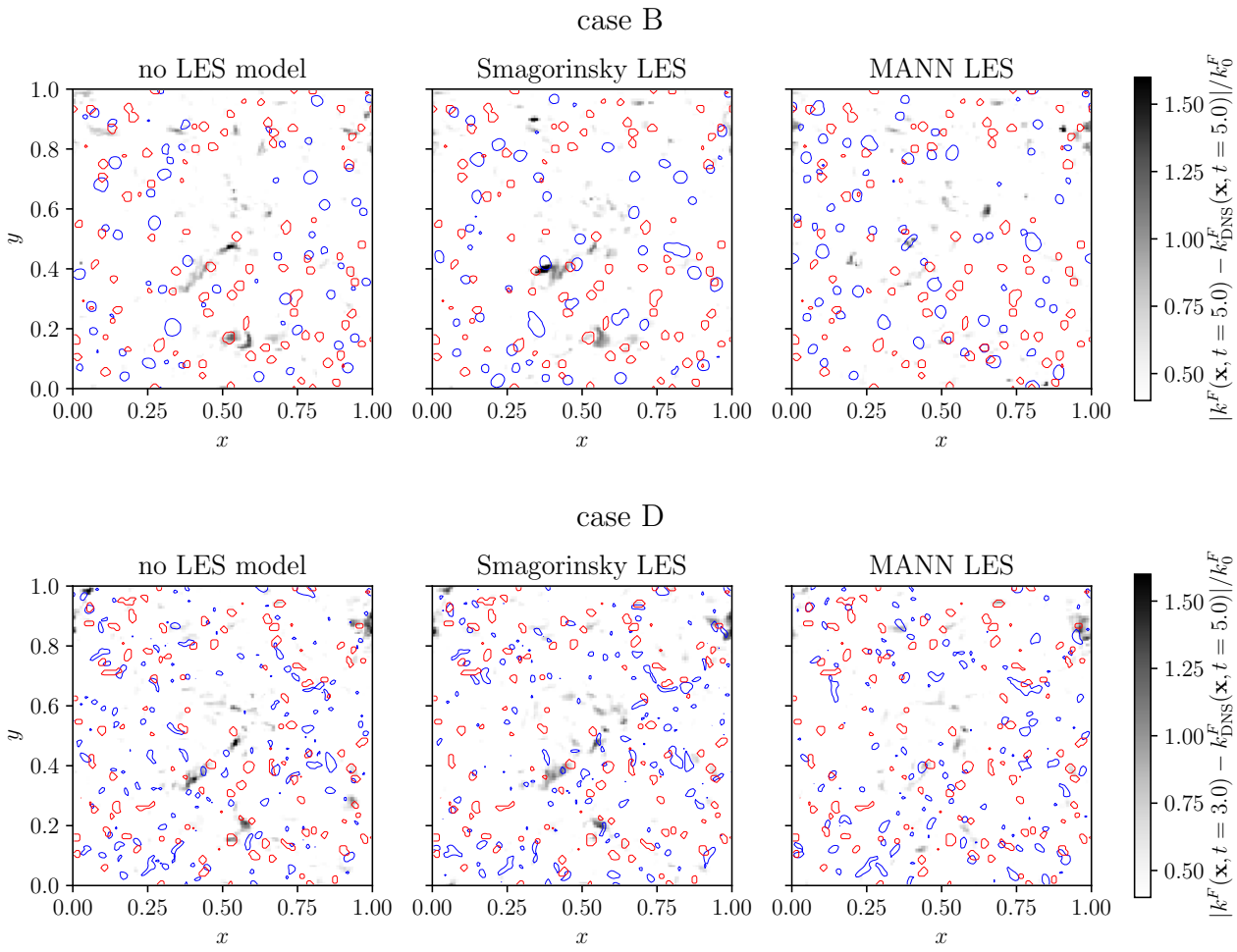

Figure 3.15: Absolute error in filtered-velocity kinetic energy, normalized by k_0^F, for the three LES cases in comparison to the filtered DNS results at $t = 5$ and $z = 0.5$, for cases B and D. The droplets are also shown as VoF contours ($\bar{\chi} = 0.5$) with fully resolved DNS results in red and the LES cases in blue.

skewed towards the lower end. MANN LES does not experience this increase in $\varepsilon^F(t)$ for case D, so the p.d.f. in Fig. 3.14 matches the filtered DNS almost as well as for case C in Fig. 3.8.

Figure 3.15 shows the absolute error in local filtered-velocity kinetic energy, $|k^F(\boldsymbol{x}, t=5) - k^F_{\mathrm{DNS}}(\boldsymbol{x}, t=5)|/k^F_0$, in a plane of the computational domain ($z = 0.5$) at $t = 5$. The MANN LES model again shows the best results in comparison to the no-LES-model and Smagorinsky cases by reducing the size of the regions of larger errors and the error magnitude for the local filtered-velocity kinetic energy. In particular, note the reduction of the error near $(x, y) = (0.6, 0.2)$ and $(0.4, 0.4)$ for both cases B and D. Table 3.9 shows the average absolute error in local filtered-velocity TKE, $\langle |k^F(\boldsymbol{x}, t=5) - k^F_{\mathrm{DNS}}(\boldsymbol{x}, t=5)|/k^F_0 \rangle$, at $t = 5$, for cases B and D. As with case C, the MANN LES model significantly reduces this error compared to using the Smagorinsky SGS stress model alone, indicating that the ANNs are able to generalize well to cases B and D.

Figure 3.15 also shows contours of the DNS VoF functions in red and LES VoF functions in blue. Again, the LES models are not able to accurately track droplet position. However, the MANN LES model does offer an improvement in maintaining the correct number of droplets, as shown in Table 3.9. In case B, the MANN LES model slows down the rate of droplet coalescence, which we already pointed out in our analysis of the power of surface tension. On the other hand, because case D has less surface tension than cases B and C, droplets are much more likely to break up than coalesce, which is reflected in the DNS results. Just as the LES cases overpredict coalescence, they also overpredict breakup, as shown in Table 3.9, though the MANN LES model is able to somewhat mitigate this effect.

Last, Figs. 3.16(a,c,e) show the wavelet energy spectra for the four methods at $t = 1.5$, 3, and 5, respectively, along with the absolute errors $|\widetilde{E}^F(k_m) - \widetilde{E}^F_{\mathrm{DNS}}(k_m)|$ in Figs. 3.16(b,d,f) for case B. The same results are shown for case D in Figs. 3.17(a,c,e) and 3.17(b,d,f). The results for cases B and D are very similar to those for case C, where the MANN LES model gives better results at wavenumbers $k_m/k_{\min} < 16$. The main difference is that in case B, by $t = 5$, the energy at these wavenumbers is slightly underpredicted (Fig. 3.16(e)). To explain why the MANN LES model is able to match the lower energy at these wavenumbers better than

