Applications in Energy and Combustion Science 12 (2022) 100087

Contents lists available at ScienceDirect

Applications in Energy and Combustion Science

journal homepage: www.elsevier.com/locate/jaecs

BLASTNet: A call for community-involved big data in combustion machine learning

Wai Tong Chung [a,*], Ki Sung Jung [b], Jacqueline H. Chen [b], Matthias Ihme [a,c]

[a] Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA
[b] Combustion Research Facility, Sandia National Laboratories, Livermore, CA 94550, USA
[c] Department of Photon Science, SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA

ARTICLE INFO

Dataset link: https://blastnet.github.io, https://github.com/IhmeGroup/lossy_ml

Keywords:
Big data
Deep learning
Direct numerical simulation
BLASTNet

ABSTRACT

Many state-of-the-art machine learning (ML) fields rely on large datasets and massive deep learning models (with $\mathcal{O}(10^9)$ trainable parameters) to predict target variables accurately without overfitting. Within combustion, a wealth of data exists in the form of high-fidelity simulation data and detailed measurements that have been accumulating since the past decade. Yet, this data remains distributed and can be difficult to access. In this work, we present a realistic and feasible framework which combines (i) community involvement, (ii) public data repositories, and (iii) lossy compression algorithms for enabling broad access to high-fidelity data via a network-of-datasets approach. This Bearable Large Accessible Scientific Training Network-of-Datasets (BLASTNet) is consolidated on a community-hosted web-platform (at https://blastnet.github.io/), and is targeted towards improving accessibility to diverse scientific data for deep learning algorithms. For datasets that exceed the storage limitations in public ML repositories, we propose employing lossy compression algorithms on high-fidelity data, at the cost of introducing controllable amounts of error to the data. This framework leverages the well-known robustness of modern deep learning methods to noisy data, which we demonstrate is also applicable in combustion by training deep learning models on lossy direct numerical simulation (DNS) data in two completely different ML problems — one in combustion regime classification and the other in filtered reaction rate regression. Our results show that combustion DNS data can be compressed by at least 10-fold without affecting deep learning models, and that the resulting lossy errors can even improve their training. We thus call on the research community to help contribute to opening a bearable pathway towards accessible big data in combustion.

1. Background

1.1. Introduction: A big view of machine learning

Combustion machine learning (CombML) offers numerous opportunities in advancing predictive modeling, scientific discoveries, and intelligent control [1]. One of the most crucial aspects of machine learning (ML) is the availability of data, which in combustion, typically exist in the form of simulation data and experimental measurements. In many ML fields outside of combustion, massive and diverse datasets are the key components in ensuring high predictive accuracy and good generalizability [2]. For example, in computer vision, a state-of-the-art ML field, massive and diverse datasets such as the image recognition dataset ImageNet [3] (170 GB, 1000 classes, 1.4M labeled images) have enabled ML methods to out-perform human capabilities in image recognition [4,5]. This achievement was made possible by the co-existence of deep learning architectures, such as the 152-layer deep ResNet [5], and the aforementioned ImageNet dataset, along with its corresponding community-involved image recognition competition [4], where researchers could develop ML methods without the laborious task of data collection, and compare results in a transparent manner via an accessible benchmark dataset.

In contrast, datasets found in flow physics, such as the (~500 TB) Johns Hopkins Turbulence Database [6], are not as diverse (currently consisting of nine flow configurations) but can be much greater in size due to increased degree-of-freedom and resolution requirements when compared to digital images. The fidelity and quality of this type of dataset is highly beneficial for applications in detailed scientific analysis, but its lack of diversity, when compared to other datasets [3,7] from the broader ML community, can be detrimental for training ML algorithms, especially for predicting in unseen configurations. In order to meet this challenge, the flow physics community has developed knowledge-guided or physics-informed ML [8], where domain

* Corresponding author.
E-mail addresses: wtchung@stanford.edu (W.T. Chung), mihme@stanford.edu (M. Ihme).

https://doi.org/10.1016/j.jaecs.2022.100087
Received 9 June 2022; Received in revised form 26 September 2022; Accepted 27 September 2022
Available online 13 October 2022

knowledge can be leveraged towards augmenting datasets, constraining optimization routines, and customizing model architectures to learn well from small scientific datasets, *i.e.*, the *small data* regime.

Outside of flow physics, ML research tends to focus on *big data*. Many improvements (including breakthroughs in model architecture such as residual blocks [5], batch normalization [9], and rectified linear units [10]) in deep learning have been tailored towards developing *big models* [11] that gain higher predictive accuracy with growing amounts of data [2]. We note that both small and big data paradigms do not necessarily compete, and remarkable results have been achieved within CombML by combining ideas from both approaches [12].

Recent developments in big data ML could inspire potential research directions for CombML. In natural language processing (NLP), *foundation models* [13] have led to state-of-the-art accuracies in a wide range of language prediction tasks. A foundation model is a broadly accessible and big ML model (typically with $\mathcal{O}(10^9)$ trainable parameters) that has been pre-trained on massive and diverse datasets, which can then be fine-tuned at later stages, by further training with smaller specific datasets (through transfer learning [14]), for application to specific problems. This eliminates the need to build and train a powerful ML model from scratch, and reduces the amount of data required to solve a tailored ML problem after the foundation model has been pre-trained and shared. With this new paradigm, one can envision a future development where only small amounts of additional data is needed to fine-tune pre-trained CombML foundation models in order to make accurate and affordable predictions of flame physics and chemistry in unseen combustion configurations. However, this ML approach is currently largely feasible only in NLP, where low-dimensional readily labeled text data can be easily mined. In computer vision, while the practice of transfer learning still persists, foundation models are comparatively nascent due to dimensionality of images (height, width, and channels, *i.e.*, $N_H \times N_W \times N_C$), and the larger cost of generating labels, which typically involves manually annotating images for image recognition or object detection.

In CombML, the massive, diverse, and labeled dataset required to eventually develop foundation models can certainly exist. A recent review [1] on CombML identified over 200 direct numerical simulation (DNS) cases, which can potentially serve as the basis of a public CombML database. We envision that this database can be further populated with a wide variety of existing experimental measurements and large-eddy simulation (LES) data, as well as future data that is expected to grow in complexity and size with advancements in measurement techniques and computational capabilities. Since simulation and experimental data are readily labeled with high-resolution quantities, CombML does not face challenges tied to laboriously annotating datasets, as seen in computer vision. Instead, this community faces the Herculean challenge of storing and accessing data with much higher degrees-of-freedom (length, height, width, time, and scalars, *i.e.*, $N_L \times N_H \times N_W \times N_t \times N_\phi$). This becomes especially true when considering the scale of data from peta/exascale simulations [15,16] and high-speed measurements [17].

In summary, massive, diverse, and public datasets for reacting and non-reacting flows are necessary to advance CombML within the big data paradigm. Specifically, the existence of these datasets would enable CombML researchers:

- To minimize the laborious task of data collection, which enables researchers to focus on advancing CombML techniques.
- To make objective and transparent evaluations of predictive accuracies from different ML approaches on *common* datasets.
- To further leverage existing architecture advances from the big data paradigm, and to foster a CombML paradigm that aligns with the broader ML community.
- To improve accessibility to state-of-the-art transfer learning practices towards eventually building CombML foundation models that can solve a wide range of scientific and engineering problems.

1.2. Requirements and pathways towards massive deep learning datasets in combml

We now discuss a set of requirements for a large and diverse CombML dataset, which we note are different to the requirements of centralized high-fidelity databases [18]. These requirements supplement scientific data management principles such as FAIR [19]:

- **Massive and diverse:** Large and diverse datasets are crucial for ensuring good accuracy and generalizability in state-of-the-art ML algorithms [2]. For example, super-resolution models [20] in computer vision, which have also been applied towards turbulence modeling [12], are typically trained with $\mathcal{O}(10^3)$ samples [7] of high-resolution images with great diversity. To establish a similar diverse dataset in CombML, we propose a *living dataset* that continuously accumulates towards at least a total of $\mathcal{O}(10^3)$ individual snapshots from $\mathcal{O}(10^2)$ different configurations. Since this volume of data cannot be easily generated from any individual researcher, a community-involved approach should be considered.
- **Accessible and consolidated:** Significant resources will be required to store and share $\mathcal{O}(10^3)$ snapshots of high-dimensional data without careful treatment. While services, such as Globus [21], currently enable researchers to share data directly from computing and storage facilities, the private permissions required for this service can hinder accessibility. Public accessibility to scientific data is typically achieved by building a centralized database, such as with the Johns Hopkins Turbulence Database [6] or the Sloan Digital Sky Survey [22]. These centralized scientific public datasets typically require dedicated storage infrastructure which consists of a database cluster, web interface system, and dedicated infrastructure for data analysis. While this approach has lead to reliable sources of scientific data, this can incur significant capital costs, as well as additional costs and human labor for maintaining and updating the centralized database. An alternate approach would be to leverage open-source and free ML repositories such as Kaggle [23], which are currently restricted by a $\mathcal{O}(100)$ GB limit that may not be sufficient for high-fidelity data, as a single snapshot of petascale DNS data can often exceed this limit.
- **Sufficient data quality:** The availability of good quality data is without a doubt important to data-driven methods. However, we must emphasize that this dataset must only be sufficiently good for training big supervised ML algorithms. A recent study [24] demonstrated that ImageNet and other popular benchmark datasets contain up to 10% label error. Despite these errors in training data, ML continues to transform numerous engineering and scientific endeavors. This is because modern deep learning algorithms are inherently robust to noisy data [25]. In fact, it is well-known that introducing small amounts of noise to a training set can be beneficial for improving the generalization of neural networks [26], and is a common form of *data augmentation* [27]. This has significant implications towards the use of compression and dimensionality reduction algorithms for mitigating storage constraints. However, since some combustion applications involve safety-critical conditions [1], we note that the use of noisy data with ML under these conditions should be treated with caution and thoroughly investigated prior to deployment.

1.3. Dimensionality reduction and lossy compression

Combustion modeling has embraced dimensionality reduction methods for chemical and manifold reduction, resulting in compact models in turbulent reacting flows with an acceptable amount of error. Interpretable data-driven dimensionality techniques such as principal component analysis (PCA) [28] have also been employed to identify

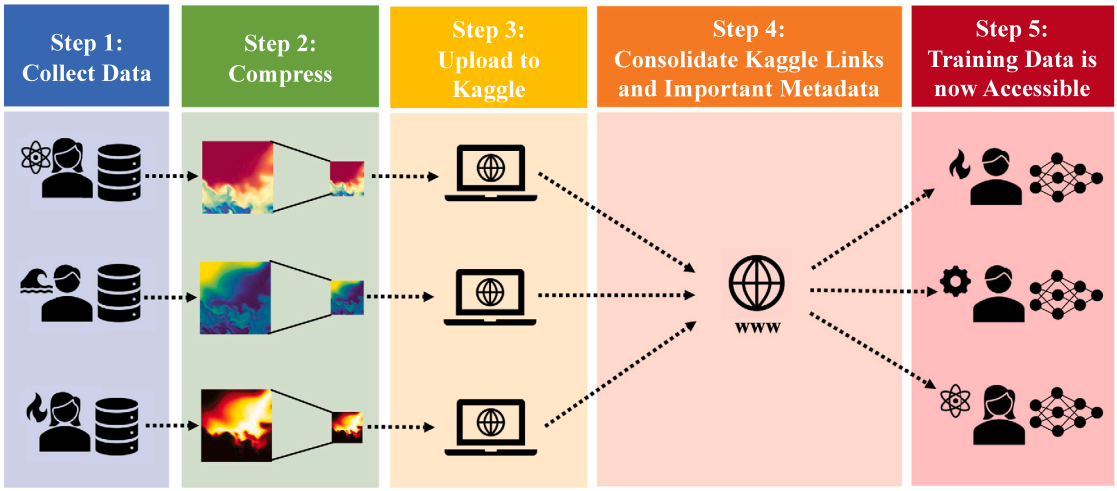


Fig. 1. BLASTNet: A community-involved pathway to big combustion data at https://blastnet.github.io/.

optimal low-dimensional manifolds that can be transported through conservation equations [29,30]. A related practice involves projecting large dimensional state spaces onto low-order manifolds by leveraging well-understood physical principles behind representative flame configurations. This approach has resulted in the formulation of models such as the Burke–Schumann solution [31], the flame-prolongation in intrinsic lower-dimensional manifold (FPI) [32], the flamelet-generated manifold (FGM) method [33], and the flamelet/progress variable (FPV) method [34,35].

