The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

# A-PRIORI ENERGY-BIASED TRAINING FOR THE PREDICTION OF CLOSURE TERMS OF THE VMS NAVIER-STOKES EQUATIONS

**A. BETTINI [1], AND S. J. HULSHOFF [2]**

Aerodynamics Section
Delft University of Technology
Kluyverweg 2, 2629 HS Delft, The Netherlands
e-mail: A.Bettini@tudelft.nl [1]
S.J.Hulshoff@tudelft.nl [2]

**Key words:** Finite Element Method (FEM), Turbulence Modelling, Machine Learning (ML), Variational Multiscale (VMS), Multilayer Perceptron (MLP), Augmented Lagrangian Method

**Summary.** The authors present an energy-biased training approach for predicting exact closure terms of the Navier-Stokes equations discretised in a Variational Multiscale framework. The approach initially involves formulating a constrained objective function, which is transformed into an unconstrained problem suitable for neural network training using the Augmented Lagrangian method. The constraints induce a prediction bias by enforcing the predicted closure term energy contributions to be smaller than the exact values, both for excessive backscatter or insufficient dissipation. Effectively, the approach controls energy contributions through a-priori measures rather than a-posteriori measures.

The approach was applied to obtain the $H^1$ projection of a $\mathrm{Re}_\tau = 180$ turbulent channel flow on a $32 \times 32 \times 32$ uniform mesh. Eight MLPs were trained to predict the closure terms associated with the weighting functions of each linear hexahedral element. Each network used 217 features, and five hidden layers with 600 neurons each. The final data-to-parameter ratio was $\sim$19.6 : 1 (30 932 992 : 1 575 604) per neural network. A-priori evaluation of the networks' outputs demonstrated its ability to predict closure terms yielding the desired behaviour in energy transfer. This was true for both closure terms that yield energy gain and energy loss. In spite of the energy bias, the closure term predictions retained correlations greater than 0.85 with their exact value for all positions between the channel walls.

## 1 INTRODUCTION

The original Variational Multiscale (VMS) formulation of the Navier-Stokes equations attempts to compute the desired projection $\mathcal{P}(\boldsymbol{U})$ of an exact solution $\boldsymbol{U}$. This requires closure terms to model the effects of scales unresolved by the projection. Normally, algebraic closure terms are used to model these effects. However, for turbulent flows, these models become expectantly worse as the mesh is coarsened, due to the complexity of the phenomena occurring in the unresolved scales.

Machine learning (ML) provides a path to determining closure terms which result in accurate projections of turbulent flow solutions. However, it must be ensured that any errors in the predictions from an ML model do not result in the net addition of kinetic energy to the resolved scales, resulting in a divergent numerical simulation. A simple approach is to clip all backscatter arising due to the closure terms. This considerably limits accuracy as accounting for backscatter, local in space and in time, becomes increasingly important as the mesh is coarsened.

Here we present an a-priori energy-biased training approach that attempts to minimise non-physical energy production stemming from erroneous ML closure term predictions. This approach is applied to create a ML closure-term model for a coarse VMS discretisation of a $\mathrm{Re}_\tau = 180$ Turbulent Channel Flow (TCF). Section 2 covers the training approach, in particular the formulation of the unconstrained objective via the Augmented Lagrangian Method (ALM). Section 3 demonstrates the a priori performance of models generated using the aforementioned energy-biased training approach, and Section 4 summarises the results of the paper.

## 2 METHODOLOGY

### 2.1 Incompressible Navier-Stokes Equations

The formulation follows that of [1, 2]. Consider a spatial domain $\Omega \in \mathbb{R}^3$ with boundary $\partial\Omega$ over a time interval $T$. The problem consists of solving the incompressible Navier-Stokes equations for the velocity $\boldsymbol{u}(\boldsymbol{x},t)$, and pressure $p(\boldsymbol{x},t)$:

$$\begin{aligned} \nabla \cdot \boldsymbol{u} &= 0 && \text{in } \Omega \ \ \forall t \in [0,T]\,, \\ \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} &= -\frac{1}{\rho}\nabla p + \nabla \cdot (2\nu \nabla^s \boldsymbol{u}) + \boldsymbol{f} && \text{in } \Omega \ \ \forall t \in [0,T]\,, \end{aligned}$$

where $\boldsymbol{f}(\boldsymbol{x},t)$ is a known body force, $\nu \in \mathbb{R}^+$ is a constant kinematic viscosity. The system of equations is subject to the following initial and no-slip boundary conditions:

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x},0) &= \boldsymbol{g}(\boldsymbol{x}) && \text{in } \Omega\,, \\ \boldsymbol{u}(\boldsymbol{x},t) &= \boldsymbol{0} && \text{on } \partial\Omega \ \ \forall t \in [0,T]\,, \end{aligned}$$

where $\boldsymbol{g}(\boldsymbol{x})$ is a known initial condition. The problem is discretised by considering a sequence of time levels $t^n$ and assuming the solution at each time level $\boldsymbol{U}^n = \{\boldsymbol{u}(\boldsymbol{x},t^n), p(\boldsymbol{x},t^n)\}$ lives in a trial solution space $\mathcal{V} = \mathcal{V}(\Omega)$. Using weighting functions $\boldsymbol{W} = \{\boldsymbol{w}, q\}$ that live in the same space, the semi-discrete form of the problem at the next time level becomes: find $\boldsymbol{U} \in \mathcal{V}$ such that $\forall \boldsymbol{W} \in \mathcal{V}$:

$$\left(\boldsymbol{w}, \frac{\partial \boldsymbol{u}^{n+1}}{\partial t} + (\boldsymbol{u}^{n+1} \cdot \nabla)\boldsymbol{u}^{n+1} + \frac{1}{\rho}\nabla p^{n+1} - \nabla \cdot (2\nu\nabla^s \boldsymbol{u}^{n+1})\right)_\Omega + \left(q, \nabla \cdot \boldsymbol{u}^{n+1}\right)_\Omega = \left(\boldsymbol{w}, \boldsymbol{f}^{n+1}\right)_\Omega ,$$