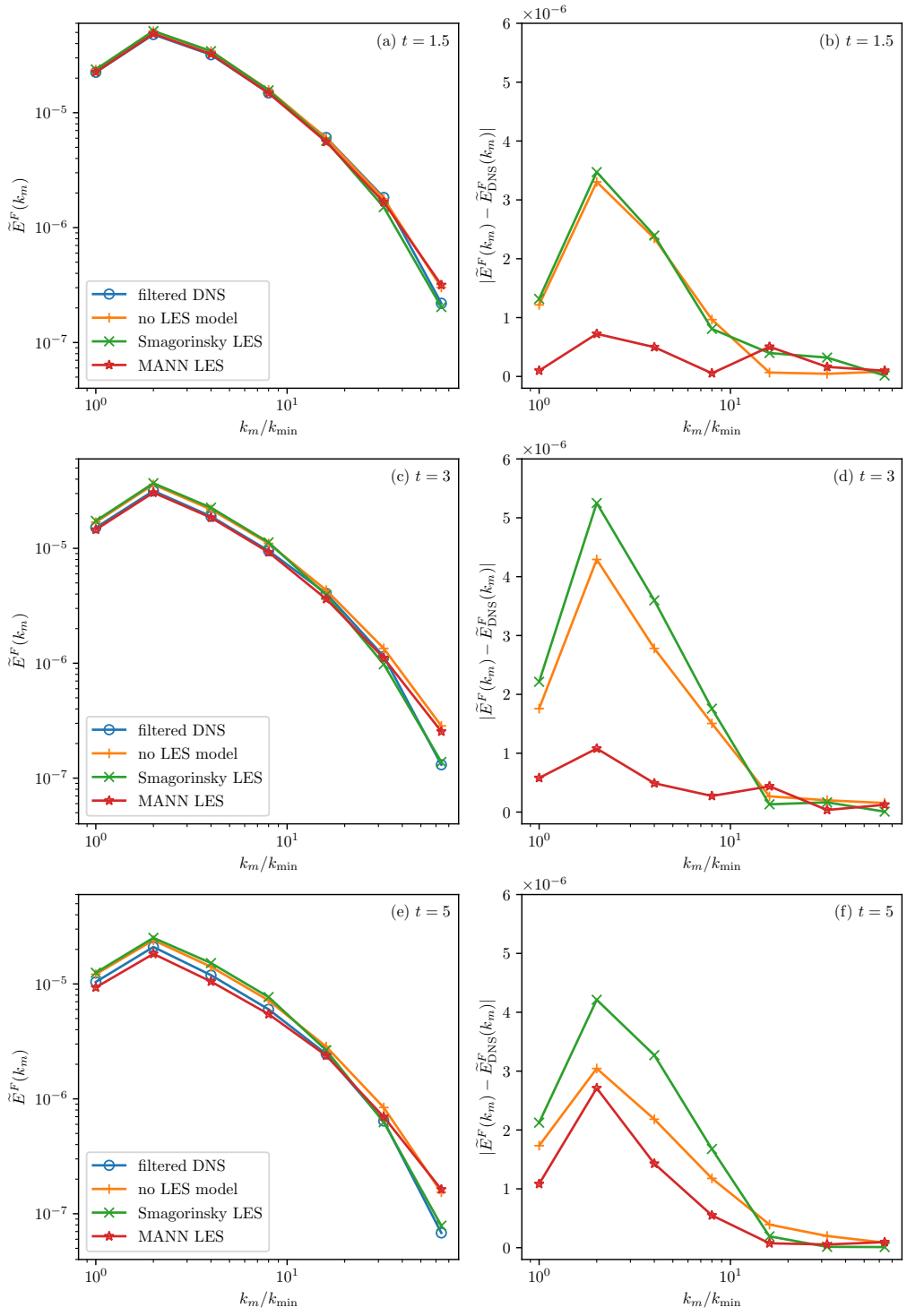

Figure 3.16: (a,c,e) Wavelet turbulence kinetic energy spectra, and (b,d,f) absolute errors in wavelet TKE spectra compared to the filtered DNS for case B at $t = 1.5$, 3, and 5.

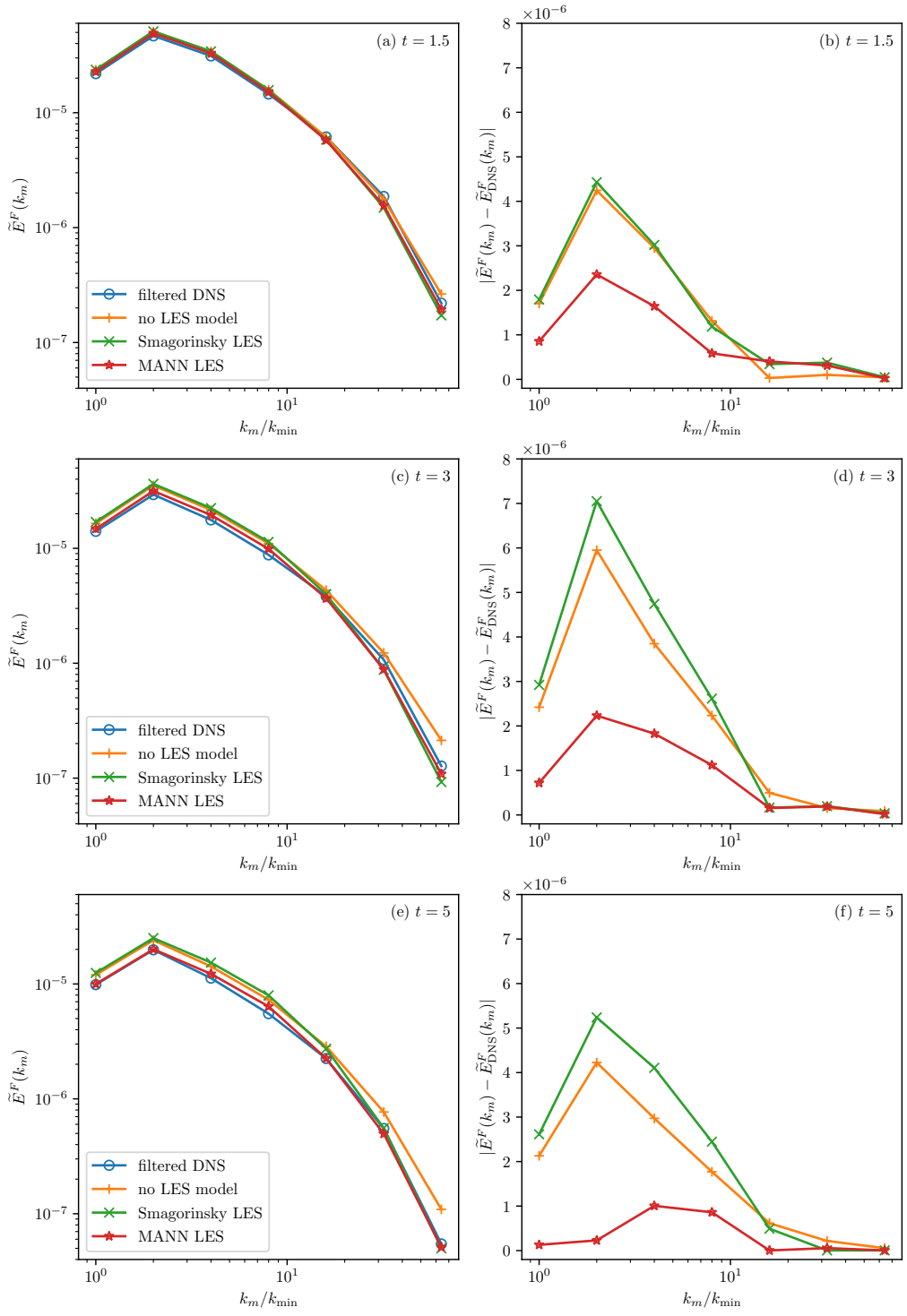

Figure 3.17: (a,c,e) Wavelet turbulence kinetic energy spectra, and (b,d,f) absolute errors in wavelet TKE spectra compared to the filtered DNS for case D at $t = 1.5$, 3, and 5.

the no-LES-model and Smagorinsky cases, we again look at the wavelet-spectrum evolution equation (3.30). As in case C, $\widetilde{P}(k_m)$ acts similarly as a sink of energy at all wavenumbers for Smagorinsky and MANN LES. In case D, $\Pi(t)$ is higher for MANN LES than for Smagorinsky LES (Fig. 3.13(d)), so $\widetilde{P}(k_m)$ is also higher on average. In case B, because $\Pi(t)$ for MANN LES drops below that of Smagorinsky LES by $t = 1.8$ (Fig. 3.13(b)), $\widetilde{P}(k_m)$ is also decreased at later times. $\widetilde{T}^F(k_m)$ acts as a sink of energy at low wavenumbers but is very close for the filtered DNS and three LES cases. In case D, $\widetilde{S}^F(k_m)$ is still one order of magnitude smaller than the other terms. In case B, $\widetilde{S}^F(k_m)$ does have a more significant effect because it scales with 1/We and contributes to the decrease in energy at low wavenumbers for MANN LES.