Since big ML algorithms are robust to noisy data [25], dimensionality reduction algorithms can be applied towards high-fidelity data that exceed size restrictions before storage in public ML repositories. However, errors obtained during PCA reduction can be difficult to control, which may result in unpredictable behavior if present in an ML dataset. More complex dimensionality reduction methods such as autoencoders [36] have been shown to be more effective (but less interpretable) than PCA at compressing data while avoiding significant information loss [37,38], but can still be difficult to control and are computationally expensive.

Recently, lossy compression algorithms [39] have gained popularity in applications with high-fidelity data due to increasing storage and I/O bottlenecks as computational capabilities and high-speed measurements outgrow disk capabilities. Similar to dimensional reduction techniques, these algorithms reduce the size of data, while introducing small errors to the compressed data. This is in contrast to lossless compression algorithms, which preserve all information during compression. As shown in Table 1, lossy compression algorithms can achieve significantly higher compression ratios (defined as the ratio between the sizes of original data and compressed data, respectively) than lossless compression. In addition, many of these lossy compression methods have been tailored towards compressing high-fidelity scientific data at tractable computational costs and include error-boundedness, which enable users to determine and control the desired error/fidelity of the compressed data. Thus, these methods can be employed towards guaranteeing a level of desired quality when compressing ML training data.

Even an $\mathcal{O}(10)$-fold compression could turn the storage of high-fidelity combustion simulation data into a bearable task. For instance, a ten-fold compression on petascale DNS data (with 200 GB per snapshot) would result in a few compressed snapshots that can be readily shared on public ML repositories such as Kaggle [23]. This process could be repeated at modest effort for multiple DNS configurations, with links to each distributed dataset curated and hosted on a single community-maintained webpage. Employing such an approach, which we detail in Section 1.4, would eliminate the time and labor required to build and maintain a centralized database by making use of the open-source nature of the broader ML community.

Table 1
Comparison of compression ratios achieved by compression algorithms on scientific datasets.
Source: Adapted from [39].

Compressor	Type	Compression Ratio
Deduplication [40]	Lossless	1.5 ~ 3
gzip [41]	Lossless	1.5 ~ 2
FPC [42]	Lossless	1.2 ~ $\mathcal{O}(10)$
ISABELA [43]	Lossy	2.1 ~ $\mathcal{O}(100)$
SZ2 [44]	Lossy	3 ~ $\mathcal{O}(100)$
ZFP [45]	Lossy	3 ~ $\mathcal{O}(100)$
TTHRESH [46]	Lossy	5.1 ~ $\mathcal{O}(100)$

1.4. BLASTNet: A big data framework for the combustion community

In this work, we propose an affordable weakly centralized framework that combines the use of lossy compression algorithms with public open-source data repositories and community involvement for sharing massive and diverse deep learning training data for combustion. In particular, this framework is targeted towards improving the diversity of accessible scientific training data for ML, and thus serves a distinct purpose when compared to existing high-fidelity databases [6].

Fig. 1 summarizes our proposed framework, *Bearable Large Accessible Scientific Training Network-of-Datasets (BLASTNet)*. BLASTNet is aimed at providing accessibility to raw simulation and measurement data (from a diverse range of configurations), which can be employed for solving a wide range of deep learning problems. This data is shared through Kaggle [23], which has an interface suitable for scientific clusters, provides the datasets with a unique Kaggle ID, and also provides the ability to register digital-object-identifiers (DOI) for each dataset. Since the data is uploaded primarily using Kaggle, any modifications are tracked via their history and version control system. In cases where a single sample of data exceeds storage limits in Kaggle (currently restricted to $\mathcal{O}(100)$ GB), this data is compressed at a desired level of error, with an error-bounded lossy compression algorithm. Here, we recommend the use of a consistent compression algorithm, SZ2 [44], so that all lossy compressed datasets can be shared in a consistent data format.

The link to, description of, and all other metadata (boundary conditions, initial conditions, fuel composition, DOI, numerics, chemical and transport properties, grid, and timesteps) from the dataset can then be shared onto a community-hosted webpage [47], at https://blastnet.github.io/, which curates all existing distributed ML datasets and provides a *centralized search interface* to enable convenient public access. An example the metadata format is provided in B.

The community-hosted webpage also provides *tutorials* for compressing, decompressing, sharing, and accessing the lossy data. BLASTNet also sets *standards* (further detailed in Section 5), and screens the data to ensure that these standards are met. Note that all data and metadata contributed to BLASTNet will adhere to FAIR principles [19] for scientific data management, as further detailed in C. A community *discussion forum* is also hosted on BLASTNet in order to receive continuous feedback from users and to provide a platform for additional support to users. Importantly, to ensure that fair attribution is provided in this open-source project, a version update will be applied to BLASTNet each time a new contribution is provided by the research community to include each individual contributor into BLASTNet's list of authors, which is a common practice in open-source software [48].

1.5. Objectives

The objectives of this work can thus be summarized as follows:

- To advocate the benefits of a massive, diverse, and distributed CombML datasets for deep learning.
- To introduce a platform, at https://blastnet.github.io/ [47], for a community-involved network-of-datasets (BLASTNet).
- To demonstrate lossy compression as an affordable and expedited pathway for storing and sharing state-of-the-art high-fidelity data.
- To quantify the compression gained from lossy algorithms and to demonstrate the robustness and limitations of deep learning algorithms to the resulting lossy errors.
- To call on the research community to contribute data to BLASTNet.

We note that a key component of BLASTNet operates under the assumption that deep learning methods are robust to controllable amounts of noise introduced during lossy compression. To investigate the validity of this assumption within combustion, we apply a lossy compression algorithm (SZ2 [44]) to DNS data of a turbulent lifted hydrogen jet flame in heated co-flow [49], and study the effects of lossy data on training deep learning models in two completely different ML problems, namely combustion regime classification and filtered reaction rate regression. The investigated DNS dataset is described further in Section 2, while the chosen lossy compression algorithm and deep learning architecture are detailed in Section 3. Results from this investigation are presented in Section 4, before concluding in Section 5.

2. DNS dataset

A three-dimensional DNS dataset from a previous study [49] of a turbulent lifted hydrogen jet flame in heated co-flow air is used to demonstrate the robustness of deep learning models to lossy errors. Fig. 2 shows a schematic of the DNS configuration. A diluted fuel mixture (65% H_2 and 35% N_2 by volume) is issued from the central slot at an inlet temperature of 400 K. This central jet is surrounded on either side by co-flowing heated air streams with an inlet temperature of 850 K, at atmospheric pressure. The mean inlet axial velocity U_{in} is given by:

$$U_{in} = U_c + \frac{U_{jet}-U_c}{2}\left(\tanh\left(\frac{y+H/2}{0.1H}\right) - \tanh\left(\frac{y-H/2}{0.1H}\right)\right), \quad (1)$$

with mean inlet jet velocity $U_{jet} = 240$ m s^{-1}, mean inlet co-flow velocity $U_c = 2$ m s^{-1}, and jet width at the inlet $H = 2$ mm. Velocity fluctuations, obtained by generating an auxiliary homogeneous isotropic turbulence field, are fed from the inlet using Taylor's hypothesis.

This $2000 \times 1600 \times 400$ computational domain is $15H \times 20H \times 3H$ in the streamwise x-, transverse y-, and spanwise z-directions, respectively, resulting in a total of 1.28 billion cells. A uniform grid size of 15 μm is placed in x- and z-directions, while the y-directional grid is algebraically stretched outside the flame and shear zones. Improved non-reflecting boundary conditions [50,51] are adopted in the x- and y-directions and periodic boundary conditions are applied in the z-direction.

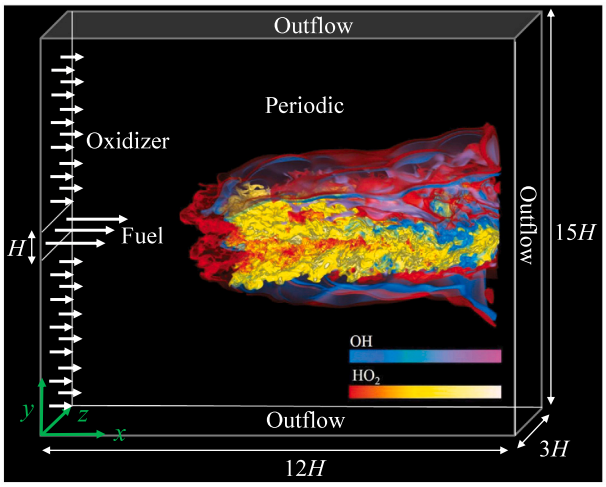


Fig. 2. H_2-air direct numerical simulation data [49] used in this study.

Table 2
Classification labels generated with flame index FI, progress variable C, and mixture fraction Z.

Label	Definition
Premixed Flame	$(C > 0.01)$ and $(\mathrm{FI} > 0)$ for all Z
Non-premixed Flame	$(C > 0.01)$ and $(\mathrm{FI} \leq 0)$ for all Z
Air	$(C \leq 0.01)$ and $(Z \leq 0.01)$
Fuel	$(C \leq 0.01)$ and $(Z > 0.90)$
Fuel–air Mixture	$(C \leq 0.01)$ and $(0.01 < Z \leq 0.90)$

The Sandia DNS code, S3D [52] was employed for solving the compressible Navier–Stokes, species continuity, and total energy equations. The employed detailed H_2-air chemical mechanism composed of nine species (H_2, O_2, H_2O, O, H, OH, HO_2, H_2O_2, and N_2) and 21 elementary reaction steps, was developed by Li et al. [18]. In the present study, a $1200 \times 300 \times 200$ sub-region of the DNS field (*i.e.*, a left half branch of the lifted jet flame) is sampled from a single 124 GB DNS snapshot, in order to reduce the computational cost during training and analysis while maintaining the fidelity of the flame structure. The size of the data subvolume was determined by ensuring that the overall structure of the partially premixed flame (i.e., lean/rich premixed flame and the trailing diffusion flame) and the upstream fuel/air mixture profiles can be clearly observed within the sub-volume throughout the simulation. This subvolume was also selected such that class imbalance [53] issues (where the proportion of one class greatly exceeds another) in the classification problems do not affect ML predictions. We employ this 72M-cell subvolume to demonstrate the robustness of deep learning models to noise from lossy compression algorithms in both classification and regression problems, as specified in Section 2.1 and Eq. (3), respectively.