where $\partial \boldsymbol{u}^{n+1}/\partial t$ is obtained from a backward finite-difference approximation. Following a direct sum decomposition of $\mathcal{V}$ into a finite-dimensional space $\overline{\mathcal{V}}$ and infinite-dimensional space $\mathcal{V}'$, and omitting the $n+1$ superscript for clarity, the variational formulation for the resolved-scale problem reads:

find the resolved solution $\overline{\boldsymbol{U}} = \{\overline{\boldsymbol{u}}, \overline{p}\} \in \overline{\mathcal{V}}$ such that $\forall\, \overline{\boldsymbol{W}} = \{\overline{\boldsymbol{w}}, \overline{q}\} \in \overline{\mathcal{V}}$:

$$\begin{aligned}
&\overbrace{\boldsymbol{B}_1^C\Big(\overline{\boldsymbol{W}},\overline{\boldsymbol{U}}\Big)}^{\left(\overline{q},\,\mathcal{L}_C(\overline{\boldsymbol{u}})\right)_\Omega} + \overbrace{\boldsymbol{B}_1^M\Big(\overline{\boldsymbol{W}},\overline{\boldsymbol{U}}\Big) + \boldsymbol{B}_2\Big(\overline{\boldsymbol{W}},\overline{\boldsymbol{U}},\overline{\boldsymbol{U}}\Big)}^{\left(\overline{\boldsymbol{w}},\,\overline{\mathcal{L}}_M(\overline{\boldsymbol{u}},\overline{p})\right)_\Omega} - \boldsymbol{L}\Big(\overline{\boldsymbol{W}}\Big) = \\
&\underbrace{-\boldsymbol{B}_1^C\Big(\overline{\boldsymbol{W}},\boldsymbol{U}'\Big)}_{-\left(\overline{q},\,\mathcal{L}_C(\boldsymbol{u}')\right)_\Omega} \underbrace{-\boldsymbol{B}_1^M\Big(\overline{\boldsymbol{W}},\boldsymbol{U}'\Big) - \boldsymbol{B}_2\Big(\overline{\boldsymbol{W}},\overline{\boldsymbol{U}},\boldsymbol{U}'\Big) - \boldsymbol{B}_2\Big(\overline{\boldsymbol{W}},\boldsymbol{U}',\overline{\boldsymbol{U}}\Big) - \boldsymbol{B}_2\Big(\overline{\boldsymbol{W}},\boldsymbol{U}',\boldsymbol{U}'\Big)}_{-\left(\overline{\boldsymbol{w}},\,\mathcal{L}'_M(\overline{\boldsymbol{u}},\boldsymbol{u}',p')\right)_\Omega},
\end{aligned} \tag{1}$$

where $\boldsymbol{U}' = \big\{\boldsymbol{u}', p'\big\} \in \mathcal{V}'$ is the unresolved solution, and we have expressed the weak form in terms of the bilinear and trilinear forms:

$$\boldsymbol{B}_1^M(\boldsymbol{W},\boldsymbol{U}) = \left(\boldsymbol{w}, \frac{\partial \boldsymbol{u}}{\partial t} + \frac{1}{\rho}\nabla p - \nabla \cdot \left(2\nu \nabla^s \boldsymbol{u}\right)\right)_\Omega \quad , \quad \boldsymbol{B}_1^C(\boldsymbol{W},\boldsymbol{U}) = \Big(q, \nabla \cdot \boldsymbol{u}\Big)_\Omega ,$$

$$\boldsymbol{B}_2(\boldsymbol{W},\boldsymbol{U},\boldsymbol{V}) = \Big(\boldsymbol{w}, (\boldsymbol{u} \cdot \nabla)\boldsymbol{v}\Big)_\Omega \quad , \quad \boldsymbol{L}(\boldsymbol{W}) = \Big(\boldsymbol{w}, \boldsymbol{f}\Big)_\Omega ,$$

with $\boldsymbol{V} = \big\{\boldsymbol{v}, \cdot\big\}$. The terms in Eq. (1) are gathered into the resolved terms on the left-hand side and the unresolved terms on the right-hand side. It is the right-hand side terms which we predict using a ML model. Equation (1) is solved by iteratively minimising its residual using a Newton-Raphson procedure. To avoid the corrector-pass issues pointed out in [2], we used the lagged feature set approach studied in [3], in which the ML model is trained to provide predictions for the sum of the closure terms at $t^{n+1}$ using data only from the time levels $t^n$ and $t^{n-1}$.

### 2.2 Model Requirements

In a finite-element setting, it is preferable that the ML model provides the closure-term predictions on the element level. The complete closure term is then computed via the assembly operator $\mathsf{A}$ as the sum of element contributions:

$$\begin{aligned}
\Big(\overline{\boldsymbol{w}}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')\Big)_\Omega &= \mathop{\mathsf{A}} \underbrace{\Big(\overline{\boldsymbol{w}}_e, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')\Big)_{\Omega_e}}_{\text{predicted}} , \\
\Big(\overline{q}, \mathcal{L}_C(\boldsymbol{u}')\Big)_\Omega &= \mathop{\mathsf{A}} \underbrace{\Big(\overline{q}_e, \mathcal{L}_C(\boldsymbol{u}')\Big)_{\Omega_e}}_{\text{predicted}} ,
\end{aligned} \tag{2}$$