To explain why the MANN LES method is able to better predict $\widetilde{E}^F(k_m)$ at $k_m/k_{\min} < 16$ for cases B and D, we also look at $\widetilde{V}^F(k_m)$ for $t = 1.5$, 3, and 5 in Figs. 3.18(a,c,e), respectively, for case B. Figures 3.18(b,d,f) show the absolute error with respect to the filtered DNS. Figures 3.19(a,c,e) and 3.19(b,d,f) show the same results for case D. As with case C, in cases B and D, the no-LES-model method experiences excessive dissipation at the three highest wavenumbers, and this error grows with time. However, with MANN LES in case B, which also exhibited an increase in $\varepsilon^F(t)$ beginning at $t = 1$, $\widetilde{V}^F(t)$ is also much higher than that of filtered DNS at $k_m/k_{\min} = 64$, and this error worsens in time. MANN LES does not have this issue in case D. Turning to the lower wavenumbers, the filtered DNS dissipation rate at $k_m/k_{\min} < 8$ for cases B and D at the three given times is best matched by the MANN LES, as in case C. Also as in case C, Smagorinsky LES performs worse than no LES model at these wavenumbers, showing that, when using the interfacial SGS closure terms of the MANN LES model, as opposed to the Smagorinsky SGS stress model alone, the droplets are able to take more energy from large-scale eddies by resisting carrier-flow motion more similarly to the DNS. In summary, in cases B and D, the MANN LES model increases the dissipation rate, $\widetilde{V}^F(k_m)$ at low wavenumbers, and, in turn, reduces the energy spectrum at those wavenumbers, $\widetilde{E}^F(k_m)$.

The results for cases B and D at $\mathrm{We}_{\mathrm{rms}} = 0.1$ and 5, respectively, show that the MANN LES method is able to perform relatively well in predicting the decay of isotropic turbulence

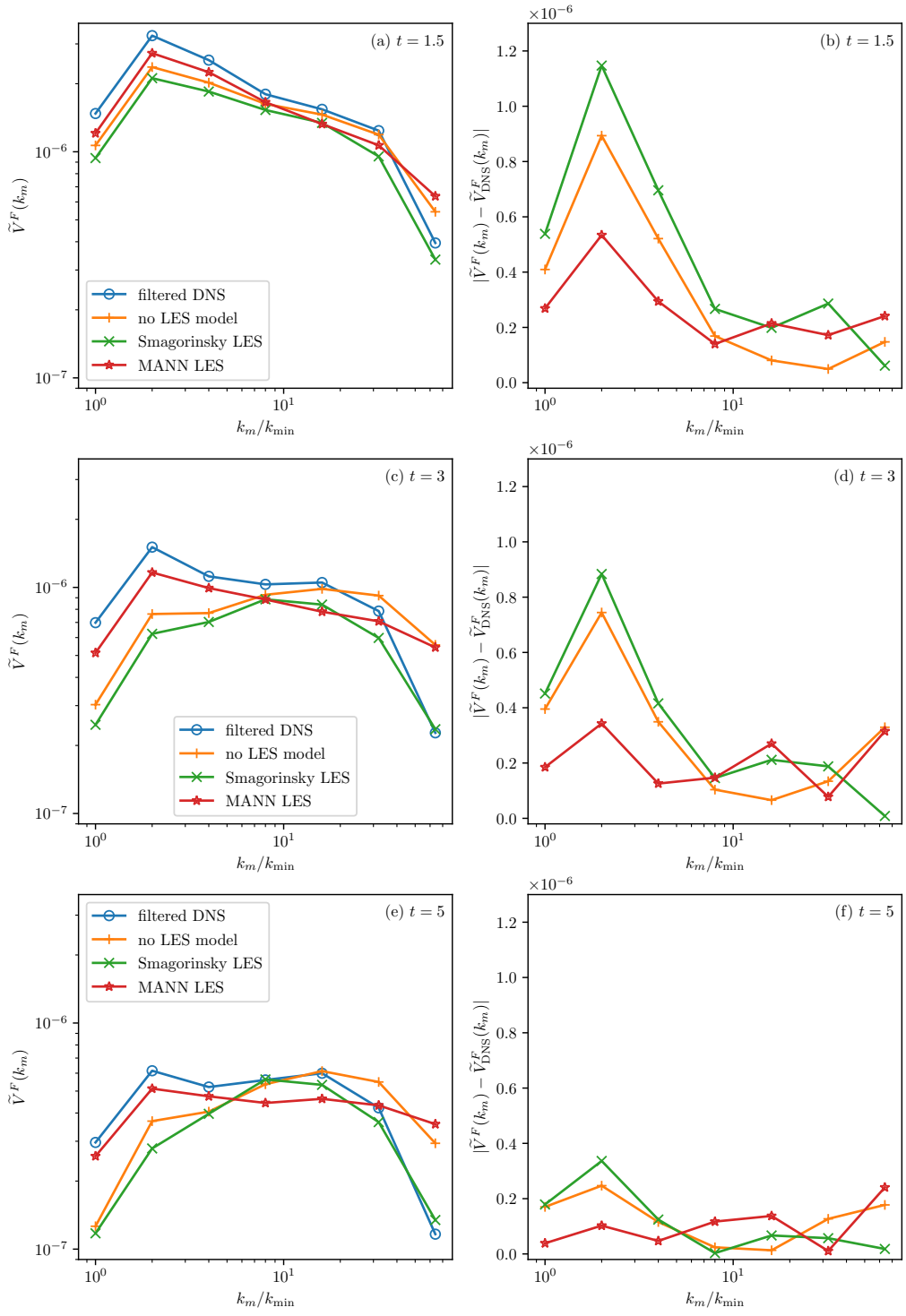

Figure 3.18: (a,c,e) Wavelet-spectral viscous dissipation rates, and (b,d,f) absolute errors compared to the filtered DNS for case B at $t = 1.5$, 3, and 5.

