Variational multiscale super-resolution: A data-driven approach for reconstruction and predictive modeling of unresolved physics

Aniruddhe Pradhan | Karthik Duraisamy

Abstract
The variational multiscale (VMS) formulation formally segregates the evolution of the coarse-scales from the fine-scales. VMS modeling requires the approximation of the impact of the fine-scales in terms of the coarse-scales. In linear problems, our formulation reduces the problem of learning the subscales to learning the projected element Green’s function basis coefficients. For this approximation, a special neural-network structure—the variational super-resolution N-N (VSRNN)—is proposed. The VSRNN constructs a super-resolution model of the unresolved scales as a sum-of-the-products of individual functions of coarse-scales and physics-informed parameters. Combined with a set of locally nondimensional features obtained by normalizing the input coarse-scale and output subscale basis coefficients, the VSRNN provides a general framework for the discovery of closures for different Galerkin discretizations. By training it on a sequence of projected data and using the subscale to compute the continuous Galerkin subgrid terms, and the super-resolved state to compute the discontinuous Galerkin fluxes, we improve the optimality and the accuracy of these methods for the convection-diffusion and linear advection problems. Finally, we demonstrate that the VSRNN allows generalization to out-of-sample initial conditions and nondimensional numbers.

KEYWORDS
coarse-grained modeling, continuous Galerkin, discontinuous Galerkin, physics-informed deep learning, super-resolution, variational multiscale method

1 | INTRODUCTION

Multiscale problems are ubiquitous in science and engineering, and many practical applications are characterized not just by a disparity of scales, but also by a complex interplay between scales. The coarse scales which are resolved in a multiscale computation constitute the macro-scale level of description. The finer, unresolved scales constitute the micro-scale level of description, and their effect on the coarse-scales needs to be modeled not just for accuracy, but also for stability. A classical application of multiscale modeling is turbulent fluid-flow at high Reynolds numbers. In the so-called...
Large Eddy Simulations (LES) of turbulent flow, the energy-containing large scales are resolved and the effect of the small scales, also known as subscales, are modeled.\textsuperscript{4,16}

Over the past few decades, several modeling approaches have been pursued to improve the performance of coarse-grained simulation of multiscale PDEs. These techniques can be broadly classified into two categories: (i) physics-inspired approaches that employ phenomenological assumptions; and (ii) structural formulations that attempt to derive subgrid models from the structure of the governing PDE. Since it is not possible to review all such approaches, we highlight representative examples in the area of turbulence simulations, and use the discussion to set the context of the present work.

Physics-based models use arguments such as the balance of energy transfer from large to small scales or scale-similarity (e.g., the Smagorinsky\textsuperscript{17} model in LES and its adaptive variants\textsuperscript{4,6,18-20}) Additional functional forms were proposed by Vreman,\textsuperscript{8} Nicoud et al.,\textsuperscript{7,9} and Lévêque et al.\textsuperscript{21} to correct asymptotic near-wall behavior. Another class of subgrid models based on the self-similarity idea was first put forward by Bardina et al.\textsuperscript{22} in its original form, and later explored by other researchers in the mixed form References \textsuperscript{22-26}. The self-similarity model in its original form was found to have a high spatial correlation with the actual subgrid stress but lacked enough dissipation to ensure the stability of the approach.

Langford et al.\textsuperscript{27} in their work on optimal-LES showed that it is possible to construct an abstract subgrid model that can obtain correct single-time, multipoint statistics and generates minimal error in the instantaneous dynamics. This is possible by minimizing the root mean square error in the time derivative of the resolved quantities. Here, the LES model is written in terms of a conditional average, an average over all the instantaneous fields that correspond to the same LES solution when filtered. As a result, the modeled subgrid stresses do not necessarily have to show a high spatial correlation with the true subgrid stresses. In addition to these modeling approaches, the interaction of the numerical method with the under-lying subgrid model also poses several challenges. The operating filter size in all these models is, in general, very close to the grid size. Consequently, the truncation error due to the numerical method can be of a similar magnitude to the subgrid model term. To decouple them, various explicit filtering approaches\textsuperscript{28-31} have also been explored in the literature. The usage of dissipative numerical methods alone to perform LES without any explicit subgrid model has also gained popularity\textsuperscript{32-36} recently in the form of implicit-LES.

Another category of subgrid models is based on the mathematical structure of the PDEs. One such mathematical formalism is the variational multiscale (VMS) method\textsuperscript{2,37,38} which formally separates the evolution equation of the coarse scales from the evolution equation of the fine-scales. The final VMS modeling step involves writing an explicit expression for the fine-scales approximately in-terms of the coarse-scale residual. These methods\textsuperscript{2,39-42} were first developed to stabilize finite element methods for linear problems involving advection but later extended to nonlinear problems\textsuperscript{10-15} as subgrid models. Analogous model forms for the subscales in terms of the coarse-scale residual can also be derived for the nonlinear problems if the Mori-Zwanzig (M-Z) formalism is used with the VMS method\textsuperscript{16,43-46} In this work, we attempt to discover such VMS model forms directly from data using deep learning tools, and use them for predictive modeling.

An alternate class of methods in structural subgrid modeling is based on the approximate deconvolution method (ADM).\textsuperscript{47,48} In the ADM approach, the filtered state variables are approximately deconvolved using ADM operators approximated by a truncated series expansion of the inverse filter. The nonlinear terms in the governing equations are then computed using these deconvolved state variables.

The availability of high resolution data from numerical simulations and experiments in the past decade has led to an interest in data-based modeling. Applications of machine learning augmentations have been used in RANS (e.g., References \textsuperscript{46,49-51} and LES (e.g., References \textsuperscript{52-55}). On the LES front, Maulik et al.\textsuperscript{56} used machine learning to classify and blend different LES models to select the most accurate model at run-time. Yang et al.\textsuperscript{57} used physics-informed features to improve the performance of equilibrium LES wall models in nonequilibrium cases. Similarly, many other notable attempts to improve LES models using data have also been made by Maulik et al.,\textsuperscript{56,58,59} Beck et al.,\textsuperscript{55} Sarghini et al.,\textsuperscript{52} Ghamara and Hattori,\textsuperscript{53} Wang et al.,\textsuperscript{60} and many more.\textsuperscript{26,54,61} These data-driven techniques have also found application in developing closures for reduced-order models (ROMs).\textsuperscript{62-65}

Very recently, super-resolution of turbulent-flow fields has been pursued using neural networks.\textsuperscript{56-71} Xie et al.\textsuperscript{66} and Fukami et al.\textsuperscript{69} appear to be the first to introduce this idea in fluid dynamic applications. These were followed by Deng et al.\textsuperscript{67} who improved traditional GAN performance by augmenting the model architecture. Improvements in flow field reconstruction were shown by Liu et al.,\textsuperscript{68} using both spatial and temporal information. Fukami et al.\textsuperscript{71} performed super-resolution and in-betweening to reconstruct a highly-resolved space-time solution using two low-resolution snapshots taken at the start $t$ and the end $t + \Delta t$ of an interval. These models have demonstrated an ability to reconstruct fine scales from highly coarse-grained data for either the same or similar data-set on which they have been trained both in
a supervised and an unsupervised setting. However, applying the trained models to super-resolve coarse flow-fields at different Reynolds numbers or another part of the flow, is relatively unexplored.

The idea of using the super-resolved field to compute the closure terms is similar to the ADM approach. However, compared to the approximately deconvolved solution that lies on the same mesh as the filtered solution, the super-resolved solution lies on a higher resolution mesh. A super-resolution model capable of reconstructing fine-space data for a case where the fine-space data already exists and the same information is used for training has no use in a predictive setting. This work attempts to improve predictive capabilities by bringing in generalizable model forms and features. In addition to the generalizability of these models, the definition of a coarse-space solution is ambiguous. This ambiguity is because a variety of low-fidelity data, including LES solution, obtained using finite difference method (FDM), finite volume method (FVM), spectrally filtered DNS solution, and stabilized finite element (FE) solution on a coarse grid qualify as coarse-solutions. The nodal or modal values in each of these methods represent different quantities. For example, Fukami et al. used the max-pooling operation to obtain coarse data. Consequently, the trained model is dependent on the type of method or filter used to generate data, and the mapping learned by the network has no formal basis. To resolve this ambiguity, we define both our coarse space and our fine space in terms of the $L_2$ projection of the DNS solution on low and high order polynomial basis functions in a similar spirit to the Variational Multiscale Method. As a result, the trained model will approximate the function that maps the $L_2$ projection of the DNS data on the two subspaces. Additionally, the model should be preferably compact and applied patch-wise rather than on the entire flow. This is because there is no guarantee that the coarse data that needs to be super-resolved has the same size as that of the input layer, and interpolating it back to the network size defeats the purpose of super-resolution. In this work, we develop N-N closures that are: (i) capable of extrapolating to unseen flow conditions and resolutions; (ii) use nondimensional features rather than dimensional features for better generalizability; and (iii) can be applied patch-wise rather than on the entire field.

The outline of this article is as follows: We introduce the VMS methodology in Section 2. In Section 3, we derive VMS consistent features. In Section 4, we propose a model form and a new network architecture for learning it. We describe the procedure of generating training data in Section 5. In Section 6, we apply our approach to the linear advection problem both in an online (numerical method) and offline (super-resolution) setting. In Section 7, we evaluate the performance of our super-resolution approach to the turbulent channel flow. Perspectives on the broader challenge of data driven modeling, and on the present work is shared in Section 8. Finally, we summarize this article in Section 9.

2 | THE VARIATIONAL MULTISCALE METHOD

This section summarizes the variational multiscale method (VMS), which was originally presented by Hughes et al. As discussed previously, this method has been extensively used for developing closures for both linear and nonlinear PDEs. This section will only discuss it in the context of a linear problem and use it as a guiding principle for feature selection and in shaping the network architecture. As discussed by Hughes et al., the development of VMS closure can be broadly categorized into two different subsections based on the type of basis functions used which are detailed below, along with a context for super resolution.

