

# Diverse and consistent multi-view networks for semi-supervised regression

Cuong Nguyen<sup>1</sup> · Arun Raja<sup>1,2</sup> · Le Zhang<sup>3</sup> · Xun Xu<sup>1</sup> · Balagopal Unnikrishnan<sup>4</sup> · Mohamed Ragab<sup>1</sup> · Kangkang Lu<sup>1</sup> · Chuan-Sheng Foo<sup>1,2</sup>

Received: 11 February 2022 / Revised: 13 January 2023 / Accepted: 19 January 2023 / Published online: 16 February 2023 © The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2023

#### Abstract

Label collection is costly in many applications, which poses the need for label-efficient learning. In this work, we present Diverse and Consistent Multi-view Networks (DiCoM)— a novel semi-supervised regression technique based on a multi-view learning framework. DiCoM combines diversity with consistency—two seemingly opposing yet complementary principles of multi-view learning—based on underlying probabilistic graphical assumptions. Given multiple deep views of the same input, DiCoM encourages a negative correlation among the views' predictions on labeled data, while simultaneously enforces their agreement on unlabeled data. DiCoM can utilize either multi-network or multi-branch architectures to make a trade-off between computational cost and modeling performance. Under realistic evaluation setups, DiCoM outperforms competing methods on tabular, time series and image data. Our ablation studies confirm the importance of having both consistency and diversity.

**Keywords** Semi-supervised regression · Multi-view learning · Diversity regularization · Probabilistic graphical models

## **1** Introduction

Deep neural networks have achieved tremendous success across several domains, ranging from computer vision, natural language processing, to audio analysis (LeCun et al., 2015). However, to train neural networks that perform well typically requires a large amount of labeled data. In many cases, this requirement for a large labeled dataset presents a challenge, because the annotation process can be labour-intensive and thus expensive, especially when specialized expertise is required. To address this challenge, semi-supervised

Chuan-Sheng Foo foo\_chuan\_sheng@i2r.a-star.edu.sg

Editors: Krzysztof Dembczynski and Emilie Devijver.

Cuong Nguyen and Arun Raja have contributed equally to this work.

Extended author information available on the last page of the article

learning methods (Van Engelen & Hoos, 2020) that can achieve similarly high performance with less labeled data by using unlabeled data have been developed.

We focus on semi-supervised learning in the regression setting. There are several approaches for semi-supervised regression, including graph-based methods (Zhur & Ghah-ramanirh, 2002), co-training (Blum & Mitchell, 1998) and entropy minimization (Jean et al., 2018). Consistency-based approaches that have been popular in the classification setting, such as Mean Teacher (Tarvainen & Valpola, 2017) and Virtual Adversarial Training (Miyato et al., 2018), which reinforce the output consistency of the network under input perturbations, have also been adapted to the regression setting (Jean et al., 2018). However, enforcing consistency alone may not be sufficient for good performance, and may lead to model collapsing (Qiao et al., 2018) or confirmation bias issues (Ke et al., 2019).

Another issue is that while the vast majority of semi-supervised learning research has been focused on semi-supervised classification (Van Engelen & Hoos, 2020), semi-supervised regression has not been receiving as much attention. Although classification and regression problems are both concerned with predicting output values for input data points, most semi-supervised classification methods cannot be naturally applied to the regression setting (Van Engelen & Hoos, 2020). This is because while some assumptions of semi-supervised learning such as the smoothness assumption or the manifold assumption (Chapelle et al., 2009) may hold for both classification and regression, other assumptions such as the cluster assumption (Chapelle et al., 2006) or the low-density separation assumption (Chapelle & Zien, 2005) do not apply to regression. Another issue with state-of-the-art classification methods such as FixMatch (Sohn et al., 2020), UDA (Xie et al., 2020) or ReMixMatch (Berthelot et al., 2019) is that they rely heavily on data augmentation techniques that are specific to visual data only.

To address these issues, we draw inspiration from ensemble learning with neural networks for regression. A necessary and sufficient condition for an ensemble of learners to be more accurate than any of its individual members is if the base learners are accurate and diverse (Dietterich, 2000). Therefore, the key component that can make or break an ensemble is the diversity (or disagreement) among its individual regressors. If this diversity is insufficient, the ensembling may not result in better performance. On the other hand, overemphasizing diversity can degrade the learnability of the ensemble members. So far, the most successful mechanism to leverage ensemble diversity in regression is Negative Correlation Learning (Liu & Yao, 1999; Zhang et al., 2019).

In this work, we propose Diverse and Consistent Multi-view Networks for Semi-supervised Regression (DiCoM) that elegantly unifies consistency and diversity in a deep multiview learning framework. Based on probabilistic graphical assumptions, we derive a loss function that integrates both consistency and diversity components. Diversity is encouraged on labeled data, while consistency is enforced on unlabeled data. Having separate optimization objectives means that both diversity and consistency can be enforced on the same representation level, which is the output label. Our approach has two advantages. First, DiCoM is less reliable on domain-specific input-level data augmentation, making it suitable for a wide range of data modalities. In our experiments, we compare DiCoM against state-of-the-art methods on eight tabular datasets, a crowd counting dataset and a remaining useful life dataset, where we show that DiCoM outperforms existing methods. Second, DiCoM is sufficiently flexible to be adapted on different network architectures. We develop two variants of DiCoM, the first uses multiple networks to achieve better performance, while the second employs a single network with multiple branches to help with scalability. Last but not least, we also perform ablation studies to analyze the importance of diversity and consistency, and the effect of varying the number of views in the model.

While other works have leveraged related ideas of complementary and consensus in multi-view classification (Xu et al., 2013); or explored commonality and individuality in multi-modal curriculum learning (Gong, 2017), these methods were developed for classification or clustering tasks, and cannot be easily modified to suit semi-supervised regression.

#### 2 Related work

Semi-supervised Regression: Semi-supervised learning is a data-efficient learning paradigm that offers the ability to learn from unlabeled data. In recent years, much work has focused on semi-supervised classification, and there have been far fewer studies on semisupervised regression. For regression tasks, graph-based methods are among the first to be developed. One example is Label Propagation (LP) (Zhur & Ghahramanirh, 2002) which defines a graph of training data and propagates ground-truth labels through high density regions of the graph. Kernel methods have also been proposed, such as Semisupervised Deep Kernel Learning (SSDKL) (Jean et al., 2018). This method minimizes the predictive variance in a posterior regularization framework to learn a more generalizable feature embedding on unlabeled data. Co-training regressors (COREG) (Zhou & Li, 2005) employs k-Nearest neighbor regressors, each of which generate pseudo-labels for the other during training; this helps to maximize their agreement on unlabeled data. In addition, tree-based methods which offer fast training and good interpretability, have also been developed for semi-supervised regression. Some examples are Self-training Tree Ensembles (ST-Tree) (Levatić et al., 2017), Semi-supervised Predictive Clustering Trees (SSL-PCT) (Levatić et al., 2018) and Semi-supervised Oblique Predictive Clustering Trees (SSL-SPYCT) (Stepišnik & Kocev, 2021). These methods have been widely adopted for regression tasks on tabular data.

Apart from the aforementioned approaches, *consistency-based methods* are also gaining traction. Mean Teacher (MT) (Tarvainen & Valpola, 2017) enforces posterior consistency between two neural networks, a student and a teacher, the latter being an exponential moving average of the former in the parameter space. An orthogonal approach is to enforce consistency on adversarially augmented input, as implemented in Virtual Adversarial Training (VAT) (Miyato et al., 2018). These methods were originally developed for classification, and were subsequently adapted to regression tasks (Jean et al., 2018). However, both MT and VAT maintain only a single trainable network, which may lead to problems such as confirmation bias (Ke et al., 2019) and overly-sensitive hyperparameters. In this paper, we show that consistency-based methods can be further improved with ensemble diversity.

*Ensemble Diversity:* Ensembles of neural networks have been extensively studied and widely used in many applications. Their effectiveness largely depends on the level of diversity (or disagreement) among members of the ensemble. It is well-understood that a good ensemble must manage the trade-off between the accuracy of the individual learners and the diversity among them (Brown et al., 2005; Tang et al., 2006). For regression tasks, a commonly-used ensemble technique is Negative Correlation Learning (NCL) (Liu & Yao, 1999; Liu et al., 2000), which formulates a diversity-promoting loss using an *ambiguity decomposition* of the squared ensemble loss (Krogh & Vedelsby, 1995). In this formulation, a correlation penalty term (also refered to as an ambiguity term) measures how much each member's prediction deviates from the ensemble output. When this penalty term is maximized, the errors of individual learners become negatively correlated. It was

theoretically proven (Brown et al., 2005) that the strategy employed by NCL is equivalent to leveraging a *bias-variance-covariance trade-off* (Ueda & Nakano, 1996) of the ensemble error.

Recently, NCL has been extended to semi-supervised learning (Chen et al., 2018), where the correlation penalty term is extended to the unlabeled data. However, this method was demonstrated only on tabular data. Another variant of NCL is Deep Negative Correlation Learning (DNCL) (Shi et al., 2018; Zhang et al., 2019), which is designed for visual regression tasks in a purely supervised learning setting.

Multi-view Learning: A dataset is considered as having multiple views when its data samples are represented by more than one feature set, each of which is sufficient for the learning task. Although each view is supposed to be sufficient for learning the task, a model trained on only one single view often faces the risk of overfitting, especially when labeled data is limited (Xu et al., 2013). To address this problem, multi-view learning assigns a modeling function to each view and jointly optimize these functions to improve overall generalization performance (Zhao et al., 2017). By analyzing the development of various techniques, Xu et al. (2013) summarized two significant principles that underpin multi-view learning: consensus and complementary. The consensus principle states that a multi-view technique must aim at maximizing the agreement on different views. This is similar to how consistency-based semi-supervised learning methods works: for instance, MT enforces agreement with its past self. The complementary principle states that in order to make improvement, each view must contain some information that the other views do not carry. In other words, the views should be sufficiently diverse. This is related to diversity regularization in ensemble learning, where individual learners are encouraged to give diverse predictions. Thus, multi-view learning offers a unifying perspective of both consistency and diversity.

#### 3 Proposed method

With DiCoM, we hope to generate and train multiple regressors that are consistent yet diverse with each other. On one hand, DiCoM applies multi-view consistency to ensure that through different augmentations and model parameters, the multiple outputs generated are still in agreement with each other. On the other hand, DiCoM encourages individual regressors to be repulsive to the average of their outputs. This diversity is much needed to enrich the total information capacity. Furthermore, as our mathematical analysis will show, diversity should be enforced on labeled data, while consistency works best on unlabeled data. Figure 1 illustrates the intuition behind DiCoM in comparison with DNCL (Zhang et al., 2014) and co-regression (Brefeld et al., 2006).

We will start this section by describing how multiple deep views can be generated from input data. Then, we propose our multi-view learning framework for regression, in which multiple deep views can be simultaneously optimized via backpropagation. We then discuss the graphical models that govern the probabilistic dependencies among the groundtruth label and the deep views. Finally, we derive the DiCoM loss function using these graphical models and provide a few insights.

*View creation:* Consider a regression task where the goal is to estimate a label  $y \in \mathbb{R}$  from an input *x*. To create multiple views, our first approach is to use *M* neural networks  $F_1, F_2, \ldots, F_M$ , each parameterized by  $\theta_1, \theta_2, \ldots, \theta_M$ , respectively. By applying different data augmentations  $\eta_1, \eta_2, \ldots, \eta_M$  on the original *x*, we generate *M* different augmented





Fig. 1 The concepts of a DNCL, **b** co-regression and **c** DiCoM, visualized on a 2-D label space. While DNCL focuses on promoting ensemble diversity and co-regression enforces mutual agreement among the views, DiCoM enables both diversity and consistency simultaneously

inputs  $x^m = \eta_m(x) \forall m = 1, ..., M$ . With each augmented input, the corresponding neural network produces a regression output  $f_m(x) = F_m(x^m, \theta_m) \forall m = 1, ..., M$ . Due to the different augmentations and network parameters, each output  $f_m$  can be treated as one *deep view* of the original input *x*. We call this multi-network setup DiCoM-N, where the N stands for 'network' (see Fig. 2a).