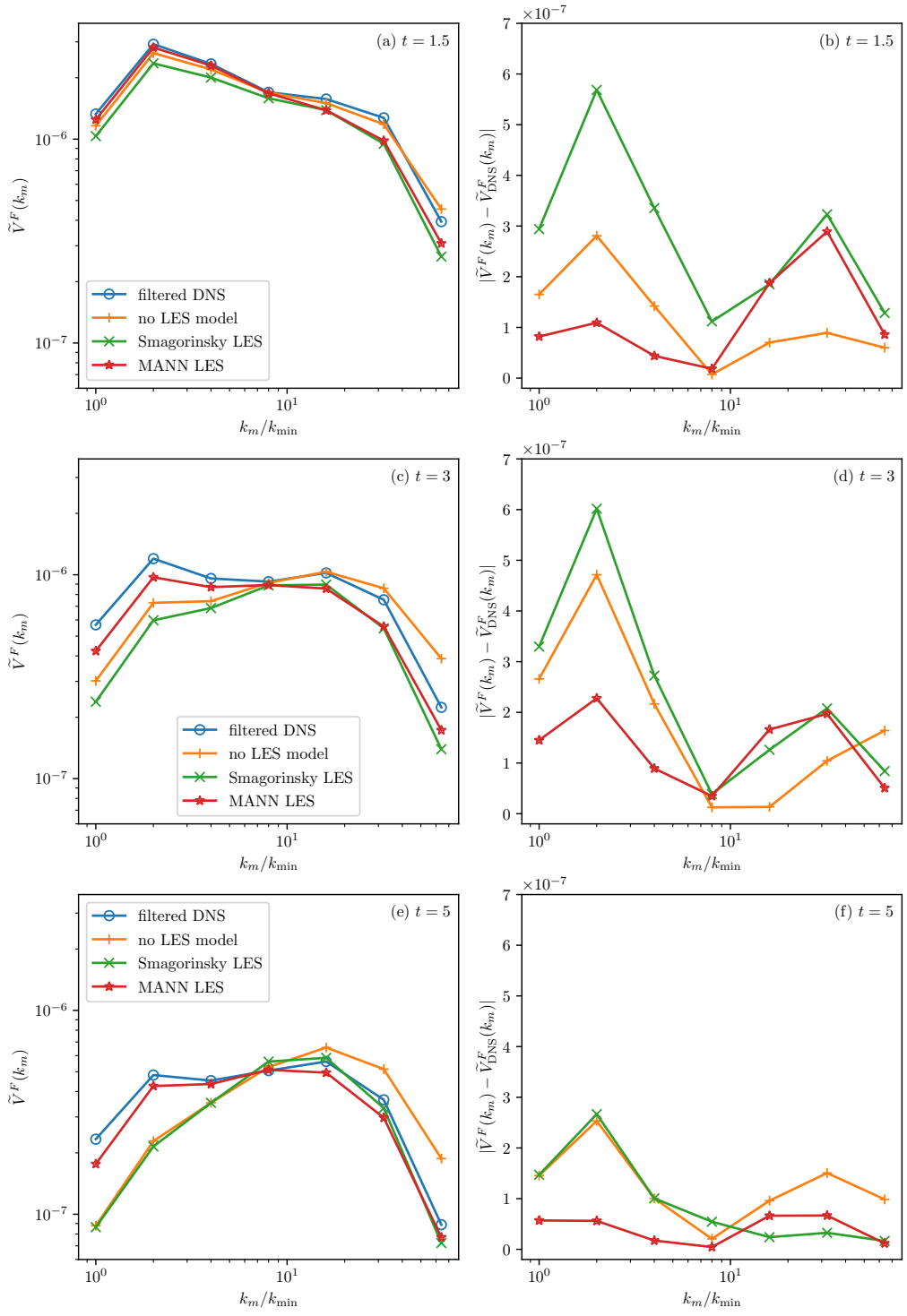

Figure 3.19: (a,c,e) Wavelet-spectral viscous dissipation rates, and (b,d,f) absolute errors compared to the filtered DNS for case D at $t = 1.5$, 3, and 5.

with respect to the filtered DNS case. Understandably, our method does perform better in case C ($\mathrm{We}_{\mathrm{rms}} = 1$) than in cases B and D, since case C is the case with which the ANNs were trained.

3.4 Conclusion

We have presented *a priori* and *a posteriori* analysis for LES of droplet-laden isotropic turbulence, and we have proposed and tested, to the best of our knowledge, the first SGS model to model the flow region near the droplet interface for droplets of diameter of Taylor length-scale of turbulence. Our method (MANN LES) splits the tasks of modeling LES closure terms in the carrier fluid and at the interface. This idea was motivated by the results of Chapter 2, where, by performing wavelet-spectral analysis of the DNS results, we showed that the droplets did not affect the wavelet energy spectrum of the small-scales of turbulence away from the droplets interface, but had effects on the wavelet energy spectrum of the interface region. Our MANN LES method uses the Smagorinsky SGS stress model in the carrier fluid and trained ANNs to predict all the closure terms in the interface region of the flow. We saw that this method has a number of benefits:

- Ease of implementation
  - We were able to use our existing multiphase Navier–Stokes solver without significant changes. We simply added to the DNS solver the components to compute the Smagorinsky SGS stress term for points contained in the carrier fluid and to evaluate the ANNs for interfacial closure terms (using existing libraries).
  - Because the ANNs are used to predict SGS effects only at the interface, we avoid the cost of ANN evaluations across the entire domain.
  - The ANNs themselves employ a simple architecture. We did not have to use any advanced machine-learning techniques, nor did we have to spend large amounts of computational resources to train these ANNs.

- Adaptability
  - Due to the simplicity of the implementation, our method is easily adaptable. For example, a different model for the carrier-fluid SGS stress term rather than Smagorinsky could be used, which wouldn't require retraining the ANNs.
  - Our method isn't limited to droplet-laden flows. A similar approach of using machine learning for closure terms at an interface could be used for turbulent multiphase flows in which the SGS stress, τ_i^{conv}, in the carrier fluid is close to that of a single-phase flow as in our case (as shown in Chapter 2).
- Performance
  - We showed that our MANN LES method was able to closely reproduce the filtered DNS results for the the filtered-velocity TKE and its p.d.f. as well the wavelet energy spectrum.
  - A minimal amount of tuning was required to achieve such results. We only put in place a limiter to ensure that the ANN predictions remained within a reasonable range.
  - We also showed that our method was easily generalized to different sets of parameters not seen by the ANN during training.

Chapter 4

FURTHER RESEARCH OPPORTUNITIES

Because we elected to keep our method as simple as possible, there are still some opportunities for improving our mixed model. One such avenue would involve upgrading to more advanced ANN architectures such as convolutional neural networks (CNNs), which are particularly common in image recognition. CNNs convolve inputs with a chosen kernel to extract certain features. CNNs also have the advantage of maintaining the spatial arrangement of their inputs such as the two-dimensional array of pixels that form an image or, as would be the case in our application, three-dimensional flow variables. Therefore, it is possible that CNNs could could improve the training of the ANN learning the relationships between the flow variables in neighboring cells and the interfacial SGS closure terms.

We are also interested in the possibility of generalizing the ANNs to work with a wider range of parameters. In the present study, we only looked at predicting closure terms for a particular set of flow parameters, namely $\mathrm{We}_{\mathrm{rms}} = 0.1$, 1, and 5 and $\rho_d/\rho_c = \mu_d/\mu_c = 10$, but this could be generalized to accurately model SGS effects for a range of density and viscosity ratios as well. Such generalization would require improving how to predict the limiter parameter L.

More importantly, as mentioned earlier, there is room for improvement in tracking droplets and maintaining the correct number, position, and shape of the droplets in the LES methodology. A possible avenue to improve that is by using the method presented in [35] to more accurately reconstruct the VoF function on the coarse grid to improve droplet advection and the calculation of the surface-tension force. Further research is needed for modeling the conditions for droplet breakup and coalescence in order to better predict droplet behavior on the coarse grid.

With these improvements in mind, we could then adapt our method for use in engineering applications, where some accuracy can be sacrificed for the ability to find numerical solutions quickly. A common application of LES in droplet-laden turbulence is the simulation of spray combustion, such as liquid jet atomization in diesel injection [9, 28], as mentioned in Section 1.1. In fact, the nondimensionalized flow parameters of Dodd & Ferrante [12] used in our study (Section 1.2.3) were already chosen for their relevance in this application. The LES must still resolve the interface in order to capture the jet breakup, and we showed in Section 3.2.4 that for our filter size the SGS closure terms are not negligible. Therefore, by adapting our method to work with the correct boundary conditions, the accurate LES of liquid jet atomization would be feasible.

Appendix A

DETAILS FOR COMPUTING THE DECOMPOSED WAVELET SPECTRUM

In this appendix, we provide step-by-step details for computing the decomposed wavelet spectrum as in Section 2.2.5. The same general steps can be used for computing the terms in the decomposed wavelet-spectrum evolution equations.