2.1 Smooth case

In the "smooth case", the basis functions are sufficiently smooth so that the distributional effects may be ignored. Both the Fourier basis and the Chebyshev spectral basis qualify as a smooth basis. For this case, consider the following PDE on an open and bounded domain $\Omega \subset \mathbb{R}^d$, where $d \geq 1$ is the dimension of the problem, with a smooth boundary $\Gamma = \partial \Omega$:

$$\mathcal{L}(u) = v \quad \text{in} \quad \Omega, \quad u = g \quad \text{on} \quad \Gamma, \quad (1)$$

where the operator $\mathcal{L}$ can be linear or nonlinear, the functions $v : \Omega \rightarrow \mathbb{R}$ and $g : \Gamma \rightarrow \mathbb{R}$ are given. The variational form of the above PDE is given by

$$(\mathcal{L}(u), w) = (v, w), \quad (2)$$
such that \( u \in \mathcal{V} \) for all \( w \in \mathcal{V} \), where \((\cdot, \cdot)\) denotes the \( L_2(\Omega) \) inner product, and \( \mathcal{V} \equiv H^1(\Omega) \) is the Sobolev space. The solution and weighting space are decomposed as follows:

\[
\mathcal{V} = \mathcal{V}_h \oplus \mathcal{V}',
\]

(3)

where \( \oplus \) represents a direct sum of \( \mathcal{V}_h \) and \( \mathcal{V}' \). Applying the VMS operation, we have

\[
(L(u_h + u'), w_h) + (L(u_h + u'), w') = (v, w_h) + (v, w'),
\]

(4)

where \( u_h, w_h \in \mathcal{V}_h \) and \( u', w' \in \mathcal{V}' \). While the above equation is valid for both nonlinear and linear equations, further simplifications can be made if the differential operator is assumed to be linear. To this end, using the linear independence of \( w_h \) and \( w' \), and taking the differential operator to be linear, we obtain the coarse and fine equations:

\[
(L(u_h), w_h) + (L(u'), w_h) = (v, w_h)
\]

(5)

\[
(L(u'), w') = -(L u_h - v, w').
\]

(6)

The coarse and fine scale equations can be re-written as:

\[
(L(u_h) - v, w_h) = -(L(u'), w_h)
\]

(7)

\[
\Pi' L(u') = -\Pi'(L u_h - v),
\]

(8)

where \( \Pi' \) is the \( L_2 \)-projector on the fine-scale basis functions. The Green’s function corresponding to the adjoint of the fine-scale problem is found by solving the following equations

\[
\Pi' L^*(g'(x,y)) = \Pi' (\delta(x - y)) \quad \forall x \in \Omega ; \quad g'(x,y) = 0 \quad \forall x \in \Gamma.
\]

(9)

The fine-scale can be obtained by super-position as follows:

\[
u'(y) = -\int_{\Omega} g'(x,y)(L u_h - v)(x)d\Omega_x.
\]

(10)

In the current super-resolution approach, we do not seek \( u' \). Rather, we seek \( u'_j \), which is the optimal projection of \( u' \) on the finer-space \( w_j \). The space of functions in \( w_j \) is finer in-comparison to \( w_h \) or represents a different kind of space. To this end, the optimal projection of \( u' \) on \( w_j \) is given by:

\[
u'_j(y) = \Pi_j u'(y) = -\int_{\Omega} \Pi_j(g'(x,y))(L u_h - v)(x)d\Omega_x.
\]

(11)

where \( \Pi_j(g'(x,y)) \) is \( L_2 \)-projection of \( g'(x,y) \) on \( w_j(y) \). For this case, the fine space can be constructed using higher wavenumber Fourier modes or higher-order Chebyshev polynomials.

2.2 Rough case

In the “rough case”, the lack of continuity of derivatives at element interfaces requires us to account for the distributional effects explicitly.2 This case is typical of the finite element method, where piece-wise continuous polynomial functions are used within each element. Similar to the “smooth case”, Hughes et al.2 showed that the exact form of subscales in the case of finite elements is given by:

\[
u'(y) = -\sum_e \left( \int_{\Omega_e} g'(x,y)(L u_h - v)(x)d\Omega_x + \int_{\Gamma_e} g'(x,y)(b u_h)(x)d\Gamma_x \right).
\]

(12)

It has to be mentioned that the “smooth case” can be considered as a limiting case of the “rough case” when a single element is used and element interfaces are not present. The subscale solution depends on the coarse-scale inside the
element and its neighbors. Applying the projection operator on both sides of Equation (12) we obtain

$$u_f'(y) = \Pi_f u'(y) = -\sum_c \left( \int_{\Omega_c} \Pi_f (g'(x, y))(L u_h - v)(x) d\Omega_x + \int_{\Gamma_c} \Pi_f (g'(x, y)) (b u_h)(x) d\Gamma_x \right).$$

(13)

An approximation to the above equation is given in the form of compact bubble functions which vanish at the element boundaries. For 1-D linear problems, solving the element Green’s function leads to the coarse-scale solution being the endpoint interpolant of the actual solution. Assuming that the coarse-scale is given in the form of the endpoint interpolant of the true solution, the fine-scale solution is given by

$$u_f'(y) = \Pi_f u'(y) = -\int_{\Omega_e} \Pi_f (g'(x, y))(L u_h - v)(x) d\Omega_x.$$  

(14)

A point to note is that application of the projection operator $\Pi_f(y)$ on the element Green’s function $g'(x, y)$ leads to the reduction of dimension only in $y$, that is,

$$g_f'(x, y) = \Pi_f g'(x, y) = \sum_i g_{e,i}'(x) \psi_i(y/h),$$

(15)

where the basis coefficients $g_{e,i}'(x)$ are functions of $x$, which are not necessarily polynomials. However, if the polynomial order $p_e$ of the coarse-scale is given, the coarse-scale residual for (e.g.,) the convection-diffusion equation will be of polynomial order $p_e - 1$. Thus, projecting $\sum g_{e,i}'(x)$ onto the space of polynomials with order $p_e - 1$ and representing $g'(x, y)$ onto tensor-product basis functions in $x$ and $y$ is sufficient. The polynomial order of $y$ is determined by the polynomial order of fine-scales, whereas the polynomial order of $x$ is decided by the maximum polynomial order arising in the coarse-scale residual. For the convection-diffusion problem given by the following differential operator:

$$L \equiv a \frac{d}{dx} - \kappa \frac{d^2}{dx^2},$$

the element Green’s function is given by:

$$g(x, y) = \begin{cases} 
C_1(y) \left( 1 - e^{-2ax/h} \right), & \text{if } x \leq y, \\
C_2(y) \left( e^{-2ax/h} - e^{-2a} \right), & \text{if } x \geq y,
\end{cases}$$

(16)

where $a$ is the cell Peclet number $a = \frac{ah}{2}$. The constants $a$ and $k$ are the advection velocity and the diffusion coefficient, respectively, and the functions $C_1(y)$ and $C_2(y)$ are defined as

$$C_1(y) = \frac{1 - e^{-2a\left( 1 - (y/h) \right)}}{a \left( 1 - e^{-2a} \right)}, \quad C_2(y) = \frac{e^{2a(y/h)} - 1}{a \left( 1 - e^{-2a} \right)}.$$  

(17)

The element’s Green’s function approximated using different order tensor-product basis functions in $x$ and $y$ is shown in Figure 1. The basis functions used here to approximate the subscale do not necessarily vanish at the element boundary. However, one can also select them to ensure that the subscales vanish at the element boundary, similar to a bubble function, as shown in Figure 2. Hence, when the input and output order is fixed, the Green’s function for the fine-scales can be represented in a finite number of dimensions. This makes it easier to learn the mapping between the coarse-scale and fine-scale solutions. On further inspection, we find that $ag'(x, y)$ is a function of $a, x/h$ and $y/h$. Consequently, $ag'(x, y)$ can be written as follows:

$$ag'(x, y) = \sum_{ij} g_{ij}'(a) \phi_i(y/h) \psi_j(x/h),$$

(18)

where $\phi_i(y/h)$ and $\psi_j(x/h)$ are 1-D basis functions constituting the tensor-product basis functions. Substituting back in Equation (14), we obtain:

$$u_f'(y) = -\sum_i \phi_i(y/h) \sum_j \int_{\Omega_e} g_{ij}'(a) \psi_j(x/h) \left( \frac{d u_h}{dx} - \kappa \frac{d^2 u_h}{a dx^2} \right) d\Omega_x.$$  

(19)
FIGURE 1  $L_2$ optimal approximation of the fine-scale Green’s function on various tensor-product polynomial basis function $g’$ for different cell Peclet number $\alpha$.

FIGURE 2  $L_2$-optimal approximation of the fine-scale Green’s function on $p_x = 0$ and $p_y = 3$ basis such that $u'_f(y)$ is zero on the boundaries.

Next we introduce a constant $u_m$ in the coarse-scale residual calculation as follows:

$$u'_f(y) = -\sum_l \phi_l(y/h) \sum_j \int_{\Omega_x} g^j_l(\alpha) \psi_j(x/h) \left( \frac{d(u_h - u_m)}{dx} - \frac{\kappa}{\alpha} \frac{d^2(u_h - u_m)}{dx^2} \right) d\Omega_x.$$  

(20)

It can be observed that subtracting $u_m$ from $u_c$ does not introduce any error in Equation (20) because we are taking derivatives of a constant. Writing the coarse-scale solution and the constant $u_m$ in terms of the nodal Lagrange basis
functions $u_h = \sum_k u_{h,k}w_{h,k}(x/h)$ and $u_m = \sum_k u_m w_{h,k}(x/h)$, and substituting in Equation (20) we get

$$u_j'(y) = -\sum_i \phi_i(y/h) \sum_k \int_{\Omega_h} \left( g_{ij}^\alpha(\alpha)\psi_j(x/h) \sum_k (u_{h,k} - u_m) \left( w_{h,k}'(x/h) - \frac{1}{\alpha} w_{h,k}''(x/h) \right) \right) \frac{d\Omega_h}{h}.$$  

(21)

Dividing both sides with a normalizing parameter $u_{rms}$ (which will be defined later) and re-arranging we obtain:

$$\frac{u_j'(0)}{u_{rms}} = -\sum_i \phi_i(y/h) \sum_k \frac{u_{h,k} - u_m}{u_{rms}} \int_{\Omega_h} \left( w_{h,k}'(x/h) - \frac{1}{\alpha} w_{h,k}''(x/h) \right) \sum_j \left( g_{ij}^\alpha(\alpha)\psi_j(x/h) \right) \frac{d\Omega_h}{h},$$

where the integral $\int_{\Omega_h} \left( w_{h,k}'(x/h) - \frac{1}{\alpha} w_{h,k}''(x/h) \right) \sum_j \left( g_{ij}^\alpha(\alpha)\psi_j(x/h) \right) \frac{d\Omega_h}{h}$ is a function of $\alpha$ which can be written as linear combinations of $g_{ij}^\alpha(\alpha)$ and $\frac{\psi_j(x/h)}{\alpha}$.

Remark 1. With this normalization, the problem of learning the subscales reduces to learning the projected element Green’s function basis coefficients $g_{ij}^\alpha(\alpha)$ which define the shapes of the surfaces plotted in Figure 1.

Equation (22) also suggests that the normalized subscale basis coefficients can be written as sum of the products of normalized coarse scales basis coefficients and functions of $\alpha$. Further, the number of such functions of $\alpha$ needed to be learnt is finite. Equation (22) also suggests for a linear problem that the subscales depend linearly on the coarse-scale basis coefficients. The dependence on $\alpha$, however, can be nonlinear. These insights will be used later in Sections 3 and 4.