The second approach is to utilize a single network with a shared backbone *B* and multiple regressor branches  $R_1, R_2, \ldots, R_M$ . We use  $\theta_B$  to denote the learnable parameters from the backbone and  $\theta_1, \theta_2, \ldots, \theta_M$  to denote the parameters of the branches. The hidden features generated by the backbone serve as input to the branches. While the backbone still applies a random augmentation to the input *x*, each regressor branch  $R_m$  applies its own random augmentation  $\eta_m$  as well. Thus, regression outputs  $f_1, f_2, \ldots, f_M$  from the branches can be considered as deep views of the original input. We name this setup DiCoM-B, where the B is short for 'branch' (see Fig. 2b). Multi-branch technique was widely adopted for supervised classification (Xie et al., 2017). In this work, it allows us to harness the power of multi-view learning with a relatively lower number of trainable parameters.

*Multi-view learning framework for regression:* Regardless of how they are generated, the deep views are used together with the true label y to compute a semi-supervised



Fig.2 The DiCoM framework with two variants: a DiCoM-N (multi-network) and b DiCoM-B (multi-branch)

loss function  $\mathfrak{Q}_{\text{DiCoM}}$ . During training phase,  $\mathfrak{Q}_{\text{DiCoM}}$  is back-propagated simultaneously through the deep views to optimize network parameters  $\theta_1, \theta_2, \ldots, \theta_M$  (including  $\theta_B$  in the case of DiCoM-B). During inference, all augmentations are removed so that the forward pass is applied on the raw input *x*. The final prediction is computed as the average of all deep views:  $\mu(x) = \sum_{i=1}^{M} \frac{1}{M} f_m(x)$ . The general DiCoM framework is illustrated in Fig. 2. In the next step, we derive  $\mathfrak{Q}_{\text{DiCoM}}$  based on a probabilistic graphical assumption.

*Probabilistic graphical models:* Since the augmented inputs are generated from the same sample, the deep views should be close to each other. Motivated by previous work in kernel learning (Yu et al., 2011) and linear regression (Nguyen et al., 2019), we consider  $f_1, f_2, \ldots, f_M$  as random variables and introduce a *consensus function*  $f_c$  as a latent variable that connects to each of the deep views. This function enforces the mutual agreement among the views. We assume that the difference between the consensus function and each view follows a zero-mean Gaussian distribution

$$f_c - f_m \propto \mathcal{N}(0, \sigma_m^2) \quad \forall m = 1, \dots, M.$$
(1)

This probabilistic relation is known as the *consensus potential* (Yu et al., 2011). Considering the whole graph, this potential implies that all views are random Gaussian variables with a shared mean  $f_c$  and variance  $\sigma_m^2$ . As a result, the views stay consistent w.r.t. each other by taking values not too far away from the shared consensus. This graphical model, shown in Fig. 3a, is assumed for each unlabeled sample. The joint density associated with the graph is given by

$$p(f_c, f_1, \dots, f_M) = \frac{1}{\mathcal{Z}} \prod_{m=1}^M \Psi(f_c, f_m)$$
<sup>(2)</sup>

where  $\mathcal{Z}$  is a normalizing constant and  $\Psi(f_c, f_m) = \exp\left[-\frac{(f_c - f_m)^2}{2\sigma_m^2}\right]$  is the potential function of the edge connecting  $f_c$  and  $f_m$ . From this model, we derive two important results. On a side note, our derivation generalizes to vector-valued labels *y*, but here we assume scalar labels for ease of exposition. The proofs of our results are provided in Appendix A.



Fig. 3 Undirected probabilistic graphical models of DiCoM:  $\mathbf{a}$  for an unlabeled sample,  $\mathbf{b}$  after marginalization of the views and  $\mathbf{c}$  for a labeled sample

(1) Marginalization of the views: By integrating the latent consensus function  $f_c$  out of the joint density, the marginal distribution of the views is

$$p(f_1, \dots, f_M) \propto \exp\left[\sum_{m=1}^M \sum_{k>m} -\lambda_{m,k} (f_m - f_k)^2\right]$$
 (3)

where  $\lambda_{m,k} = \left[2\sigma_m^2 \sigma_k^2 \left(\sum_m \frac{1}{\sigma_m^2}\right)\right]^{-1}$ . This result implies that the marginal likelihood can be factorized as a product of  $\begin{pmatrix} M \\ 2 \end{pmatrix}$  terms. Each term is an isotropic Gaussian distribution on the difference between a pair of views  $(f_m, f_k)$ , with zero mean and variance  $(2\lambda_{m,k})^{-1}$ . The equivalent graphical model is shown in Fig. 3b.

(II) Conditional of the consensus function: By applying Bayes' theorem, the conditional distribution of the consensus function  $f_c$  given all the views is a Gaussian

$$f_c \mid f_1, \dots, f_M \sim \mathcal{N}\left(\tilde{\mu}, \sigma_{\mu}^2\right) \tag{4}$$

where  $\sigma_{\mu}^2 = \left(\sum_{m} \frac{1}{\sigma_m^2}\right)^{-1}$  and  $\tilde{\mu} = \sigma_{\mu}^2 \sum_{m} \frac{f_m}{\sigma_m^2}$ . This result highlights that the conditional distribution of  $f_c$  depends only on the weighted average  $\tilde{\mu}$ , and the values of individual views are not required. Furthermore,  $\tilde{\mu}$  can be treated as a view itself, with a variance that is smaller than any of the variances of the views.

*Derivation of DiCoM loss function:* For simplicity, we assume equal variance for different deep views, i.e.,  $\sigma_m^2 = \sigma_v^2 \quad \forall m$ . For an unlabeled sample  $(x_n)$ , we directly apply the first result **(I)** to obtain the following negative log likelihood function

$$L_{unl} = \sum_{m=1}^{M} \sum_{k>m} \frac{1}{2M\sigma_v^2} \left[ f_m(x_n) - f_k(x_n) \right]^2$$
(5)

For a labeled sample  $(x_n, y_n)$ , since the ground-truth is given, we assume a graphical model that involves the final DiCoM prediction, i.e., the averaged output  $\mu$ . This graph is shown in Fig. 3c. Since we assume a shared variance  $\sigma_v^2$ , the weighted output now reduces to an equal-weight average, following from result (II)

$$\tilde{\mu}(x_n) = \sum_{m=1}^{M} \frac{f_m(x_n)}{M} = \mu(x_n) \quad \sigma_{\mu}^2 = \frac{\sigma_{\nu}^2}{M}$$
(6)

Subsequently, we apply result (I) on this graph to get the negative log likelihood as follows

$$L_{lab} = \frac{M}{2M\sigma_{y}^{2} + 2\sigma_{v}^{2}} \left[ y_{n} - \mu(x_{n}) \right]^{2}$$
(7)

$$= \frac{1}{2M\sigma_y^2 + 2\sigma_y^2} \sum_{m=1}^M \left\{ \left[ f_m(x_n) - y_n \right]^2 - \left[ f_m(x_n) - \mu(x_n) \right]^2 \right\}$$
(8)

$$\approx \frac{1}{2\sigma_{v}^{2}} \sum_{m=1}^{M} \left\{ \left[ f_{m}(x_{n}) - y_{n} \right]^{2} - \left[ f_{m}(x_{n}) - \mu(x_{n}) \right]^{2} \right\}$$
(9)

where in Eq. (8), we have applied the *ambiguity decomposition* (Krogh & Vedelsby, 1995) and in Eq. (9), we have assumed that the label is accurate, i.e.,  $\sigma_y^2 \ll \sigma_v^2$ . Given a training batch of labeled samples  $\{(x_n, y_n)\}_{n=1}^L$  and unlabeled samples  $\{(x_n)\}_{n=1}^U$ .

Given a training batch of labeled samples  $\{(x_n, y_n)\}_{n=1}^{L}$  and unlabeled samples  $\{(x_n)\}_{n=1}^{U}$ , assuming that the samples are independently generated, we can add the log-likelihood functions across all training samples. This can be done via simply adding up two Eqs. (5) and (9)

$$\mathfrak{L}_{\text{DiCoM}} = \frac{1}{L} \sum_{n=1}^{L} \sum_{m=1}^{M} \left\{ \left[ f_m(x_n) - y_n \right]^2 - \kappa_{\text{div}} \left[ f_m(x_n) - \mu(x_n) \right]^2 \right\} + \frac{1}{U} \sum_{n=1}^{U} \sum_{m=1}^{M} \sum_{k>m} \kappa_{\text{csc}} \left[ f_m(x_n) - f_k(x_n) \right]^2$$
(10)

where we introduce two hyperparameters  $\kappa_{div}$  and  $\kappa_{csc}$  to absorb other constants and to enable a trade-off between diversity and consistency components of the loss.

The DiCoM loss encourages *diversity on labeled data*, while enforcing *consistency on unlabeled data*. These two seemingly opposing components can both be derived from the same underlying graphical assumptions. Furthermore, they should not be weighted equally. In fact, we have shown that it depends on the number of views: when *M* increases, diversity grows in O(M), while consistency grows in  $O(M^2)$ . It is worth noting that our method is fundamentally different from other extensions of NCL such as Semi-supervised NCL (Chen et al., 2018), which enforces diversity on both labeled and unlabeled data. Last but not least, since both diversity and consistency are incorporated in the DiCoM objective function, our method is highly adaptable to different implementations such as multi-network or multi-branch, as long as the views are provided.

#### 4 Experiments

In this section, we study the proposed method in different settings, including (i) regression tasks on eight tabular datasets, (ii) crowd counting on image data and (iii) remaining useful life prediction on time series data. We provide additional experiment results in Appendix C and an additional experiment on toy data in Appendix D.

*Datasets:* We evaluate DiCoM on eight datasets from the UCI repository (Dua & Graff, 2017): skillcraft, parkinsons, elevators, protein, blog, ctslice, buzz, and electric. These datasets are collected from real-world regression scenarios, with varying sample sizes and input dimensions. For each dataset, we keep 1000 labeled samples as a hold-out test set; further retain N = 300 samples for the labeled training set, and keep the rest as the unlabeled training set. We follow the realistic evaluation setup in (Oliver et al., 2018) and use a 90%-10% train-validation split, i.e., 270 samples are used for training, leaving only 30 for validation.

*Experiment Setup:* We implement both variants of DiCoM. Our DiCoM-N networks adopt the same architecture as (Jean et al., 2018; Wilson et al., 2016), which is a fully-connected multilayer perceptron with four hidden layers, containing 100, 50, 50 and 2 hidden nodes, respectively. Our DiCoM-B also utilizes this architecture, but branch out after the third hidden layer, i.e., the backbone contains hidden layers of 100, 50 and 50 nodes, while the regressor branches each contains one hidden layers of 2 nodes. This model is trained end-to-end, the backbone is trained together with the branches. DiCoM hyperparameters ( $\kappa_{div}$ ,  $\kappa_{csc}$ ) are chosen from a grid of values based on validation errors. Across 10 random seeds, we report the root-mean-squared errors (RMSE) statistics on the test set. For simplicity, we append '-*M*' to the end of our method name to denote the number of views, e.g., DiCoM-B-4 represents the multi-branch DiCoM network with 4 branches. In addition to RMSE, we also report the relative-root-mean-squared error (RRMSE), which is computed as the percentage between RMSE score of the model and the RMSE of the mean predictor (mean of labels from training set). Following (Levatić et al., 2018), the formula for RRMSE is as follows:

RRMSE = 
$$\sqrt{\frac{\sum_{n=1}^{T} (y_n - \hat{y}_n)^2}{\sum_{n=1}^{T} (y_n - \bar{y})^2}} \times 100$$
 (11)

where *T* is the number of samples in test set;  $y_n$  and  $\hat{y}_n$  are the ground truth label and the model's prediction on the *n*-th test sample, respectively;  $\bar{y}$  is the mean of target values on the labeled training set. Unlike RMSE which is domain-specific and difficult to interpret, RRMSE is a domain-independent metric.

*Data Augmentation:* we apply zero-mean Gaussian noise which is commonly used for tabular data. For the DiCoM-B model, Gaussian noise is applied on the input and on the features at the beginning of each regressor branch, right after branching out. Since the independent Gaussian noise is added during the forward pass, it does not affect the gradient values during backpropagation. The exact amount of Gaussian noise can be found in Appendix B.