where $\overline{\boldsymbol{w}}_e$ and $\overline{q}_e$ are the portions of the weighting functions within each element, and $\Omega_e$ is the element domain. Therefore, to satisfy that the closure terms are predicted exactly, the individual element predictions have to match their exact values, or at least sum to the exact sum. Here the former requirement is chosen, since the correct local predictions imply that the $L_2$-inner product is carried out using a unique unresolved solution. This results in four requirements for the $s$-th element weighting functions $\{\overline{\boldsymbol{w}}_{e,s}, \overline{q}_{e,s}\}$:

$$\begin{aligned}
E_x^{\mathrm{M}} &:= \Big[\Big(\overline{w}_{e,s,x}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')_x\Big)_{\Omega_e}\Big]_{\text{pred.}} - \Big[\Big(\overline{w}_{e,s,x}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')_x\Big)_{\Omega_e}\Big]_{\text{truth}} = 0\,,\\
E_y^{\mathrm{M}} &:= \Big[\Big(\overline{w}_{e,s,y}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')_y\Big)_{\Omega_e}\Big]_{\text{pred.}} - \Big[\Big(\overline{w}_{e,s,y}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')_y\Big)_{\Omega_e}\Big]_{\text{truth}} = 0\,,\\
E_z^{\mathrm{M}} &:= \Big[\Big(\overline{w}_{e,s,z}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')_z\Big)_{\Omega_e}\Big]_{\text{pred.}} - \Big[\Big(\overline{w}_{e,s,z}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')_z\Big)_{\Omega_e}\Big]_{\text{truth}} = 0\,,\\
E^{\mathrm{C}} &:= \Big[\Big(\overline{q}_{e,s}, \mathcal{L}_C(\boldsymbol{u}')\Big)_{\Omega_e}\Big]_{\text{pred.}} - \Big[\Big(\overline{q}_{e,s}, \mathcal{L}_C(\boldsymbol{u}')\Big)_{\Omega_e}\Big]_{\text{truth}} = 0\,.
\end{aligned} \tag{3}$$

Since the predicted terms above are supplied by a ML model, they will be subject to errors. The distribution of these errors can be controlled by imposing a constraint upon the predictions themselves. Consider the resolved-scale kinetic energy equation:

$$\Big(\overline{\boldsymbol{u}}, \overline{\mathcal{L}}_M(\overline{\boldsymbol{u}}, \overline{p})\Big)_{\Omega} - \Big(\overline{\boldsymbol{u}}, \boldsymbol{f}\Big)_{\Omega} = -\Big(\overline{\boldsymbol{u}}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')\Big)_{\Omega}\,.$$

We propose that the predicted closure-term resolved-scale kinetic-energy contributions must be equivalent or smaller than the truth values, i.e.

$$\Big[-\Big(\overline{\boldsymbol{u}}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')\Big)_{\Omega}\Big]_{\text{pred.}} \leq \Big[-\Big(\overline{\boldsymbol{u}}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')\Big)_{\Omega}\Big]_{\text{truth}}\,.$$

The above can be expressed similarly to Eqs. (2) by noting that $\overline{\boldsymbol{u}}$ is a weighted sum of solution bases and coefficients $\overline{a}_{\,x}$, $\overline{a}_{\,y}$, $\overline{a}_{\,z}$. As the $\overline{a}_{\,(\cdot)}$ are constants at a given time level, they can be factored out of the inner products, e.g. for the energy contribution originating from the $x$ momentum component:

$$\Big[\mathsf{A}\; -\overline{a}_{\,e,x}\Big(\overline{w}_{e,x}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')_x\Big)_{\Omega_e}\Big]_{\text{pred.}} \leq \Big[\mathsf{A}\; -\overline{a}_{\,e,x}\Big(\overline{w}_{e,x}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')_x\Big)_{\Omega_e}\Big]_{\text{truth}}\,.$$

To provide suitable constraints for training at the element level, we decompose the three resulting inequalities into more restrictive versions that require the contributions from each element weighting function to adhere to the inequality:

$$\begin{aligned}
C_x &:= \Big[-\overline{a}_{\,e,s,x}\Big]_{\text{truth}}\Big[\Big(\overline{w}_{e,s,x}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')_x\Big)_{\Omega_e}\Big]_{\text{pred.}} - \Big[-\overline{a}_{\,e,s,x}\Big(\overline{w}_{e,s,x}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')_x\Big)_{\Omega_e}\Big]_{\text{truth}} \leq 0\,,\\
C_y &:= \Big[-\overline{a}_{\,e,s,y}\Big]_{\text{truth}}\Big[\Big(\overline{w}_{e,s,y}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')_y\Big)_{\Omega_e}\Big]_{\text{pred.}} - \Big[-\overline{a}_{\,e,s,y}\Big(\overline{w}_{e,s,y}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')_y\Big)_{\Omega_e}\Big]_{\text{truth}} \leq 0\,,\\
C_z &:= \Big[-\overline{a}_{\,e,s,z}\Big]_{\text{truth}}\Big[\Big(\overline{w}_{e,s,z}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')_z\Big)_{\Omega_e}\Big]_{\text{pred.}} - \Big[-\overline{a}_{\,e,s,z}\Big(\overline{w}_{e,s,z}, \mathcal{L}'_M(\overline{\boldsymbol{u}}, \boldsymbol{u}', p')_z\Big)_{\Omega_e}\Big]_{\text{truth}} \leq 0\,.
\end{aligned} \tag{4}$$

where we have also loosened the constraint on the prediction to rely on the truth $\overline{a}_{\,(\cdot)}$, since the generated model will not predict $\overline{a}_{\,(\cdot)}$. Equations (3) and Eqs. 4 form the necessary components of a constrained optimisation problem $\forall s$, which reads:

$$\begin{aligned}
&\arg\min && E_x^{\mathrm{M}} + E_y^{\mathrm{M}} + E_z^{\mathrm{M}} + E^{\mathrm{C}}\,,\\
&\text{subject to} && C_x \leq 0\,,\, C_y \leq 0\,,\, C_z \leq 0\,.
\end{aligned}$$

The $\overline{a}_{\,(\cdot)}$ constants in Eqs. (4) are not factored out of the truth values due to the method chosen to tackle the constrained problem (see Section 2.5). Correspondingly, the coefficients provide bias such that larger closure-term resolved-scale kinetic-energy contributions are penalised more than smaller contributions are.