2.1. Classification dataset

Within CombML, classification can be useful for optimizing numerical computations [54], detecting catastrophic events [55], and identifying combustion regimes [56]. As detailed in Table 2, we generate five classes of labels for the present classification problem, with the use of the progress variable $C = Y_{H_2O}$, mixture fraction Z as defined

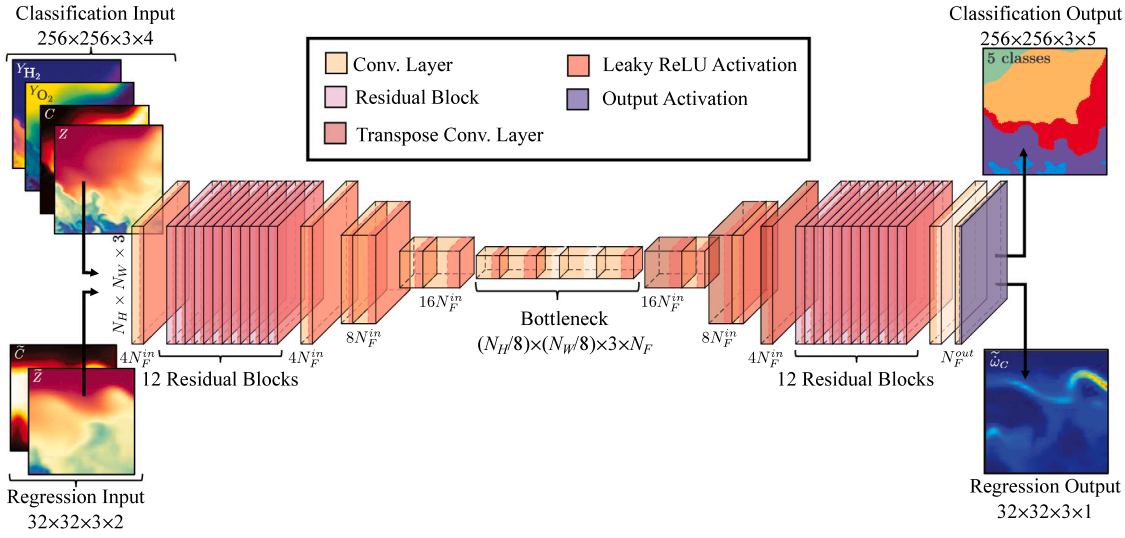


Fig. 3. Present 3-D CNN architecture. The number of filters in each layer N_F is represented in terms of the number of input channels N_F^{in}.

by Bilger [57], and flame index FI [58], which is defined by:

$$\mathrm{FI} = \frac{\nabla Y_{\mathrm{H}_2} \cdot \nabla Y_{\mathrm{O}_2}}{\|\nabla Y_{\mathrm{H}_2}\| \cdot \|\nabla Y_{\mathrm{O}_2}\|}. \tag{2}$$

These five labels were chosen (i) to account for a well-balanced proportion of classes, (ii) to investigate the effects of lossy compression on fine thresholds, and (iii) to investigate the effects of the gradient operator in Eq. (2) in magnifying lossy errors. We note that these classes have been defined for an illustrational purpose, and that more refined classes should be considered in contexts related to scientific discovery, if needed. For each label, we extract four flow features $\{Z, C, Y_{\mathrm{H}_2}, Y_{\mathrm{O}_2}\}$, and then divide the data into 268 sub-volumes, each with $256 \times 256 \times 3$ cells. Note that three cells in the z-axis is sufficient for preserving spatial information in these samples, since this configuration is homogeneous in the spanwise direction.

2.2. Regression dataset

Within CombML, regression is particularly popular for constructing turbulence closure [59], modeling thermodynamics and chemistry [60], and parameterizing combustion manifolds [30]. Here, we generate our regression label by filtering and down-sampling the DNS data to evaluate the Favre-filtered progress variable reaction rate $\widetilde{\dot{\omega}}_C$:

$$\widetilde{\dot{\omega}}_C(\boldsymbol{x}) = \frac{1}{\bar{\rho}} \int_V \rho \dot{\omega}_C^{DNS}(\boldsymbol{y}) G(\boldsymbol{x}-\boldsymbol{y}, \Delta_F)\, d\boldsymbol{y}, \tag{3a}$$

and

$$G(\boldsymbol{x}-\boldsymbol{y}, \Delta_F) = \left(\frac{6}{\pi \Delta_F^2}\right)^{3/2} \exp\left[\frac{-6(\boldsymbol{x}-\boldsymbol{y})^2}{\Delta_F^2}\right], \tag{3b}$$

where $\bar{\cdot}$ denotes a filtered quantity, $\widetilde{\cdot}$ is a Favre-filtered quantity, G is a Gaussian filter, and $\Delta_F = 8\Delta$ is the filter width, which is prescribed to be eight times larger than the DNS cell width Δ. This filter width corresponds to three cells (in a corresponding LES) for sufficiently resolving a laminar flame thickness of 0.3 mm, which is evaluated through a stoichiometric 1D premixed flame calculation. The quantity $\widetilde{\dot{\omega}}_C$ from turbulence-chemistry interaction is of interest within CombML, as shown in other studies [61,62], and is a suitable quantity to test the robustness of ML models to lossy errors, due to the presence of the exponential operator in the Arrhenius term, which can significantly magnify lossy errors. For each label, we extract two flow features $\{\widetilde{Z}, \widetilde{C}\}$ from this dataset, and then divide the data into 177 sub-volumes (each with $32 \times 32 \times 3$ cells) that encompass the flame region.

3. Methods

3.1. Deep learning

Fig. 3 shows the 3-D convolutional neural network (CNN) architecture used in both classification and regression problems. CNNs use a small moving window, known as a filter, that performs a mathematical operation (typically convolution and pooling [1]) on a neighborhood of pixels. When learning from spatial data, CNNs can be easier to train and achieve higher prediction accuracies, compared to other ML algorithms, due to the preservation of spatial information through the use of the CNN filter. In this work, we employ a deep learning architecture based on the convolutional autoencoder architecture by Glaws et al. [63], with the input channel N_F^{in} of the model modified to suit the number of features in the present classification ($N_F^{in} = 4$) and regression ($N_F^{in} = 2$) datasets and the CNN filter width reduced to three, which is a popular choice for CNN architectures [64]. A key component of this architecture is its autoencoder structure. Autoencoder networks can be thought of as a non-linear PCA [36], where raw features are automatically processed by the encoder into an embedded form which can then be forward-propagated by the decoder to generate complex predictions. The present network contains 93 layers and approximately 1M trainable parameters, with weights initialized via Xavier initialization [65], and contains 12 residual blocks [5] near the input and output, for improving training and avoiding vanishing gradients during back-propagation.

For the classification problem, a softmax output activation with five filters $N_F^{out} = 5$ (for the five classes) is used together with a categorical cross-entropy loss function, while a linear output activation with a single filter $N_F^{out} = 1$ is used for the regression problem with a mean-absolute-error (MAE) loss function. Train and validation procedures are further detailed in Appendix A.

3.2. Lossy compression

In this work, we employ the SZ2 compressor [44], which combines curve-fitting, the Lorenzo predictor, and data quantization for compressing scientific data. In principle, SZ2 (i) partitions field variables into clusters, (ii) iteratively searches for regression functions

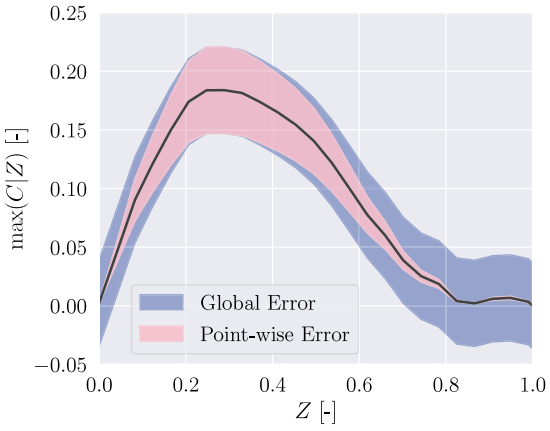


Fig. 4. Difference between global and point-wise error bounded control modes in SZ2, illustrated on the maximum conditioned progress variable $(C|Z)_{max}$.

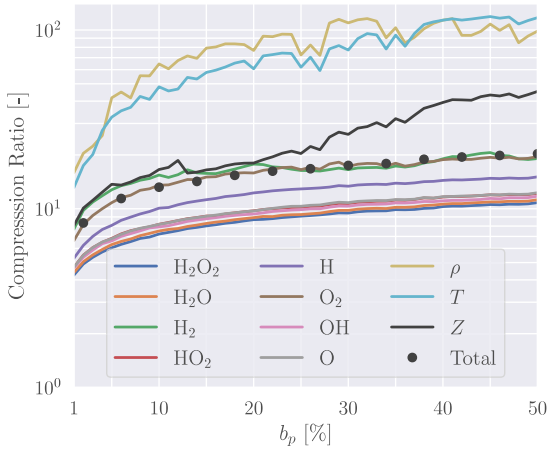


Fig. 5. SZ2 compression ratio of different scalar flowfields at varying point-wise error bounds b_p. Species quantities are expressed in mass fractions.

that can approximate each cluster with a guaranteed error-bound, and (iii) stores the quantized regression coefficients of the function and indices of the field variables for recomputing the original data during decompression. Data can be compressed effectively since the quantized coefficients and indices are much smaller than the original variables. Compression and decompression of the twelve quantities (temperature, density, pressure, as well as mass fractions of H_2, O_2, H_2O, O, H, OH, HO_2, H_2O_2, and N_2) in the thermo-chemical state–space for the present 72M subvolume requires a total of approximately 35 s wall-clock-time on a single Intel Xeon CPU. We note that SZ2 has been reported to be at least 2-times faster than the other lossy compressors listed in Table 1 [44,46].

Thus, SZ2 meets the criteria described in Section 1.3 for compressing high-fidelity data for a large public training database: (i) capable of high compression ratios, (ii) fast, and (iii) allows for bounded error control. While a global error bound is typically used for controlling errors in other compressors [43,46], SZ2 allows for control via both global error bound, which guarantees that the lossy error in all cells do not exceed a single user-defined value, as well as the point-wise relative error bound [66] b_p, which guarantees that the lossy error in each cell does not exceed a user-defined percentage of the compressed value. Fig. 4 demonstrates the range of lossy data obtained via point-wise relative error control and a corresponding global relative error control, on a curve obtained from the maximum conditional progress variable $\max(C|Z)$. The use of point-wise error control is seen to ensure that values near zero are preserved, with positive values guaranteed to stay positive after lossy compression. Point-wise error control also preserves steep gradients more effectively than global error control between $Z = 0$ and $Z = 0.3$. Both these properties are important for preserving the fidelity of steep gradients and small values of chemical species seen commonly within combustion.

3.3. Evaluation metrics

For the results discussed in Section 4, we employ several statistical metrics of accuracy to evaluate the effects of lossy compression on data fidelity, and to assess the predictive accuracy of deep learning models. We quantify the quality of lossy compressed data through two popular image quality metrics, *i.e.*, the structural similarity index measure [67] (SSIM) and the peak-signal-to-noise-ratio [68] (PSNR). In this work, SSIM is evaluated by passing a sliding window $\boldsymbol{\Omega}$ ($\Delta_{\boldsymbol{\Omega}} = 6\Delta$) across a scalar from the uncompressed data ϕ and a scalar from the lossy compressed data ψ, and volume-averaging their statistical quantities:

$$\begin{aligned}\mathrm{SSIM}(\phi,\psi) &= \langle l(\phi,\psi)s(\phi,\psi)r(\phi,\psi)\rangle,\\ &= \left\langle \left(\frac{2\mu_\phi\mu_\psi + c_1}{\mu_\phi^2+\mu_\psi^2+c_1}\right)\left(\frac{2\sigma_\phi\sigma_\psi + c_1}{\sigma_\phi^2+\sigma_\psi^2+c_1}\right)\left(\frac{\sigma_{\phi\psi}+c_3}{\sigma_\phi\sigma_\psi+c_3}\right)\right\rangle,\end{aligned} \tag{4a}$$

where mean μ_ϕ and variance σ_ϕ^2 of the sliding window are:

$$\mu_\phi = \frac{1}{N_{\boldsymbol{\Omega}}}\int_{\boldsymbol{\Omega}} \phi \, d\boldsymbol{\Omega}, \tag{4b}$$

$$\sigma_\phi^2 = \frac{1}{N_{\boldsymbol{\Omega}}}\int_{\boldsymbol{\Omega}} (\phi-\mu_\phi)^2 \, d\boldsymbol{\Omega}, \tag{4c}$$

while l measures the similarity of $\mu_{\phi,\psi}$, s measures the similarity of $\sigma^2_{\phi,\psi}$, r measures correlation of the $\{\phi,\psi\}$, and constants $c_{\{1,2,3\}}$ ensure numerical stability.

PSNR is related to mean-squared-error (MSE):

$$\mathrm{PSNR}(\phi,\psi) = -10\log_{10}\left(\frac{\max(\phi,\psi)^2}{\mathrm{MSE}(\phi,\psi)}\right). \tag{5}$$

Note that for both metrics, higher values are indicative of higher post-compression quality, with SSIM bounded between −1 and 1, while the highest possible value for PSNR is restricted by the maximum value of a data type, *i.e.*, 48 dB for 8-bit images.

For the classification problem, we evaluate the class accuracy score via a one-versus-all approach, *i.e.*, the number of sample points that have been predicted correctly for a given class divided by the total number of sample points. In the regression problem, SSIM is employed to compare the similarity between filtered progress variable reaction rates from the DNS and from the deep learning models. The normalized mean-squared-error for N number of cells is also employed to measure the difference between the ground truth ϕ and model predictions ψ:

$$\text{Norm. MSE}(\phi,\psi) = \frac{\sum_{i=1}^{N}(\phi-\psi)^2}{\sum_{i=1}^{N}\phi^2}. \tag{6}$$

4. Results

4.1. Effects of lossy compression on data

This section describes the effects of lossy compression on the training data, while Section 4.2 discusses the effects of training deep learning models with this lossy data.