3 \ VMS-INSPIRED FEATURE SELECTION

In this section, we will derive an appropriate feature set and the network architecture for our model. To demonstrate the action of the super-resolution operator, we assume the coarse space to be composed of piece-wise linear polynomials and the governing PDE to be again the linear convection-diffusion equation given by the following differential operator and boundary condition

$$\mathcal{L} \equiv \frac{d}{dx} - \kappa \frac{d^2}{dx^2} \quad \text{in} \quad \Omega = [0, L], \quad u = 0 \quad \text{on} \quad \Gamma = \{0, L\}.$$  

(23)

For a linear 1-D element of size $h$ existing between $0 \leq x \leq h$ in its local co-ordinates, the coarse solution $u_h$ in terms of the end-point values $u(0)$ and $u(h)$ is given by:

$$u_h(x) = u(0)(1 - x/h) + u(h)(x/h).$$

(24)

The space in which the fine scales are approximated can be a discontinuous finite element space or a bubble function space. The $L_2$-optimal approximation of the fine scales $u'$ on any polynomial space existing inside an element is given by the projection $\Pi_f$:

$$u_j'(y) = \Pi_f u'(y) = -\int_{\Omega_h} \Pi_f (g'(x,y)\alpha) \left( \frac{u(h) - u(0)}{h} \right) d\Omega_h.$$  

(25)

The coarse solution considered here is an endpoint interpolant of the true solution. In that case, one can determine the exact form of the subscale inside the element by assuming the endpoint values as Dirichlet boundary conditions and by solving the equation inside the element:

$$u'(y) = (u(h) - u(0)) \left( \frac{1 - e^{2\alpha\xi}}{1 - e^{2\alpha}} - \frac{y}{h} \right).$$

(26)
The simplest approximation of the bubble function is obtained by projecting it on a \( p_0 \) discontinuous space inside the element that is,

\[
\mu'_0(y) = \int_0^h (u(h) - u(0)) \left( \frac{1 - e^{2\alpha y}}{1 - e^{2\alpha h}} - \frac{y}{h} \right) dy \\
= -a \frac{(u(h) - u(0))}{h} \frac{h}{2a} \left( \coth(\alpha) - \frac{1}{\alpha} \right),
\]

(27)

where the first part is the residual when linear basis functions are used, and the second part is the form of the stabilization parameter \( \tau \) commonly used in stabilized methods. For the linear CG finite element method, Equation (27) represents the closure for obtaining the nodally exact solution. Equation (27) can also be obtained by first projecting the elements Green’s function on \( p \) element that is,

\[
g'_{f,0}(x,y) = \frac{1}{h^2} \int_{\Omega} \int_{\Omega} g'(x,y) d\Omega_x d\Omega_y = \frac{1}{2a} \left( \coth(\alpha) - \frac{1}{\alpha} \right),
\]

(28)

and using this result in Equation (25) to evaluate the subscale as follows:

\[
\mu'_0(y) = \Pi_{f,0} \mu'_0(y) = -\int_{\Omega} \frac{1}{2a} \left( \coth(\alpha) - \frac{1}{\alpha} \right) a \frac{(u(h) - u(0))}{h} d\Omega_x = -a \frac{(u(h) - u(0))}{h} \frac{h}{2a} \left( \coth(\alpha) - \frac{1}{\alpha} \right).
\]

(29)

Next, we define the mean and r.m.s quantities of the coarse solution in an element as follows:

\[
u_m = \frac{\int_0^h u_0 d\Omega}{h}; \quad u_{\text{rms}} = \sqrt{\frac{\int_0^h (u_0 - u_m)^2 d\Omega}{h}}.
\]

(30)

An important observation is that the solution is independent of the mean \( u_m \):

\[
u'(y) = ((u(h) - u_m) - (u(0) - u_m)) \left( \frac{1 - e^{2\alpha y}}{1 - e^{2\alpha h}} - \frac{y}{h} \right).
\]

(31)

If our approximation space is linear, then \( u_m \) and \( u_{\text{rms}} \) are given by:

\[
u_{m,1} \triangleq (u(0) + u(h))/2 \quad u_{\text{rms},1} \triangleq \frac{|u(h) - u(0)|}{\sqrt{12}}.
\]

(32)

Re-arranging this form we get:

\[
u'_0(y) \quad \frac{u'_0(y)}{u_{\text{rms},1}} = \sqrt{3} \left( \frac{1}{\alpha} - \coth(\alpha) \right) \text{sgn}(u(h) - u(0)),
\]

(33)

where \( \text{sgn} \) denotes the sign function. The above equation can be simplified as follows:

\[
u'_0(y) \quad \frac{u'_0(y)}{u_{\text{rms},1}} = \begin{cases} 
\sqrt{3} \left( \frac{1}{\alpha} - \coth(\alpha) \right), & \frac{u(h) - u(0)}{|u(h) - u(0)|} \geq 0, \\
-\sqrt{3} \left( \frac{1}{\alpha} - \coth(\alpha) \right), & \frac{u(h) - u(0)}{|u(h) - u(0)|} \leq 0.
\end{cases}
\]

(34)

If we compute the the mean-subtracted basis-coefficients of the coarse solution \( u_h \) and re-scale them with the r.m.s \( u_{\text{rms},1} \) we get:

\[
u(0) - u_m \quad \frac{u(0) - u_m}{u_{\text{rms},1}} = \sqrt{3} \frac{u(0) - u(h)}{|u(h) - u(0)|}, \quad u(h) - u_m \quad \frac{u(h) - u_m}{u_{\text{rms},1}} = \sqrt{3} \frac{u(h) - u(0)}{|u(h) - u(0)|}.
\]

(35)
These parameters determine the sign of the subscale in Equation (34). Hence, the magnitude of the subscales is fully determined by the physics-informed parameter \( \alpha \) and its sign (phase) is determined by these parameters. Consequently, an appropriate choice of the feature will be:

\[
\frac{u'_0(y)}{u_{\text{rms},1}} = f\left( \alpha, \frac{u(0) - u_m}{u_{\text{rms}}}, \frac{u(h) - u_m}{u_{\text{rms}}} \right).
\]

(36)

The two extra parameters, in this case, are redundant because they are the same in magnitude and opposite in sign. Hence, only one parameter can be used:

\[
\frac{u'_0(y)}{u_{\text{rms},1}} = f\left( \alpha, \frac{u(h) - u(0)}{u_{\text{rms}}} \right).
\]

(37)

Although Equations (27) and (37) are identical, the generalizability of a neural-network model is considerably affected by the choice of the feature set and the normalization process. For example, the model form

\[
\frac{u'_0(y)}{u_{\text{rms},1}} = p\left( \alpha, \kappa, h, u(h), u(0) \right),
\]

(38)

does not utilize the idea that only specific combinations of \( \alpha, \kappa, h \) that is, cell Peclet number \( \alpha \) affect the subscale solution distinctly. To train this model, a big data-set with a large range in the values of the input parameters is required. This equation is also not invariant to the units or the scaling used for the input parameters. The proposed model form in Equation (37) tries to address some of these challenges. An important point to note is that the functional form presented in Equation (37) is derived for the advection-diffusion problem. However, this structure will serve as the inspiration for applying this method to other nonlinear problems, as detailed in the following section.

### 4 LEARNING VMS-CONSISTENT SUBSCALES

In this section, we will attempt to derive a general model structure that can be used to learn subscales arising in a wide range of multiscale problems. In case of super-resolution, the subscales contain the fine-scale information that is absent in the lower-resolution image. When used as the closure, they are responsible for modeling the effect of unresolved fine-scales on the coarse-scales. Irrespective of the mode of application, the model structure should not be different. As a first step, we learn the mapping presented in Equation (37) and compare it to the analytical solution. Data is first obtained by solving the equation at different Peclet numbers \( Pe \) on very fine meshes (we refer to this as the DNS) for training the network. Similarly, we can also generate data by dividing a single high \( Pe \) DNS case into multiple cases with different element sizes, that is, different cell Peclet numbers. The coarse-scale is obtained in the form of end-point interpolant, and the approximation to the subscale is then computed for each such element numerically as shown in Figure 3. A small network with a size of \( 3 \times 8 \times 8 \times 8 \times 1 \) is then trained using this data. The learning rate was set to 0.0005 and the optimization was run for 10,000 epochs. Figure 4 shows the comparison between the subscale obtained analytically versus that learned purely from data using neural network (N-N). It can be observed that the small network could learn the analytical solution accurately. The discussion in this section was mainly focused on learning the subscales. In appendix A, we demonstrate how these subscales can be used as closures for the CG finite element method and further extend them to high order discretizations.

One way of obtaining the exact form of the subscale is by keeping our approximating space of the coarse-scale the same (linear) and increasing the order of the discontinuous space \( p \) in which \( u' \) is approximated. In limit, \( p \to \infty \) the approximate subscale should approach \( u'(y) \). However, the order of the polynomial \( p \) required to learn the function increases when \( Pe \) is increased. By limiting the output order \( p \) to the order of super-resolution, we are reducing the complexity and size of the network, as discussed in the last section. This is because, when \( \alpha \to \infty \), \( f(\alpha) = \frac{u'_0}{u_{\text{rms}}} = \sqrt{3} \left( \coth(\alpha) - \frac{1}{\alpha} \right) \) is well-behaved, whereas if \( u'(y) \) given by the Equation (26) is learnt directly as a function of \( y \) and \( \alpha \), the function becomes steep at \( y = h \) for \( \alpha \to \infty \). A solution to this problem is to use features such as \( \frac{1-e^{2\alpha}}{1-e^{2\alpha}} \) as inputs. However, this restricts the method to one kind of problem. Similarly, the optimal form of the subscale on discontinuous \( p = 1 \) basis
FIGURE 3  Computation of \( u'_0 \) by element-wise \( L_2 \)-projection of the subscale \( u' \) on the \( p = 0 \) polynomial space.

FIGURE 4  Comparison of subscales magnitude as a function of cell Peclet number \( \alpha \) obtained analytically versus that learnt from data using a N-N.

functions is given by:

\[
\frac{u'_1(y)}{u_{\text{rms},1}} = C'_1(\alpha)\psi_1(y/h) + C'_2(\alpha)\psi_2(y/h). \tag{39}
\]

The form of the function to be learned for this case is:

\[
[C'_1, C'_2] = f_{\text{NN}}(\alpha). \tag{40}
\]