### 2.3 Test Case

We consider a $\mathrm{Re}_\tau = 180$ TCF with constant forcing on a domain of size $L_x \times L_y \times L_z = 6\delta \times 4\delta \times 2\delta$, with $\delta$ being the channel half height, and $x, y, z$ being the streamwise, spanwise and wall-normal directions, respectively. The resolved scales, $\overline{\boldsymbol{U}}$, are represented using a P1/P1 finite-element space on a $32 \times 32 \times 32$ uniform hexahedral mesh indexed using $i, j$ and $k$ for the $x, y$ and $z$ directions. Reference DNS data, $\boldsymbol{U}_{\mathrm{DNS}}$, was generated using a spectral/finite-volume code by starting from a statistically-converged initial condition and proceeding for four eddy turnovers. 480 snapshots of this data were used to generate reference resolved-scale solutions using $H_1$ projections of the DNS data, $\overline{\boldsymbol{U}} = \mathcal{P}_{H_1}(\boldsymbol{U}_{\mathrm{DNS}})$. The corresponding unresolved scales $\overline{\boldsymbol{U}}' = \boldsymbol{U}_{\mathrm{DNS}} - \overline{\boldsymbol{U}}$, were used to compute truth/exact values for the closure terms. These were used to train a ML model to predict $t^{n+1}$ closure terms. Note that in this setting, exact ML predictions used in LES computations with Eq. (1) will produce the desired projection of the DNS solution, $\mathcal{P}_{H_1}(\boldsymbol{U}_{\mathrm{DNS}})$, as the resolved scales. Correspondingly, ML prediction error is the only source of error in the formulation.

### 2.4 Architecture Selection

For simplicity, our ML model is comprised of eight multilayer perceptrons $(\mathrm{MLP})_s$. The $(\mathrm{MLP})_s$ network predicts the closure terms associated with the $\{\overline{\boldsymbol{w}}_{e,s}, \overline{q}_{e,s}\}$ element weighting function for all elements in the mesh. For an initial baseline version of the ML model, we use dense and wide MLP networks, each with five hidden layers and 600 ReLU-activated neurons.

### 2.5 Objective Function

The ALM takes a constrained objective and transforms it to an unconstrained problem, whilst replacing the row associated with the Lagrange multiplier in the system of equations with an update scheme instead. This is useful for constrained neural network training as the backpropagation step does not have to be modified to account for additional minimisation terms that are part of the cost function (i.e. the Lagrange multiplier). Instead, the implementation details are relegated to member functions that are called at the start-of-training step (setting variables), or end-of-epoch step (update scheme), and are coupled to the loss function object. The ALM is best described schematically for a simple non-ML constrained problem, as shown in Fig. 1. Given an initial guess for parameters $\mu^k$, $\lambda^k$, and a user-chosen scaling parameter $\beta$, the $k$-th iterate minimisation problem is solved iteratively until its residual is lower than the preset tolerance. Then an update for $\mu^k$, $\lambda^k$ occurs, for which the $(k+1)$-th iterate minimisation problem is solved for. This is repeated until the solution does not change as a function of $k$.

$$\begin{tikzcd}
 & \text{Given } \mu^0\,,\ \lambda^0\,,\ \beta \arrow[d, "\text{start}"'] & \\
\begin{array}{ll} \underset{x}{\arg\min} & f(x) \\ \text{subject to} & g(x) = a \end{array} \arrow[r, "\text{ALM}"] & \underset{x\,;\,\mu^k\,,\lambda^k}{\arg\min} \quad f(x) + \dfrac{\mu^k}{2}\varepsilon_g^2 - \lambda^k \varepsilon_g \arrow[d, shift right=2, "\text{res.} < \text{tol.}"'] & \equiv \quad \underset{x\,;\,\mu^k\,,\lambda^k}{\arg\min} \quad f(x) + \rho_g^k \\
 & \begin{aligned} \mu^{k+1} &= \beta\mu^k \\ \lambda^{k+1} &= \lambda^k - \mu^k \varepsilon_g \end{aligned} \arrow[u, shift right=2, "k \to k+1"'] &
\end{tikzcd}$$

Figure 1: Schematical representation of the ALM applied to a simple constrained problem. $\varepsilon_g = g(x) - a,\ \rho_g^k = (\mu^k/2)\varepsilon_g^2 - \lambda^k \varepsilon_g$, and the subscript references the constraint function of interest. Inspired by [4].

We can extend the definition for the constraint error metric $\varepsilon_g$ to inequalities constraints using max functions (such that inequality constraints transform to equality constraints):

$$\begin{array}{ccccc}
\text{old constraint} & \longrightarrow & \text{redefinition} & \longrightarrow & \text{new constraint} \\
g(x) \leq a & \longrightarrow & \varepsilon_g = \max\big(0,\, g(x) - a\big) & \longrightarrow & \varepsilon_g = 0 \\
g(x) \geq a & \longrightarrow & \varepsilon_g = \max\big(0,\, a - g(x)\big) & \longrightarrow & \varepsilon_g = 0
\end{array}$$

A further extension to ML problems can be achieved by using MSE metrics, given a batch size $N_b$,

$$C < 0 \underset{\text{ML}}{\longrightarrow} \mathcal{C} = \frac{1}{N_b} \sum_{i=1}^{N_b} \Big[ \max\Big(0, C_i\Big) \Big]^2 ,$$

where we have specifically tailored this definition to be used with Eqs. (4). We convert Eqs. (3) to MSE metrics as well:

$$E = 0 \underset{\text{ML}}{\longrightarrow} \mathcal{E} = \frac{1}{N_b} \sum_{i=1}^{N_b} E_i^2 \,.$$

We tailor the ALM to update $\mu^k$ and $\lambda^k$ each epoch, although this is not entirely necessary. This was done to hasten the training process. We found better success by updating $\lambda^k$ by $\mathcal{C}$ calculated using the validation data, which makes the ALM scheme mathematically inconsistent with [4]. The end schematic is found in Fig. 2.