1. Load $u_i(\boldsymbol{x}, t)$, $i = 1, 2, 3$, and $\chi(\boldsymbol{x}, t)$ from DNS data at some time t.

2. Compute the three-dimensional multilevel DWTs of u_i using a chosen wavelet (sym4 in our case) and periodization handling of the boundaries. We used the PyWavelet [34] function `wavedecn`. The result is the wavelet coefficients $w_i^{(m,q)}[\boldsymbol{l}]$.

3. Loop over wavelet scales m.

   (a) Calculate the width of the chosen wavelet at the given scale m.

   (b) Initialize empty sets $\mathcal{D}_C^{(m)} = \mathcal{D}_D^{(m)} = \mathcal{D}_I^{(m)} = \emptyset$.

   (c) Loop over each index $\boldsymbol{l}' \in \mathcal{D}^{(m)}$ at which the wavelet coefficients at scale m are defined.

      i. Given the wavelet width, find the sum of χ for all points where u_i was used to compute $w_i^{(m,q)}[\boldsymbol{l}']$.

         A. If this sum is zero, the wavelet was contained in the carrier fluid, so add $\boldsymbol{l}'$ to $\mathcal{D}_C^{(m)}$.

         B. Otherwise, if this sum is the cube of the wavelet width, the wavelet was contained in a droplet, so add $\boldsymbol{l}'$ to $\mathcal{D}_D^{(m)}$.

C. Otherwise, the wavelet contained both droplets and carrier fluid, so add $\boldsymbol{l}'$ to $\mathcal{D}_i^{(m)}$.

(d) Compute

$$\sum_{i=1}^{3}\sum_{q=1}^{7}\frac{1}{2}w_i^{(m,q)}[\boldsymbol{l}]^2. \tag{A.1}$$

i. Average this sum over $\boldsymbol{l}\in\mathcal{D}^{(m)}$, $\boldsymbol{l}\in\mathcal{D}_C^{(m)}$, $\boldsymbol{l}\in\mathcal{D}_D^{(m)}$, and $\boldsymbol{l}\in\mathcal{D}_I^{(m)}$.

ii. Multiply these means by C_m to find $\widetilde{E}(k_m)$, $\widetilde{E}_C(k_m)$, $\widetilde{E}_D(k_m)$, and $\widetilde{E}_I(k_m)$.

Appendix B

DETAILS FOR COMPUTING LES TRAINING DATA

B.1 Discretization of two-fluid LES closure terms

In this section, we provides discretizations of the closure terms τ_1^{conv}, τ_1^{accel}, τ_{1j}^{visc}, τ_1^{surf}, and τ^{adv} defined in (3.5). These discretizations are used to compute the exact values of the closure terms using DNS computed on the fine 1024^3 grid. Every overbar $\bar{\cdot}$ denotes the filtering with the uniform filter with width $\Delta = 8\,\Delta x_{\text{DNS}}$. In the following, the i, j, k subscripts denote field variables as defined at the grid cell indexed by i, j, k.

Note that we are using a staggered computational grid. This means that $\bar{\chi}$ (and therefore also $\bar{\rho}$ and $\bar{\mu}$) as well as $\bar{p}$ are defined at the cell center, but $\bar{u}_i$, $i = 1, 2, 3$, are defined at the cell faces. Therefore, if $\bar{\chi}_{i,j,k}$ is located at $(x, y, z) = (i\,\Delta x, j\,\Delta x, k\,\Delta x)$, then $\bar{u}_1$ is located at $((i + \frac{1}{2})\,\Delta x, j\,\Delta x, k\,\Delta x)$, $\bar{u}_2$ at $(i\,\Delta x, (j + \frac{1}{2})\,\Delta x, k\,\Delta x)$, and $\bar{u}_3$ at $(i\,\Delta x, j\,\Delta x, (k + \frac{1}{2})\,\Delta x)$. A schematic of a single computational cell from the staggered grid is provided in Fig. B.1.

First we have the convective closure term in the x direction:

$$
\begin{aligned}
\tau^{\text{conv}}_{1;i,j,k} = \frac{\bar{\rho}_{i,j,k}+\bar{\rho}_{i+1,j,k}}{2}\Bigg(&\frac{(\bar{u}_{1;i+1,j,k}+\bar{u}_{1;i,j,k})^2-(\bar{u}_{1;i,j,k}+\bar{u}_{1;i-1,j,k})(\bar{u}_{1;i,j,k}+\bar{u}_{1;i-1,j,k})}{4\,\Delta x}\\
&+\frac{(\bar{u}_{1;i,j+1,k}+\bar{u}_{1;i,j,k})(\bar{u}_{2;i+1,j,k}+\bar{u}_{2;i,j,k})-(\bar{u}_{1;i,j,k}+\bar{u}_{1;i,j-1,k})(\bar{u}_{2;i+1,j-1,k}+\bar{u}_{2;i,j-1,k})}{4\,\Delta x}\\
&+\frac{(\bar{u}_{1;i,j,k+1}+\bar{u}_{1;i,j,k})(\bar{u}_{3;i+1,j,k}+\bar{u}_{3;i,j,k})-(\bar{u}_{1;i,j,k}+\bar{u}_{1;i,j,k-1})(\bar{u}_{3;i+1,j,k-1}+\bar{u}_{3;i,j,k-1})}{4\,\Delta x}\Bigg)\\
-\overline{\frac{\rho_{i,j,k}+\rho_{i+1,j,k}}{2}\Bigg(}&\overline{\frac{(u_{1;i+1,j,k}+u_{1;i,j,k})^2-(u_{1;i,j,k}+u_{1;i-1,j,k})(u_{1;i,j,k}+u_{1;i-1,j,k})}{4\,\Delta x}}\\
&\overline{+\frac{(u_{1;i,j+1,k}+u_{1;i,j,k})(u_{2;i+1,j,k}+u_{2;i,j,k})-(u_{1;i,j,k}+u_{1;i,j-1,k})(u_{2;i+1,j-1,k}+u_{2;i,j-1,k})}{4\,\Delta x}}\\
&\overline{+\frac{(u_{1;i,j,k+1}+u_{1;i,j,k})(u_{3;i+1,j,k}+u_{3;i,j,k})-(u_{1;i,j,k}+u_{1;i,j,k-1})(u_{3;i+1,j,k-1}+u_{3;i,j,k-1})}{4\,\Delta x}\Bigg)}.
\end{aligned}
\tag{B.1}
$$