The above analysis was performed for the continuous Galerkin (CG) method, but these ideas can be extended to the discontinuous Galerkin (DG). Retaining the same structure, we extend the technique to nonlinear/linear problems for both CG/DG types of basis functions as follows:

\[
[C'_{1,p_1}, C'_{2,p_1}, \ldots, C'_{(p_1+1)d,p_1}, \ldots, \bar{C}'_{1,p_s}, \bar{C}'_{2,p_s}, \bar{C}'_{(p_s+1)d,p_s}, \ldots, \bar{C}'_{(p_s+1)d,p_s}]_m, \ldots, \tag{41}
\]

where \( p_1 \) and \( p_s \) are the polynomial orders of the spaces in which the subscale and coarse-scale are optimally represented, and \( d \) denotes the dimension of the problem. This function, apart from \( \alpha \) (equivalent to cell \( Re/\text{cell Pe} \)) also contains the basis coefficients of the current element and its neighbors. The term \([\bar{C}'_{1,p_s}, \bar{C}'_{2,p_s}, \ldots, \bar{C}'_{(p_s+1)d,p_s}]_m\) with subscript \( m \) denotes the mean subtracted normalized basis coefficient of \( m^{th} \) neighbor. The neighbor information is critical when
discontinuous basis is used, or when bubble function approximation are not employed in CG, or nonlocal transfer of information happens from outside the element. These coefficients are first subtracted with the coarse scale mean $u_m$ and then normalized with the coarse scale r.m.s $u_{\text{rms}}$ as done previously. The output of the function $\left[ C^r_{1,p_1}, C^r_{2,p_1}, \ldots, C^r_{(p_r-1)^d,p_1} \right]$ denotes the basis coefficients of the subscale that has been normalized with $u_{\text{rms}}$ only that is,

$$\left[ \tilde{C}_{1,p_1}, \tilde{C}_{2,p_1}, \ldots, \tilde{C}_{(p_r-1)^d,p_1} \right]_m = \left[ C_{1,p_1} - u_m, C_{2,p_1} - u_m, \ldots, C_{(p_r-1)^d,p_1} - u_m \right]_m / u_{\text{rms}}. \quad (42)$$

where $C_{ij}$ denotes the actual basis coefficients. The quantities used for shifting and nondimensionalizing the input parameters, that is, $u_m$ and $u_{\text{rms}}$ respectively, and nondimensionalizing the output parameters $u_{\text{rms}}$ are calculated using the coarse-scale solution in the center element only. As will be seen later in this article, the nondimensionalization process is critical for the N-N model to generalize. The output coefficients are finally re-scaled with $u_{\text{rms}}$ and added to the coarse-scale to obtain the super-resolved solution as follows:

$$u_{sr} = u_{pc} + u'_{pc} = u_{pc} + u_{\text{rms}} \sum_{i} C^r_{i,p_i} \psi_{i,p_i}, \quad (43)$$

where, $\psi_{i,p_i}$ denotes basis function corresponding to the $i$th node or mode. Division by $u_{\text{rms}}$ in Equation (42) is a problem when $u_{\text{rms}}$ is precisely equal to zero. However, adding a small positive number $\epsilon$ to $u_{\text{rms}}$ while dividing was sufficient for all the cases presented below.

**4.1 The variational super-resolution network architecture**

In addition to the model features, the model architecture can be made consistent with the VMS formulation. As proposed in Equation (61), the input to the model are the physics-informed parameters such as the cell Péclet number $\alpha$, along with the normalized mean-subtracted coarse-scale basis coefficients of an element and its neighbor. The output to the network are the normalized subscale basis coefficients in that particular element. The physics-informed parameter can also be other nondimensional numbers such as the CFL number or the cell Reynolds number $Re_A$ specific to the problem. Given these sets of input and output features, many possible ways of embedding them into the model exist. Figure 5 shows two different kinds of network architectures to achieve this.

The traditional approach is based on the idea of training one single fully-connected N-N (FNN) with both the normalized coarse-scale basis coefficients and the physics informed-parameter as inputs. If the normalized subscale basis coefficients are denoted by $u'$, the normalized input coarse basis coefficients of the element and its neighbors as $u_c$ and the physics-informed parameter $\alpha$, then the traditional model is given by:

$$u' = f_{\text{FNN}}(\alpha, u_c), \quad (44)$$

where $f_{\text{FNN}}$ denotes a FNN. Another approach also called the variational super-resolution N-N (VSRNN), is based on a multiplicative strategy in which the fine-scales are approximated by a sum of products of individual functions of the parameters and the coarse-scales. The model form can be summarized as follows:

$$u' = f_{\text{FNN}}(g_\alpha \odot g_u), \quad ; \quad g_\alpha = h_{\text{FNN}}(\alpha), \quad ; \quad g_u = k_{\text{FNN}}(u_c). \quad (45)$$

where $f_{\text{FNN}}$, $h_{\text{FNN}}$ and $k_{\text{FNN}}$ denote three different FNNs. The symbol $\odot$ denotes element-wise multiplication between two vectors of the same size. This architecture is inspired by Equation (22) and the analytical solution of the subscale provided in Equation (33). In this case, $g_\alpha$ (Part B) learns the dependence of $\alpha$ that is, $\sqrt{3} \left( \frac{1}{a} - \coth(\alpha) \right)$ and $g_u$ (Part A) learns the dependence of the normalized coarse-scale basis coefficients, that is, $\text{sgn}(u(h) - u(0))$.

In Sections 2–4 and Appendix A, we tried to develop insights into the working of the VSRNN and demonstrated its application as a closures for the CG method. In Sections 5–7, we use the VSRNN to perform super-resolution and subgrid modeling for the DG method. This does not imply that the exact subscales used for the CG method in Sections 2–4 are re-used in Sections 5–7. Only the model form has been assumed to be the same which is finally trained on the correct subscale that is consistent with the DG approach. Details about the scale-decomposition used for the DG method are detailed in Section 5.
For training these networks, the Adam optimizer\textsuperscript{75} that is part of the Keras library\textsuperscript{76} has been used. The loss function is chosen to be the mean squared error (MSE).

## 5 | DATA GENERATION

The generation of proper training and testing data is as critical as the model architecture and features. For example, low-resolution data can be obtained from a variety of high fidelity sources (simulations and experiment) and can be coarse-grained. There is no guarantee that a model trained to perform super-resolution of the filtered solution will be useful unless the filtering operation is consistent with the underlying numerics. To this end, consider the VMS decomposition of the full-order solution $u$ as follows:

$$u = u_h + u', \quad (46)$$

where $u_h \in \mathcal{V}_h$ and $u' \in \mathcal{V}'$. The vector space of functions $\mathcal{V} \equiv \mathcal{H}^1(\Omega)$ is a Sobolev space where the functions and their derivative are square-integrable. This space is now decomposed as follows:

$$\mathcal{V} = \mathcal{V}_h \oplus \mathcal{V}', \quad (47)$$
where \( \oplus \) represents a direct sum of \( \mathcal{V}_h \) and \( \mathcal{V}' \). Let us also define \( \mathcal{T}_h \) to be a tessellation of domain \( \Omega \) into a set of nonoverlapping elements, \( K \), each having a subdomain \( \Omega_k \) and boundary \( \Gamma_k \). \( \mathcal{V}_h \) is now defined as:

\[
\mathcal{V}_h \triangleq \{ u \in L^2(\Omega) : u|_{T} \in P^k(T), T \in \mathcal{T}_h \},
\]

where the space of polynomials up to degree \( k \) is denoted as \( P^k \). Defining \( \mathcal{V}_h \) in this manner allows discontinuity in the solution across element boundaries. Given \( u \) from the high-fidelity simulation, our goal is to find the optimal representation of \( u \) in the coarse subspace \( \mathcal{V}_h \). In our case, we will use the \( L^2 \) projection to obtain \( u_h \) which minimizes the value of \( ||u - u_h||_2^2 \). This problem is equivalent to the problem of finding \( u_h \in \mathcal{V}_h \) such that

\[
(u, w_h) = (u_h, w_h) \quad \forall w_h \in \mathcal{V}_h.
\]

For example, to generate training data for Section 7, we use direct simulation (DNS) results for a channel flow at \( Re \approx 950 \). The 3-D data is sliced into many 2-D planes at different \( y^+ \) locations, and projection is performed in 2-D for simplicity. The fine space and coarse space's polynomial orders are chosen to be 3 and 1, respectively, that is, we are super-resolving \( p = 1 \) results to \( p = 3 \) as shown in Figure 6.

The computation of terms on the RHS of the Equation (49) requires special care. Although \( u \) has been assumed to exist in an infinitely high dimensional space, in reality, it is not. For example, the Kolmogorov scale \( \eta \) dictates the size of the smallest size eddy and the size of \( u \). Although the dimension of \( u \) is finite, it is enormous when compared to \( u_h \) because the size of our finite element grid \( h \) is much greater than \( \eta \). Hence, to accurately compute these terms, the DNS data is first
interpolated using cubic-splines to a much finer-grid $O(\eta)$ and then the inner products with the coarse finite element basis function (having dimension $O(h)$) are computed on these fine-meshes using the Simpson’s Rule. Interpolation of DNS was done to ensure that the projected solution $u_h$ did not change significantly due to the numerical integration scheme. Sample $L_2$-projected snapshots of the DNS data on elements of different sizes and orders are shown in Figure 6.

6 | APPLICATION TO LINEAR ADVECTION

In this section, we apply our super-resolution methodology to the linear advection equation in the domain $\Omega \subset \mathbb{R}$ with the boundary $\Gamma = \partial \Omega$ as follows

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0, \quad (50)$$

with time $t \in (0, T]$, and spatially periodic boundary conditions on $\Gamma$. The parameter $a$ denotes the advection velocity. The required training data is generated by $L_2$-projecting the true solution on coarse and fine spaces. Unlike traditional methods, in which only the spatial term in the PDE is discretized using finite elements, we will consider 2-D space-time finite elements spanned by $p = 1, 2$ degree tensor-product Lagrange basis functions in space and time. The word “traditional” that has been used in this section to denote the original DG method should not be confused with the traditional NN architecture described in Section 4.1. The goal is to investigate the application of our super-resolution method in two different settings. First, as a model to super-resolve coarse low-order finite element data to high-order finer finite element data. Second, as a method to improve the existing finite element method for this problem in a predictive setting on a problem with a very different initial condition in comparison to the training data.

6.1 | Super-resolution

To train the super-resolution model, we will generate the true solution on a fine grid. The initial condition for this case is $\sin(x) + \sin(2x) + \sin(4x)$, and the size of the grid is taken to be $N_x \times N_t : 512 \times 512$. This high-resolution mesh is to ensure that the true solution remains highly resolved on this grid. For simplicity, periodic boundary conditions are also applied in the $x$ direction. The true solution is then evaluated on all the grid points to obtain the DNS solution. The next step is to obtain the coarse $p = 1$ and fine $p = 2$ $L_2$-projected solution for different meshes having spatial and temporal element sizes $\Delta x$ and $\Delta t$ respectively as shown in Figure 7. A nondimensional parameter naturally arising in this case is the $CFL = \frac{a\Delta t}{\Delta x}$ number which is similar to Peclet number in the 1-D convection-diffusion problem. This solution is then projected on grids of various sizes corresponding to CFL numbers of 0.25, 0.5, 1.0, 2 and 4. These CFL numbers correspond to three sets of grid sizes: (i) $N_x \times N_t : 32 \times 128, 32 \times 64, 32 \times 32, 32 \times 16$, and $32 \times 8$; (ii) $N_x \times N_t : 16 \times 64, 16 \times 32, 16 \times 16$, $16 \times 8$, and $16 \times 4$; and (iii) $N_x \times N_t : 24 \times 96, 24 \times 48, 24 \times 24, 24 \times 12$, and $24 \times 6$. The choice of the initial condition, the number of mesh elements ($N_x$ and $N_t$), the wave-speed $a$, and the domain sizes can be arbitrary as long as large spread in the CFL number is present and the initial condition contains many frequencies both resolvable and not-resolvable on these meshes. For each element in these grids, the basis coefficients corresponding to the coarse-space is extracted along with its neighbors, excluding those which are part of the future time step, that is, only space-time elements present in the south, south-east, south-west, east, and west of the central coarse element are extracted. The corresponding fine-space basis coefficients are also extracted for the central element. As a first step toward normalization, a mean value $u_m$ is first computed inside the element as follows:

$$u_m = \frac{\int_{\Omega} u_t(x, t) \, dx \, dt}{\int_{\Omega} \, dx \, dt}. \quad (51)$$

Similarly, an r.m.s value is also computed inside the element:

$$u_{rms} = \sqrt{\frac{\int_{\Omega} (u_t(x, t) - u_m)^2 \, dx \, dt}{\int_{\Omega} \, dx \, dt}}. \quad (52)$$
A model is then sought in the following form

$$
\left[ C'_{1,p_1}, C'_{2,p_1}, \ldots, C'_{p_1+1,p_1} \right] = f_{\text{NN}}(\log(\text{CFL})), \left[ \tilde{C}_{1,p_1}, \tilde{C}_{2,p_1}, \ldots, \tilde{C}_{p_1+1,p_1} \right], \ldots
$$

where $C'_{ij}$ and $\tilde{C}_{ij}$ are defined in terms of $u_m$ and $u_{\text{rms}}$, similar to Equation (42). For learning this function, the VSRNN architecture is adopted with size $24 \times 12$ for part A, $1 \times 6 \times 12$ for part B, and $12 \times 9$ for the postmultiplication part, respectively. The network is first trained on data obtained by projecting a highly resolved DNS solution, as shown in Figure 7. The learning rate was set to $1\text{E}-4$ and training was performed for 5500 epochs.