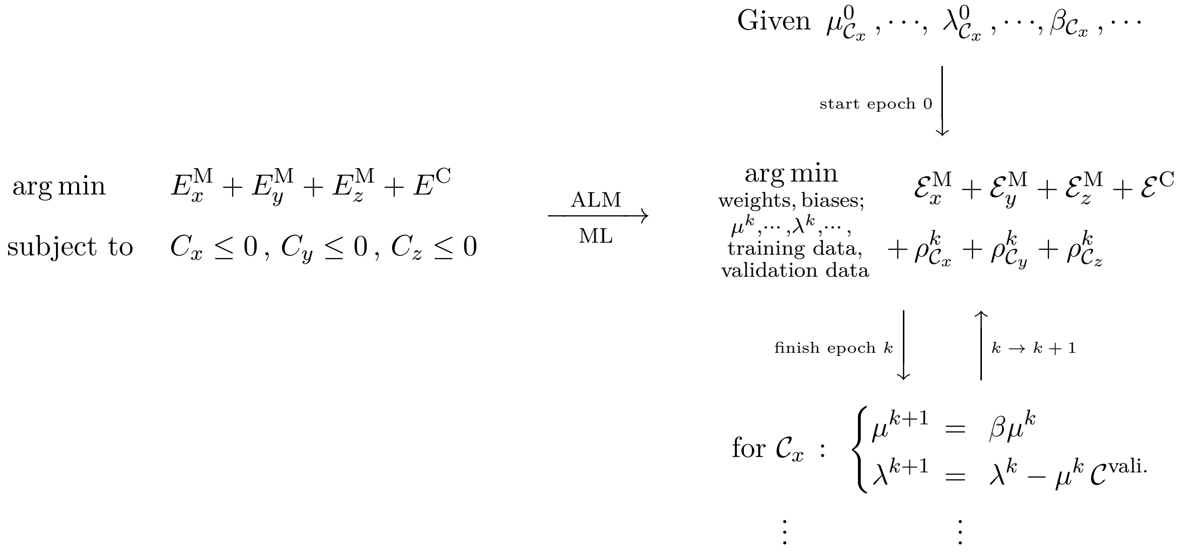

Given $\mu^0_{\mathcal{C}_x}, \cdots, \lambda^0_{\mathcal{C}_x}, \cdots, \beta_{\mathcal{C}_x}, \cdots$
start epoch 0
arg min $E^{\mathrm{M}}_x + E^{\mathrm{M}}_y + E^{\mathrm{M}}_z + E^{\mathrm{C}}$
subject to $C_x \leq 0\,, C_y \leq 0\,, C_z \leq 0$
ALM
ML
arg min weights, biases; $\mu^k, \cdots, \lambda^k, \cdots$, training data, validation data
$\mathcal{E}^{\mathrm{M}}_x + \mathcal{E}^{\mathrm{M}}_y + \mathcal{E}^{\mathrm{M}}_z + \mathcal{E}^{\mathrm{C}} + \rho^k_{\mathcal{C}_x} + \rho^k_{\mathcal{C}_y} + \rho^k_{\mathcal{C}_z}$
finish epoch $k$
$k \to k+1$
for $\mathcal{C}_x$ : $\begin{cases} \mu^{k+1} = \beta\mu^k \\ \lambda^{k+1} = \lambda^k - \mu^k\, \mathcal{C}^{\text{vali.}} \end{cases}$
$\vdots$ $\vdots$

Figure 2: Schematical representation of the ML model's constrained objective, posed by Eqs. (3) and Eqs. (4), in its ML-ALM formulation.

### 2.6 Feature Selection

We propose that Eq. (1) contains candidate features to predict the closure terms. Specifically, its left-hand side evaluated at previous time levels provides the ML model the history of the closure terms, which is useful for their prediction at $t^{n+1}$. Correspondingly, we include the resolved-scale weak-form terms in the features, except the time-derivative term. These are supplemented with the sums of the weak-form terms, which do include the time-derivative terms. Since the energy constraint Eqs. (4) are dependent on the velocity basis coefficients, we include these as part of the features. Finally, we also include the wall-normal distance, $d$, as the solution is anisotropic in the wall-normal direction.

The features mentioned above are associated with specific locations in space and time. We include these features, for each element weighting function, from two spatiotemporal element stencils, one for the features in the momentum equations and one for the features in the continuity equation. To specify the extent of the momentum-equation stencil in space, we note that local unresolved scales are dependent on inflow conditions, and thus it is logical to include upstream elements. More precisely, in [5] it is demonstrated that the local unresolved scales associated with $H^1$ projections of exact 2D advective solutions are only strong functions of the residual in a limited number of upstream locations, due to the form of the corresponding fine-scale Green's function. To capture this effect while limiting the size of the total feature set, we use a spatial stencil which includes the considered element, and four elements upstream of the considered element. We also include one downstream element. For the continuity-equation stencil, we note that the fine-scale Green's function for the $H^1$ projection of exact 2D elliptic operators is symmetric in nature [6]. Therefore, we use the momentum-equation stencil and add both spanwise elements immediately adjacent to the element under consideration. For the wall-normal direction, we include only one adjacent element that is dependent on the closest channel wall, with elements closest to the $z = 0$ wall including the $k + 1$ element, and elements closest to the $z = 2\delta$ wall including the $k - 1$ element. Finally, we include the velocity basis coefficients from the continuity-equation stencil. Given that the problem at hand is unsteady, we must also account for inflow dependency in time. We consider element-length CFL numbers of $O(1)$, so we provide the networks with the above-mentioned features from both time levels $t^n$ and $t^{n-1}$.