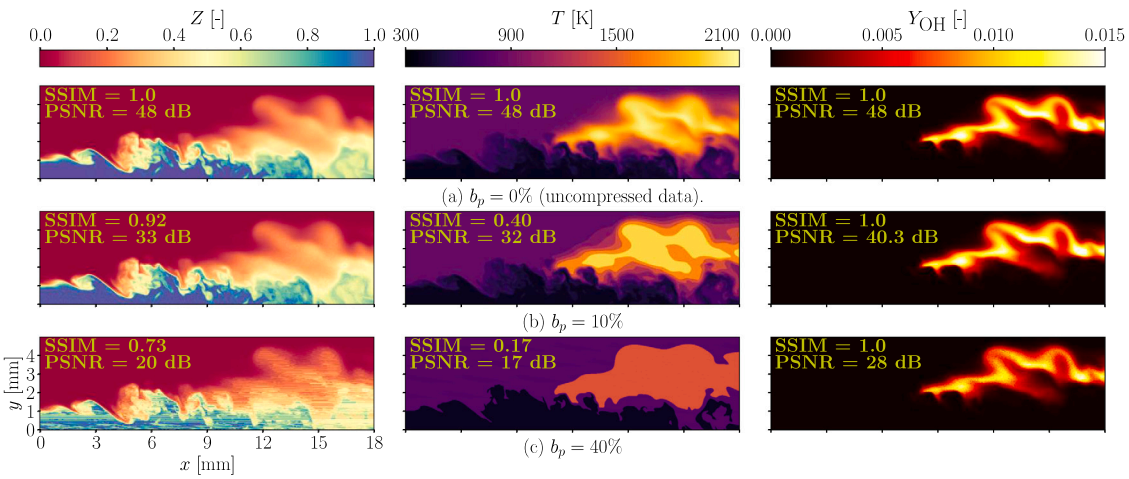


Fig. 6. Mixture fraction Z (left), temperature T (center), and OH mass fraction Y_{OH} (right) from the train set at different levels of maximum point-wise error bound b_p specified during compression. Quality metrics such as PSNR and SSIM are included in-panel.

We first compress flowfields required to solve both regression and classification problems with SZ2. Fig. 5 demonstrates that the total compression ratio, from 1% to 50% point-wise error bound b_p, ranges from 7- to 20-fold compression. Even if we consider only the lowest compression ratio seen in compressing the H_2O_2 mass fraction, a four-fold compression of the 124 GB DNS solution file, would enable at least three snapshots of this data to be shared as a single dataset on Kaggle. Data compression could be repeated on other flow configurations of a similar scale, and shared via the framework presented in Fig. 1 for building a diverse network-of-datasets. Fig. 5 also shows that greater compression ratios are seen in state variables, such as density ρ and temperature T, compared to mixture fraction Z and chemical species. We note that SZ2 has consistently resulted in large compression ratios across numerous scientific configurations in weather modeling, climate modeling, and cosmology modeling [39,66]. Thus, similarly large compression ratios should be expected for other potential configurations for BLASTNet.

Visualization of the uncompressed and lossy-compressed flowfields in Fig. 6 provides better insight into the different compression ratios exhibited by different quantities. In mixture fraction (left), temperature T (center), and OH mass fraction Y_{OH} (right), PSNR is shown to decrease with increasing point-wise error bound b_p when compared to the uncompressed flowfields in Fig. 6a. SSIM also decreases for Z and T with increasing b_p, but is preserved for Y_{OH}. At large settings of b_p, distortions first appear in regions with large magnitudes and small gradients, as is expected from the point-wise error control, as discussed with Fig. 4. As such, large field distortions are first clearly observable in temperature and mixture fraction Z at $b_p = 20\%$ and $b_p = 40\%$ in Figs. 6b and 6c, respectively, with no temperature fluctuations visible at $b_p = 40\%$ in Fig. 6c. In these cases, compression should be performed with smaller values of b_p to preserve flow structure. Note that for lossy compressed Y_{OH}, PSNR decreases with increasing b_p, while SSIM remains constant. This is due to the preservation of small magnitudes outside the small regions of the flame, which preserve the statistical quantities used to evaluate SSIM in Eq. (4). These results also demonstrate that the point-wise error bound is suited for preserving the large gradients and small magnitudes as seen in Z at $b_p = 10\%$ in Fig. 6b, and in all b_p for the OH mass fraction (Fig. 6a, b, c). This property is useful for preserving the flowfields of many scalar quantities encountered in reacting flow configurations.

4.1.1. Classification

When developing an ML dataset, one can either (i) identify and target a specific supervised learning problem or (ii) share raw data that can then be processed for a target ML problem just before training.

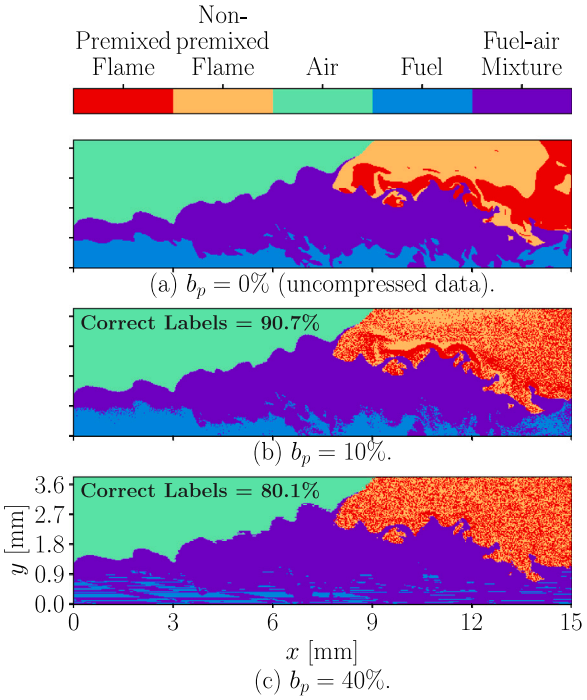


Fig. 7. Classification labels generated from lossy data at different levels of point-wise error bounds b_p.

Thus, in the present classification problem, we investigate two corresponding scenarios: (i) training with lossy features and clean labels, and (ii) training with lossy features and *post-processed labels* generated from lossy data.

Fig. 7 compares the labels used to train the deep learning models at different levels of point-wise error bound b_p, with Fig. 7a showing original uncompressed labels. Significant noise is seen at $b_p = 10\%$ (Fig. 7b), especially in the premixed and non-premixed flame regions, with a 9.3% total label error introduced to the data. This noise is present because lossy errors are magnified by the cell width Δ when evaluating scalar gradients used to determine the flame index (Eq. (2)). We demonstrate this on a central-differencing scheme:

$$f(X + \epsilon^{\ell}(X)) = \frac{(X_{i+1} + \epsilon^{\ell}_{i+1}) - (X_{i-1} + \epsilon^{\ell}_{i-1})}{2\Delta} \tag{7a}$$

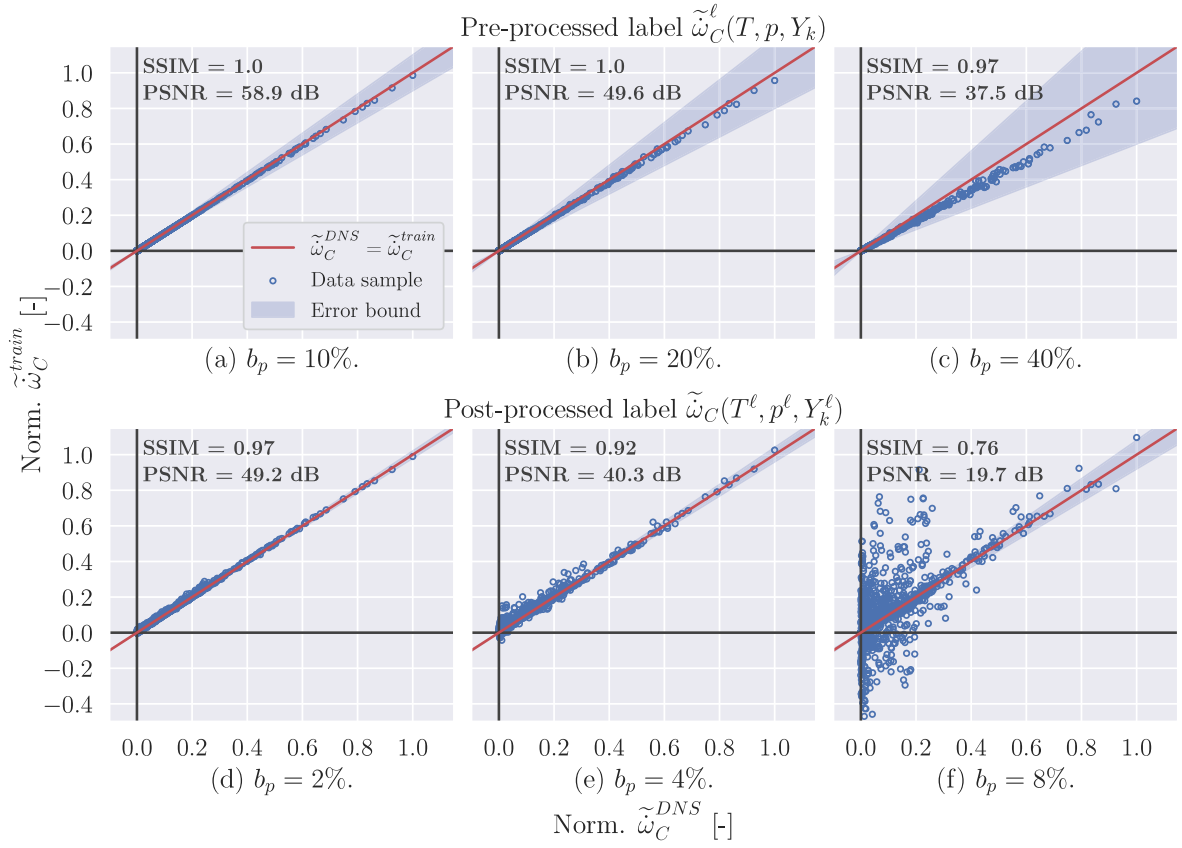


Fig. 8. Scatter plots of labels (filtered progress variable reaction rate $\widetilde{\omega}_C$) used for training $\widetilde{\omega}_C^{train}$ against true labels $\widetilde{\omega}_C^{DNS}$ for different cases of lossy labels. $\widetilde{\omega}_C$ is normalized by $\widetilde{\omega}_{C,max}^{DNS} = 10933\ \mathrm{s}^{-1}$.

$$= \frac{X_{i+1} - X_{i-1}}{2\Delta} + \frac{\epsilon_{i+1}^{\ell} - \epsilon_{i-1}^{\ell}}{2\Delta}. \tag{7b}$$

Note that in this text, we use the superscript $\cdot^{\ell}$ to denote lossy terms. In the worst case, where lossy errors $\epsilon_{i+1}^{\ell} = b_p X_{i+1}$ and $\epsilon_{i-1}^{\ell} = -b_p X_{i-1}$:

$$f(X + \epsilon^{\ell}(X)) = f(X) + b_p \frac{(X_{i+1} + X_{i-1})}{2\Delta}, \tag{7c}$$

which could be significantly larger than:

$$f(X) + \epsilon^{\ell}(f) = f(X) + b_p \frac{(X_{i+1} - X_{i-1})}{2\Delta}. \tag{7d}$$

Fig. 7b shows that the fuel labels also become distorted at $b_p = 40\%$ as the lossy errors obfuscate the threshold ($Z \leq 0.01$) in generating the labels, as discussed with Table 2, resulting in a total label error of nearly 20%.