In the next step, the super-resolution model is evaluated on an unseen coarse solution obtained by projecting the DNS solution for a different set of initial conditions, as shown in Figure 8. It can be observed that unless an extremely coarse model was used, the model could efficiently super-resolve unseen coarse solution to its fine-solution with minimal reconstruction error. This reconstruction error for the cases is reported in Table 1. Except for the case when the CFL number was as high as 8, the error in reconstruction $\|u_{2,\text{NN}} - u_2\|_2$ is orders of magnitude smaller in comparison to the magnitude of the subscale $\|u_{2} - u_1\|_2$. Hence, the super-resolution model is very efficient in reconstructing the fine-solution as long
as the CFL is not large. A separate model for super-resolving $p = 1$ solution to $p = 3$ solution is also trained by repeating the steps above. As reported in Table 2, a similar trend is again observed for this case where efficient reconstruction was obtained at lower CFL values. Irrespective of the large reconstruction error at CFL values of 8 and above, the magnitude of $||u_3 - u_1||_2$ is still very close to $||u_{3,NN} - u_1||_2$ for all CFL numbers. This indicates that the super-resolution model can also act as an efficient error indicator that can be used for mesh adaption. The trend in the error versus CFL is hard to understand due to the black-box nature of the data-driven methods. However, these are some of the possible reasons that can contribute to this trend: (1) the number of data points obtained from a low CFL number solution is relatively less compared to a high CFL case resulting in a biased training data-set; (2) can be purely due to the physics of the problem leading to a requirement for more number of neighboring cells in the SR model.

### 6.2 Subgrid modeling

In the previous section, we showed that the neural network could efficiently predict the subscales as long as the grid is not highly under-resolved. In this section, we will use the trained model from the previous section to improve the existing space-time based numerical method. To this end, we start with the linear advection equation in the domain $\Omega \subset \mathbb{R}$ with


| Case | $N_x$ | $N_t$ | $\Delta x$ | $\Delta t$ | CFL | $\|u_2 - u_1\|_2$ | $\|u_{2,NN} - u_1\|_2$ | $\|u_{2,NN} - u_2\|_2$ | $\frac{\|u_2 - u_1\|_2}{\|u_{2,NN} - u_1\|_2}$ | $\frac{\|u_{2,NN} - u_1\|_2}{\|u_{2,NN} - u_2\|_2}$ |
|------|------|------|------|------|------|------|------|------|----------------|----------------|
| L1   | 16   | 32   | 0.3927 | 0.19635 | 0.5 | 0.41684 | 0.41684 | 0.0018975 | 0.0663444 | 0.066345 | 0.0030201 |
| L2   | 16   | 16   | 0.3927 | 0.3927  | 1   | 0.56984 | 0.57053 | 0.0044046 | 0.090699  | 0.090809 | 0.00070106 |
| L3   | 16   | 8    | 0.3927 | 0.7854  | 2   | 1.4709  | 1.4681  | 0.0066213 | 0.23456   | 0.23411 | 0.0010559 |
| L4   | 16   | 4    | 0.3927 | 1.5708  | 4   | 3.3222  | 3.3223  | 0.016548  | 0.56392   | 0.56394 | 0.0028089 |
| L5   | 16   | 2    | 0.3927 | 3.1416  | 8   | 3.115   | 2.9975  | 1.7061   | 0.75135   | 0.723   | 0.41152 |
| L6   | 32   | 64   | 0.19635| 0.098175 | 0.5 | 0.1075  | 0.1075  | 0.00075342 | 0.017117  | 0.017117 | 0.00011991 |
| L7   | 32   | 32   | 0.19635| 0.19635 | 1   | 0.14736 | 0.14738 | 0.002046 | 0.023453  | 0.023457 | 0.00032576 |
| L8   | 32   | 16   | 0.19635| 0.3927  | 2   | 0.41684 | 0.41517 | 0.005024 | 0.0663444 | 0.066163 | 0.00079962 |
| L9   | 32   | 8    | 0.19635| 0.7854  | 4   | 1.4235  | 1.4245  | 0.009266 | 0.22699   | 0.22716  | 0.0014776 |
| L10  | 32   | 4    | 0.19635| 1.5708  | 8   | 3.3146  | 2.723   | 1.3326   | 0.5626    | 0.46219  | 0.2262 |

Table 2: Reconstruction error when super-resolved from $p = 1$ to $p = 2$ for the linear advection problem with an unseen initial condition at different CFL numbers.

The boundary $\Gamma = \partial \Omega$ as follows

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0,$$

with a periodic boundary condition at the boundary $\Gamma$ and time $t \in (0, T]$. The weak form of the above equation is obtained by multiplying it with a test function $w$ and integrating it over the space-time element as follows

$$\int_{\Omega_t} \left( \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} \right) w \, d\Omega = 0,$$

such that $u \in \mathcal{V}$ for all $w \in \mathcal{V}$. Let us also define $\mathcal{T}_h$ as a tessellation of the domain $\Omega$ into a set of nonoverlapping elements, $K$, each having a subdomain $\Omega_K$ and boundary $\Gamma_K$. The vector space of functions $\mathcal{V} \equiv H^1(\mathcal{T}_h)$ is a Sobolev space where the functions and their derivatives are square-integrable inside each element. Simplifying the previous equation, we obtain
the following:
\[
\int_{\Omega_t} \left( \frac{\partial u w}{\partial t} + \frac{\partial a u w}{\partial x} \right) d\Omega - \int_{\Omega_t} \left( u \frac{\partial w}{\partial t} + a u \frac{\partial w}{\partial x} \right) d\Omega = 0. \tag{56}
\]

Application of the divergence theorem leads to the following:
\[
\int_{\Gamma_n} (a u w + u w j)(n_t \hat{i} + n_j \hat{j}) d\Gamma - \int_{\Omega_t} \left( u \frac{\partial w}{\partial t} + a u \frac{\partial w}{\partial x} \right) d\Omega = 0. \tag{57}
\]

where \(n_t\) and \(n_s\) denote the components of the outward normal on the surface of the element along the space and time axis, respectively. The first term in the DG method is replaced with a numerical flux as follows:
\[
\int_{\Gamma_n} (F^*_x(a u, a u^-)\hat{i} + F^*_t(u, u^-)\hat{j})(n_t \hat{i} + n_j \hat{j}) n_t d\Gamma - \int_{\Omega_t} \left( u \frac{\partial w}{\partial t} + a u \frac{\partial w}{\partial x} \right) d\Omega = 0. \tag{58}
\]

The traditional space-time DG method can be obtained by applying the Galerkin approximation to the previous equation as follows:
\[
\int_{\Gamma_n} (F^*_x(a u, a u^-)\hat{i} + F^*_t(u, u^-)\hat{j})(n_t \hat{i} + n_j \hat{j}) n_t d\Gamma - \int_{\Omega_t} \left( u_h \frac{\partial w_h}{\partial t} + a u_h \frac{\partial w_h}{\partial x} \right) d\Omega = 0, \tag{59}
\]

where \(F^*_x = a u_h^-\) when \(n_t < 0\) and \(a u_h^-\) when \(n_t > 0\). The temporal flux on the bottom face is based on previous space-time slab that is, \(F^*_t = u_h^-\). The effect of the numerical fluxes is similar to that of a closure, which is dissipative in action due to the jump term, ensuring the stability of the method. The numerical fluxes were originally developed for solving the exact 1-D problem at the interface, and application to the DG method is generally made by applying it along the normal direction of the interface. However, this might not be the most optimal choice for the flux. To this end, we revisit the strong form of the DG that is, Equation (58) through the VMS approach. The coarse-scale equation corresponding to Equation (58) is given by:
\[
\int_{\Gamma_n} \left( F^*_x(a u_h + u'), a(u_h^- + u^-)\hat{i} + F^*_t(u_h + u', u_h^- + u^-)\hat{j} \right) (n_t \hat{i} + n_j \hat{j}) n_t d\Gamma 
- \int_{\Omega_t} \left( (u_h + u') \frac{\partial w_h}{\partial t} + a(u_h + u') \frac{\partial w_h}{\partial x} \right) d\Omega = 0. \tag{60}
\]

If \(V_h \bot V'\), then the effect of the subscale on the interior flux is negligible that is,
\[
\int_{\Gamma_n} \left( F^*_x(a u_h + u'), a(u_h^- + u^-)\hat{i} + F^*_t(u_h + u', u_h^- + u^-)\hat{j} \right) (n_t \hat{i} + n_j \hat{j}) n_t d\Gamma 
- \int_{\Omega_t} \left( (u_h \frac{\partial w_h}{\partial t} + a u_h \frac{\partial w_h}{\partial x} \right) d\Omega = 0, \tag{61}
\]

and the effect of un-resolved subscales is only through the flux. The true solution \(u_h + u'\) is infinite-dimensional. However, only a few of its moments are required in the form of inner-products with low-order basis functions on element faces. In this limit, we assume that the following approximation can be made: \(u_s \approx u_h + u'\), where \(u_s\) denotes the super-resolved solution of \(u_h\) that is,
\[
\int_{\Gamma_n} \left( F^*_x(a u_h + u'), a(u_h^- + u^-)\hat{i} + F^*_t(u_h + u', u_h^- + u^-)\hat{j} \right) (n_t \hat{i} + n_j \hat{j}) n_t d\Gamma 
\approx \int_{\Gamma_n} \left( F^*_x(a u_s, a u_{s}^-)\hat{i} + F^*_t(u_s, u_{s}^-)\hat{j} \right) (n_t \hat{i} + n_j \hat{j}) n_t d\Gamma, \tag{62}
\]

where \(F^*_x\) and \(F^*_t\) are traditional up-wind numerical fluxes but computed using the super-resolved state \(u_s\) instead of \(u_h\). In this work, we choose the spaces of \(u_h\) and \(u_s\) to be \(p = 1\) and \(p = 2\) respectively.