We compare DiCoM against eight semi-supervised regression methods: SSDKL (Jean et al., 2018), COREG (Zhou & Li, 2005), LP (Zhur & Ghahramanirh, 2002), VAE (Jean et al., 2018), MT (Tarvainen & Valpola, 2017), VAT (Miyato et al., 2018), ST-Tree (Levatić et al., 2017), SSL-PCT (Levatić et al., 2018) and a Supervised baseline. These methods span a wide range of approaches such as consistency regularization (MT, VAT), entropy minimization (SSDKL), multi-view learning (COREG), graph-based (LP), generative modeling (VAE), or tree-based ensembling (ST-Tree, SSL-PCT). In addition, these competing methods also cover both conventional machine learning methods (LP, COREG, ST-Tree, SSL-PCT) as well as methods based on deep learning (VAE, MT, VAT, SSDKL).

Last but not least, the Supervised baseline uses the same network architecture as the backbone network. The detailed information about the datasets and experiment setup can be found in Appendix B.

*Results:* Figure 4 shows the experiment results. We observe significant improvements compared to the state-of-the-art semi-supervised regression methods. The largest performance gains are achieved on parkinsons and ctslice, where DiCoM-N-2 improves upon the best competing method by 32.4% and 42.5%, respectively. DiCoM-N-2 also outperforms DiCoM-B-2 on all datasets, except for protein. This is expected since DiCoM-N-2 has almost twice as many learnable parameters as DiCoM-B-2. For elevators, the most frequently selected hyperparameters across 10 random seeds are  $(\kappa_{div}, \kappa_{csc}) = (1, 0.01)$ , while for ctslice, the most frequently selected values are  $(\kappa_{div}, \kappa_{csc}) = (0.1, 1)$ . This shows that different datasets require different trade-offs between consistency and diversity. We also notice that LP (a graph-based method) and COREG (a nearest-neighbors-based method) performs relatively well on blog, ctslice and buzz. Meanwhile, MT and VAT, which are based on consistency regularization (without diversity regularization), did not perform well on these regression datasets, even though they have been shown to be effective on classification tasks. Tree-based ensembling methods (ST-Tree, SSL-PCT) yield higher errors than DiCoM variants on all datasets. They have relatively comparable performance to MT and VAT in most cases, except for ctslice and buzz. In general, the Supervised baseline performs worse than deep learning-based methods (SSDKL, VAE, MT, VAT, DiCoM-N-2 and DiCoM-B-2) and two non deep-learning methods (COREG, LP), but it can outperform tree-based methods in some cases. Table 1 shows the relative RMSE errors on UCI datasets. While some datasets such as parkinsons and electric have low RRMSE scores, many other datasets (skillcraft, protein and blog) caused high relative errors for all competing methods, including DiCoM. This implies that some of the UCI datasets remain as challenging regression tasks, even for the state-of-the-art methods.

Ablation study on the components: We analyse the effect of different components of the DiCoM-N-2 model by individually removing them from the model. For the first model, Ablation-1, we remove data augmentation. In Ablation-2, we remove the diversity loss on the unlabeled data. Next, the consistency loss on labeled data is set to zero for Ablation-3 model. Lastly, we apply diversity loss to both labeled and unlabeled training data for model Ablation-4. Note that our Ablation-4 implementation closely resembles the SSNCL



Fig. 4 Test RMSE on UCI datasets: each subplot shows the results for one dataset. Lower value indicates better performance

| Dataset    | skillcraft | parkinsons | elevators | protein | blog   | ctslice | buzz   | electric |
|------------|------------|------------|-----------|---------|--------|---------|--------|----------|
| SSDKL      | 74.45      | 32.01      | 63.79     | 91.44   | 104.09 | 44.02   | 37.55  | 11.98    |
| COREG      | 77.52      | 40.59      | 82.60     | 90.10   | 97.70  | 45.53   | 36.89  | 26.78    |
| LP         | 73.50      | 50.88      | 79.75     | 87.82   | 92.36  | 46.47   | 41.76  | 42.94    |
| VAE        | 85.03      | 100.93     | 96.46     | 99.93   | 93.76  | 74.05   | 53.84  | 100.32   |
| MT         | 89.87      | 82.65      | 110.33    | 97.82   | 101.41 | 76.53   | 83.49  | 71.53    |
| VAT        | 97.67      | 100.52     | 110.16    | 98.13   | 98.70  | 77.19   | 85.49  | 94.12    |
| ST-Tree    | 94.32      | 92.83      | 107.26    | 107.00  | 113.90 | 158.16  | 162.58 | 129.53   |
| SSL-PCT    | 97.52      | 88.63      | 106.34    | 100.56  | 107.90 | 155.45  | 161.84 | 129.01   |
| Supervised | 119.55     | 99.00      | 175.72    | 104.80  | 98.07  | 100.47  | 98.35  | 107.49   |
| DiCoM-N-2  | 67.29      | 21.65      | 61.48     | 82.79   | 84.45  | 25.33   | 35.08  | 11.73    |
| DiCoM-B-2  | 67.72      | 31.16      | 61.71     | 80.48   | 84.82  | 39.22   | 40.66  | 28.67    |

Table 1 Relative test errors (RRMSE, in %) on UCI datasets

Lower value indicates better performance

Bold values indicate the best performance for each setup

model (Chen et al., 2018), which also promotes ensemble diversity on both labeled and unlabeled data. Using the results of DiCoM-N-2 as the baseline, we also report the percentage reduction in test RMSE (% Redc.) for the other methods. This metric allows us to comprehend the impact made by each ablation model: a positive reduction in test error represents a positive impact and vice versa. The RMSE percentage reduction is computed as follows

Percentage Reduction = 
$$\frac{\text{(baseline score - new score)} \times 100}{\text{baseline score}}$$
(12)

Table 2 reports the ablation results. The average percentage reduction scores tell us the importance of each component. Augmentation has a relatively small impact on the performance of DiCoM-N-2, since the test errors of Ablation-1 are 5.07% larger than that of DiCoM-N-2. This supports our understanding that diversity can be enforced backward from the output-level representation, so that DiCoM only requires a minimal amount of input-level augmentation. Moreover, diversity is also promoted by the stochasticity of network parameters across different views. Mean while, without diversity, the model Ablation-2 performs much worse than the baseline with an average error increase of 14.5%. We suspect that the consistency enforcement is too strong, creating a risk of individual views collapsing into a single model. On the other hand, the model without consistency (Ablation-3) also suffer an error increase of 10.3%. Here, the lack of consistency regularization might have caused the individual views to have higher variance on test samples. Thus, both diversity and consistency regularization are important. DiCoM-N-2 outperforms Ablation-4 in all cases, which suggests that a mere reliance on diversity, such as the SSNCL approach, is insufficient. In Appendix D, we provide an additional experiment to study the impacts of consistency and diversity on the performance of DiCoM.

Varying the number of views: We further evaluate the impact of the number of views M. Figure 5 shows the performance of three DiCoM-N models with increasing number of views  $M \in \{2, 4, 8\}$ . The results show that a larger value of M leads to an improvement in the performance. When M increases from 4 to 8, the average reduction in test RMSE is 3.48%, larger than the average reduction rate of 1.70% when M increases from 2 to 4. While

|             |                       | •               |          |                   |          |                   |          |                   |          |
|-------------|-----------------------|-----------------|----------|-------------------|----------|-------------------|----------|-------------------|----------|
|             | DiCoM-N-2             | Ablation-1      |          | Ablation-2        |          | Ablation-3        |          | Ablation-4        |          |
| Aug.        | >                     | -               |          | >                 |          | >                 |          | ^                 |          |
| Lab. Div.   | >                     | >               |          |                   |          | >                 |          | `                 |          |
| Unlab. Csc. | >                     | >               |          | >                 |          |                   |          |                   |          |
| Unlab. Div. |                       |                 |          |                   |          |                   |          | `                 |          |
|             | RMSE R1               | MSE             | % Redc.  | RMSE              | % Redc.  | RMSE              | % Redc.  | RMSE              | % Redc.  |
| skillcraft  | $0.313 \pm 0.005$ 0.  | $330 \pm 0.017$ | - 5.559  | $0.327 \pm 0.008$ | - 4.610  | $0.333 \pm 0.012$ | - 6.446  | $0.330 \pm 0.011$ | - 5.326  |
| parkinsons  | $2.285 \pm 0.208$ 2.  | $437 \pm 0.280$ | - 6.666  | $2.390 \pm 0.278$ | - 4.593  | $2.563 \pm 0.299$ | - 12.159 | $2.438 \pm 0.286$ | - 6.692  |
| elevators   | $0.145 \pm 0.025$ 0.  | $142 \pm 0.031$ | 2.082    | $0.149 \pm 0.022$ | - 2.328  | $0.155 \pm 0.027$ | - 6.225  | $0.157 \pm 0.028$ | - 7.936  |
| protein     | $0.646 \pm 0.031$ 0.  | $679 \pm 0.029$ | - 5.153  | $0.654 \pm 0.025$ | - 1.313  | $0.669 \pm 0.034$ | - 3.546  | $0.661 \pm 0.030$ | - 2.327  |
| blog        | $0.930 \pm 0.040$ 1.  | $014 \pm 0.039$ | - 8.998  | $1.021 \pm 0.051$ | - 9.752  | $1.027 \pm 0.036$ | - 10.455 | $1.007 \pm 0.045$ | - 8.261  |
| ctslice     | $5.575 \pm 0.606$ 6.  | $976 \pm 0.984$ | - 25.120 | $7.154 \pm 1.176$ | - 28.306 | $7.461 \pm 1.014$ | - 33.813 | $7.398 \pm 0.662$ | - 32.699 |
| puzz        | $0.715 \pm 0.136$ 0.  | $757 \pm 0.064$ | - 5.825  | $0.938 \pm 0.489$ | -31.184  | $0.830 \pm 0.131$ | - 16.045 | $1.038 \pm 0.550$ | - 45.085 |
| electric    | $0.114 \pm 0.025$ 0.0 | $097 \pm 0.026$ | 14.658   | $0.153 \pm 0.124$ | - 33.833 | $0.107 \pm 0.011$ | 6.327    | $0.150\pm0.117$   | - 31.543 |
| Average     |                       |                 | - 5.073  |                   | - 14.490 | ·                 | - 10.295 |                   | - 17.484 |
|             |                       |                 |          |                   |          |                   |          |                   |          |

 Table 2
 Test RMSE from ablation study on UCI datasets



Fig. 5 Test RMSE of DiCoM-N on UCI datasets with varying number of views  $M \in \{2, 4, 8\}$ . The x-axis shows number of views M, the y-axis shows test RMSE.

varying the number of views, we also monitor the changes in the model hyperparameters. Using the values that were selected to minimize the validation error of DiCoM-N, Table 3 shows the log ratio of  $\log_{10}(\kappa_{div}/\kappa_{csc})$ . For most datasets, we see that this log ratio tends to increase for larger number of views. This is because in  $\mathfrak{Q}_{\text{DiCoM}}$ , the number of diversity terms grows in O(M) while the number of consistency terms grows in  $O(M^2)$ . Thus, as *M* increases, a larger ( $\kappa_{div}/\kappa_{csc}$ ) ratio is required to keep those terms balanced.