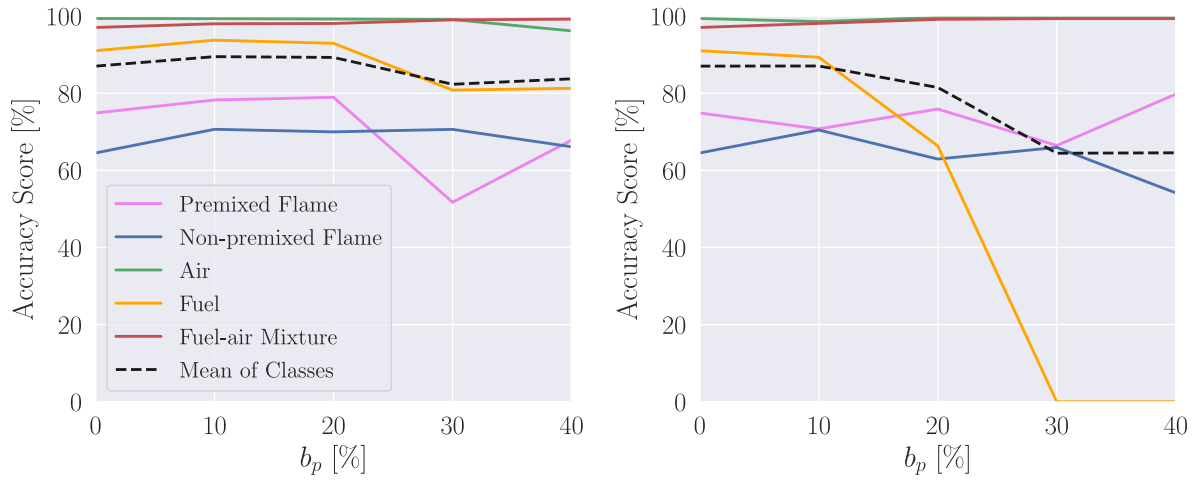
4.1.2. Regression

We consider the same two scenarios from Section 4.1.1: (i) targeting a specific supervised learning problem or (ii) generating labels from shared lossy simulation data. Specifically, we explore scenarios where (i) pre-processed filtered progress variable reaction rate $\widetilde{\omega}_C^{\ell}(T,p,Y_k)$ are compressed and shared, and where (ii) post-processed filtered progress variable reaction rate $\widetilde{\omega}_C(T^{\ell},p^{\ell},Y_k^{\ell})$ are generated directly from shared lossy data. The pre-processed label $\widetilde{\omega}_C^{\ell}$ is generated by (i) evaluating $\dot{\omega}_C^{DNS}$ through inputting the thermo-chemical vector $[T,p,Y_k]^{\mathrm{T}}$ from each cell into the chemical mechanism, (ii) applying Favre-filtering (Eq. (3)) to form $\widetilde{\omega}_C^{DNS}$, and (iii) applying lossy compression to form $\widetilde{\omega}_C^{l}$. In contrast, the post-processed label $\widetilde{\omega}_C(T^{\ell},p^{\ell},Y_k^{\ell})$ is generated by (i) applying lossy compression on thermo-chemical vector to form $[T^{\ell},p^{\ell},Y_k^{\ell}]^{\mathrm{T}}$, (ii) evaluating $\dot{\omega}_C(T^{\ell},p^{\ell},Y_k^{\ell})$ using the chemical mechanism, and (iii) applying Favre-filtering to form $\widetilde{\omega}_C(T^{\ell},p^{\ell},Y_k^{\ell})$.

Fig. 8 compares scatter plots of labels (filtered progress variable reaction rate $\widetilde{\omega}_C$ normalized by $\widetilde{\omega}_{C,max}^{DNS} = 10933\ \mathrm{s}^{-1}$) used for training $\widetilde{\omega}_C^{train}$ *vs.* true labels $\widetilde{\omega}_C^{DNS}$ for different cases of lossy labels. Fig. 8a, b, c shows that in cases with pre-processed labels $\widetilde{\omega}_C^{\ell}$, the lossy labels never exceeds the point-wise error bound $b_p = 10\%$, $b_p = 20\%$, and $b_p = 40\%$, respectively. In contrast, Fig. 8d, e, f shows that the post-processed labels $\widetilde{\omega}_C(T^{\ell},p^{\ell},Y_k^{\ell})$ can exceed $b_p = 2\%$, $b_p = 4\%$, and $b_p = 8\%$, respectively. These errors can even result in unphysical labels, as seen in Fig. 8e, f, where negative values of $\widetilde{\omega}_C$ are observed even at relatively small $b_p = 4\%$, and $b_p = 8\%$. This is because exponential operators in the Arrhenius term can magnify the lossy errors, which is also seen with gradient operators in Eq. (7c). These results highlight the importance of choosing a sufficiently low b_p during lossy compression, as significant errors could form in labels generated from lossy data. In addition, the large errors that can be seen in the post-processed quantities indicate that lossy data in BLASTNet should not be used for conventional DNS analysis.

4.2. Deep learning predictions

We now explore the effects of lossy data on deep learning. In general, validation and test data do not necessarily match the distribution of the training data, and are usually sampled to represent data encountered after deployment. For instance, when building a



(a) Trained on lossy features and clean labels. (b) Trained on lossy features and lossy labels.

Fig. 9. Class accuracy score at different levels of maximum point-wise error specified during compression.

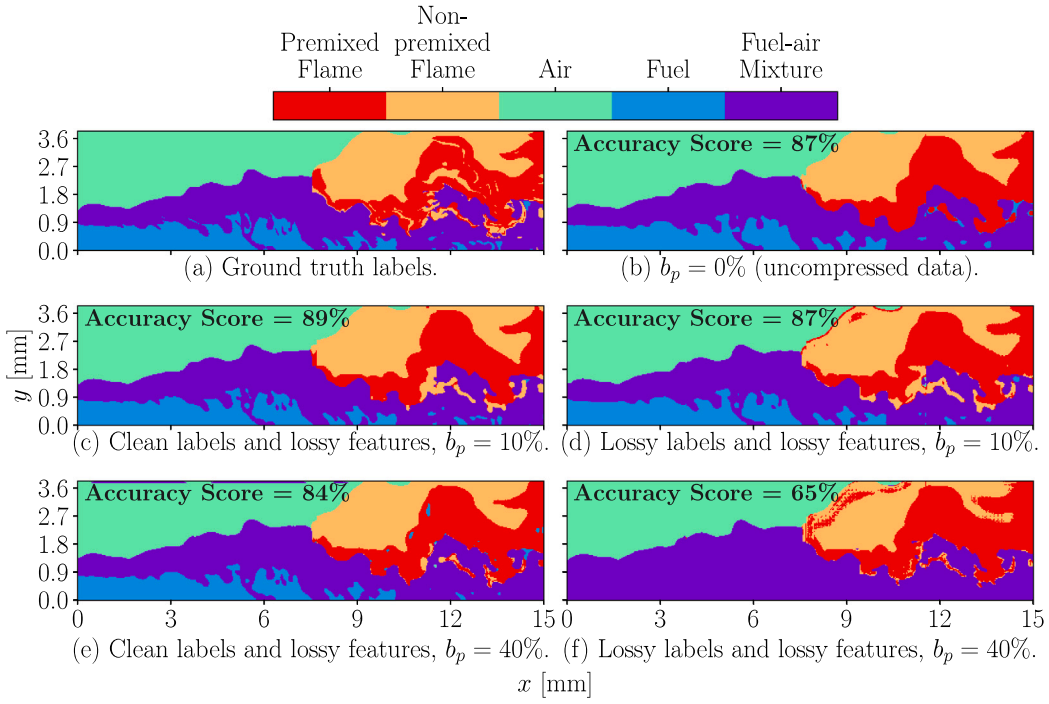
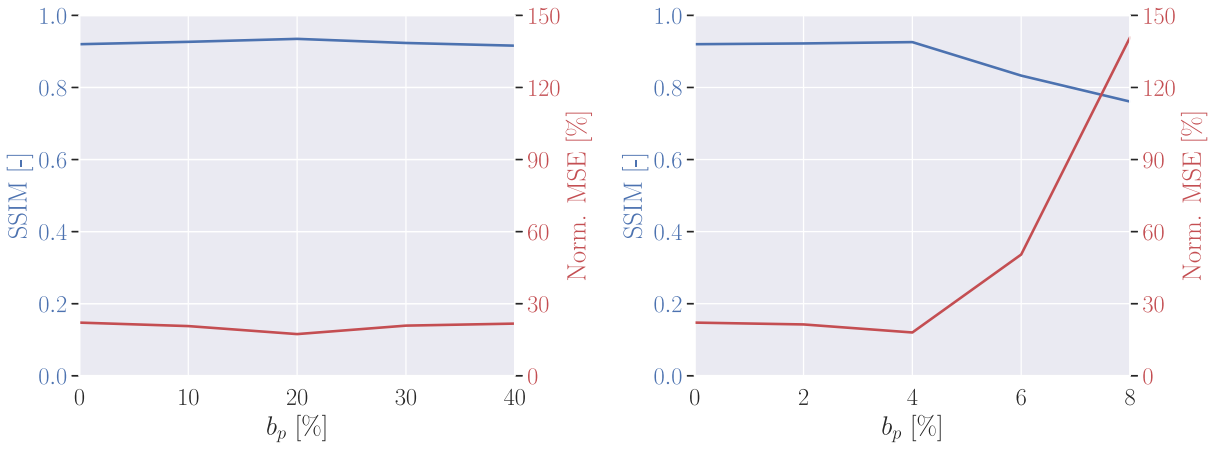


Fig. 10. Visualization of ground truth and predictions from ML models in the classification problem.

data-driven turbulence model in a numerical solver, training data can be extracted from as many different sources as possible to improve generalizability, while validation and test data should match the flow conditions simulated by in the numerical solver [1]. Thus, in the big data framework proposed in Fig. 1, we envision a scenario where large quantities of lossy compressed training data can be easily obtained from public repositories, with small quantities of clean test and validation data sampled personally by a user. As such, for the classification and regression problems in Section 4.2.1 and Section 4.2.2, only the training data are lossy-compressed, while validation and test sets are uncompressed.

4.2.1. Classification

Fig. 9(a) compares class accuracy scores for different levels of point-wise error bounds b_p, for ML models trained on lossy features and clean labels. A mean class accuracy score of 87% is seen in the baseline case of $b_p = 0\%$, which is typical in other classification/segmentation problems [54,69]. The mean accuracy scores are robust up to $b_p = 20\%$, corresponding to a 13-fold compression. At $b_p > 40\%$, a high mean accuracy (84%) is still observed, which is in agreement with the well-known observation [70] that ML algorithms are reasonably robust to feature noise. Note that non-monotonic behavior can be observed in the accuracy for the premixed flame class. This behavior is also



(a) Trained on pre-processed labels $\widetilde{\omega}_C^{\ell}$. (b) Trained on post-processed labels $\widetilde{\omega}_C(T^{\ell}, p^{\ell}, Y_k^{\ell})$.

Fig. 11. Structural similarity index measure (SSIM) and normalized mean-squared-error (MSE) at different values of point-wise error bounds b_p. Tested on clean features and clean labels.

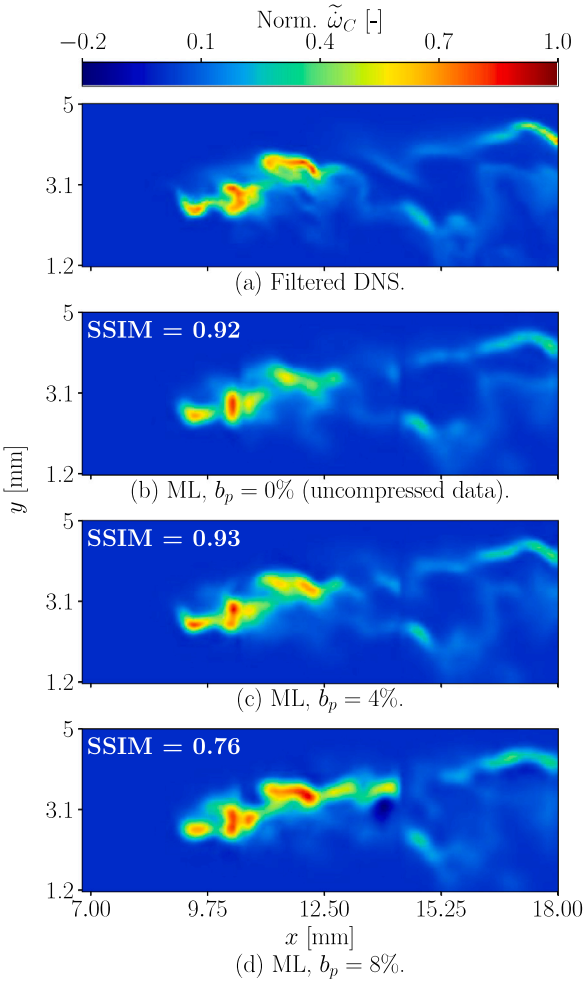
observed in another study [71] investigating the effects of increasing noise in labels for classification problems involving numerous image and sequential datasets. This behavior is expected in ML algorithms that rely on stochastic first-order gradient descent optimization when training deep learning algorithms.