Based on the statistical histograms placed in Appendix A of [7], we observed strong correlations between the weak-form terms against the closure terms, with stronger correlations from upstream elements than from the weak-form terms of the current element. This confirms the previous observations used to construct the spatiotemporal stencils. We also observed different correlational strengths across the various element weighting functions, with various symmetries between the functions. However, the correlational symmetries of one function with another function is not the same for all features, making it difficult to categorise any of the functions together. This motivated the use of one MLP per element weighting function instead of a single MLP in order to simplify any of the relations between the weak-form terms and closure terms. We did not omit any feature that showed no correlation to the closure terms, as a seemingly useless feature may show significant correlation when paired with another feature [8]. The summary of the 217 features per MLP can be found in Tab. 1.

### 2.7 Training Routine

The 217 features and network architecture yields 1 575 604 free parameters to fit, per MLP. One data example requires three time steps from the data set for construction, thereby removing the first and last time steps from each eddy-turnover data set as a source of data examples. This reduces the usable snapshots for data examples to 472 data snapshots, as we gathered the 480 snapshots in four segments. The data-to-parameter ratio is then ∼19.6 : 1 (30 932 992 : 1 575 604) per MLP when considering a 2× data augmentation related to the wall-normal distance feature being calculable from either of the two channel walls.

The MLP training was conducted using Tensorflow 2.11. The data was randomly split between test data and validation/training data of 25%−75%. The latter was further split 15%−85% into validation and training data respectively. The batch size was kept to 2500.

Each MLP was initialised with $\mu^0 = 0$ and $\lambda^0 = 0$ for all constraints, default weights and biases, and trained for 100 epochs with a cyclical learning rate. This yields a suitable initial condition for the constrained optimisation problem. The next 200 epochs began with $\mu^{99} = 2$ and $\lambda^{99} = 0$, and a $\beta$ such that $\mu^k$ doubles every 25 epochs. These 200 epochs also used a cyclical learning rate. These choices were mostly arbitrary in nature, chosen to be fairly “aggressive” so that constraint effects can be observed while maintaining low training times. The change between $\mu^{98}$ and $\mu^{99}$ was made internally in the beginning-of-epoch member function using a conditional statement.

## 3 RESULTS AND DISCUSSION

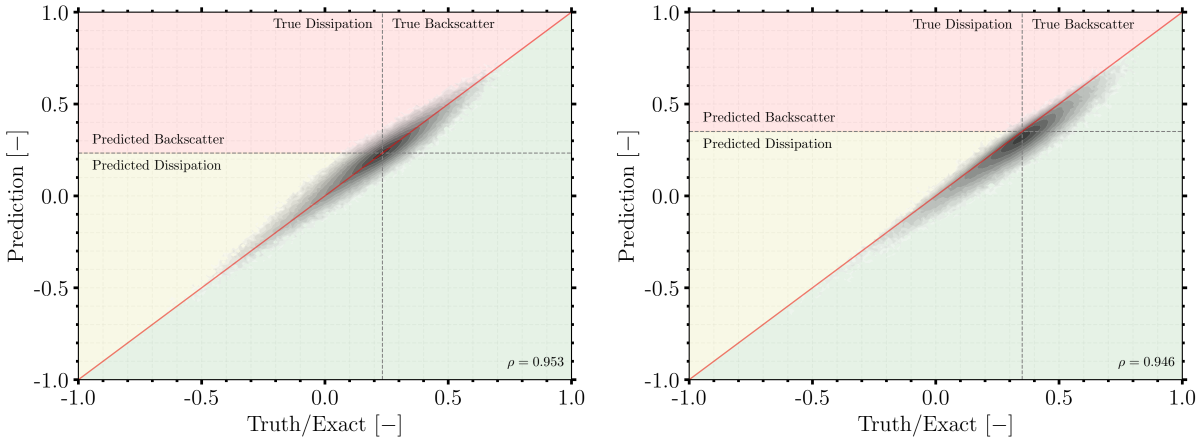
### 3.1 Network Histograms

We limit our discussion to the momentum term predictions from the $s = 1$ MLP, and consider the amount of energy these predictions drain/add to the test case. Figure 3 and Fig. 4 display the $t^{n+1}$ closure-term resolved-scale kinetic-energy contribution predictions compared with truth values in histogram format based on 500 000 samples. The data was normalised between -1.0 and 1.0 relative to the truth. The Pearson correlation coefficient is given in the bottom right corner of the graphs.

The results in Fig. 3a were calculated using a network with both $\mu^k$ and $\lambda^k$ set to zero, for all constraints, for the entire training procedure, producing a data footprint centred around the $y = x$ curve. In comparison, the results in Fig. 3b were calculated using a network as mentioned in Section 2.7, which produces a footprint centred below the $y = x$ curve. The latter is in the excessive dissipation region, thus demonstrating the efficacy of the aforementioned energy constraints. However, both the data footprint in Fig. 4a and Fig. 4b remain nearly centred along the $y = x$ curve. This likely results from the dominating magnitude of the $x$ component of the energy constraint.