*Comparing multi-network and multi-branch:* In this experiment, we compare the two variants of DiCoM against each other, by reporting both their performance and execution time (adding train and test time). Table 4 shows the percentage reduction computed using Eq. (12) by treating DiCoM-N's results as the baseline scores and DiCoM-B's corresponding results as the new scores. It can be seen that DiCoM-N is consistently outperforming

|       | skillcraft | parkinsons | elevators | protein | blog  | ctslice | buzz  | electric |
|-------|------------|------------|-----------|---------|-------|---------|-------|----------|
| M = 2 | 1.849      | - 0.151    | 1.151     | 0.500   | 1.000 | - 1.000 | 0.151 | 1.849    |
| M = 4 | 2.000      | - 0.301    | 1.151     | 0.301   | 1.151 | 0.199   | 0.349 | 2.000    |
| M = 8 | 1.699      | - 0.500    | 1.500     | 1.151   | 1.849 | 0.349   | 0.500 | 2.000    |

**Table 3** Median values of  $\log_{10}(\kappa_{div}/\kappa_{csc})$  across 10 Seeds from DiCoM-N

| Table 4 | From I | DiCoM-N | to DiCoM-B: | percentage re | eduction in | test RMSE a | nd execution time |
|---------|--------|---------|-------------|---------------|-------------|-------------|-------------------|
|---------|--------|---------|-------------|---------------|-------------|-------------|-------------------|

| Dataset    | Test RMSE          |          |          | Execution tir | ne     |        |
|------------|--------------------|----------|----------|---------------|--------|--------|
|            | $\overline{M} = 2$ | M = 4    | M = 8    | M = 2         | M = 4  | M = 8  |
| skillcraft | - 0.702            | 2.012    | - 2.945  | - 33.336      | 23.505 | 48.462 |
| parkinsons | - 43.903           | - 24.177 | - 57.821 | 36.845        | 55.590 | 72.481 |
| elevators  | - 0.436            | - 13.126 | - 21.811 | - 52.633      | 19.389 | 60.346 |
| protein    | 2.798              | - 0.536  | - 0.520  | - 45.076      | 19.876 | 62.177 |
| blog       | - 0.430            | - 6.908  | - 18.729 | - 39.355      | 39.847 | 47.129 |
| ctslice    | - 54.862           | - 34.924 | - 73.619 | 45.016        | 81.627 | 83.731 |
| buzz       | - 15.885           | - 4.984  | - 17.976 | 4.194         | 33.759 | 51.035 |
| Average    | - 16.203           | - 11.806 | - 27.632 | - 12.049      | 39.085 | 60.766 |

DiCoM-B in terms of test RMSE. The multi-network variant is also the faster option (by 12.0%) when M = 2. However, as the number of views increases, the execution time of DiCoM-B is significantly faster, by 39.1% when M = 4 and by 60.8% when M = 8. This shows that while DiCoM-N achieves better test performance, DiCoM-B demonstrates better scalability.

#### 4.2 Crowd counting on image data

In order to show the versatility of DiCoM, we further conduct experiments on a crowd counting task. Crowd counting is a fundamental question in the vision community due to its far-reaching applications in many scenarios, including video surveillance, metropolis security, human behavior analysis. Crowd counting has been recently used as a benchmark for deep regression algorithms (Zhang et al., 2019); for this task, counting by regression has been perceived as the state-of-the-art approach.

*Dataset:* We study the ShanghaiTech Part-A dataset (Zhang et al., 2016). This is a new large-scale crowd counting dataset that contains extremely congested scenes, with varying perspective and unfixed resolution. The data are split into 300 training and 182 test samples. Among the 300 training samples, we randomly select  $N \in \{30, 120, 210\}$  samples as the labeled set and use the remaining data as the unlabeled set. We follow the common practice to report both mean absolute errors (MAE) and root-mean-squared errors (RMSE) on the test set. Similar to the UCI experiment, we also report the errors relative to the mean-absolute-value of the ground-truth labels in the test set, including RRMSE and RMAE. Here, RMAE score is computed by replacing RMSE in (11) with MAE, i.e., by dividing the MAE of the model by the MAE of the mean predictor. We note that this dataset inevitably contains personally identifiable information, which has been made public by the owner of the dataset.

*Experiment Setup:* In our experiments, we adopt the network architecture of CSRNet B (Li et al., 2018) and implement DiCoM-B-4. More specifically, we use a pre-trained VGG16 network as the encoder and append another decoder on top of it. In the penultimate layer of the decoder, we enlarge the number of hidden channels by M times. We then apply a group-convolutional layer as the last layer, setting both the number of output channels and group size to M. Thus, the backbone of DiCoM-B-4 includes the pre-trained VGG16 and the decoder up to its penultimate layer.

We compare DiCoM with the following competing methods: (i) the supervised baseline which uses only the labeled training samples and standard MSE loss; (ii) the DNCL model (Zhang et al., 2019); and (iii) the Co-Regression model (Brefeld et al., 2006). Since we are not running for multiple random seeds, we remove all random data augmentations (e.g., cropping, flipping) to enable fair comparison between different methods.

*Results:* From the results in Fig. 6, we observe that both DNCL and Co-Regression outperform the supervised baseline by enforcing either diversity on labeled data or consistency on unlabeled data, and that overall, DiCoM-B-4 outperforms other methods by incorporating both diversity and consistency on the unlabeled data. For example, at labeling budget N = 30, DiCoM-B improves MAE by 27.3% and RMSE by 33.9% compared to Co-Regression, the second best method. At the large budget N = 210, the improvements are 1.25% and 5.79% for MAE and RMSE, respectively. Table 5 shows the relative errors on ShanghaiTech data. It is interesting to note that these errors are smaller compared to those



Fig. 6 Test results on ShanghaiTech dataset: a RMSE and b MAE

Table 5 Relative test errors (RRMSE and RMAE, in %) on ShanghaiTech

|                | Supervise | ed baseline | DNCL  |       | Co-regree | ssion | DiCoM-H | 3-4   |
|----------------|-----------|-------------|-------|-------|-----------|-------|---------|-------|
|                | RMAE      | RRMSE       | RMAE  | RRMSE | RMAE      | RRMSE | RMAE    | RRMSE |
| N = 30         | 71.93     | 101.75      | 60.68 | 66.52 | 54.32     | 63.53 | 39.52   | 41.94 |
| N = 120        | 32.08     | 38.80       | 29.03 | 34.38 | 31.24     | 38.08 | 27.77   | 33.48 |
| <i>N</i> = 210 | 26.14     | 30.54       | 25.23 | 30.55 | 25.03     | 30.38 | 24.72   | 28.62 |
|                |           |             |       |       |           |       |         |       |

Lower value indicates better performance

Bold values indicate the best performance for each setup

obtained in some of the UCI datasets. This implies that the crowd counting task is better resolved, possibly thanks to the utilization of a pre-trained network.

#### 4.3 Remaining useful life prediction on time series data

In modern advanced manufacturing, predictive maintenance provides effective tools to reduce operational cost of industrial machinery. Among the sub-fields of predictive maintenance, remaining useful life prediction is a core pillar. By definition, remaining useful life (RUL) refers to the time duration from the current condition to the failure condition of an industrial machine. Precise prediction of RUL plays a crucial role in scheduling maintenance operations.

Recently, deep learning methods have shown state-of-the-art performance for RUL prediction task (Deutsch & He, 2017). Several deep learning architectures have been adopted for RUL task such as deep belief networks (Deutsch & He, 2017), convolutional neural networks (Li et al., 2018; Zhu et al., 2018), and long-short-term memory (LSTM) (Huang et al., 2019; Chen et al., 2020). However, labeled data for RUL task can be only obtained after machine failure, which is costly and time-consuming, especially for complex machines (Ragab et al., 2020). Meanwhile, unlabeled data are more readily available as they can be acquired under normal operation. Thus, an accurate semi-supervised regression model would further improve the cost-effectiveness of RUL prediction. *Dataset:* We use the NASA C-MAPSS dataset (Saxena et al., 2008), a benchmark dataset that describes the run-to-failure behaviours of aircraft engines. C-MAPSS contains four smaller subsets, from FD001 to FD004, each with different failure modes, operating conditions, lifespan and number of engines. These details can be found in Appendix C. Different 21 sensors have been used to monitor condition for each engine. Following Ragab et al. (2020), only most informative sensors that show clear degradation trend have been selected. Each data sample is a time series signal consisting of a number of selected pre-processed sensory features through 30 consecutive cycles. Given the current conditions of an engine, the task is to predict the number of remaining cycles until the engine fails. Ground-truth label is a positive integer indicating the number of remaining useful cycles. For model evaluation, we employ three evaluation metrics: root-mean-squared error (RMSE), relative error to the mean-absolute-value of ground-truth label (RRMSE) and RUL-Score (Heimes, 2008). Unlike RMSE which treats early and late predictions equally, the RUL-Score penalizes late predictions heavily, which is more suitable as a domain-specific metric. The equation to compute RUL-Score is as follows

$$\text{RUL-Score}_{i} = \begin{cases} e^{-\frac{\Delta \text{RUL}_{i}}{13}} - 1 & \text{for } \Delta \text{RUL}_{i} < 0\\ e^{\frac{\Delta \text{RUL}_{i}}{10}} - 1 & \text{for } \Delta \text{RUL}_{i} \ge 0 \end{cases}$$
(13)

where  $\Delta RUL_i$  is the difference between the predicted and true RUL for sample  $x_i$ . Since the predictions from RUL models are usually used to schedule maintenance operations, it would be undesirable if the engine fails before the predicted number of remaining cycles. Thus, the RUL-Score applies heavier penalty on positive  $\Delta RUL_i$  errors compared to negative errors. The reported RUL-Score for the whole dataset is computed as the sum of RUL-Score from individual samples.

*Experiment Setup:* In our experiments, we adopt the network architecture used in Ragab et al. (2020). It contains an LSTM encoder followed by a regressor network to facilitate the regression involved in predicting the RUL. The LSTM is bi-directional with 3 hidden layers and a window-length of 30. The regressor is a multi-layer perceptron consisting of one hidden layer mapping the output of the LSTM encoder to the final RUL prediction output. Upon this architecture, we implement a DiCoM-N model with M = 2 views, i.e., DiCoM-N-2. We compare our method against the following methods which are developed for semi-supervised RUL prediction: Self-supervised Learning (SelfSL) (Krokotsch et al., 2022), Variational Autoencoder (VAE) (Yoon et al., 2017) and Restricted Boltzmann Machines (RBM) (Ellefsen et al., 2019).

*Results:* From the results in Table 6, we can observe that DiCoM-N-2 performs better compared to the competing methods, except for a few setups such as FD001 at budget 2% and FD003 at budgets 10% and 20%. In the best scenario (FD001 at budget 10%), DiCoM-N-2 outperforms the best competing method by 35.7% in test RMSE and by 93.2% in RUL-Score. Another interesting point to note is the RMSE improvements of DiCoM-N-2 when the amount of labeled data increases. From 2% to 20% labeling budget, we can notice the test RMSE being halved (e.g., 35.6 to 15.8 for FD001), while the RMSE reductions of other methods are more gradual. The Supervised baseline performs worse than all semi-supervised methods in general, except for the lowest labeling budget (2%), where Supervised yields better RUL-Score than RBM and SelfSL. We believe this is due to the bias and volatile nature of the RUL-Score metric, which heavily penalizes positive errors by an exponential function. The RMSE score, being a more balanced metric, shows that the Supervised baseline is more error-prone than

| Table 6 T | Fest results on | C-MAPSS datase | ts    |       |        |            |                 |            |            |            |                |
|-----------|-----------------|----------------|-------|-------|--------|------------|-----------------|------------|------------|------------|----------------|
| Budget    | Subset          | Test RMSE      |       |       |        |            | RUL-score       |            |            |            |                |
|           |                 | DiCoM-N-2      | VAE   | RBM   | SelfSL | Supervised | DiCoM-N-2       | VAE        | RBM        | SelfSL     | Supervised     |
| 2%        | FD001           | 33.62          | 30.57 | 31.86 | 31.69  | 45.57      | 4.56E+03        | 1.01E+04   | 1.62E+04   | 1.50E+04   | 9.94E+03       |
|           | FD002           | 22.83          | 31.55 | 26.98 | 24.99  | 44.11      | 7.72E+03        | 7.85E+04   | 4.95E+04   | 3.75E+04   | $3.08E \pm 04$ |
|           | FD003           | 28.86          | 33.37 | 36.48 | 31.79  | 39.91      | 7.33E+03        | 3.22E+04   | 8.56E+04   | 1.31E+04   | 1.93E+04       |
|           | FD004           | 24.13          | 35.95 | 35.88 | 31.07  | 49.42      | <b>1.55E+04</b> | 1.58E+05   | 1.46E + 05 | 5.05E+04   | 4.76E+05       |
| 10%       | FD001           | 16.70          | 26.04 | 25.05 | 25.20  | 40.17      | 7.14E+02        | 1.25E+04   | 1.37E+04   | 1.06E + 04 | 2.30E+04       |
|           | FD002           | 15.63          | 23.70 | 22.55 | 21.48  | 44.52      | 2.02E + 03      | 1.48E + 04 | 1.03E+04   | 8.11E+03   | 7.37E+04       |
|           | FD003           | 20.99          | 20.14 | 23.47 | 21.19  | 43.45      | 2.10E + 03      | 3.40E + 03 | 5.24E+03   | 3.44E + 03 | 6.22E+04       |
|           | FD004           | 21.80          | 27.57 | 27.84 | 24.20  | 44.84      | 9.23E+03        | 3.28E + 04 | 2.51E+04   | 1.14E + 04 | 1.17E+05       |
| 20%       | FD001           | 15.24          | 21.84 | 25.01 | 23.52  | 40.26      | 4.94E+02        | 2.99E+03   | 1.22E+04   | 7.26E+03   | 2.47E+04       |
|           | FD002           | 15.70          | 22.30 | 20.99 | 20.26  | 44.78      | 2.01E + 03      | 8.13E+03   | 7.04E+03   | 3.91E + 03 | 8.61E+04       |
|           | FD003           | 19.89          | 17.10 | 17.26 | 16.34  | 43.66      | 3.27E+03        | 1.73E+03   | 1.97E+03   | 1.14E + 03 | 6.53E+04       |
|           | FD004           | 17.93          | 25.66 | 25.69 | 24.17  | 45.12      | 3.18E + 03      | 1.59E+04   | 1.61E+04   | 1.17E+04   | 1.35E+05       |
|           |                 |                |       |       |        |            |                 |            |            |            |                |

| datasets        |  |
|-----------------|--|
| C-MAPSS         |  |
| Test results on |  |
| e 6             |  |