Fig. 9(b) compares class accuracy scores for different b_p, when training with both lossy features and (post-processed) labels generated from lossy data. The mean accuracy scores are robust to errors up to only $b_p = 10\%$, which still corresponds to a 11-fold compression in the original data. At $b_p \geq 20\%$, class accuracy for fuel begins to decrease towards 0. This is caused by the distorted fuel labels shown in Fig. 7c. Nevertheless, the deep learning model demonstrates reasonably robust behavior in the other classes, especially in the flame regions, up until $b_p = 40\%$.

Fig. 10 visualizes predictions from the deep learning model trained on lossy features and post-processed labels. Fig. 10b shows that the model predictions at $b_p = 0\%$ are in reasonable agreement with the ground truth labels in Fig. 10a. A slightly higher class accuracy of 89% is seen when training with clean labels and lossy features at $b_p = 10\%$, as shown in Fig. 10c. Increase in accuracy is commonly observed in ML models with the introduction of small amounts of noise during data-augmentation [27], which is well-known to improve neural network models [26]. However, Fig. 10d shows that non-premixed flame samples are misclassified as premixed flame near the flame boundary with air when the ML model is trained with lossy labels and lossy features at $b_p = 10\%$. This is likely caused by the excessive label noise between the premixed and non-premixed flame regions, as seen in Fig. 7b. Similarly, misclassification is seen in the air and premixed flame labels in Fig. 10e, with feature noise at $b_p = 40\%$. The aforementioned failure in classifying fuel is clearly observed when the ML model is trained with lossy labels and lossy features at $b_p = 40\%$ (Fig. 10f). Nevertheless, coherent classification is still observed in the flame regions at $b_p = 40\%$, despite the high label noise seen in Fig. 7c.

4.2.2. Regression

Fig. 11(a) compares regression accuracy and error metrics, namely SSIM and the normalized MSE, respectively, for different levels of point-wise error bounds b_p, for ML models trained on lossy features and lossy pre-processed labels $\widetilde{\omega}_C^{\ell}$. For $b_p = 0\%$, a normalized MSE of 22% is similar to results from another study [62]. These values and the high SSIM ≈ 0.92 are reasonably consistent up to $b_p = 40\%$, corresponding to a 20-fold compression.



(a) Filtered DNS. (b) ML, $b_p = 0\%$ (uncompressed data). (c) ML, $b_p = 4\%$. (d) ML, $b_p = 8\%$.

Fig. 12. Visualization of filtered DNS and predictions from model trained on lossy features and post-processed labels $\widetilde{\omega}_C(T^{\ell}, p^{\ell}, Y_k^{\ell})$, and tested on clean features and clean labels. $\widetilde{\omega}_C$ is normalized by $\widetilde{\omega}_{C,max}^{DNS} = 10933\ \mathrm{s}^{-1}$.

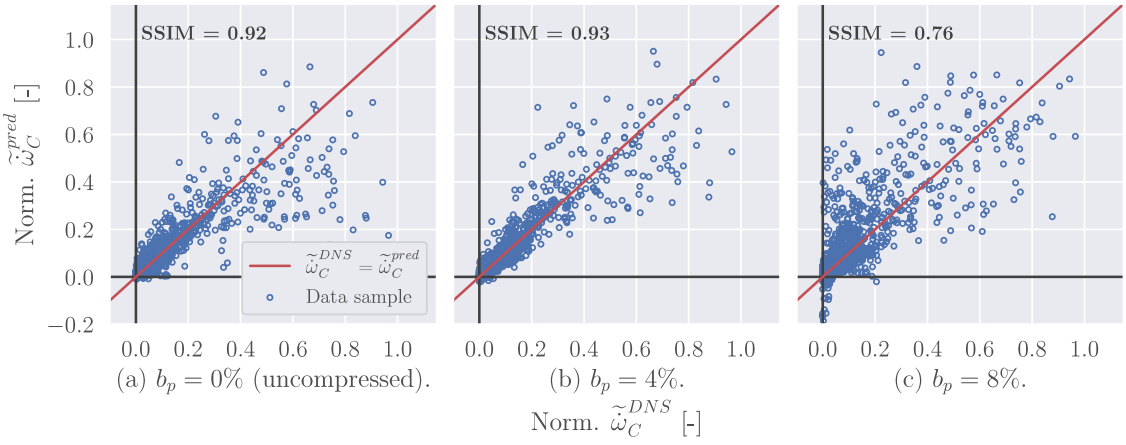


Fig. 13. Scatter plots of predictions of filtered progress variable reaction rate $\widetilde{\omega}_C^{pred}$ (normalized by $\widetilde{\omega}_{C,max}^{DNS} = 10933\ s^{-1}$) from the ML model (trained with post-processed labels) *vs.* true labels $\widetilde{\omega}_C^{DNS}$ for different cases of lossy labels.

Fig. 11(b) compares SSIM and normalized MSE when training with both lossy features and post-processed labels $\widetilde{\omega}_C(T^\ell, p^\ell, Y_k^\ell)$ generated from lossy data, as a function of b_p. SSIM and normalized MSE are robust to errors up to only $b_p = 4\%$, which still corresponds to a 10-fold compression. After $b_p \geq 4\%$, SSIM begins to decrease while normalized MSE increases significantly due to the magnification of errors during label generation, as shown in Fig. 8.

Fig. 12 visualizes the predictions (normalized by $\widetilde{\omega}_{C,max}^{DNS} = 10933\ s^{-1}$) from the ML model trained on lossy features and post-processed labels $\widetilde{\omega}_C(T^\ell, p^\ell, Y_k^\ell)$, along with the filtered DNS. Fig. 12b, c shows that the model predictions at $b_p = 0\%$ and $b_p = 4\%$ are in reasonable agreement with the ground truth labels in Fig. 12a. Over-prediction and under-prediction of $\widetilde{\omega}_C$ is observed in Fig. 12d, where $b_p = 8\%$.

Fig. 13 compares normalized predictions of filtered progress variable reaction rate $\widetilde{\omega}_C^{pred}$ from the ML model (trained with post-processed labels) *vs.* normalized true labels $\widetilde{\omega}_C^{DNS}$ for different cases of lossy labels. The distribution of scatter points shown in Fig. 13a for $b_p = 0\%$ (uncompressed data) is qualitatively similar to distributions from another *a priori* ML study [62] involving $\widetilde{\omega}_C$. The high SSIM = 0.92 indicates good correlation between the ML predictions and the true labels. Fig. 13b shows a narrower distribution of scatter points and higher SSIM = 0.93 for $b_p = 4\%$. Here, small amounts of noise in the training data is seen to improve the regression model, which is also observed in Fig. 10c for the classification problem. This behavior is also observed in a recent regression study involving turbulent flows [72]. However, Fig. 13c shows that for $b_p = 8\%$, significant errors (such as the negative predictions for $\widetilde{\omega}_C^{pred}$) are observed, which also lead to a large decrease in SSIM = 0.76.

Fig. 14 compares mean conditional filtered progress variable reaction rate $\langle\widetilde{\omega}_C|\widetilde{Z}\rangle$ (normalized by $\langle\widetilde{\omega}_C|\widetilde{Z}\rangle_{max} = 1122\ s^{-1}$) from the ML model (trained with post-processed labels) with ground truth labels from the filtered DNS. The misprediction observed at $b_p = 8\%$ in Fig. 8d can also be observed here, where a two-fold over-prediction in $\langle\widetilde{\omega}_C|\widetilde{Z}\rangle$ occurs at $\widetilde{Z} = 0.24$. $\langle\widetilde{\omega}_C|\widetilde{Z}\rangle$ at $b_p = 4\%$ is seen to be in better agreement with the filtered DNS than at $b_p = 2\%$ and $b_p = 0\%$. Introducing small amounts of noise is also seen to improve the classification model, as discussed with Fig. 13b and Fig. 10c.

We note that while the addition of noise can improve training (as commonly done via data augmentation [27]), an excessive amount of noise can lead to poor predictions as shown by $b_p = 8\%$ (Fig. 12c and Fig. 14), and thus caution should be exercised when dealing with noisy data. Hence, for the purposes of compressing excessively large data files for BLASTNet, we recommend a soft-constraint of $b_p = 1\%$ (seven-fold compression), unless necessary to achieve higher compression ratio in very large datasets. Any further augmentation with noise should be performed after downloading and during training at a user's discretion. Nevertheless, these results demonstrate that controlled amounts of noise does not affect deep learning models, and in some cases can even be beneficial.

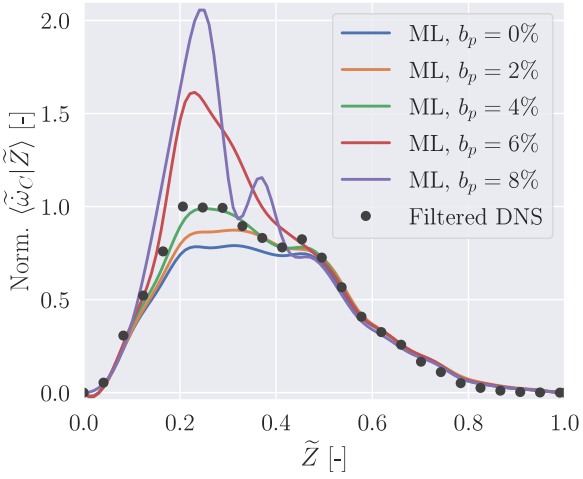


Fig. 14. Comparison of mean conditional filtered progress variable $\langle\widetilde{\omega}_C|\widetilde{Z}\rangle$ from the filtered DNS and predictions from the ML model, trained with post-processed labels $\widetilde{\omega}_C(T^\ell, p^\ell, Y_k^\ell)$. $\langle\widetilde{\omega}_C|\widetilde{Z}\rangle$ is normalized by $\langle\widetilde{\omega}_C|\widetilde{Z}\rangle_{max} = 1122\ s^{-1}$.

5. Conclusions

In this paper, we propose BLASTNet, a realistic framework that combines (i) community involvement, (ii) public data repositories, and (iii) lossy compression algorithms for accessing the wealth of combustion data that already exists in the form of high-fidelity simulations and detailed measurements. Alongside this, we introduce a web-platform, at https://blastnet.github.io/, for consolidating the proposed network-of-datasets.

Given the potential limitations in public storage capacity, a key component of this framework involves the use of lossy compression algorithms for enabling access to petascale simulation data and large

experimental measurements. Thus, we evaluate effects of lossy compression algorithms on data quality and deep learning performance on a H_2-air lifted flame DNS. To this end, we train CNN models with labels and features, extracted from lossy DNS data in two completely different regression and classification problems.

In scientific supervised learning, two broad categories of datasets can be encountered: (i) a dataset targeted at a specific problem, i.e., with clean/pre-processed labels, and (ii) raw simulation data or measurements, i.e., with lossy/post-processed labels. For the classification problem, we trained ML models to predict 5 different classes. In the clean label scenario, the classification model is robust to lossy errors in the features up to a point-wise error bound of $b_p = 20\%$, which corresponds to a 13-fold compression. In the case of lossy labels and features, the CNN is robust up to $b_p = 10\%$, corresponding to a 11-fold compression ratio.

For the regression problem, we trained the ML models to predict the Favre-filtered progress variable reaction rate $\widetilde{\dot{\omega}}_c$. In the pre-processed scenario, the performance of the regression model is unhindered even at $b_p = 40\%$. Due to the magnification of the lossy errors by Arrhenius term calculations, large lossy errors are seen in the post-processed training labels even at $b_p = 4\%$. Nevertheless, the regression model still predicts $\widetilde{\dot{\omega}}_c$ accurately, and the presence of small amounts of noise is even seen to improve the performance of the deep learning model. However, model predictive accuracy drops sharply at $b_p > 4\%$. In both regression and classification problems, our results demonstrate that deep learning models applied to combustion can be robust to small amounts of noise.