To obtain the super-resolved state, we will re-use the super-resolution network trained in the previous subsection. Two different initial conditions are chosen: (i.) Case A with an initial condition \(\sin(4x) + 2\sin(8x)\) (ii.) Case B with an initial condition \(2.5e^{-20(x-x^*)^2}\). These initial conditions are different from those used for training and testing. A comparison of different initial conditions used in training, reconstruction, and online evaluation is summarized in Figure 9. The
space-time slab is then discretized into 32 elements in the spatial direction, and the CFL value is taken to be 1.0. Figures 10 and 11 show results obtained for the super-resolution model and the traditional model compared to the optimal solution obtained by $L_2$-projection of the DNS solution for the two different initial conditions, respectively. The super-resolution model is far more accurate than the traditional method, where the subscales were recovered with a high level of accuracy for both cases. In the case of the traditional method, extrema can be seen decreasing with time considerably in comparison to the super-resolution space-time method both in Figures 10 and 11. This shows a higher dissipation characteristic of the traditional numerical method over the super-resolution method. This can also be quantitatively seen in both the Figures 12A and 13A, where the red line denoting time evolution of energy that is, $E(t) = \int_{\Omega} u_h(x,t)^2 \, d\Omega_x$ decreases in time for the traditional approach in comparison to the optimal $p = 1$ solution which is oscillatory but energy conservative. Hence, the optimal solution oscillates about a fixed value because the true solution itself conserves energy. The reason $E(t)$ oscillates for the optimal solution is because it is computed by integrating $u_h(x,t)^2$ over only space $x$ and can still vary in time across the space-time slab.

In the next step, we obtain the space-time solution by first solving the problem using the traditional approach and applying our super-resolution model on this coarse solution. In this case, the super-resolution network has no contribution to the numerical simulation stage. Rather, it is used offline when the solution is made available. As shown in Figures 12A and 13A, application of the super-resolution in an offline stage does not improve the results. On the other hand, when the super-resolved states were used to compute the flux in the numerical method online, a high level of $L_2$-optimality was also obtained in the coarse solution as shown in Figures 12B and 13B. Consequently, the corresponding super-resolved solution was also accurate and close to the $p = 2$ optimal solution.

As observed previously in this section, the super-resolution did not improve the result when it was used on the data obtained using the traditional DG method. However, when used in an online setting, it improved the performance of the numerical method. This can be explained by Figure 14, which shows the evolution of the DNS solution (red line) in an infinite-dimensional space. When one trains the super-resolution model, the mapping from a point on the green line (optimal LES) to its corresponding point on the red line (DNS) is learnt. However, when running an online numerical simulation using a traditional approach, the trajectory (blue line) taken by the solution (LES) is entirely different. The model encounters input parameters that it has not encountered during training and outputs an incorrect super-resolved solution in the evaluation stage. The blue line represents another optimal representation of DNS solution on the coarse solution space and not the $L_2$-optimal solution for which the method has been trained. However, when the super-resolved state is used to compute the fluxes, it forces the coarse resolution toward $L_2$-optimality because the closure has been formulated using the VMS method, where the coarse and fine spaces are formally defined. As shown in the second part of Figure 14, consistent numerical methods are required for the super-resolution models to work correctly. The VMS method is an ideal candidate to help us in achieving this consistency.

Remark 2. The above discussion is only applicable when $L_2$-projected coarse and fine-scale solutions are used to train the super-resolution model. The coarse-scale solution can also be chosen to be the end-point/nodal
interpolate as used for the CG cases presented in Section 4 and Appendix A of this article. For these cases, the super-resolved solution/subscale from our model used to compute the residual terms will force the coarse solution to be the end-point/nodal interpolate rather than the $L_2$-optimally projected solution.

7 | P-SUPER-RESOLUTION OF TURBULENT CHANNEL FLOW

The assessment of super-resolution models for turbulent flows poses a stern challenge. This is because, given limited measurements from a severely under-resolved coarse-grained solution such as LES, there are infinitely many possible solutions for the fine-scales. This problem is especially true for filtering using the sharp spectral filter, where the fine-scale solution is lost after filtering, and it is impossible to recover the original field from filtered data. The exact fine-scales are both functions of the coarse space and their time-history.\textsuperscript{16,43-45,78} As shown by Langford and Moser\textsuperscript{27} in their work on optimal LES, to compute the correct single-time multipoint statistical quantity of the large-scale field exact fine-scales might not be required. The Smagorinsky and the VMS models,\textsuperscript{2,10-13,15,45} which perform well online, are well-known to perform poorly in an a priori setting. Similarly, the N-N generated super-resolved field is not point-wise exact. Rather it is an optimal representation of the fine-space generating correct single-time multipoint statistics.

Our model being compact, the $L_2$ error is computed only locally in a single element. The training data consists of data from several elements, part of a 2-D DNS slice having homogeneity in stream-wise and span-wise directions. During the optimization, these errors from each element are averaged. As a result, the model output, in some sense, is an optimal representation of the fine-scales for all possible realizations. To this end, we will be using one-dimensional energy spectra that have been averaged over homogeneous directions as a measure for model accuracy in-place of the $L_2$ norm for the full field. The contours of the reconstructed fine-space solution will only be presented for qualitative purposes.
FIGURE 11  Comparison of solution obtained using the traditional space-time method and the super-resolved method to the projected DNS solution for initial condition $u(x, 0) = 2.5e^{-20x^2}$. 

The first step is to generate data for training the model. As described in Section 5, a single 2-D DNS snapshot (x-z plane) is extracted at a wall-normal height of $y^+ \approx 850$ and is $L_2$-projected on discontinuous polynomial spaces spanned by order $p = 1,3$ tensor-product Lagrange basis functions on meshes of different sizes. In this case, we project the DNS solution on meshes with elements $N_x \times N_z: 8 \times 4, 16 \times 8, 32 \times 16, 64 \times 32$ in the (stream-wise) and (span-wise) directions respectively. Once the $p = 1,3$ projected solutions are obtained, for each element present in these meshes, $p = 1$ coarse-scale basis coefficients are extracted for both the element and its immediate neighbors along with the $p = 3$ fine-scale basis coefficients. The next step is to evaluate the normalizing parameter for each element that is, $u_{\text{rms}} = \sqrt{\frac{\int_{\Omega_e} (u_i - \bar{u}_m)^2 d\Omega_e}{\int_{\Omega_e} d\Omega_e}}$, where the mean velocity $\bar{u}_m$ inside an element is given by $\bar{u}_m = \frac{\int_{\Omega_e} u_i d\Omega_e}{\int_{\Omega_e} d\Omega_e}$. Finally, a functional form similar to Equation (61) is assumed except the parameter $\alpha$ is replaced with the logarithm of cell Reynolds number that is, $\log(Re_\Delta)$. The physics-informed feature log($Re_\Delta$) ensures that different orders of magnitude of $Re_\Delta$ in training data is accounted for. Finally, the normalized input and output basis coefficient data and the logarithmic cell Reynolds number log($Re_\Delta$) for each element are assembled into a single table as a training data-set.

A VSRNN architecture for the N-N model is then assumed with sizes: $37 \times 32 \times 32 \times 32$ for the part A, $16 \times 32 \times 32$ for the part B, and a $32 \times 64 \times 64 \times 16$ sized postmultiplication part, respectively. The learning rate was set to $1E-4$ and optimization was run for 5000 iterations. Finally, the model performance is evaluated in Figures 16 and 15 by comparing the stream-wise and span-wise energy spectra obtained for the super-resolved $p = 3$ solution and the $p = 3$ $L_2$-optimal solution at different wall-normal heights of $y^+ \approx 500, 800, 850$. To compute the energy spectra, the solution in first obtained on a uniform mesh with size $(p + 1)N_x \times (p + 1)N_z$ where the factor $p + 1$ accounts for the effective grid-size at higher orders. This also prevents the over-sampling of the data.
As can be observed in Figures 15 and 16, the network can successfully recreate the correct energy distribution across different wave-numbers both in the stream-wise and the span-wise directions. This is true for both the cases: the plane at $y^+ \approx 850$, which was used for training, and the unseen planes at $y^+ \approx 500$ and $y^+ \approx 800$. However, the energy at the high wave-number modes for all these cases was slightly higher than the $L_2$-projected $p = 3$ optimal solution suggesting that a small amount of de-aliasing would be helpful. A qualitative plot of the coarse $p = 1$ solution, the super-resolved $p = 3$ solution and the $L_2$-optimally projected $p = 3$ solution for different grid sizes at a wall-normal distance of $y^+ \approx 500$ is shown in Figure 17. It can be observed that the super-resolved solution, similar to the optimal $p = 3$ projected solution, has finer structures when compared to the coarse $p = 1$ solution at different mesh resolutions.

The generalizability of the model trained using a single snapshot of DNS data stems from the fact that when the DNS image is projected on several finite element meshes with different element sizes, the average value of the cell Reynolds number changes. As a result, the training data contains an extensive range of cell Reynolds numbers. When the trained model is evaluated at different wall-normal distances while retaining the same the grid size, the cell Reynolds number changes due to changes in $\Delta_{\text{max}}$. However, this new cell Reynolds number can also be obtained at a previous wall-normal height by changing the grid-size alone. This can also be observed in Equation (34) for the normalized subscales. The normalized subscales only depend on the cell Peclet number $\alpha$ and the nondimensionalized inputs rather than the grid size or the diffusion coefficient separately.
**FIGURE 14** Sources of errors in offline and online super-resolution.

**FIGURE 15** Stream-wise energy spectra obtained for the $L_2$-projected stream-wise velocity solution on $p=1$, $L_2$-projected stream-wise velocity solution on $p=3$, N-N super-resolved $p=3$ solution and DNS at different wall normal height $y^+$ and mesh resolutions.
In Section 6, we demonstrated that the trained super-resolution model can also be used as a subgrid model. When this procedure was extended to the turbulent channel flow problem, the resulting subgrid model was found to be unstable and required under-relaxation of the super-resolved state to regain stability.\textsuperscript{79} Although partially successful in its current form, this approach is a topic of further research. Additional perspectives are provided in the following section.

8 | PERSPECTIVES

Inspired by successes in the machine vision community, there has recently been considerable interest in the use of super-resolution in the physical sciences. Much of the existing literature has, however, focused on reconstruction performance and not on predictive modeling. Truly predictive models should not be restricted to a single mesh or flow configuration, and should generalize to a class of flows. Despite the success in the canonical problem in Section 6, the results in Section 7 suggest that there is much to be done before a truly predictive capability can be realized for a problem as challenging as turbulent flow. We view our work as a first step in moving toward a predictive LES capability. Along these lines, we outline the following ingredients for the discovery of subgrid closures:

1. The model should be constructed using features that lend themselves to generalization
2. The structure of the learning model should allow one to efficiently embed physics-informed parameters
3. The closure model should be intimately linked to the underlying numerical discretization.