🙆 Springer

Bold values indicate the best performance for each setup

Lower value indicates better performance

| Table 7Relative test errors(RRMSE, in %) on C-MAPSS | Budget | Subset | DiCoM-N-2 | VAE   | RBM   | SelfSL | Supervised |
|---|--------|--------|-----------|-------|-------|--------|------------|
| Datasets  | 2%     | FD001  | 39.67     | 36.07 | 37.59 | 37.39  | 53.76      |
|   |        | FD002  | 26.43     | 36.52 | 31.23 | 28.93  | 51.05      |
|   |        | FD003  | 34.37     | 39.74 | 43.45 | 37.86  | 47.53      |
|   |        | FD004  | 26.80     | 39.93 | 39.85 | 34.51  | 54.89      |
|   | 10%    | FD001  | 19.72     | 30.75 | 29.58 | 29.76  | 47.43      |
|   |        | FD002  | 18.11     | 27.45 | 26.12 | 24.88  | 51.56      |
|   |        | FD003  | 25.01     | 24.01 | 27.97 | 25.26  | 51.79      |
|   |        | FD004  | 24.19     | 30.58 | 30.88 | 26.85  | 49.74      |
|   | 20%    | FD001  | 17.99     | 25.79 | 29.53 | 27.77  | 47.54      |
|   |        | FD002  | 18.19     | 25.83 | 24.32 | 23.47  | 51.88      |
|   |        | FD003  | 23.71     | 20.38 | 20.57 | 19.48  | 52.05      |
|   |        | FD004  | 19.90     | 28.47 | 28.50 | 26.82  | 50.06      |

#### Lower value indicates better performance

Bold values indicate the best performance for each setup



Fig. 7 Test results on C-MAPSS FD001 dataset: a RMSE and b RUL-Score on log scale. Lower value indicates better performance

semi-supervised methods by a large margin. Table 7 shows the relative errors (RRMSE, in %). Compared to UCI and ShanghaiTech results, DiCoM-N-2 performs relatively better on C-MAPSS, yielding percentage errors less than 30% when labeling budget is 10% or 20%. Figure 7 shows the test RMSE and RUL-Score for the subset FD001. Plots from the remaining subsets can be found in Appendix C.

## 5 Conclusion

In this paper, we proposed novel Diverse and Consistent Multi-view Networks for Semisupervised Regression (DiCoM), that elegantly combines ensemble diversity with consistency regularization to tackle generic deep semi-supervised regression tasks. DiCoM utilizes probabilistic graphical models to control the underlying dependencies among multiple regression outputs and label. Experiments on tabular, visual and sequential data demonstrated the effectiveness of the proposed method across different domains. We also show that DiCoM is highly flexible, it can be adopted for multi-network or multi-branch implementations, the latter significantly improves the scalability of the method. On tabular data, our ablation studies validated the importance of both consistency and diversity. In the future, one may extend the DiCoM framework by introducing asymmetric views (of non-identical architectures), which naturally causes the final output  $\mu$  to be an unequally-weighted average. Another interesting direction is to explore the potential impact of data augmentation techniques, since multi-view learning often benefits from diversified inputs.

#### Appendix A Mathematical proofs

Consider again the general model, where there are *M* deep views, i.e.,  $\{\mathbf{f}_m\}_{m=1}^M$ . Graphically, these functions are represented by nodes that are connected not directly, but only via the consensus function  $\mathbf{f}_c$  using isotropic Gaussian potentials

$$\mathbf{f}_{c} - \mathbf{f}_{m} \propto \mathcal{N}(\mathbf{0}, \sigma_{m}^{2} \mathbf{I}).$$
(A1)

We note that in this general case, each deep view is a vector (instead of a scalar) and is assigned a separate variance  $\sigma_m^2$ , which are not necessarily equal to each other. In terms of notation, we use italic letters for scalar variables and boldface letters for vectors and matrices.

#### A.1 Marginal density of the views

In this proof, we derive the marginal distribution of the views. Given the DiCoM graphical model, it is necessary to integrate  $\mathbf{f}_c$  out of the joint density distribution of the graph, because  $\mathbf{f}_c$  is a latent variable. The joint density distribution function of this graphical model is as follows

$$p(\mathbf{f}_c, \mathbf{f}_1, \dots, \mathbf{f}_M) = \frac{1}{\mathcal{Z}_1} \prod_{m=1}^M \exp\left(-\frac{\|\mathbf{f}_c - \mathbf{f}_m\|^2}{2\sigma_m^2}\right)$$
(A2)

$$= \frac{1}{\mathcal{Z}_{1}} \exp\left(-\sum_{m} \frac{\mathbf{f}_{c}^{\mathsf{T}} \mathbf{f}_{c}}{2\sigma_{m}^{2}} + \sum_{m} \frac{\mathbf{f}_{c}^{\mathsf{T}} \mathbf{f}_{m}}{\sigma_{m}^{2}} - \sum_{m} \frac{\mathbf{f}_{m}^{\mathsf{T}} \mathbf{f}_{m}}{2\sigma_{m}^{2}}\right)$$
(A3)

$$= \frac{1}{\mathcal{Z}_1} \exp\left(-\frac{\psi}{2}\mathbf{f}_c^{\mathsf{T}}\mathbf{f}_c + \boldsymbol{\phi}^{\mathsf{T}}\mathbf{f}_c + \boldsymbol{\chi}\right),\tag{A4}$$

where the normalizing factor  $\mathcal{Z}_1$  is a constant w.r.t.  $\mathbf{f}_c, \mathbf{f}_1, \dots, \mathbf{f}_M$  and

$$\psi = \sum_{m=1}^{M} \frac{1}{\sigma_m^2} \qquad \mathbf{\Phi} = \sum_{m=1}^{M} \frac{\mathbf{f}_m}{\sigma_m^2} \qquad \chi = \sum_{m=1}^{M} -\frac{\mathbf{f}_m^\top \mathbf{f}_m}{2\sigma_m^2}.$$
 (A5)

Notice that  $\psi$ ,  $\phi$ ,  $\chi$  are constants w.r.t.  $\mathbf{f}_c$ . By applying the following integration rule for a multivariate Gaussian variable **x** Petersen and Pedersen (2008)

$$\int \exp\left(-\frac{1}{2}\mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x} + \mathbf{c}^{\mathsf{T}}\mathbf{x}\right) d\mathbf{x} = \sqrt{\det\left(2\pi\mathbf{A}^{-1}\right)} \exp\left(\frac{1}{2}\mathbf{c}^{\mathsf{T}}\mathbf{A}^{-\mathsf{T}}\mathbf{c}\right),\tag{A6}$$

we can integrate  $\mathbf{f}_c$  out of the joint distribution in (A4) to obtain the following marginal likelihood

$$p(\mathbf{f}_1, \dots, \mathbf{f}_M) = \int p(\mathbf{f}_c, \mathbf{f}_1, \dots, \mathbf{f}_M) d\mathbf{f}_c$$
(A7)

$$= \frac{1}{\mathcal{Z}_2} \exp\left(\frac{\boldsymbol{\Phi}^{\mathsf{T}} \boldsymbol{\Phi}}{2\psi} + \chi\right) \tag{A8}$$

$$= \frac{1}{\mathcal{Z}_2} \exp\left[\frac{1}{2\psi} \left(\sum_m \frac{\mathbf{f}_m^{\mathsf{T}} \mathbf{f}_m}{\sigma_m^4} + 2\sum_m \sum_{k>m} \frac{\mathbf{f}_m^{\mathsf{T}} \mathbf{f}_k}{\sigma_m^2 \sigma_k^2} - \psi \sum_m \frac{\mathbf{f}_m^{\mathsf{T}} \mathbf{f}_m}{\sigma_m^2}\right)\right]$$
(A9)

$$= \frac{1}{\mathcal{Z}_2} \exp\left[\frac{1}{2\psi} \left(\sum_{m} \sum_{k>m} -\frac{\mathbf{f}_m^{\mathsf{T}} \mathbf{f}_m - 2\mathbf{f}_m^{\mathsf{T}} \mathbf{f}_k + \mathbf{f}_k^{\mathsf{T}} \mathbf{f}_k}{\sigma_m^2 \sigma_k^2}\right)\right]$$
(A10)

$$= \frac{1}{\mathcal{Z}_2} \exp\left(\sum_m \sum_{k>m} \frac{-\|\mathbf{f}_m - \mathbf{f}_k\|^2}{2\rho_{m,k}^2}\right)$$
(A11)

$$= \frac{1}{\mathcal{Z}_2} \exp\left(\sum_{m} \sum_{k>m} -\lambda_{m,k} \|\mathbf{f}_m - \mathbf{f}_k\|^2\right),$$
(A12)

where  $\mathcal{Z}_2$  is another constant w.r.t.  $\mathbf{f}_c, \mathbf{f}_1, \dots, \mathbf{f}_M$ , and

$$\rho_{m,k} = \sigma_m^2 \sigma_k^2 \psi = \sigma_m^2 \sigma_k^2 \left(\sum_{i=1}^M \frac{1}{\sigma_i^2}\right)$$
(A13)

$$\lambda_{m,k} = (2\rho_{m,k})^{-1} = \left[2\sigma_m^2 \sigma_k^2 \left(\sum_{i=1}^M \frac{1}{\sigma_i^2}\right)\right]^{-1}.$$
 (A14)

#### A.2 Conditional density of the consensus function

In this proof, we derive the conditional density distribution of the consensus function  $\mathbf{f}_c$  given the views. Consider again the general model with M views, i.e.,  $\{\mathbf{f}_m\}_{m=1}^M$ . Each view is represented by a random variable connected only to the consensus function  $\mathbf{f}_c$  via an isotropic Gaussian potential as defined in (A1). From (A2), (A11), the conditional distribution of  $\mathbf{f}_c$ given the views  $\mathbf{f}_1, \ldots, \mathbf{f}_M$  is

$$p(\mathbf{f}_c \mid \mathbf{f}_1, \dots, \mathbf{f}_M) = \frac{p(\mathbf{f}_c, \mathbf{f}_1, \dots, \mathbf{f}_M)}{p(\mathbf{f}_1, \dots, \mathbf{f}_M)}$$
(A15)

$$= \frac{1}{\mathcal{Z}_{3}} \exp\left(\sum_{m=1}^{M} \frac{-\|\mathbf{f}_{c} - \mathbf{f}_{m}\|^{2}}{2\sigma_{m}^{2}} + \sum_{m=1}^{M} \sum_{k>m}^{M} \frac{\|\mathbf{f}_{m} - \mathbf{f}_{k}\|^{2}}{2\rho_{m,k}^{2}}\right)$$
(A16)

$$= \frac{1}{\mathcal{Z}_3} \exp\left(\sum_m \frac{-\mathbf{f}_c^{\mathsf{T}} \mathbf{f}_c + 2\mathbf{f}_m^{\mathsf{T}} \mathbf{f}_c - \mathbf{f}_m^{\mathsf{T}} \mathbf{f}_m}{2\sigma_m^2} + \sum_m \sum_{k>m} \frac{\|\mathbf{f}_m - \mathbf{f}_k\|^2}{2\rho_{m,k}^2}\right)$$
(A17)

$$= \frac{1}{\mathcal{Z}_{3}} \exp\left(\frac{-\mathbf{f}_{c}^{\mathsf{T}}\mathbf{f}_{c}}{2\sigma_{\mu}^{2}} + \frac{\tilde{\boldsymbol{\mu}}^{\mathsf{T}}\mathbf{f}_{c}}{\sigma_{\mu}^{2}} + \boldsymbol{\aleph}\right),\tag{A18}$$

where the normalizing factor  $Z_3$  is a constant w.r.t.  $\mathbf{f}_c, \mathbf{f}_1, \dots, \mathbf{f}_M$  and

$$\sigma_{\mu}^{2} = \left(\sum_{m=1}^{M} \frac{1}{\sigma_{m}^{2}}\right)^{-1}$$
(A19)

$$\tilde{\boldsymbol{\mu}} = \sigma_{\mu}^2 \sum_{m=1}^{M} \frac{\mathbf{f}_m}{\sigma_m^2}$$
(A20)

$$\boldsymbol{\aleph} = \sum_{m=1}^{M} \frac{-\mathbf{f}_{m}^{\mathsf{T}} \mathbf{f}_{m}}{2\sigma_{m}^{2}} + \sum_{m=1}^{M} \sum_{k>m}^{M} \frac{\|\mathbf{f}_{m} - \mathbf{f}_{k}\|^{2}}{2\rho_{m,k}^{2}}.$$
 (A21)