We now summarize the findings from all sections of this paper towards recommendations for standards in BLASTNet. Based on the requirements for a useful training dataset listed in Section 1.2, we envision DNS and LES data, covering $\mathcal{O}(10^2)$ different configurations with a total of $\mathcal{O}(10^3)$ different snapshots for the first iteration of BLASTNet, with later versions considering experimental data. Since this work demonstrates that deep learning models can train on labels that are post-processed from lossy data, the flowfield in these datasets should at least contain $[\rho, \boldsymbol{u}, T, p, Y_k]^{\mathrm{T}}$, with additional information required for evaluating thermodynamic and transport properties provided to BLASTNet as metadata (in standardized Chemkin or Cantera formats), so users can recreate any labels required for training in the wide range of supervised learning problems that are of interest to combustion. For the choice of the public repository for BLASTNet, we recommend the use of Kaggle [23], due to the platform's command-line interface that can enable data access from computing clusters, and ability to provide each data contribution a unique digital-object-identifier (DOI),

We recommend the use of a consistent lossy compressor (SZ2 [44]) to allow for a consistent data format that would expedite the construction of a data pipeline during training. Since caution should be exercised as the performance of deep learning algorithms are seen here to degrade rapidly in the presence of excessive noise, lossy compression should only be applied when necessary, which is largely applicable to bigger DNS cases that exceed 100 GB per snapshot. While the results from this study demonstrate the merit of introducing small amounts of noise to the data prior to training, noise addition should be performed under the discretion of the ML practitioner as a data augmentation procedure [27]. In these cases, we recommend a soft-constraint of approximately $b_p = 1\%$ with SZ2 so that a few snapshots from a petascale simulation can be stored onto Kaggle. Since the total compression ratio observed in this study (7-fold compression) is limited by compression of the chemical species, we expect that a 5 to 10 compression ratio would also be observed in other flame configurations since the volumetric ratio of reacting to non-reacting gases should be relatively similar across different simulation configurations. To ensure that data fidelity is preserved after lossy compression, we propose that lossy datasets should be tested with the classification and regression tasks presented in this paper with ML prediction results compared with an uncompressed counterpart prior to sharing.

To help facilitate these standards and guidelines, tutorials on Kaggle, SZ2, and reading/writing with the recommended data format are provided in BLASTNet. BLASTNet also curates information (boundary conditions, initial conditions, fuel composition, chemical mechanism, DOI) regarding individual simulation configurations, and provides a centralized search interface that enables users to download individual cases, along with scripts that enable batch access to all shared data. In this web-platform, a BLASTNet discussion forum is also hosted in order to receive community feedback and to provide user support. We remind the readers that each BLASTNet contributor will be included to the list-of-authors in order to cultivate a truly community-involved big training database for combustion. Thus, we call on the combustion community to contribute to this bearable large accessible scientific training network-of-datasets.

Supplementary material

The web-platform for consolidating BLASTNet [47] can be found in https://blastnet.github.io/, which also provides standards for contributing data and tutorials on reading and accessing shared data. The code and ML models used for this study can be found in https://github.com/IhmeGroup/lossy_ml.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All data and ML models are made available at https://blastnet.github.io and https://github.com/IhmeGroup/lossy_ml

Acknowledgments

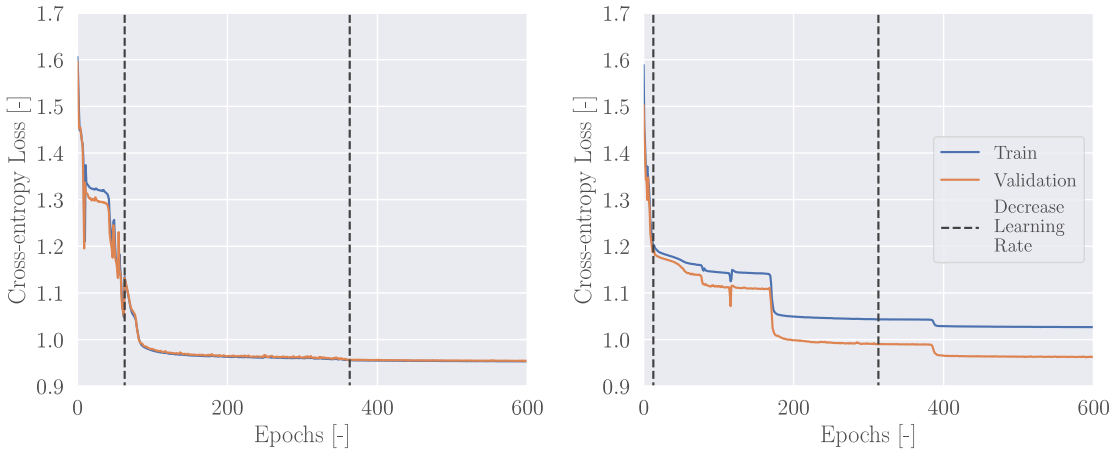
The authors acknowledge financial support and computing resources from the Department of Energy (DoE), USA, under award DE-NA0003968. We are also thankful for funding support from the DoE Office of Basic Energy Sciences under award DE-SC0022222. The work at Sandia National Laboratories was supported by the DoE, Office of Basic Energy Sciences, Division of Chemical Sciences, Geosciences, and Biosciences. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for DoE, USA National Nuclear Security Administration under contract DE-NA0003525. We are grateful for fruitful discussions with the Kaggle team, which have greatly aided the progress of this work.

Appendix A. Training and validation

In both classification and regression problems, training is performed with the Adam [73] optimizer. In the classification problem, we employ raw learning rates of 1E-4, 1E-5, and 1E-6 for 100, 300, and 300 epochs, respectively, and early-stopping is employed when necessary. Prior to training, the raw learning rates are multiplied by the square root of the batch size. Here, the batch size is 24.

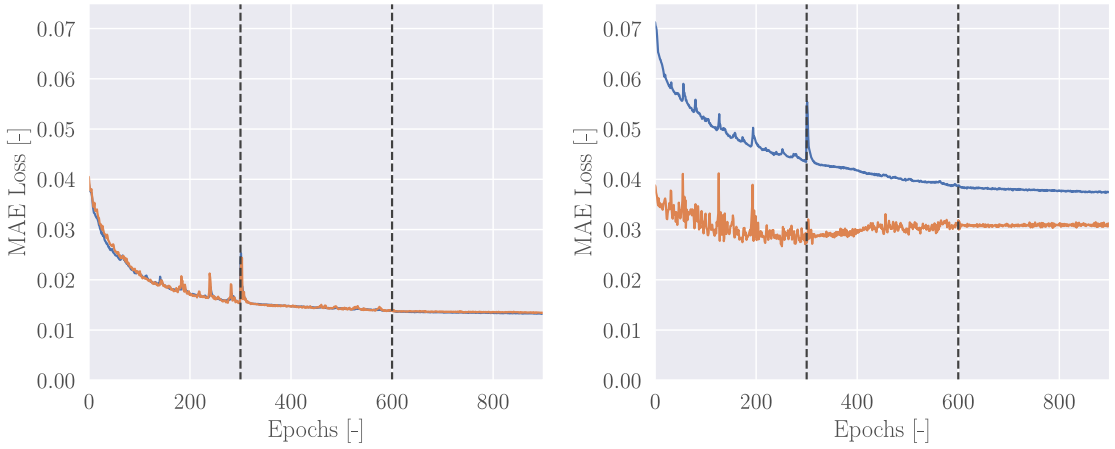
In the regression problem, we employ raw learning rates of 1E-4, 5E-5, and 1E-5, for 300 epochs each, with batch size of 36. Training both regression and classification models on four Tesla V100 GPUs requires a total of approximately 4 h of wall-clock-time for each case.

Training and validation losses for selected cases are shown in Fig. A.15. In Figs. A.15(b) and A.15(d), the converged validation loss can be lower than the training loss, leading to higher validation accuracy than training accuracy. This is caused by the absence of lossy errors in the validation set, as described in Section 4. Otherwise, training shows no sign of overfitting in Figs. A.15(a) and A.15(c)



(a) Classification: Lossy and clean labels, $b_p = 20\%$.

(b) Classification: Lossy features and lossy labels, $b_p = 10\%$.



(c) Regression: Lossy features and lossy pre-processed labels, $b_p = 10\%$.

(d) Regression: Lossy features and post-processed labels, $b_p = 8\%$

Fig. A.15. Loss during training.

Appendix B. Metadata

Metadata, containing additional information of simulation configurations, is stored as a JSON file and can be generated with the Python commands shown in Listings 1 and 2, which are for global and local metadata, respectively. Global metadata contains information on the flow configuration (initial conditions, chemistry, numerics, *etc.*), while the local metadata contains information on a specific snapshot from the configuration.

Appendix C. FAIR principles

The datasets contributed to BLASTNet have and will adhere to the FAIR principles [19] for scientific data management, with the specific details as follows:

- **Findable:** Each dataset has been assigned a unique and persistent Kaggle identifier. Each dataset is defined by rich metadata for its specific configuration and snapshots as shown in B. Metadata explicitly includes the Kaggle identifier. Both data and metadata are indexed and can be easily searched via both Kaggle and BLASTNet platforms.
- **Accessible:** Both data and metadata are retrievable via standardized Kaggle API. The protocol is free and available at https://github.com/Kaggle/kaggle-api. The protocol requires authentication and authorization via a Kaggle account. Metadata will remain hosted on https://blastnet.github.io/ even when the data is no longer available.
- **Interoperable:** The metadata uses a formal, accessible, shared, and broadly applicable for representation in the JSON files. In addition, vocabulary used follows FAIR principles. Further, metadata are referenced to other metadata through DOI from source publications associated with the data.
- **Reusable:** The metadata contains richly described information on the flow configuration (initial conditions, chemistry, numerics, *etc.*). In addition, all Kaggle submissions default to a CC BY-SA 4.0 license. The metadata contains information on the source publication associated with the data. All data and metadata are presented in consistent little-endian single-precision binaries and JSON files, respectively.

```
metadata['global'] = {
         "dataset_id": "waitongchung/inert-ch4o2-hit-dns",
              "Nxyz": [129,129,129],
          "snapshots": 98,
          "variables": ["UX_ms-1","UY_ms-1","UZ_ms-1",
                    "P_Pa","T_K","RHO_kgm-3",
                    "YO2","YCH4"],
       "compression": "None",
              "grid": {"x": "./grid/X_m.dat",
                    "y": "./grid/Y_m.dat",
                    "z": "./grid/Z_m.dat"},
        "numerics": {"spatial": "4th order central-differencing
                        with 2nd order ENO",
                "temporal": "3rd-order SSP-RK3 (non-stiff)
                         and semi-implicit ROWPLUS (stiff)",
                 "solver": "CharlesX"},
                 "bc": "Periodic in x-, y-, and z-directions.",
                 "ic": {"U": "HIT Von Karman Pao with Re_t = 80 and
                        integral lengthscale of 62.5E-6m",
                  "T [K]": 300,
                "P [Pa]": 101325,
               "Mixture": "CH4-O2 inert branch from 1D
                           cantera counterflow calculations."},
               "doi": "https://doi.org/10.1016/j.combustflame.2021.111758",
      "contributors": "Wai Tong Chung and Matthias Ihme",
      "description": "Compressible Inert CH4-O2 Homogeneous
                 Isotropic Turbulence DNS",
  "chem_thermo_tran": {"description": "FRC and Mixture-Averaged Transport
                                       with constant lewis number",
                       "cantera_xml": "./chem_thermo_tran/bfer.xml"}
}
```

Listing 1: Python command for generating global metadata for a BLASTNet contribution.

```
metadata['local'] = [
                  {"id": 0,
              "time [s]": 6.88389e-06,
    "UX_ms-1 filename": "./data/UX_ms-1_id000.dat",
    "UY_ms-1 filename": "./data/UY_ms-1_id000.dat",
    "UZ_ms-1 filename": "./data/UZ_ms-1_id000.dat",
    "  P_Pa filename": "./data/P_Pa_id000.dat",
      "T_K filename": "./data/T_K_id000.dat",
 "RHO_kgm-3 filename": "./data/RHO_kgm-3_id000.dat",
      "YO2 filename": "./data/YO2_id000.dat",
     "YCH4 filename": "./data/YCH4_id000.dat"},
                  {"id": 1, ...},
                       ...,
                  {"id": 97, ...}
]
```

Listing 2: Python command for generating local metadata for a BLASTNet contribution.