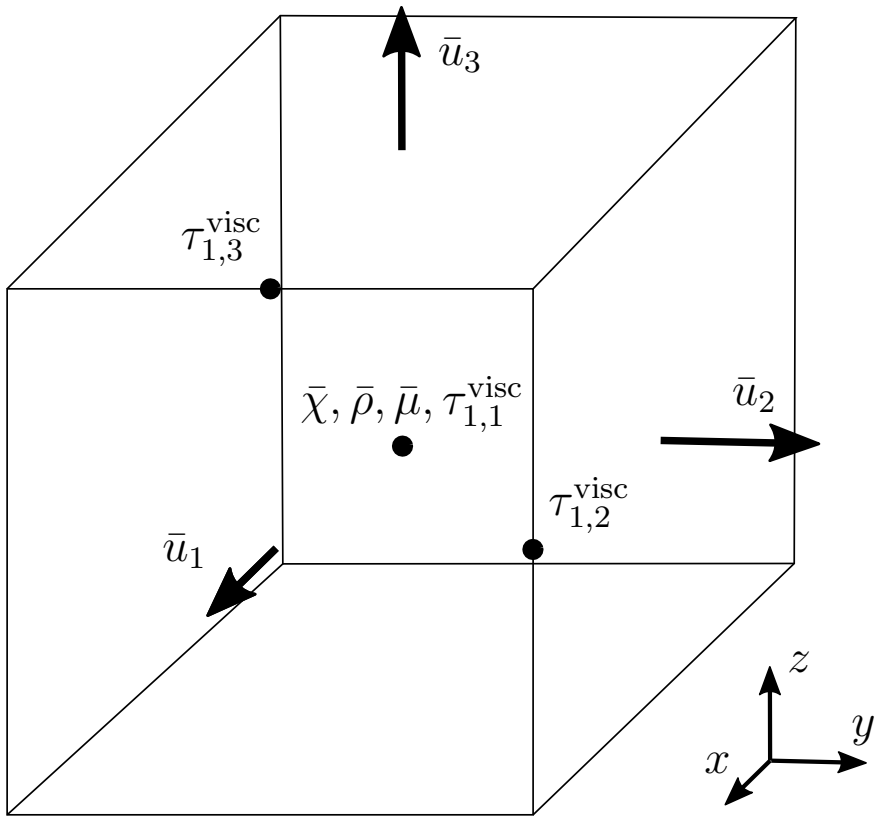

Figure B.1: Positions of flow variables at a cell i, j, k.

Note that the overbar denoting the filtering operation extends across the last three lines of the equation. Because this and the following equations will be used to compute the closure terms using DNS data, we can take $\Delta x = \Delta x_{\text{DNS}}$.

Next is the acceleration closure term:

$$\tau_{1;i,j,k}^{\text{accel}} = \frac{\bar{\rho}_{i,j,k} + \bar{\rho}_{i+1,j,k}}{2} \cdot \frac{\bar{u}_{1;i,j,k}^{n} - \bar{u}_{1;i,j,k}^{n-1}}{\Delta t} - \overline{\frac{\rho_{i,j,k} + \rho_{i+1,j,k}}{2} \cdot \frac{u_{1;i,j,k}^{n} - u_{1;i,j,k}^{n-1}}{\Delta t}}. \tag{B.2}$$

Note that this equation depends on the previous time step $n-1$ to compute the time derivative of velocity. Variables are assumed to be at time step n if not otherwise indicated.

Then we compute the viscous closure term, which has three components:

$$\tau^{\text{visc}}_{1,1;i,j,k} = \overline{2\mu_{i,j,k}\frac{u_{1;i,j,k}-u_{1;i-1,j,k}}{\Delta x}} - 2\bar{\mu}_{i,j,k}\frac{\bar{u}_{1;i,j,k}-\bar{u}_{1;i-1,j,k}}{\Delta x}, \tag{B.3}$$

$$\begin{aligned}\tau^{\text{visc}}_{1,2;i,j,k} = {} & \overline{\frac{\mu_{i,j,k}+\mu_{i+1,j,k}+\mu_{i,j+1,k}+\mu_{i+1,j+1,k}}{4}\left(\frac{u_{1;i,j+1,k}-u_{1;i,j,k}}{\Delta x}+\frac{u_{2;i+1,j,k}-u_{2;i,j,k}}{\Delta x}\right)} \\ & - \frac{\bar{\mu}_{i,j,k}+\bar{\mu}_{i+1,j,k}+\bar{\mu}_{i,j+1,k}+\bar{\mu}_{i+1,j+1,k}}{4}\left(\frac{\bar{u}_{1;i,j+1,k}-\bar{u}_{1;i,j,k}}{\Delta x}+\frac{\bar{u}_{2;i+1,j,k}-\bar{u}_{2;i,j,k}}{\Delta x}\right),\end{aligned} \tag{B.4}$$

$$\begin{aligned}\tau^{\text{visc}}_{1,3;i,j,k} = {} & \overline{\frac{\mu_{i,j,k}+\mu_{i+1,j,k}+\mu_{i,j,k+1}+\mu_{i+1,j,k+1}}{4}\left(\frac{u_{1;i,j,k+1}-u_{1;i,j,k}}{\Delta x}+\frac{u_{3;i+1,j,k}-u_{3;i,j,k})}{\Delta x}\right)} \\ & - \frac{\bar{\mu}_{i,j,k}+\bar{\mu}_{i+1,j,k}+\bar{\mu}_{i,j,k+1}+\bar{\mu}_{i+1,j,k+1}}{4}\left(\frac{\bar{u}_{1;i,j,k+1}-\bar{u}_{1;i,j,k}}{\Delta x}+\frac{\bar{u}_{3;i+1,j,k}-\bar{u}_{3;i,j,k})}{\Delta x}\right).\end{aligned} \tag{B.5}$$

Note that, while $\tau^{\text{visc}}_{1,1}$ is defined at the cell center, $\tau^{\text{visc}}_{1,2}$ and $\tau^{\text{visc}}_{1,3}$ are located at cell edges as shown in Figure B.1. This is done so that we can more easily compute $\hat{\tau}^{\text{visc}}_{1}$ as in (3.7c) with

$$\hat{\tau}^{\text{visc}}_{1;i,j,k} = \frac{1}{\text{Re}}\cdot\frac{2}{\bar{\rho}_{i,j,k}+\bar{\rho}_{i+1,j,k}}\left(\frac{\tau^{\text{visc}}_{1,1;i+1,j,k}-\tau^{\text{visc}}_{1,1;i,j,k}}{\Delta x}+\frac{\tau^{\text{visc}}_{1,2;i,j,k}-\tau^{\text{visc}}_{1,2;i,j-1,k}}{\Delta x}+\frac{\tau^{\text{visc}}_{1,3;i,j,k}-\tau^{\text{visc}}_{1,3;i,j,k-1}}{\Delta x}\right). \tag{B.6}$$

The next closure term is for surface tension:

$$\tau^{\text{surf}}_{1,i,j,k} = \overline{\kappa(\chi)n(\chi)_i} - \kappa(\bar{\chi})n(\bar{\chi})_i. \tag{B.7}$$

And lastly, we have the advective closure term:

$$\begin{aligned}\tau^{\text{adv}}_{i,j,k} = {} & \frac{\bar{u}_{1;i,j,k}+\bar{u}_{1;i-1,j,k}}{2}\cdot\frac{\bar{\chi}_{i+1,j,k}-\bar{\chi}_{i-1,j,k}}{2\,\Delta x}+\frac{\bar{u}_{2;i,j,k}+\bar{u}_{2;i,j-1,k}}{2}\cdot\frac{\bar{\chi}_{i,j+1,k}-\bar{\chi}_{i,j-1,k}}{2\,\Delta x} \\ & + \frac{\bar{u}_{3;i,j,k}+\bar{u}_{1;i,j,k-1}}{2}\cdot\frac{\bar{\chi}_{i,j,k+1}-\bar{\chi}_{i,j,k-1}}{2\,\Delta x} \\ & - \overline{\frac{u_{1;i,j,k}+u_{1;i-1,j,k}}{2}\cdot\frac{\chi_{i+1,j,k}-\chi_{i-1,j,k}}{2\,\Delta x}+\frac{u_{2;i,j,k}+u_{2;i,j-1,k}}{2}\cdot\frac{\chi_{i,j+1,k}-\chi_{i,j-1,k}}{2\,\Delta x}} \\ & \overline{+\frac{u_{3;i,j,k}+u_{1;i,j,k-1}}{2}\cdot\frac{\chi_{i,j,k+1}-\chi_{i,j,k-1}}{2\,\Delta x}}.\end{aligned} \tag{B.8}$$

B.2 Generation of training data

Here we provide step-by-step details for generating the data needed to train the ANNs in Section 3.3.3. We compute τ^{conv}_i here, but the same general steps apply for the other closure terms.