Using the definitions (A13), (A19) and (A20), ℵ can be rewritten as follows

$$\boldsymbol{\aleph} = \sigma_{\mu}^{2} \left( \sum_{m} \frac{-\mathbf{f}_{m}^{\mathsf{T}} \mathbf{f}_{m}}{2\sigma_{m}^{2} \sigma_{\mu}^{2}} + \sum_{m} \sum_{k>m} \frac{\|\mathbf{f}_{m} - \mathbf{f}_{k}\|^{2}}{2\sigma_{m}^{2} \sigma_{k}^{2}} \right)$$
(A22)

$$= \sigma_{\mu}^{2} \left[ \left( \sum_{m} \frac{-\mathbf{f}_{m}^{\mathsf{T}} \mathbf{f}_{m}}{2\sigma_{m}^{2}} \right) \left( \sum_{k} \frac{1}{\sigma_{k}^{2}} \right) + \sum_{m} \sum_{k>m} \frac{\|\mathbf{f}_{m} - \mathbf{f}_{k}\|^{2}}{2\sigma_{m}^{2} \sigma_{k}^{2}} \right]$$
(A23)

$$= \sigma_{\mu}^{2} \left( \sum_{m} \sum_{k} \frac{-\mathbf{f}_{m}^{\mathsf{T}} \mathbf{f}_{m}}{2\sigma_{m}^{2} \sigma_{k}^{2}} + \sum_{m} \sum_{k \neq m} \frac{\mathbf{f}_{m}^{\mathsf{T}} \mathbf{f}_{m}}{2\sigma_{m}^{2} \sigma_{k}^{2}} + \sum_{m} \sum_{k > m} \frac{-\mathbf{f}_{m}^{\mathsf{T}} \mathbf{f}_{k}}{\sigma_{m}^{2} \sigma_{k}^{2}} \right)$$
(A24)

$$= \sigma_{\mu}^{2} \left( \sum_{m} \frac{-\mathbf{f}_{m}^{\mathsf{T}} \mathbf{f}_{m}}{2\sigma_{m}^{4}} + \sum_{m} \sum_{k>m} \frac{-\mathbf{f}_{m}^{\mathsf{T}} \mathbf{f}_{k}}{\sigma_{m}^{2} \sigma_{k}^{2}} \right)$$
(A25)

# 

$$= -\frac{\sigma_{\mu}^{2}}{2} \left(\sum_{m} \frac{\mathbf{f}_{m}^{\mathsf{T}}}{\sigma_{m}^{2}}\right) \left(\sum_{m} \frac{\mathbf{f}_{m}}{\sigma_{m}^{2}}\right)$$
(A26)

$$=\frac{-\tilde{\mu}^{\top}\tilde{\mu}}{2\sigma_{\mu}^{2}}.$$
(A27)

Thus, we can rewrite (A18) in its Gaussian form

$$p(\mathbf{f}_c \mid \mathbf{f}_1, \dots, \mathbf{f}_M) = \frac{1}{\mathcal{Z}_3} \exp\left(\frac{-\mathbf{f}_c^{\mathsf{T}} \mathbf{f}_c}{2\sigma_\mu^2} + \frac{\tilde{\mathbf{\mu}}^{\mathsf{T}} \mathbf{f}_c}{\sigma_\mu^2} - \frac{\tilde{\mathbf{\mu}}^{\mathsf{T}} \tilde{\mathbf{\mu}}}{2\sigma_\mu^2}\right)$$
(A28)

$$= \frac{1}{\mathcal{Z}_3} \exp\left(-\frac{\|\mathbf{f}_c - \tilde{\boldsymbol{\mu}}\|^2}{2\sigma_{\mu}^2}\right).$$
(A29)

Therefore,

$$\mathbf{f}_{c} \mid \mathbf{f}_{1}, \dots, \mathbf{f}_{M} \sim \mathcal{N}\left(\tilde{\boldsymbol{\mu}}, \sigma_{\mu}^{2} \mathbf{I}\right).$$
(A30)

#### Appendix B Experiment setup details

In this section, we provide the detailed setups for our benchmarking experiments. All experiments are run on NVIDIA GeForce GTX 1080Ti GPUs, using an Anaconda virtual environment installed with CUDA version 10.1, Python version 3.7.10 and Pytorch version 1.7.0. In each experiment, all the competing methods are evaluated using the same training/validation/test split.

Experiment setup for DiCoM in UCI experiments:

- Network architecture: for DiCoM-N, fully-connected multilayer perceptron with hidden layers of size [100, 50, 50, 2]. For DiCoM-B, the backbone includes the first 3 hidden layers of size [100, 50, 50] and the branches include one hidden layer of size 2. Note that we are not counting the input and output layers.
- Parallelization: for DiCoM-B, we use group convolution in order to back-propagate through all branches simultaneously.
- Random seeds: 20, 40, ..., 200 (10 seeds in total).
- Training: 2000 epochs with 250 epoch patience for early stopping (stop if no improvement is observed on validation set for 250 consecutive epochs).
- Optimizer: Stochastic Gradient Descent with momentum 0.95 and weight decay 10<sup>-9</sup>. Learning rate is 10<sup>-4</sup> for ctslice and is 10<sup>-3</sup> for other datasets.
- Augmentation: additive random Gaussian noise with mean 0 and standard deviation 0.05 for DiCoM-N and 0.01 for DiCoM-B.
- Diversity hyperparameter search range:  $\kappa_{div} \in \{0.01, 0.05, 0.1, 0.5, 1\}$ .
- Consistency hyperparameter search range:  $\kappa_{csc} \in \{0.01, 0.05, 0.1, 0.5, 1\}$ .

- Network architecture: CSRNet B (Li et al., 2018).
- Random seed: 9999 (only 1 seed).
- Training: 1000 epochs with no early stopping.
- Optimizer: Adam with weight decay  $10^{-5}$  and learning rate  $10^{-5}$ .
- Augmentation: None.
- Diversity hyperparameter search range:  $\kappa_{div} \in \{10^{-5}, 10^{-4}, 10^{-3}\}.$
- Consistency hyperparameter search range:  $\kappa_{csc} \in \{10^{-5}, 10^{-4}, 10^{-3}\}$ .

Experiment setup for DiCoM in C-MAPSS experiments:

- Network architecture: bi-directional LSTM (Ragab et al., 2020).
- Random seeds: 190, 210, 230.
- Training: 100 epochs with 60 epochs of early stopping.
- Optimizer: AdamW with weight decay  $10^{-2}$  and learning rate  $10^{-4}$ .
- Augmentation: additive random Gaussian noise with mean 0 and standard deviation 0.05.
- Diversity hyperparameter search range:  $\kappa_{div} \in \{10^{-5}, 10^{-2}, 10^{-1}\}.$
- Consistency hyperparameter search range:  $\kappa_{csc} \in \{10^{-5}, 10^{-2}, 10^{-1}\}$ .

Experiment setup for competing methods in UCI experiments:

We follow the setup in Jean et al. (2018) for SSDKL, COREG, LP, VAE, MT and VAT, publicly available at: https://github.com/ermongroup/ssdkl. For ST-Tree and SSL-PCT, we follow the setup in Levatić et al. (2017, 2018). The setup details for each method are as follows:

- For SSDKL:
  - Network architecture: Fully-connected multilayer perceptron with hidden layers of size [100, 50, 50, 2].
  - Random seeds: 20, 40, ..., 200 (10 seeds in total).
  - Training: 2000 epochs.
  - Optimizer: Adam with momentum 0.9,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ . Learning rate is  $10^{-4}$  with decay rate of 0.9 every 50 epochs.
  - Kernel: squared exponential kernel with  $\sigma_l = 1.0$ ,  $\sigma_f = 1.0$ ,  $\sigma_n = 1.5$ .
- For COREG:
  - Random seeds: 20, 40, ..., 200 (10 seeds in total).
  - COREG parameters  $k_1 = 3$ ,  $k_2 = 3$ ,  $p_1 = 2$ ,  $p_2 = 5$ , pool size is 100.
- For LP:
  - Initialization: k-NN with k = 5.
  - Kernel: squared Euclidean, search for  $\sigma^2$  in a range of {0.8, 1.35, 1.9, 2.45, 3.0}.
- For VAE:
  - Network architecture: Fully-connected multilayer perceptron with hidden layers of size [100, 50, 50, 2, 50, 50, 100].
  - Random seeds: 20, 40, ..., 200 (10 seeds in total).

- Training: 2000 epochs.
- Optimizer: Adam with momentum 0.9,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ . Learning rate is  $10^{-3}$ .
- For MT:
  - Network architecture: Fully-connected multilayer perceptron with hidden layers of size [100, 50, 50, 2].
  - Random seeds: 20, 40, ..., 200 (10 seeds in total).
  - Training: 2000 epochs.
  - Optimizer: Stochastic Gradient Descent with momentum 0.95. Learning rate is 10<sup>-3</sup>.
  - Augmentation: additive random Gaussian noise with mean 0 and standard deviation 0.4.
  - Exponential moving average parameter  $\alpha = 0.999$ .
- For VAT:
  - Network architecture: Fully-connected multilayer perceptron with hidden layers of size [100, 50, 50, 2].
  - Random seeds: 20, 40, ..., 200 (10 seeds in total).
  - Training: 2000 epochs.
  - Optimizer: Adam with momentum 0.9,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ . Learning rate is  $10^{-4}$  with decay rate of 0.9 every 50 epochs.
  - VAT parameters  $\epsilon = 2.0$ , number of power iterations is 1.
- For ST-Tree:
  - Ensemble method: random forest with 100 base-level trees.
  - Number of semi-supervised learning iterations: 10.
  - Unlabeled criteria: threshold. Unlabeled samples with confidence of prediction greater than threshold 0.8 are added to the training set.
  - Other parameters are set at their default values.
- For SSL-PCT:
  - Ensemble method: random forest with 100 base-level trees.
  - Number of semi-supervised learning iterations: 10.
  - Unlabeled criteria: automatic-OOB-initial. After the initial iteration, unlabeled threshold is automatically selected on the basis of reliability scores of out-of-bag labeled examples.
  - Other parameters are set at their default values.

Experiment setup for competing methods in ShanghaiTech experiments:

In ShanghaiTech experiments, for DNCL and Co-Regression: We follow the setup in Zhang et al. (2019). Publicly available at: https://github.com/shizenglin/Deep-NCL. The setup details for each method are as follows:

- For DNCL:
  - Network architecture: CSRNet B Li et al. (2018).
  - Random seed: 9999.

- Training: 1000 epochs with no early stopping.
- Optimizer: Adam with weight decay  $10^{-5}$  and learning rate  $10^{-5}$ .
- Augmentation: None.
- DNCL correlation parameter  $\lambda = 10^{-5}$ .
- For Co-Regression:
  - Network architecture: CSRNet B (Li et al., 2018).
  - Random seed: 9999.
  - Training: 1000 epochs with no early stopping.
  - Optimizer: Adam with weight decay  $10^{-5}$  and learning rate  $10^{-5}$ .
  - Augmentation: None.
  - Co-Regression pairwise-disagreement parameter  $\lambda = 10^{-5}$ .