References

[1] Ihme M, Chung WT, Mishra AA. Combustion machine learning: Principles, progress and prospects. Prog Energy Combust Sci 2022;91:101010.

[2] Sun C, Shrivastava A, Singh S, Gupta AK. Revisiting unreasonable effectiveness of data in deep learning era. In: Proc IEEE Int Conf Comput Vis. 2017, p. 843–52.

[3] Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 2009;248–55.

[4] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet large scale visual recognition challenge. Int J Comput Vis 2015;115(3):211–52.

[5] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proc IEEE Conf Comput Vis Pattern Recognit 2016;770–8.

[6] Li Y, Perlman E, Wan M, Yang Y, Meneveau C, Burns R, et al. A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence. J Turbul 2008;9:No. 31.

[7] Agustsson E, Timofte R. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In: IEEE Conf Comput Vis Pattern Recognit Workshop. 2017.

[8] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nat Rev Phys 2021;3:422–40.

[9] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proc Int Conf Mach Learn 2015;37:448–56.

[10] Nair V, Hinton G. Rectified linear units improve restricted Boltzmann machines. Proc Int Conf Mach Learn 2010;27:807–14.

[11] Yuan S, Zhao H, Zhao S, Leng J, Liang Y, Wang X, et al. A roadmap for big model. 2022, arXiv Pre-Print 2203.14101.

[12] Bode M, Gauding M, Lian Z, Denker D, Davidovic M, Kleinheinz K, et al. Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows. Proc Combust Inst 2021;38:2617–25.

[13] Bommasani R, Hudson DA, Adeli E, Altman R, Arora S, von Arx S, et al. On the opportunities and risks of foundation models. 2021, arXiv Pre-Print 2108.07258.

[14] Thrun S. Lifelong learning algorithms. In: Thrun S, Pratt L, editors. Learning to learn. Boston, MA: Springer US; 1998, p. 181–209.

[15] Chen JH. Petascale direct numerical simulation of turbulent combustion—fundamental insights towards predictive models. Proc Combust Inst 2011;33(1):99–123.

[16] Treichler S, Bauer M, Bhagatwala A, Borghesi G, Sankaran R, Kolla H, et al. S3D-Legion: An exascale software for direct numerical simulation of turbulent combustion with complex multicomponent chemistry. In: Straatsma TP, Antypas KB, Williams TJ, editors. Exascale scientific applications. New York, NY: Chapman and Hall/CRC; 2017, p. 257–78.

[17] Frank JH. Advances in imaging of chemically reacting flows. J Chem Phys 2021;154(4):040901.

[18] Li J, Zhao Z, Kazakov A, Dryer FL. An updated comprehensive kinetic model of hydrogen combustion. Int J Chem Kinet 2004;36(10):566–75.

[19] Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, et al. The FAIR guiding principles for scientific data management and stewardship. Sci Data 2016;3:160018.

[20] Wang X, Yu K, Wu S, Gu J, Liu Y, Dong C, et al. ESRGAN: Enhanced super-resolution generative adversarial networks. In: Proc Euro Conf Comput Vis. 2018, p. 63–79.

[21] Foster I. Globus online: Accelerating and democratizing science through cloud-based services. IEEE Internet Comput 2011;15(3):70–3.

[22] Blanton MR, Bershady MA, Abolfathi B, Albareti FD, Prieto CA, Almeida A, et al. Sloan digital sky survey IV: Mapping the milky way, nearby galaxies, and the distant universe. Astron J 2017;154(1):28–62.

[23] Goldbloom A, Hamner B. Kaggle: Your machine learning and data science community. 2010, https://www.kaggle.com.

[24] Northcutt CG, Athalye A, Mueller J. Pervasive label errors in test sets destabilize machine learning benchmarks. In: Proc Neural Inf Process Syst Track Datasets Benchmarks, vol. 1. 2021.

[25] Mahajan D, Girshick R, Ramanathan V, He K, Paluri M, Li Y, et al. Exploring the limits of weakly supervised pretraining. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, editors. Proc Euro Conf Comput Vis. 2018, p. 185–201.

[26] Bishop CM. Training with noise is equivalent to Tikhonov regularization. Neural Comput 1995;7(1):108–16.

[27] Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. J Big Data 2019;6(1):60.

[28] Jolliffe IT, Cadima J. Principal component analysis: A review and recent developments. Phil Trans R Soc A 2016;374(2065):20150202.

[29] Malik MR, Obando Vega P, Coussement A, Parente A. Combustion modeling using principal component analysis: A posteriori validation on Sandia flames D, E and F. Proc Combust Inst 2021;38:2635–43.

[30] Gitushi KM, Ranade R, Echekki T. Investigation of deep learning methods for efficient high-fidelity simulations in turbulent combustion. Combust Flame 2022;236:111814.

[31] Burke SP, Schumann TEW. Diffusion flames. Ind Eng Chem 1928;20(10):998–1004.

[32] Gicquel O, Darabiha N, Thévenin D. Laminar premixed hydrogen/air counterflow flame simulations using flame prolongation of ILDM with differential diffusion. Proc Combust Inst 2000;28(2):1901–8.

[33] van Oijen J, de Goey L. Modelling of premixed laminar flames using flamelet-generated manifolds. Combust Sci Technol 2000;161(1):113–37.

[34] Pierce CD, Moin P. Progress-variable approach for large-eddy simulation of non-premixed turbulent combustion. J Fluid Mech 2004;504:73–97.

[35] Ihme M, Cha CM, Pitsch H. Prediction of local extinction and re-ignition effects in non-premixed turbulent combustion using a flamelet/progress variable approach. Proc Combust Inst 2005;30:793–800.

[36] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science 2006;313(5786):504–7.

[37] Glaws A, King R, Sprague M. Deep learning for in situ data compression of large turbulent flow simulations. Phys Rev Fluids 2020;5(11):114602.

[38] Lu Y, Jiang K, Levine JA, Berger M. Compressive neural representations of volumetric scalar fields. Comput Graph Forum 2021;40(3):135–46.

[39] Liu T, Wang J, Liu Q, Alibhai S, Lu T, He X. High-ratio lossy compression: Exploring the autoencoder to compress scientific data. IEEE Trans Big Data 2021. [in press].

[40] Meister D, Kaiser J, Brinkmann A, Cortes T, Kuhn M, Kunkel J. A study on data deduplication in HPC storage systems. Proc Int Conf High Perform Comput Netw Storage Anal 2012;7:1–11.

[41] Gailly J-L, Adler M. GNU gzip. 1992, https://www.gnu.org/software/gzip/.

[42] Burtscher M, Ratanaworabhan P. FPC: A high-speed compressor for double-precision floating-point data. IEEE Trans Comput 2009;58(1):18–31.

[43] Lakshminarasimhan S, Shah N, Ethier S, Ku S-H, Chang CS, Klasky S, et al. ISABELA for effective in situ compression of scientific data. Concurr Comput 2013;25(4):524–40.

[44] Liang X, Di S, Tao D, Li S, Li S, Guo H, et al. Error-controlled lossy compression optimized for high compression ratios of scientific datasets. In: Proc IEEE Int Conf Big Data. 2018, p. 438–47.

[45] Lindstrom P. Fixed-rate compressed floating-point arrays. IEEE Trans Vis Comput Graphics 2014;20(12):2674–83.

[46] Ballester-Ripoll R, Lindstrom P, Pajarola R. TTHRESH: Tensor compression for multidimensional visual data. IEEE Trans Vis Comput Graphics 2019;26:2891–903.

[47] Chung WT, Ihme M, Jung KS, Chen JH, Guo J, Brouzet D, et al. BLASTNet simulation dataset. 2022, https://blastnet.github.io/.

[48] Goodwin DG, Moffat HK, Schoegl I, Speth RL, Weber BW. Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes. 2022, https://www.cantera.org.

[49] Jung KS, Kim SO, Lu T, Chen JH, Yoo CS. On the flame stabilization of turbulent lifted hydrogen jet flames in heated coflows near the autoignition limit: A comparative DNS study. Combust Flame 2021;233:111584.

[50] Yoo CS, Wang Y, Trouvé A, Im HG. Characteristic boundary conditions for direct simulations of turbulent counterflow flames. Combust Theor Model 2005;9(4):617–46.

[51] Yoo CS, Im HG. Characteristic boundary conditions for simulations of compressible reacting flows with multi-dimensional, viscous and reaction effects. Combust Theor Model 2007;11(2):259–86.

[52] Chen JH, Choudhary A, de Supinski B, DeVries M, Hawkes ER, Klasky S, et al. Terascale direct numerical simulations of turbulent combustion using S3D. Comput Sci Discov 2009;2(1):015001.

[53] Johnson JM, Khoshgoftaar TM. Survey on deep learning with class imbalance. J Big Data 2019;6:27.

[54] Chung WT, Mishra AA, Perakis N, Ihme M. Data-assisted combustion simulations with dynamic submodel assignment using random forests. Combust Flame 2021;227:172–85.

[55] Cellier A, Lapeyre C, Öztarlik G, Poinsot T, Schuller T, Selle L. Detection of precursors of combustion instability using convolutional recurrent neural networks. Combust Flame 2021;233:111558.

[56] Wan K, Hartl S, Vervisch L, Domingo P, Barlow RS, Hasse C. Combustion regime identification from machine learning trained by Raman/Rayleigh line measurements. Combust Flame 2020;219:268–74.

[57] Bilger R. Turbulent jet diffusion flames. Prog Energy Combust Sci 1976;1(2):87–109.

[58] Yamashita H, Shimada M, Takeno T. A numerical study on flame stability at the transition point of jet diffusion flames. Proc Combust Inst 1996;26(1):27–34.

[59] Chung WT, Mishra AA, Ihme M. Interpretable data-driven methods for subgrid-scale closure in LES for transcritical LOX/GCH4 combustion. Combust Flame 2022;239:111758.

[60] Nakazawa R, Minamoto Y, Inoue N, Tanahashi M. Species reaction rate modelling based on physics-guided machine learning. Combust Flame 2022;235:111696.

[61] Nikolaou ZM, Chrysostomou C, Vervisch L, Cant S. Progress variable variance and filtered rate modelling using convolutional neural networks and flamelet methods. Flow Turbul Combust 2019;103(2):485–501.

[62] Yellapantula S, de Frahan MTH, King R, Day M, Grout R. Machine learning of combustion LES models from reacting direct numerical simulation. In: Pitsch H, Attili A, editors. Data analysis for direct numerical simulations of turbulent combustion: from equation-based analysis to machine learning. Cham, Switzerland: Springer International Publishing; 2020, p. 273–92.

[63] Glaws A, King R, Sprague M. Deep learning for in situ data compression of large turbulent flow simulations. Phys Rev Fluid 2020;5:114602.

[64] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proc Int Conf Learn Represent. 2015.

[65] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proc Int Conf Artif Intell Stat. 2010, p. 249–56.

[66] Liang X, Di S, Tao D, Chen Z, Cappello F. An efficient transformation scheme for lossy data compression with point-wise relative error bound. In: Proc IEEE Int Conf Clust Comput. 2018, p. 179–89.

[67] Wang Z, Bovik A, Sheikh H, Simoncelli E. Image quality assessment: From error visibility to structural similarity. IEEE Trans Image Process 2004;13(4):600–12.

[68] Horé A, Ziou D. Image quality metrics: PSNR vs. SSIM. In: Proc IEEE Int Conf Pattern Recognit. 2010, p. 2366–9.

[69] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF, editors. Proc Med Image Comput Comput-assist Interv.. 2015, p. 234–41.

[70] Zhu X, Wu X. Class noise vs. Attribute noise: A quantitative study. Artif Intell Rev 2004;22(3):177–210.

[71] Patrini G, Rozza A, Krishna Menon A, Nock R, Qu L. Making deep neural networks robust to label noise: A loss correction approach. In: Proc IEEE Conf Comput Vis Pattern Recognit. 2017.

[72] Stachenfeld K, Fielding DB, Kochkov D, Cranmer M, Pfaff T, Godwin J, et al. Learned simulators for turbulence. In: Proc Int Conf Learn Represent. 2022.

[73] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv Pre-Print 1412.6980.