1. Load $u_i(\boldsymbol{x}, t)$, $i = 1, 2, 3$, and $\chi(\boldsymbol{x}, t)$ from DNS data at some time t.

2. Calculate $\rho(\boldsymbol{x}, t)$ using χ.

3. Filter u_i, χ, and ρ using a chosen filter and width (the uniform filter with width $\Delta = 8\,\Delta x_{\text{DNS}}$ in our case) to obtain $\bar{u}_i$, $\bar{\chi}$, and $\bar{\rho}$. We used the `uniform_filter` function in SciPy's `ndimage` submodule [58].

4. Compute τ_i^{conv} across the entire domain using (B.1) with $\Delta x = \Delta x_{\text{DNS}}$.

5. Create an array of all the training data. Each row represents one point on the interface corresponding to the indices i, j, k. The first column is τ_i^{conv}, and the remaining columns are all of the features: $\bar{u}_1$, $\bar{u}_2$, $\bar{u}_3$, and $\bar{\chi}$. After adding all of the data corresponding to τ_1^{conv} to this array, we we rotate the remaining data so that it shares the same frame of reference as the $i = 1$ data and we only need one ANN for the three dimensions. When shifting the features, we jump by 8 grid points because during LES the ANNs will only have access to information on the coarse grid.

   (a) First we add $\tau_{1;i,j,k}^{\text{conv}}$ and $\bar{u}_{1;i+l,j+m,k+n}$, $\bar{u}_{2;i+l,j+m,k+n}$, $\bar{u}_{3;i+l,j+m,k+n}$, and $\bar{\chi}_{i+l,j+m,k+n}$ for $l, m, n \in \{-8, 0, 8\}^3$.

   (b) Then we add $\tau_{2;i,j,k}^{\text{conv}}$ and $\bar{u}_{2;i+m,j+n,k+l}$, $\bar{u}_{3;i+m,j+n,k+l}$, $\bar{u}_{1;i+m,j+n,k+l}$, and $\bar{\chi}_{i+m,j+n,k+l}$ for $l, m, n \in \{-8, 0, 8\}^3$.

   (c) Lastly we add $\tau_{3;i,j,k}^{\text{conv}}$ and $\bar{u}_{3;i+n,j+l,k+m}$, $\bar{u}_{1;i+n,j+l,k+m}$, $\bar{u}_{2;i+n,j+l,k+m}$, and $\bar{\chi}_{i+n,j+l,k+m}$ for $l, m, n \in \{-8, 0, 8\}^3$.

BIBLIOGRAPHY

[1] M. Abadi et al. TensorFlow: large-scale machine learning on heterogeneous systems. `https://www.tensorflow.org/`, 2015.

[2] E. Aulisa, S. Manservisi, R. Scardovelli, and S. Zaleski. Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry. *J. Comp. Phys.*, 225(2):2301–2319, 2007. `doi:10.1016/j.jcp.2007.03.015`.

[3] S. Balachandar and J. K. Eaton. Turbulent dispersed multiphase flow. *Annu. Rev. Fluid Mech.*, 42:111–133, 2010. `doi:10.1146/annurev.fluid.010908.165243`.

[4] A. Baraldi, M. S. Dodd, and A. Ferrante. A mass-conserving volume-of-fluid method: volume tracking and droplet surface-tension in incompressible isotropic turbulence. *Comput. Fluid.*, 96:322–337, 2014. `doi:10.1016/j.compfluid.2013.12.018`.

[5] M. Bassenne, P. Moin, and J. Urzay. Wavelet multiresolution analysis of particle-laden turbulence. *Phys. Rev. Fluid.*, 3(8):084304, 2018. `doi:10.1103/PhysRevFluids.3.084304`.

[6] A. Beck, D. Flad, and C.-D. Munz. Deep neural networks for data-driven les closure models. *J. Comput. Phys*, 398:108910, 2019. `doi:10.1016/j.jcp.2019.108910`.

[7] F. Chollet et al. Keras. `https://keras.io`, 2015.

[8] T. Dairay, E. Lamballais, S. Laizet, and J. C. Vassilicos. Numerical dissipation vs. subgrid-scale modelling for large eddy simulation. *J. Comput. Phys.*, 337:252–274, 2017. `doi:10.1016/j.jcp.2017.02.035`.

[9] E. de Villiers, A. D. Gosman, and H. G. Weller. Large eddy simulation of primary diesel spray atomization. *SAE Trans.*, 113(3):193–206, 2004. `doi:10.4271/2004-01-0100`.

[10] M. Derakhti and J. T. Kirby. Bubble entrainment and liquid–bubble interaction under unsteady breaking waves. *J. Fluid Mech.*, 761:464–506, 2014. `doi:10.1017/jfm.2014.637`.

[11] M. S. Dodd and A. Ferrante. A fast pressure-correction method for incompressible two-fluid flows. *J. Comput. Phys.*, 273:416–434, 2014. `doi:10.1016/j.jcp.2014.05.024`.

[12] M. S. Dodd and A. Ferrante. On the interaction of Taylor lengthscale size droplets and isotropic turbulence. *J. Fluid Mech.*, 806:356–412, 2016. doi:10.1017/jfm.2016.550.

[13] M. S. Dodd and L. Jofre. Tensor-based analysis of the flow topology in droplet-laden homogeneous isotropic turbulence. Annual research brief, Center for Turbulence Research, 2018. URL: https://ctr.stanford.edu/annual-research-briefs-2018.

[14] S. Elghobashi. Direct numerical simulation of turbulent flows laden with droplets or bubbles. *Annu. Rev. Fluid Mech.*, 51:217–244, 2019. doi:10.1146/annurev-fluid-010518-040401.

[15] S. Elghobashi and G. C. Truesdell. On the two-way interaction between homogeneous turbulence and dispersed solid particles, I: Turbulence modification. *Phys. Fluid.*, 5(7):1790–1801, 1993. doi:10.1063/1.858854.

[16] A. Ferrante and S. Elghobashi. On the physical mechanisms of two-way coupling in particle-laden isotropic turbulence. *Phys. Fluid.*, 15(2):315–329, 2003. doi:10.1063/1.1532731.

[17] A. Freund and A. Ferrante. Wavelet-spectral analysis of droplet-laden isotropic turbulence. *J. Fluid Mech.*, 875:914–928, 2019. doi:10.1017/jfm.2019.515.

[18] A. Freund and A. Ferrante. Large-eddy simulation of droplet-laden decaying isotropic turbulence using neural networks. In preparation, 2020. To be submitted to *Int. J. Multiphase Flow*.

[19] A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly, Sebastopol, CA, 2017.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE, 2016. doi:10.1109/CVPR.2016.90.

[21] T. Hermann. frugally-deep. https://github.com/Dobiasd/frugally-deep, 2020.

[22] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. Preprint, 2012. URL: https://arxiv.org/abs/1207.0580v1.

[23] S. Ioffe and C. Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In F. Bach and D. Blei, editors, *International Conference on Machine Learning*, number 37 in Proceedings of Machine Learning Research,

pages 448–456. Machine Learning Research, 2015. URL: http://proceedings.mlr.press/v37/ioffe15.html.

[24] T. Ishihara, T. Gotoh, and Y. Kaneda. Study of high-Reynolds number isotropic turbulence by direct numerical simulation. *Annu. Rev. Fluid Mech.*, 41:165–180, 2009. doi:10.1146/annurev.fluid.010908.165203.

[25] S. S. Jain and S. K. Lele. *A priori* testing of subgrid-scale models for a two-phase turbulent flow: droplets in homogeneous-isotropic turbulence. Course project, Stanford University, 2018. URL: http://web.stanford.edu/~sjsuresh/me461-project-report.pdf.

[26] B. Ji, X. Luo, X. Peng, Y. Wu, and H. Xu. Numerical analysis of cavitation evolution and excited pressure fluctuation around a propeller in non-uniform wake. *Int. J. Multiphase Flow*, 43:13–21, 2012. doi:10.1016/j.ijmultiphaseflow.2012.02.006.