FIGURE 16 Span-wise energy spectra obtained for the $L_2$-projected stream-wise velocity solution on $p = 1$, $L_2$-projected stream-wise velocity solution on $p = 3$, N-N super-resolved $p = 3$ solution and DNS at different wall normal height $y^+$ and mesh resolutions.
4. The training should be performed in manner that the super-resolution is consistent with the coarse scales during the prediction.

5. The resulting model should not make the simulation unstable.

In our work, we addressed points #1 #2 above by choosing nondimensional features that are inspired by VMS closures, and by choosing a compositional neural network structure. Further work is required to design features that satisfy additional physics-informed invariances.

Regarding point #3, in contrast to implicitly filtered approaches in which coarse space is defined ambiguously, the VMS approach formally segregates the coarse and fine spaces, thus setting a clean environment for the super-resolution. Other candidates include explicitly filtering with a large test filter.

Point #4 refers to establishing consistency between the learning and prediction environments. In essence, the training is performed on DNS data, that is, $u' = f(u_{DNS}^{h})$, whereas in the online prediction stage, it is used as $u' = f(u_{LES}^{h})$. As the error between the coarse scales in the LES and DNS grows, the super-resolution becomes less accurate. In other words, the parameters of the learning model have not been inferred for online performance. Model-consistent training has been successfully demonstrated in RANS closures, the authors are aware of only one such attempt in the context of LES. However, as mentioned above, and in more detail in Reference 82, implicitly filtered approaches are associated with other challenges. The VMS approach, on the other hand, allows for both numerics-consistent and model-consistent training, but the implementation of such a capability is a major undertaking is yet to be pursued by the authors in an LES context.

Regarding point #5, it is possible that when the exact subscales are not used, the simulation can become both unstable and inaccurate. The inaccuracy present in the subscales can be tracked down to the approximations used for deriving the subscale model, incorrect choice of features used, or error in learning the subscale from data. We observed in Section 6 and Appendix A that the N-N predicted subscale did not affect the stability of the problem. However, for the turbulent channel...
flow problem the super-resolution state was required to be under-relaxed to ensure the stability of the simulation. The present approach presented in this article does not guarantee stability which is something that needs to be addressed in future work.

As a final point, while the appeal of VMS is the segregation of scales and the prospects of deriving closures with few phenomenological assumptions, structural models (e.g., Reference 45) generally perform poorly when the simulation is severely under-resolved. Several attempts have been made to combine traditional VMS approaches with phenomenological models like Smagorinsky in the form of mixed models. The use of data-driven techniques potentially allows us to account for these phenomenological relationships present in the data directly into the VMS model, thus, bridging the gap between phenomenological and structural modeling.

9 | CONCLUSIONS

We proposed a strategy for multiscale modeling in which the coarse and the fine scales are defined in terms of projection onto their respective finite element spaces, and segregated using a variational multiscale formulation. Existing variational multiscale formulations provide guiding principles for the construction of consistent features and network architecture to define a super-resolution model of the fine scales. Particularly, we define an architecture—called the Variational super-resolution neural network (VSRNN)—which approximates the subscales as a sum of products of individual functions of coarse-scales and the physics-informed parameters. This model form and network structure is inspired by analytical expression for the subscales as given by the convection-diffusion equation. It is emphasized that traditional architectures—such as a fully connected neural network—are not ideal for this purpose because they combine heterogeneous quantities (e.g., coarse-scale basis coefficients and physics informed-parameters) as inputs. The input features and output quantities are obtained by appropriately nondimensionalizing the coarse-scales and the subscale basis coefficients. By applying the super-resolved state to compute the Discontinuous Galerkin (DG) fluxes, we ensure that the online coarse-scale solution is forced toward its $L_2$-optimal state.

We verify that when the present approach is applied to the convection-diffusion problem, it can learn the analytical solution to a high degree of accuracy. Similarly, for the 1-D linear-advection space-time problem, the model could accurately super-resolve low-order coarse solutions to high-order fine-solution. The network could also reproduce super-resolved velocity fields with the proper energy distribution across different wave-numbers in the stream-wise and the span-wise direction for the turbulent channel case.

Next, we assessed the predictive capability of these models. Super-resolution was used to determine the DG fluxes for the linear-advection problem, and shown to result in higher accuracy and optimality of the method over traditional space-time methods for the same number of degrees of freedom. This improvement stems from the fact that the present model has been trained using $L_2$-optimal fine and coarse solutions, leading to subgrid models that are consistent with the type of optimality sought. The present method was found to generalize to out-of-sample nondimensional numbers.

Perspectives were provided on data-driven closure modeling in general, and particularly how model-consistent training could improve the prospects of developing truly predictive models. In addition to reconstruction and subgrid modeling, the super-resolution model can be used as an error indicator for adaptive grid refinement: Regions in which the high magnitude of the subscale values can be used as a measure for under-resolution. Finally, the authors would like to point out that effective implementation of this approach solvers requires the development of efficient nonlinear solvers and preconditioners to handle the additional nonlinearity and stiffness due to the model.

ACKNOWLEDGMENTS

This research was funded by NASA under the project “Scale-resolving turbulence simulations through adaptive high-order discretizations and data-enabled model refinements”, grant number 80NSSC18M0149 (Technical monitor: Gary Coleman). We gratefully acknowledge Dr. Krzysztof Fidkowski for the valuable discussions.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.
REFERENCES