#### Experiment setup for competing methods in C-MAPSS experiments:

In C-MAPSS experiments, for VAE, RBM and SelfSL: We follow the setup in Krokotsch et al. (2022). Publicly available at: https://github.com/tilman151/self-supervised-ssl. The setup details for each method are as follows:

- For VAE:
  - Network architecture: Encoder is a 1-D CNN with 6 hidden layers of size 64 each. Decoder network is a mirror of the encoder network.
  - Random seeds: 190, 210, 230.
  - Training: 200 epochs with 100 epochs of pre-training.
  - Optimizer: Stochastic Gradient Descent with learning rate 10<sup>-4</sup>.
- For RBM:
  - Network architecture: RBM with 1 hidden layer of size 64 and ReLU activation.
  - Random seeds: 190, 210, 230.
  - Training: 200 epochs with 5 epochs of pre-training.
  - Optimizer: Adam with learning rate 10<sup>-4</sup>.
- For SelfSL:
  - Network architecture: 1-D CNN (6 hidden layers of size 64) with an MLP prediction head.
  - Random seeds: 190, 210, 230.
  - Training: 200 epochs with 100 epochs of pre-training.
  - Optimizer: Adam with learning rate 10<sup>-4</sup>.

# Appendix C Additional experiment results

*Information of UCI datasets:* The detailed statistics of the eight UCI datasets are given in Table 8.

| Dataset    | No. of samples | Input dim. | No. of unique<br>Label values | Prediction target  |
|------------|----------------|------------|-------------------------------|--|
| skillcraft | 3325           | 18         | 7                             | Skill level of gamers (ordinal classification)             |
| parkinsons | 5875           | 20         | 1129                          | Unified Parkinson's Disease Rating Scale<br>(UPDRS) scores |
| elevators  | 16,599         | 18         | 61                            | Aileron control of F16 aircraft                            |
| protein    | 45,730         | 9          | 15,903                        | Physicochemical properties of protein tertiary structure   |
| blog       | 52,397         | 280        | 438                           | Number of comments received within 24 hrs                  |
| ctslice    | 53,500         | 384        | 53,347                        | Relative location of the image on the axial axis           |
| buzz       | 583,250        | 77         | 8123                          | Popularity of a topic in social media                      |
| electric   | 2,049,280      | 6          | 4186                          | Power consumption in one household per minute              |

 Table 8
 UCI regression datasets

Additional experiment results on UCI datasets with labeling budget N = 300: Please see additional results from Sect. 4.1 in Tables 9 and 10.

Additional experiment results on UCI datasets with labeling budget N = 100: Using the same UCI datasets, we conduct experiments similar to the ones in Sect. 4.1 with a smaller labeling budget of N = 100 samples. The results are provided in Fig. 8, Tables 11, 12, 13, 14 and 15.

Additional experiment results on ShanghaiTech dataset: The numerical results accompanying Fig. 6 are provided in Table 16.

*Information of C-MAPSS dataset:* The detailed statistics of the four C-MAPSS subsets are given in Table 17.

Additional experiment results on C-MAPSS datasets: For subsets FD002, FD003 and FD004, the plots of test results are provided in Figs. 9, 10 and 11.

#### Appendix D Additional experiment on toy data

We conduct an experiment on a synthetic toy dataset to illustrate how multi-view diversity and consistency work together to affect the training and inference of DiCoM-N.

*Dataset:* We synthesize a regression dataset where inputs  $x \in \mathbb{R}^{30}$  and labels  $y \in \mathbb{R}^2$ . The labels are related to the inputs by  $y = Ax + \epsilon$ , where A is a fixed  $2 \times 30$  coefficient matrix and  $\epsilon \sim \mathcal{N}(\mathbf{0}, 0.3^2 \mathbf{I})$ . Each coordinate of x is drawn from the standard normal distribution. We generate a training set of 100 labeled and 1000 unlabeled samples, and a hold-out test set of 1000 labeled samples.

*Experiment Setup:* Our DiCoM-N model has M = 5 views, each uses a simple neural network with a single hidden layer containing two hidden nodes. We train the model with SGD for 50 epochs with a learning rate of  $5 \times 10^{-2}$ , then evaluate the mean-squared-error (MSE) of the model on the test set.

*Results:* We keep  $\kappa_{div} = 1$  and vary  $\kappa_{csc}$  on a log scale:  $\kappa_{csc} \in \{0.01, 0.1, 1\}$ . Both quantitative and qualitative results are shown in Fig. 12. We plot the training losses on the top row and visually show the predictions of each network on eight random test samples. On the left scenario (Fig. 12a) when  $\kappa_{div} \gg \kappa_{csc}$ , the diversity loss dominates the consistency component. Even though the total loss converges on the training set, the

| Table 9 Test R1 | MSE on UCI Datase | sts with Labeling Buc | lget $N = 300$ .  |                   |                   |                    |                   |                   |
|-----------------|-------------------|-----------------------|-------------------|-------------------|-------------------|--------------------|-------------------|-------------------|
| Dataset         | skillcraft        | parkinsons            | elevators         | protein           | blog              | ctslice            | puzz              | electric          |
| SSDKL           | $0.346 \pm 0.036$ | $3.379 \pm 0.387$     | $0.151 \pm 0.013$ | $0.714 \pm 0.043$ | $1.146 \pm 0.072$ | $9.691 \pm 1.135$  | $0.766 \pm 0.040$ | $0.117 \pm 0.011$ |
| COREG           | $0.361 \pm 0.009$ | $4.285 \pm 0.200$     | $0.195 \pm 0.006$ | $0.703 \pm 0.020$ | $1.076 \pm 0.040$ | $10.023 \pm 1.235$ | $0.752 \pm 0.035$ | $0.261 \pm 0.008$ |
| LP              | $0.342 \pm 0.012$ | $5.371 \pm 0.994$     | $0.189 \pm 0.005$ | $0.685 \pm 0.011$ | $1.017 \pm 0.048$ | $10.230 \pm 1.410$ | $0.851 \pm 0.030$ | $0.418 \pm 0.010$ |
| VAE             | $0.396 \pm 0.014$ | $10.655 \pm 0.151$    | $0.228 \pm 0.010$ | $0.780\pm0.017$   | $1.033 \pm 0.045$ | $16.301 \pm 0.773$ | $1.098 \pm 0.026$ | $0.977 \pm 0.017$ |
| MT              | $0.418 \pm 0.038$ | $8.725 \pm 0.904$     | $0.261 \pm 0.031$ | $0.763 \pm 0.018$ | $1.117 \pm 0.036$ | $16.845 \pm 1.996$ | $1.702 \pm 0.160$ | $0.696 \pm 0.023$ |
| VAT             | $0.454 \pm 0.025$ | $10.612 \pm 0.198$    | $0.261 \pm 0.012$ | $0.766 \pm 0.010$ | $1.087 \pm 0.063$ | $16.992 \pm 4.839$ | $1.743 \pm 0.186$ | $0.916 \pm 0.127$ |
| ST-Tree         | $0.439 \pm 0.015$ | $9.800 \pm 0.662$     | $0.254 \pm 0.008$ | $0.835 \pm 0.019$ | $1.254 \pm 0.104$ | $34.815 \pm 0.662$ | $3.315 \pm 0.045$ | $1.261 \pm 0.011$ |
| SSL-PCT         | $0.454 \pm 0.022$ | $9.356 \pm 0.089$     | $0.252 \pm 0.008$ | $0.785 \pm 0.002$ | $1.188 \pm 0.034$ | $34.219 \pm 0.322$ | $3.300 \pm 0.038$ | $1.256 \pm 0.006$ |
| Supervised      | $0.556 \pm 0.091$ | $10.451 \pm 0.226$    | $0.416 \pm 0.124$ | $0.818\pm0.056$   | $1.080 \pm 0.056$ | $22.116 \pm 0.316$ | $2.005 \pm 0.034$ | $1.046 \pm 0.055$ |
| DiCoM-N-2       | $0.313 \pm 0.005$ | $2.285 \pm 0.208$     | $0.145 \pm 0.025$ | $0.646 \pm 0.031$ | $0.930 \pm 0.040$ | $5.575 \pm 0.606$  | $0.715 \pm 0.136$ | $0.114 \pm 0.025$ |
| DiCoM-B-2       | $0.315\pm0.013$   | $3.289 \pm 0.481$     | $0.146 \pm 0.035$ | $0.628 \pm 0.014$ | $0.934 \pm 0.036$ | $8.634 \pm 0.923$  | $0.829 \pm 0.065$ | $0.279 \pm 0.021$ |
|                 |                   |                       |                   |                   |                   |                    |                   |                   |

| •••           |
|---------------|
| Ш             |
| $\geq$        |
| Budget        |
| Labeling      |
| Datasets with |
| Ŋ             |
| u             |
| Test RMSE     |
| 6             |

| Dataset    | DiCoM-N-2         | DiCoM-N-4         |                           | DiCoM-N-8         |                           |
|------------|-------------------|-------------------|---------------------------|-------------------|---------------------------|
|            | RMSE              | RMSE              | % Redc. $2 \rightarrow 4$ | RMSE              | % Redc. $4 \rightarrow 8$ |
| Skillcraft | $0.313 \pm 0.005$ | 0.319 ± 0.025     | - 1.917                   | $0.304 \pm 0.008$ | 4.702                     |
| Parkinsons | $2.285 \pm 0.208$ | $2.291 \pm 0.355$ | - 0.263                   | $2.200 \pm 0.219$ | 3.972                     |
| Elevators  | $0.145 \pm 0.025$ | $0.135 \pm 0.020$ | 7.187                     | $0.125 \pm 0.010$ | 7.407                     |
| Protein    | $0.646 \pm 0.031$ | $0.636 \pm 0.027$ | 1.548                     | $0.635 \pm 0.029$ | 0.157                     |
| Blog       | $0.930 \pm 0.040$ | $0.912 \pm 0.030$ | 1.935                     | $0.897 \pm 0.020$ | 1.645                     |
| Ctslice    | $5.575 \pm 0.606$ | $6.233 \pm 0.524$ | - 11.795                  | $6.174 \pm 0.587$ | 0.947                     |
| Buzz       | $0.715 \pm 0.136$ | $0.688 \pm 0.078$ | 3.804                     | $0.654 \pm 0.041$ | 4.942                     |
| Electric   | $0.114 \pm 0.025$ | $0.099 \pm 0.008$ | 13.132                    | $0.095 \pm 0.012$ | 4.058                     |
| Average    |                   |                   | 1.704                     |                   | 3.479                     |

Table 10 Test RMSE of DiCoM-N on UCI Datasets with Varying Number of Views, N = 300

![](_page_27_Figure_3.jpeg)