[27] L. Jofre, M. S. Dodd, J. Grau, and R. Torres. Near-interface flow modeling in large-eddy simulation of two-phase turbulence. *Int. J. Multiphase Flow*, 132:103406, 2020. doi:10.1016/j.ijmultiphaseflow.2020.103406.

[28] S. Ketterl, M. Reißmann, and M. Klein. Large eddy simulation of multiphase flows using the volume of fluid method, 2 : A-posteriori analysis of liquid jet atomization. *Exp. Comput. Multiphase Flow*, 1(3):201–211, 2019. doi:10.1007/s42757-019-0026-x.

[29] J. Kim, M. Bassenne, C. A. Z. Towery, P. E. Hamlington, A. Y. Poludnenko, and J. Urzay. Spatially localized multi-scale energy transfer in turbulent premixed combustion. *J. Fluid Mech.*, 848:78–116, 2018. doi:10.1017/jfm.2018.371.

[30] D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. Preprint, 2017. Appeared as conference paper at the 2015 International Conference for Learning Representations. URL: https://arxiv.org/abs/1412.6980v9.

[31] M. Klein, S. Ketterl, and J. Hasslberger. Large eddy simulation of multiphase flows using the volume of fluid method, 1: Governing equations and a priori analysis. *Exp. Comput. Multiphase Flow*, 1(2):130–144, 2019. doi:10.1007/s42757-019-0019-9.

[32] J. N. Kutz. Deep learning in fluid dynamics. *J. Fluid Mech.*, 814:1–4, 2017. doi:10.1017/jfm.2016.803.

[33] D. Lakehal. LEIS for the prediction of turbulent multifluid flows applied to thermal-hydraulics applications. *Nuclear Eng. Des.*, 240(9):2096–2106, 2010. doi:10.1016/j.nucengdes.2009.11.030.

[34] G. Lee, R. Gommers, F. Wasilewski, K. Wohlfahrt, A. O'Leary, H. Nahrstaedt, and Contributors. PyWavelets: wavelet transforms in Python, 2006. Available at `https://github.com/PyWavelets/pywt`. URL: `https://github.com/PyWavelets/pywt`.

[35] P. Liovic and D. Lakehal. Subgrid-scale modelling of surface tension within interface tracking-based Large Eddy and Interface Simulation of 3D interfacial flows. *Comp. Fluid.*, 60:27–46, 2012. `doi:10.1016/j.compfluid.2012.03.019`.

[36] J. López and J. Hernández. Analytical and geometrical tools for 3D volume of fluid methods in general grids. *J. Comp. Phys.*, 227(12):5939–5948, 2008. `doi:10.1016/j.jcp.2008.03.010`.

[37] F. Lucci, A. Ferrante, and S. Elghobashi. Modulation of isotropic turbulence by particles of Taylor length-scale size. *J. Fluid Mech.*, 650:5–55, 2010. `doi:10.1017/S0022112009994022`.

[38] M. R. Maxey. Droplets in turbulence: a new perspective. *J. Fluid Mech.*, 816:1–4, 2017. `doi:10.1017/jfm.2017.96`.

[39] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.*, 5:115–133, 1943. `doi:10.1007/BF02478259`.

[40] C. Meneveau. Analysis of turbulence in the orthonormal wavelet representation. *J. Fluid Mech.*, 232:469–520, 1991. `doi:10.1017/S0022112091003786`.

[41] R. Mittal, R. Ni, and J.-H. Seo. The flow physics of COVID-19. *J. Fluid Mech.*, 894:F2, 2020. `doi:10.1017/jfm.2020.330`.

[42] V. Perrier, T. Philipovitch, and C. Basevant. Wavelet spectra compared to Fourier spectra. *J. Math. Phys.*, 36(3):1506–1519, 1995. `doi:10.1063/1.531340`.

[43] S. B. Pope. *Turbulent flows*. Cambridge University, 2000.

[44] M. Raissi and G. E. Karniadakis. Hidden physics models: machine learning of nonlinear partial differential equations. *J. Comp. Phys.*, 357:125–141, 2018. `doi:10.1016/j.jcp.2017.11.039`.

[45] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comp. Phys.*, 378:686–707, 2019. `doi:10.1016/j.jcp.2018.10.045`.

[46] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat. Deep learning and process understanding for data-driven Earth system science. *Nature*, 566:195–204, 2019. doi:10.1038/s41586-019-0912-1.

[47] F. Risso. Agitation, mixing, and transfers induced by bubbles. *Annu. Rev. Fluid Mech.*, 50:25–48, 2018. doi:10.1146/annurev-fluid-122316-045003.

[48] M. Sadek and H. Aluie. Extracting the spectrum by spatial filtering. *Phys. Rev. Fluid.*, 3(12):124610, 2018. doi:10.1103/PhysRevFluids.3.124610.

[49] R. Scardovelli, E. Aulisa, S. Manservisi, and V. Marra. A marker-VOF algorithm for incompressible flows with interfaces. In *Joint U.S.-European Fluids Engineering Division Conference*, volume 1, pages 905–910, New York, 2002. American Society of Mechanical Engineers.

[50] R. Scardovelli and S. Zaleski. Interface reconstruction with least-square fit and split Eulerian–Lagrangian advection. *Int. J. Numer. Method. Fluid.*, 41(3):251–274, 2003. doi:10.1002/fld.431.

[51] R. A. Shaw. Particle-turbulence interactions in atmospheric clouds. *Annu. Rev. Fluid Mech.*, 35:183–227, 2003. doi:10.1146/annurev.fluid.35.101101.161125.

[52] J. Sirignano, J. F. MacArt, and J. B. Freund. DPM: a deep learning PDE augmentation method with application to large-eddy simulation. *J. Comp. Phys.*, 423:109811, 2020. doi:10.1016/j.jcp.2020.109811.

[53] W. A. Sirignano. Fuel droplet vaporization and spray combustion theory. *Prog. Energy Combust. Sci.*, 9(4):291–322, 1983. doi:10.1016/0360-1285(83)90011-4.

[54] J. Smagorinsky. General circulation experiments with the primitive equations, I: The basic experiment. *Month. Weather Rev.*, 91:3, 1963. doi:10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2.

[55] A. Toutant, E. Labourasse, O. Lebaigue, and O. Simonin. DNS of the interaction between a deformable buoyant bubble and a spatially decaying turbulence: *a priori* tests for LES, two-phase flow modelling. *Comp. Fluid.*, 37:877–886, 2008. doi:10.1016/j.compfluid.2007.03.019.

[56] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkens-Diehr. XSEDE: accelerating scientific discovery. *Comput. Sci. Eng.*, 16(5):62–74, 2014. doi:10.1109/MCSE.2014.80.

[57] S. Vincent, M. Tavares, S. Fleau, S. Mimouni, M. Ould-Rouiss, and J.-L. Estivalezes. *A priori* filtering and les modeling of turbulent two-phase flows application to phase separation. *Comp. Fluid.*, 176:245–259, 2018. doi:10.1016/j.compfluid.2016.10.021.

[58] P. Virtanen et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Method.*, 17:261–272, 2020. doi:10.1038/s41592-019-0686-2.

[59] X. I. A. Yang, S. Zafar, J.-X. Wang, and H. Xiao. Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Phys. Rev. Fluids*, 4:034602, 2019. doi:10.1103/PhysRevFluids.4.034602.

[60] D. L. Youngs. Time-dependent multi-material flow with large fluid distortion. In K. W. Morton and M. J. Baines, editors, *Numerical methods in fluid dynamics*, pages 273–285. Academic, New York, 1982.