### 3.2 Wall-Normal Correlations

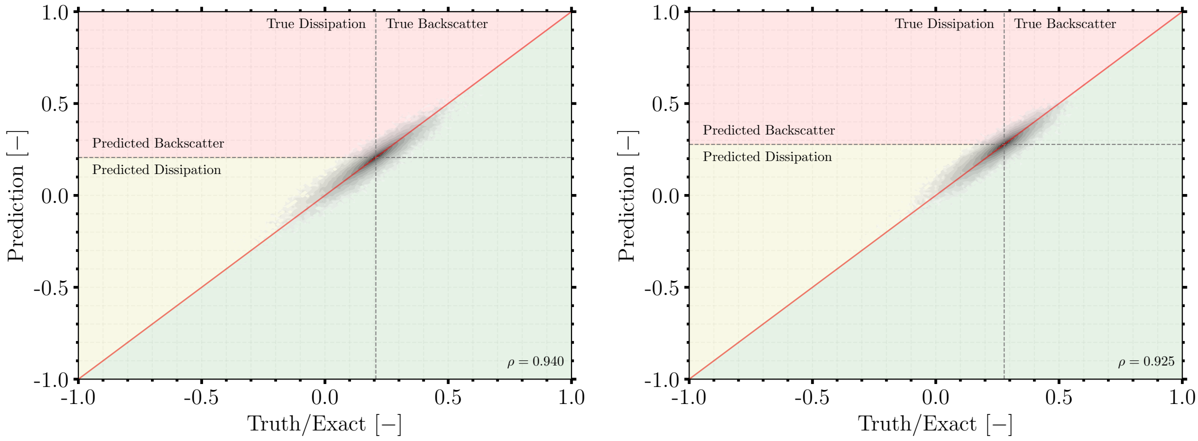
The Pearson correlation coefficient between the ML predictions and their truth values is also plotted as a function of wall-normal coordinates in Fig. 5. It is notable that the penalty term does not induce a change in the profiles between the two subfigures, suggesting that the ALM loss function is inducing a bias in the predictions, while not degrading prediction capability. The slightly asymmetrical profiles are due to the transient nature of the relatively short time period considered for data collection. Generally, the centre-channel correlations are the weakest, and are attributed to the lack of energy production/dissipation in that region (hence, smaller in magnitude to zero). Therefore, the correlations calculation in the channel centre are more sensitive to erroneous predictions than those in the near-wall region.



(a) $\left[-\,\overline{a}_{\,e,s,x}\right]_{\text{truth}}\left[\left(\overline{w}_{e,s,x}, \mathcal{L}'_M\left(\overline{\boldsymbol{u}}, \boldsymbol{u}', p'\right)_x\right)_{\Omega_e}\right]$

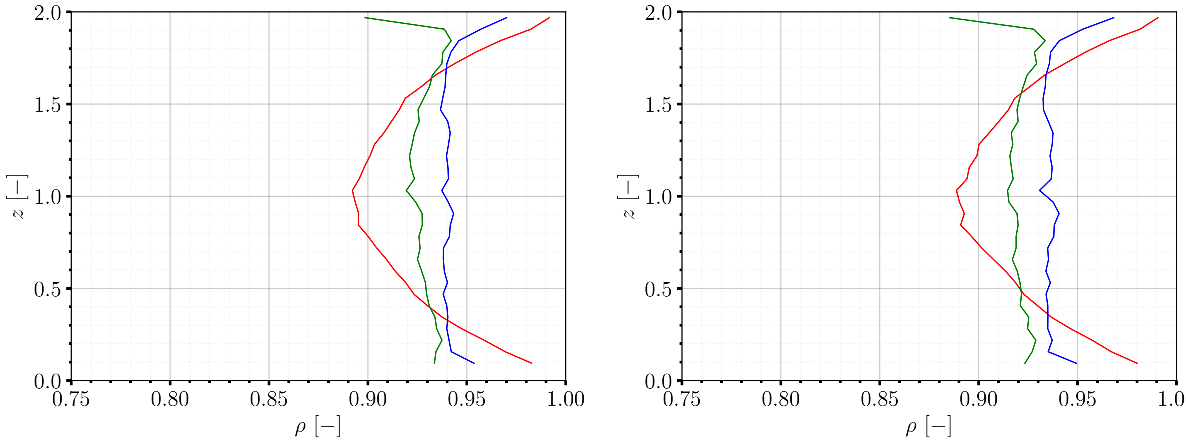
(b) $\left[-\,\overline{a}_{\,e,s,x}\right]_{\text{truth}}\left[\left(\overline{w}_{e,s,x}, \mathcal{L}'_M\left(\overline{\boldsymbol{u}}, \boldsymbol{u}', p'\right)_x\right)_{\Omega_e}\right]$

Figure 3: $x$-momentum closure-term kinetic-energy contributions without penalty (a) and with penalty (b). —— Perfect-relation line, - - - - - Denormalised data axes



(a) $\left[-\,\overline{a}_{e,s,y}\right]_{\text{truth}}\left[\left(\overline{w}_{e,s,y}, \mathcal{L}'_M\left(\overline{\boldsymbol{u}}, \boldsymbol{u}', p'\right)_y\right)_{\Omega_e}\right]$ (b) $\left[-\,\overline{a}_{e,s,z}\right]_{\text{truth}}\left[\left(\overline{w}_{e,s,z}, \mathcal{L}'_M\left(\overline{\boldsymbol{u}}, \boldsymbol{u}', p'\right)_z\right)_{\Omega_e}\right]$

Figure 4: $y$-momentum (a) and $z$-momentum (b) closure-term kinetic-energy contributions with penalty. — Perfect-relation line, - - - - Denormalised data axes



(a) without penalty (b) with penalty

Figure 5: Closure-term kinetic-energy contribution correlations as a function of wall-normal coordinate. — $x$-momentum, — $y$-momentum, — $z$-momentum

## 4 CONCLUSION

We introduced an energy-biased ML training approach for the prediction of VMS Navier-Stokes closure terms, such that excessive backscatter / insufficient dissipation predictions are reduced. This was achieved by formulating model requirements from which a constrained problem is generated, then modifying the ALM to suit ML training methods. As a test case, the closure terms associated with a $\mathrm{Re}_\tau = 180$ TCF flow case were predicted using MLPs. The a-priori results successfully showed the desired biased energy predictions.