1. Weinan E. Principles of Multiscale Modeling. Cambridge University Press; 2011.
2. Hughes TJ, Peijjó GR, Mazzei L, Quincy JB. The variational multiscale method—a paradigm for computational mechanics. *Comput Methods Appl Mech Eng*. 1998;166(1-2):3-24.
3. Pope SB. Turbulent Flows. Cambridge University Press; 2000.
4. Germano M, Piomelli U, Moin P, Cabot WH. A dynamic subgrid-scale eddy viscosity model. *Phys Fluids A Fluid Dyn*. 1991;3(7):1760-1765.
5. You D, Moin P. A dynamic global-coefficient subgrid-scale eddy-viscosity model for large-eddy simulation in complex geometries. *Phys Fluids*. 2007;19(6):065110.
6. Meneveau C, Lund TS, Cabot WH. A Lagrangian dynamic subgrid-scale model of turbulence. *J Fluid Mech*. 1996;319:353-385.
7. Nicoud F, Toda HB, Cabrit O, Bose S, Lee J. Using singular values to build a subgrid-scale model for large eddy simulations. *Phys Fluids*. 2011;23(8):085106.
8. Vreman A. An eddy-viscosity subgrid-scale model for turbulent shear flow: algebraic theory and applications. *Phys Fluids*. 2004;16(10):3670-3681.
9. Nicoud F, Ducros F. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow Turbul Combust*. 1999;62(3):183-200.
10. Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Comput Methods Appl Mech Eng*. 2002;191(39-40):4295-4321.
11. Codina R, Principe J, Guasch O, Badia S. Time dependent subscales in the stabilized finite element approximation of incompressible flow problems. *Comput Methods Appl Mech Eng*. 2007;196(21-24):2413-2430.
12. Bazilevs Y, Calo V, Cottrell J, Hughes T, Reali A, Scovazzi G. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Comput Methods Appl Mech Eng*. 2007;197(1-4):173-201.
13. Wang Z, Oberai A. Spectral analysis of the dissipation of the residual-based variational multiscale method. *Comput Methods Appl Mech Eng*. 2010;199(13-16):810-818.
14. Gravemeier V, Gee MW, Kronbichler M, Wall WA. An algebraic variational multiscale–multigrid method for large eddy simulation of turbulent flow. *Comput Methods Appl Mech Eng*. 2010;199(13-16):853-864.
15. Masud A, Calderer R. A variational multiscale method for incompressible turbulent flows: bubble functions and fine scale fields. *Comput Methods Appl Mech Eng*. 2011;200(33-36):2577-2593.
16. Parish EJ, Duraisamy K. A unified framework for multiscale Modeling using the Mori-Zwanzig formalism and the Variational multiscale method. *arXiv preprint arXiv:1712.09669*. 2017.
17. Smagorinsky J. General circulation experiments with the primitive equations: I the basic experiment. *Mon Weather Rev*. 1963;91(3):99-164.
18. Lilly DK. A proposed modification of the Germano subgrid-scale closure method. *Phys Fluids A Fluid Dyn*. 1992;4(3):633-635.
19. Zang Y, Street RL, Koseff JR. A dynamic mixed subgrid-scale model and its application to turbulent recirculating flows. *Phys Fluids A Fluid Dyn*. 1993;5(12):3186-3196.
20. Armenio V, Piomelli U. A Lagrangian mixed subgrid-scale model in generalized coordinates. *Flow Turbul Combust*. 2000;65(1):51-81.
21. Lévéque E, Toschi F, Shao L, Bertoglio JP. Shear-improved Smagorinsky model for large-eddy simulation of wall-bounded turbulent flows. *J Fluid Mech*. 2007;570:491-502.
22. Bardina J, Ferziger J, Reynolds W. *Improved Subgrid-Models for Large-Eddy Simulation*. American Institute of Aeronautics and Astronautics; 1980:1357.
23. Liu S, Meneveau C, Katz J. On the properties of similarity subgrid-scale models as deduced from measurements in a turbulent jet. *J Fluid Mech*. 1994;275:83-119.
24. Liu S, Meneveau C, Katz J. Experimental study of similarity subgrid-scale models of turbulence in the far-field of a jet. *Appl Sci Res*. 1995;54(3):177-190.
25. Vreman B, Geurts B, Kuerten H. On the formulation of the dynamic mixed subgrid-scale model. *Phys Fluids*. 1994;6(12):4057-4059.
26. Xie C, Wang J, Li H, Wan M, Chen S. Artificial neural network mixed model for large eddy simulation of compressible turbulence. *Phys Fluids*. 2019;31(8):085112.
27. Langford JA, Moser RD. Optimal LES formulations for isotropic turbulence. *J Fluid Mech*. 1999;398:321-346.
28. Bose ST, Moin P, You D. Grid-independent large-eddy simulation using explicit filtering. *Phys Fluids*. 2010;22(10):105103.
29. Lund T. The use of explicit filters in large eddy simulation. *Comput Math Appl*. 2003;46(4):603-616.
30. Lund TS, Kaltenbach HJ. *Experiments with Explicit Filtering for LES Using a Finite-Difference Method*. Center for Turbulence Research, Stanford University; 1995.
31. Mathew J, Lechner R, Foysi H, Sesterhenn J, Friedrich R. An explicit filtering method for large eddy simulation of compressible flows. *Phys Fluids*. 2003;15(8):2279-2289.
32. Moura RC, Mengaldo G, Péiró J, Sherwin SJ. On the eddy-resolving capability of high-order discontinuous Galerkin approaches to implicit LES/under-resolved DNS of Euler turbulence. *J Comput Phys*. 2017;330:615-623.
33. Flad D, Gassner G. On the use of kinetic energy preserving DG-schemes for large eddy simulation. *J Comput Phys*. 2017;350:782-795.
34. Fernandez P, Nguyen NC, Peraire J. Subgrid-Scale Modeling and Implicit Numerical Dissipation in DG-Based Large-Eddy Simulation. American Institute of Aeronautics and Astronautics; Vol 3951; 2017.
35. Hickel S, Adams NA, Domaradzki JA. An adaptive local deconvolution method for implicit LES. J Comput Phys. 2006;213(1):413-436.
36. Sun G, Domaradzki JA. Implicit LES using adaptive filtering. J Comput Phys. 2018;359:380-408.
37. Hughes TJ, Calo VM, Scovazzi G. Variational and multiscale methods in turbulence. Mechanics of the 21st Century. Springer; 2005:153-163.
38. Hughes TJ, Oberal AA, Mazzei L. Large eddy simulation of turbulent channel flows by the variational multiscale method. Phys Fluids. 2001;13(6):1784-1799.
39. Hughes TJ, Franca LP, Hulbert GM. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. Comput Methods Appl Mech Eng. 1989;73(2):173-189.
40. Brooks AN, Hughes TJ. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-stokes equations. Comput Methods Appl Mech Eng. 1982;32(1-3):199-259.
41. Codina R. On stabilized finite element methods for linear systems of convection–diffusion-reaction equations. Comput Methods Appl Mech Eng. 2000;188(1-3):61-82.
42. Hughes TJ, Franca LP, Balestra M. A new finite element formulation for computational fluid dynamics: V. circumventing the Babuška-Brezzi condition: a stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations. Comput Methods Appl Mech Eng. 1986;59(1):85-99.
43. Parish EJ, Duraisamy K. Non-Markovian closure models for large eddy simulations using the Mori-Zwanzig formalism. Phys Rev Fluids. 2017;2(1):014604.
44. Parish EJ, Duraisamy K. A dynamic subgrid scale model for large Eddy simulations based on the Mori-Zwanzig formalism. J Comput Phys. 2017;349:154-175.
45. Pradhan A, Duraisamy K. Variational multiscale closures for finite element discretizations using the Mori-Zwanzig approach. Comput Methods Appl Mech Eng. 2020;368:113152.
46. Parish EJ, Duraisamy K. A paradigm for data-driven predictive modeling using field inversion and machine learning. J Comput Phys. 2016;305:758-774.
47. Stolz S, Adams NA. An approximate deconvolution procedure for large-eddy simulation. Phys Fluids. 1999;11(7):1699-1701.
48. Stolz S, Adams NA, Kleiser L. An approximate deconvolution model for large-eddy simulation with application to incompressible wall-bounded flows. Phys Fluids. 2001;13(4):997-1015.
49. Singh AP, Duraisamy K. Using field inversion to quantify functional errors in turbulence closures. Phys Fluids. 2016;28(4):045110.
50. Ling J, Kurzawski A, Templeton J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. J Fluid Mech. 2016;807:155-166.
51. Wang JX, Wu JL, Xiao H. Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. Phys Rev Fluids. 2017;2(3):034603.
52. Sarghini F, De Felice G, Santini S. Neural networks based subgrid scale modeling in large eddy simulations. Phys Fluids. 2003;15(2):32-39.
53. Gamahara M, Hattori Y. Searching for turbulence models by artificial neural network. Phys Rev Fluids. 2017;2(5):054604.
54. Xie C, Li K, Ma C, Wang J. Modeling subgrid-scale force and divergence of heat flux of compressible isotropic turbulence by artificial neural network. Phys Rev Fluids. 2019;4(10):104605.
55. Beck A, Flad D, Munz CD. Deep neural networks for data-driven LES closure models. J Comput Phys. 2019;398:108910.
56. Maulik R, San O, Jacob JD, Crick C. Sub-grid scale model classification and blending through deep learning. J Fluid Mech. 2019;870:151-181.
57. Yang X, Zafar S, Wang JX, Xiao H. Predictive large-eddy-simulation wall modeling via physics-informed neural networks. Phys Rev Fluids. 2019;4(3):034602.
58. Maulik R, San O. A neural network approach for the blind deconvolution of turbulent flows. J Fluid Mech. 2017;831:151-181.
59. Maulik R, San O, Rasheed A, Vedula P. Data-driven deconvolution for large eddy simulations of Kraichnan turbulence. Phys Fluids. 2018;30(12):125109.
60. Wang Z, Luo K, Li D, Tan J, Fan J. Investigations of data-driven closure for subgrid-scale stress in large-eddy simulation. Phys Fluids. 2018;30(12):125101.
61. Xie C, Wang J, Weinan E. Modeling subgrid-scale forces by spatial artificial neural networks in large eddy simulation of turbulence. Phys Rev Fluids. 2020;5(5):054606.
62. Mou C, Koc B, San O, Rebholz LG, Iliescu T. Data-driven variational multiscale reduced order models. Comput Methods Appl Mech Eng. 2021;373:113470.
63. Xie X, Mohdubujiaman M, Rebholz LG, Iliescu T. Data-driven filtered reduced order modeling of fluid flows. SIAM J Sci Comput. 2018;40(3):B834-B857.
64. Mohdubujiaman M, Rebholz LG, Iliescu T. Physically constrained data-driven correction for reduced-order modeling of fluid flows. Int J Numer Methods Fluids. 2019;89(3):103-122.
65. Wang Q, Ripamonti N, Hesthaven JS. Recurrent neural network closure of parametric POD-Galerkin reduced-order models based on the Mori-Zwanzig formalism. J Comput Phys. 2020;410:109402.
66. Xie Y, Franz E, Chu M, Thuerey N. tempoGAN: a temporally coherent, volumetric GAN for super-resolution fluid flow. ACM Trans Graph. 2018;37(4):1-15.
How to cite this article: Pradhan A, Duraisamy K. Variational multiscale super-resolution: A data-driven approach for reconstruction and predictive modeling of unresolved physics. Int J Numer Methods Eng. 2023;124(19):4339-4370. doi: 10.1002/nme.7310

APPENDIX A. “NODALLY EXACT” HIGH-ORDER CG SCHEMES FOR 1-D CONVECTION-DIFFUSION

To demonstrate the action of subscales in 3, we assumed that the coarse space to be composed of piece-wise linear polynomials. However, this approach can be extended to higher-order polynomials as well. In this section, we will use the VSRNN architecture to learn closures for high order CG discretizations where the coarse-scale is 'nodally exact'. The governing PDE is again taken to be the linear convection-diffusion equation as follows:

$$\mathcal{L} = a \frac{d}{dx} - \nu \frac{d^2}{dx^2} \text{ in } \Omega = [0, L]$$

(A1)

with Dirichlet boundary conditions: \( u(0) = u_0 \) and \( u(L) = u_L \). To derive VSRNN closures for 'nodally exact' coarse-scales, we start with the variational form:

$$\left( a \frac{du}{dx} - \nu \frac{d^2u}{dx^2}, w \right) = 0.$$  

(A2)
The weak form after integration by parts is obtained as follows:

\[
\left( a \frac{du}{dx}, w \right) + \kappa \left( \frac{du}{dx}, \frac{dw}{dx} \right) = 0. \tag{A3}
\]

The next step is to apply the VMS decomposition such that the coarse-scale is exactly the interpolate of the true solution at nodal points that is,

\[
\left( a \frac{du_h}{dx}, w_h \right) + \kappa \left( \frac{du_h}{dx}, \frac{dw_h}{dx} \right) + \left( a \frac{du'}{dx}, w_h \right) + \kappa \left( \frac{du'}{dx}, \frac{dw_h}{dx} \right) = 0. \tag{A4}
\]
FIGURE A3  Discretizations of the 1-D Convection-Diffusion equation at $Pe = 40$ using two CG elements $N_{el} = 2$ and different polynomial orders $p = 3, 4, 7, 8$.

Since the coarse-scale is the true interpolant of the solution, the subscales should vanish at the nodal points. Hence, integration by parts can be performed as follows:

$$
\left( a \frac{du_h}{dx}, w_h \right) + \kappa \left( \frac{du_h}{dx}, \frac{dw_h}{dx} \right) + \sum_e \int_{\Omega_e} u' \left( -a \frac{dw_h}{dx} - \kappa \frac{d^2 w_h}{dx^2} \right) d\Omega = 0.
$$

It can be recognized that the subscale lies in an infinite dimensional space. However, for each element only its inner-product with $-a \frac{dw_h}{dx} - \kappa \frac{d^2 w_h}{dx^2}$ needs to be computed inside each element. Hence, if $p$ is the order of the polynomial used to describe the coarse-scales, $L_2$-projecting $u'$ in a discontinuous polynomial space inside the element consisting of polynomials up to order $p - 1$ is sufficient. To learn these projected subscales, the VSRNN is used as follows:

$$
\left[ C'_{1,p-1}, C'_{2,p-1}, \ldots, C'_{p,p-1} \right] = f_{\text{NN}}(\log(\alpha), [\tilde{C}_{1,p}, \tilde{C}_{2,p}, \ldots, \tilde{C}_{p+1,p}]),
$$

where, $\left[ C'_{1,p-1}, C'_{2,p-1}, \ldots, C'_{p,p-1} \right]$ represents the subscale basis coefficients normalized by coarse-scale r.m.s $u_{rms}$, $[\tilde{C}_{1,p}, \tilde{C}_{2,p}, \ldots, \tilde{C}_{p+1,p}]$ represents the mean $u_{mean}$ subtracted and $u_{rms}$ normalized coarse-scale basis coefficients, and $\alpha$ is the Peclet number.

It is noted that although $u'$ is infinite dimensional, $\left[ C'_{1,p-1}, C'_{2,p-1}, \ldots, C'_{p,p-1} \right]$ is not. It is sufficient to learn these projected subscales to precisely compute the required inner-products. For example, when the coarse solution is linear ($p = 1$), the subscale can be represented by $p = 0$ constant functions which corresponds to commonly used $\tau = \frac{A}{2a} \left( \coth(\alpha) - \frac{1}{\alpha} \right)$. The online evaluation of the $p = 1$ closure presented in Figure 4 is shown in Figure A1. As expected, the VSRNN is able to precisely recreate the results obtained using the analytical expression for $\tau$. 
The data generation procedure used here for deriving closures for higher order polynomials is same as that used in Section 4 for the $p = 1$ case. The model is trained and applied to Equation (A5). Two different cases with global Peclet numbers $Pe_g = \frac{aL}{\nu} = 20$ and 40 are considered here. For each case, two CG elements with different polynomial orders $p = 3, 4, 7, \text{ and } 8$ are used to discretize the domain. Figures A2 and A3 shows the comparison of the present VSRNN closure to existing closures and no-model for global Peclet numbers of 20 and 40 respectively. The VSRNN model in both the cases accurately learns the subscales and ensures that the coarse-scale is the interpolate of the true solution. As expected, $\tau = \frac{h}{2a} \left( \coth(a) - \frac{1}{a} \right)$ based on low-order discretization is not accurate at high-orders and the no-model discretization (Galerkin) is oscillatory when resolution is not sufficient. The no-model performance increases at high-order because effective resolution increases with $p$. 