## REFERENCES

[1] Bazilevs, Y., Calo, V. M., Cottrell, J. A., Hughes, T. J. R., Reali, A., & Scovazzi, G. (2007). Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering, 197*(1–4), 173–201. doi:10.1016/j.cma.2007.07.016

[2] Janssens, M. and Hulshoff, S.J., (2022). Advancing artificial neural network parameterization for atmospheric turbulence using a variational multiscale model. *Journal of Advances in Modeling Earth Systems, 14*(1). doi:10.1029/2021MS002490

[3] S. Rajampeta, *Stabilization Strategies for a Variational Multiscale Method Coupled with an Artificial Neural Network.* [Master's thesis, Delft University of Technology]. TUDelft Education Repository. url:https://repository.tudelft.nl/record/uuid:2114d9a4-7333-47bd-9793-77721a8303b0

[4] Nocedal, J., & Wright, S. (2006). *Numerical Optimization* (2nd ed.). Springer. doi:10.1007/978-0-387-40065-5

[5] Hughes, T. J. R., & Sangalli, G. (2007). Variational multiscale analysis: The fine-scale green's function, projection, optimization, localization, and stabilized methods. *SIAM Journal on Numerical Analysis, 45*(2), 539–557. doi:10.1137/050645646

[6] Shrestha, S. , Dekker, J. , Gerritsma, M., Hulshoff, S. J., and Akkeman, I. (2024), Construction and application of an algebraic dual basis and the Fine-Scale Greens' Function for computing projections and reconstructing unresolved scales, *Computer Methods in Applied Mechanics and Engineering, 422*. doi:10.1016/j.cma.2024.116833

[7] Bettini, A. (2023). *Energy-Conservative Data-Driven Modelling for the two-scale Navier-Stokes Equations* [Master's thesis, Delft University of Technology]. TUDelft Education Repository. url:https://repository.tudelft.nl/record/uuid:bd34fb15-9b2d-4cb0-bec9-0ca76ef98cc7

[8] Guyon, I., & Elisseef, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning, 3*, 1157-1182. url:https://ei.is.mpg.de/publications/2685

Table 1: Feature set associated to $(\text{MLP})_s$ and the $(i,j,k)$-th element. ✓ indicates inclusion in feature set. Temporal index of the relevant term is given by the relevant column entry. See Section 2.3 for $(i,j,k)$ index definitions.

| | **Feature** | **Stencil** | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $(i,j,k)$ | | $(i-1,j,k)$ | | $(i-2,j,k)$ | | $(i-3,j,k)$ | | $(i-4,j,k)$ | | $(i+1,j,k)$ | | $(i,j-1,k)$ | | $(i,j+1,k)$ | | $(i,j,k\pm 1)$ | |
| | | $t^n$ | $t^{n-1}$ | $t^n$ | $t^{n-1}$ | $t^n$ | $t^{n-1}$ | $t^n$ | $t^{n-1}$ | $t^n$ | $t^{n-1}$ | $t^n$ | $t^{n-1}$ | $t^n$ | $t^{n-1}$ | $t^n$ | $t^{n-1}$ | $t^n$ | $t^{n-1}$ |
| **Coar. Scale Sums** | $\left(\overline{w}_{e,s,x},\, \overline{\mathcal{L}}_M\left(\overline{\boldsymbol{u}},\overline{p}\right)_x\right)_{\Omega_e}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | $\left(\overline{w}_{e,s,y},\, \overline{\mathcal{L}}_M\left(\overline{\boldsymbol{u}},\overline{p}\right)_y\right)_{\Omega_e}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | $\left(\overline{w}_{e,s,z},\, \overline{\mathcal{L}}_M\left(\overline{\boldsymbol{u}},\overline{p}\right)_z\right)_{\Omega_e}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| **Conv. Terms** | $\left(\overline{w}_{e,s,x},\, \left(\overline{\boldsymbol{u}}\cdot\nabla\overline{\boldsymbol{u}}\right)_x\right)_{\Omega_e}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | $\left(\overline{w}_{e,s,y},\, \left(\overline{\boldsymbol{u}}\cdot\nabla\overline{\boldsymbol{u}}\right)_y\right)_{\Omega_e}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | $\left(\overline{w}_{e,s,z},\, \left(\overline{\boldsymbol{u}}\cdot\nabla\overline{\boldsymbol{u}}\right)_z\right)_{\Omega_e}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| **Pres. Terms** | $\left(\overline{w}_{e,s,x},\, \left(\nabla\overline{p}\right)_x\right)_{\Omega_e}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | $\left(\overline{w}_{e,s,y},\, \left(\nabla\overline{p}\right)_y\right)_{\Omega_e}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | $\left(\overline{w}_{e,s,z},\, \left(\nabla\overline{p}\right)_z\right)_{\Omega_e}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| **Diff. Terms** | $\left(\nabla\overline{w}_{e,s,x},\, \left(\nabla^s\overline{\boldsymbol{u}}\right)_x\right)_{\Omega_e}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | $\left(\nabla\overline{w}_{e,s,y},\, \left(\nabla^s\overline{\boldsymbol{u}}\right)_y\right)_{\Omega_e}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | $\left(\nabla\overline{w}_{e,s,z},\, \left(\nabla^s\overline{\boldsymbol{u}}\right)_z\right)_{\Omega_e}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| **Div. Term** | $-\left(\nabla q_{e,s},\, \overline{\boldsymbol{u}}\right)_{\Omega_e}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **Vel. Basis Coefs.** | $\overline{a}_{e,s,x}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | $\overline{a}_{e,s,y}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | $\overline{a}_{e,s,z}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **Norm. dist.** | $d$ | ✓ | | | | | | | | | | | | | | | | | |