Fig. 8 Test RMSE on UCI datasets: each subplot shows the results for one dataset, N = 100

| Table 11 Test R | MSE on UCI datas  | ets with labeling budg | get $N = 100$     |                   |                   |                    |                   |                   |
|-----------------|-------------------|------------------------|-------------------|-------------------|-------------------|--------------------|-------------------|-------------------|
| Dataset         | skillcraft        | parkinsons             | elevators         | protein           | blog              | ctslice            | puzz              | electric          |
| SSDKL           | $0.360 \pm 0.029$ | $4.701 \pm 0.687$      | $0.182 \pm 0.019$ | $0.741 \pm 0.050$ | $1.129 \pm 0.081$ | $12.646 \pm 1.059$ | $0.904 \pm 0.117$ | $0.152 \pm 0.030$ |
| COREG           | $0.360 \pm 0.013$ | $5.552 \pm 0.358$      | $0.213 \pm 0.009$ | $0.751 \pm 0.015$ | $1.067 \pm 0.078$ | $12.911 \pm 0.983$ | $0.830 \pm 0.063$ | $0.352 \pm 0.020$ |
| LP              | $0.345 \pm 0.016$ | $7.037 \pm 1.373$      | $0.202 \pm 0.008$ | $0.746 \pm 0.022$ | $1.115 \pm 0.061$ | $13.726 \pm 1.178$ | $0.911 \pm 0.066$ | $0.484 \pm 0.032$ |
| VAE             | $0.412 \pm 0.020$ | $10.694 \pm 0.216$     | $0.251 \pm 0.029$ | $0.789 \pm 0.036$ | $1.051 \pm 0.102$ | $19.750 \pm 3.178$ | $1.114 \pm 0.036$ | $0.982 \pm 0.034$ |
| MT              | $0.447 \pm 0.039$ | $8.998 \pm 0.778$      | $0.269 \pm 0.039$ | $0.757 \pm 0.025$ | $1.112 \pm 0.053$ | $17.752 \pm 1.352$ | $1.786 \pm 0.125$ | $0.838 \pm 0.105$ |
| VAT             | $0.454 \pm 0.034$ | $10.738 \pm 0.254$     | $0.261 \pm 0.015$ | $0.784 \pm 0.050$ | $1.136 \pm 0.111$ | $18.318 \pm 3.720$ | $1.625 \pm 0.339$ | $0.937 \pm 0.071$ |
| ST-Tree         | $0.433 \pm 0.032$ | $9.387 \pm 0.368$      | $0.254 \pm 0.009$ | $0.851\pm0.030$   | $1.144 \pm 0.128$ | $30.681 \pm 0.487$ | $3.104 \pm 0.079$ | $1.356 \pm 0.017$ |
| SSL-PCT         | $0.461 \pm 0.039$ | $9.357 \pm 0.319$      | $0.256 \pm 0.011$ | $0.799 \pm 0.023$ | $1.132 \pm 0.055$ | $30.774 \pm 1.309$ | $3.095 \pm 0.121$ | $1.350 \pm 0.011$ |
| Supervised      | $0.564\pm0.110$   | $10.419 \pm 0.158$     | $0.429 \pm 0.131$ | $0.829 \pm 0.066$ | $1.072 \pm 0.047$ | $22.167 \pm 0.396$ | $2.034 \pm 0.051$ | $1.052 \pm 0.061$ |
| DiCoM-N-2       | $0.342 \pm 0.011$ | $3.580 \pm 0.662$      | $0.179 \pm 0.030$ | $0.691 \pm 0.021$ | $1.031 \pm 0.055$ | $10.742 \pm 0.629$ | $0.836 \pm 0.093$ | $0.185 \pm 0.116$ |
| DiCoM-B-2       | $0.330 \pm 0.021$ | $4.442 \pm 0.929$      | $0.194 \pm 0.029$ | $0.675 \pm 0.013$ | $0.998 \pm 0.052$ | $12.062 \pm 1.413$ | $0.974 \pm 0.059$ | $0.308 \pm 0.043$ |

| I        | I         |
|----------|-----------|
| 2        | 2         |
| hudaat   | Duugot    |
| labaling | Iaucillig |
| with     | M ILLI    |
| datacate | uatascis  |
| E LI     | 5         |
| 5        | 5         |
| DMCE.    | TOWN      |
| Tact     | TCOL      |
| 5        |           |
| 4        | μ         |

| Dataset | skillcraft | parkinsons | elevators | protein | blog   | ctslice | buzz   | electric |
|---------|------------|------------|-----------|---------|--------|---------|--------|----------|
| SSDKL   | 79.81      | 44.72      | 77.97     | 94.88   | 104.74 | 56.33   | 44.94  | 15.72    |
| COREG   | 79.85      | 52.81      | 91.37     | 96.17   | 98.99  | 57.50   | 41.25  | 36.55    |
| LP      | 76.64      | 66.93      | 86.44     | 95.61   | 103.50 | 61.13   | 45.29  | 50.21    |
| VAE     | 91.30      | 101.71     | 107.60    | 101.04  | 97.54  | 87.96   | 55.38  | 101.88   |
| MT      | 99.18      | 85.59      | 115.03    | 97.03   | 103.22 | 79.07   | 88.77  | 86.94    |
| VAT     | 100.66     | 102.14     | 111.67    | 100.43  | 105.39 | 81.59   | 80.80  | 97.21    |
| ST-Tree | 96.16      | 89.29      | 108.73    | 109.02  | 106.14 | 136.65  | 154.33 | 140.79   |
| SSL-PCT | 102.20     | 89.00      | 109.75    | 102.31  | 105.06 | 137.07  | 153.84 | 140.15   |

106.22

88.53

86.47

99.46

95.66

92.60

98.73

47.84

53.72

101.13

41.56

48.42

109.17

19.20

31.97

**Table 12** Relative test errors (RRMSE, in %) on UCI datasets with labeling budget N = 100

183.65

76.76

83.07

73.22 Lower value indicates better performance

125.17

75.95

Bold values indicate the best performance for each setup

99.10

34.05

42.25

individual views' losses do not, resulting in their large variance on test samples. This can be an issue if the DiCoM-N model contains a smaller number of views. On the other hand, the consistency enforcement is too strong on the right scenario (Fig. 12c). The individual views and the averaged output seem to have all collapsed into a single point. Where the models collapse to is limited by the individual networks' capacity and may not necessarily be the global optimum for the averaged output. Finally, in the middle scenario (Fig. 12b), the effects of diversity and consistency losses are balanced, yielding a good trade-off. The averaged model output is able to perform better than each individual view, and is also the best among three scenarios. These results also suggest that even though diversity and consistency are contradicting forces, they can still be applied simultaneously on the regression outputs to produce the desirable behaviours.

Supervised

DiCoM-N-2

DiCoM-B-2

| Table 13 Test | RMSE from abl      | ation study on U(  | CI datasets, $N = 1$ | 00                 |          |                    |          |                    |          |
|---------------|--------------------|--------------------|----------------------|--------------------|----------|--------------------|----------|--------------------|----------|
|               | DiCoM-N-2          | Ablation-1         |                      | Ablation-2         |          | Ablation-3         |          | Ablation-4         |          |
| Aug.          | >                  |                    |                      | >                  |          | >                  |          | >                  |          |
| Lab. Div.     | >                  | `                  |                      |                    |          | `                  |          | \$                 |          |
| Unlab. Csc.   | >                  | >                  |                      | >                  |          |                    |          |                    |          |
| Unlab. Div.   |                    |                    |                      |                    |          |                    |          | >                  |          |
|               | RMSE               | RMSE               | % Redc.              | RMSE               | % Redc.  | RMSE               | % Redc.  | RMSE               | % Redc.  |
| skillcraft    | $0.342 \pm 0.011$  | $0.354 \pm 0.030$  | - 3.404              | $0.361 \pm 0.032$  | - 5.372  | $0.355 \pm 0.031$  | - 3.756  | $0.363 \pm 0.032$  | - 6.124  |
| parkinsons    | $3.580\pm0.662$    | $4.044 \pm 0.623$  | - 12.969             | $3.688 \pm 0.674$  | - 3.039  | $3.674 \pm 0.326$  | - 2.643  | $3.516 \pm 0.374$  | 1.783    |
| elevators     | $0.179 \pm 0.030$  | $0.186\pm0.033$    | - 3.844              | $0.192 \pm 0.032$  | - 6.864  | $0.193 \pm 0.029$  | - 7.574  | $0.197 \pm 0.026$  | - 9.775  |
| protein       | $0.691 \pm 0.021$  | $0.705 \pm 0.023$  | - 1.959              | $0.711 \pm 0.013$  | - 2.818  | $0.723 \pm 0.040$  | - 4.603  | $0.714 \pm 0.023$  | - 3.268  |
| blog          | $1.031\pm0.055$    | $1.074 \pm 0.052$  | - 4.214              | $1.090 \pm 0.043$  | - 5.721  | $1.124 \pm 0.072$  | -9.030   | $1.117 \pm 0.081$  | - 8.295  |
| ctslice       | $10.742 \pm 0.629$ | $10.822 \pm 0.564$ | - 0.747              | $11.341 \pm 1.114$ | - 5.575  | $10.290 \pm 0.911$ | 4.209    | $10.673 \pm 1.070$ | 0.642    |
| buzz          | $0.836\pm0.093$    | $0.918\pm0.085$    | - 9.760              | $0.933 \pm 0.263$  | - 11.599 | $1.019 \pm 0.183$  | - 21.893 | $1.017 \pm 0.242$  | - 21.613 |
| electric      | $0.185\pm0.116$    | $0.176 \pm 0.103$  | 4.602                | $0.198 \pm 0.115$  | - 6.798  | $0.158 \pm 0.023$  | 14.693   | $0.193 \pm 0.104$  | - 4.460  |
| Average       |                    |                    | - 4.037              |                    | - 5.973  |                    | - 3.825  |                    | - 6.389  |

| >          |
|------------|
| asets, /   |
| dat        |
| nci        |
| uo /       |
| study      |
| olation    |
| rom al     |
| <b>MSE</b> |
| Test F     |
| 13         |
| e          |

| Dataset    | DiCoM-N-2         | DiCoM-N-4          |                           | DiCoM-N-8          |                           |
|------------|-------------------|--------------------|---------------------------|--------------------|---------------------------|
|            | RMSE              | RMSE               | % Redc. $2 \rightarrow 4$ | RMSE               | % Redc. $4 \rightarrow 8$ |
| skillcraft | $0.342 \pm 0.011$ | $0.349 \pm 0.022$  | - 1.977                   | $0.334 \pm 0.018$  | 4.250                     |
| parkinsons | $3.580 \pm 0.662$ | $3.343 \pm 0.394$  | 6.610                     | $3.250 \pm 0.323$  | 2.782                     |
| elevators  | $0.179 \pm 0.030$ | $0.171 \pm 0.026$  | 4.605                     | $0.169 \pm 0.028$  | 1.185                     |
| protein    | $0.691 \pm 0.021$ | $0.681 \pm 0.021$  | 1.456                     | $0.679 \pm 0.024$  | 0.294                     |
| blog       | $1.031 \pm 0.055$ | $0.975 \pm 0.038$  | 5.432                     | $0.961 \pm 0.053$  | 1.436                     |
| ctslice    | $10.742\pm0.629$  | $11.164 \pm 1.972$ | - 3.924                   | $10.457 \pm 1.109$ | 6.333                     |
| buzz       | $0.836 \pm 0.093$ | $0.832 \pm 0.083$  | 0.447                     | $0.796 \pm 0.052$  | 4.350                     |
| electric   | $0.185 \pm 0.116$ | $0.171 \pm 0.059$  | 7.650                     | $0.137 \pm 0.029$  | 20.096                    |
| Average    |                   |                    | 2.537                     |                    | 5.091                     |

Table 14 Test RMSE of DiCoM-N on UCI datasets with varying number of views, N = 100

Table 15 From DiCoM-N to DiCoM-B: Percentage reduction in test RMSE and execution time, N = 100

| Dataset    | Test RMSE |          |          | Execution tir | ne     |              |
|------------|-----------|----------|----------|---------------|--------|--------------|
|            | M = 2     | M = 4    | M = 8    | M = 2         | M = 4  | <i>M</i> = 8 |
| skillcraft | 3.503     | - 1.323  | - 10.419 | - 72.882      | 18.200 | 59.470       |
| parkinsons | - 24.102  | - 30.058 | - 57.663 | 28.085        | 28.085 | 65.291       |
| elevators  | - 8.069   | - 13.635 | - 20.037 | - 11.291      | 37.481 | 37.481       |
| protein    | 2.260     | - 5.401  | - 15.882 | - 57.697      | 20.049 | 63.618       |
| blog       | 3.219     | - 4.069  | - 11.128 | 23.903        | 23.812 | 56.539       |
| ctslice    | - 7.157   | - 6.599  | - 45.748 | 22.144        | 42.888 | 77.162       |
| buzz       | - 16.619  | - 9.784  | - 25.956 | 0.524         | 46.706 | 68.124       |
| Average    | - 6.709   | - 10.124 | - 26.690 | - 9.602       | 31.032 | 61.098       |

#### Table 16 Test results on ShanghaiTech

|         | Supervise | ed baseline | DNCL   |        | Co-regres | sion   | DiCoM-I | 3-4    |
|---------|-----------|-------------|--------|--------|-----------|--------|---------|--------|
|         | MAE       | RMSE        | MAE    | RMSE   | MAE       | RMSE   | MAE     | RMSE   |
| N = 30  | 297.45    | 551.68      | 250.91 | 360.66 | 224.63    | 344.46 | 163.41  | 227.42 |
| N = 120 | 132.05    | 210.82      | 119.48 | 186.80 | 128.57    | 206.90 | 114.31  | 181.91 |
| N = 210 | 107.76    | 165.91      | 104.00 | 165.97 | 103.20    | 165.04 | 101.90  | 155.49 |

| Table 17    | C-MAPSS remaining |
|-------------|-------------------|
| useful life | e subsets         |

| Dataset                     | FD001 | FD002 | FD003 | FD004 |
|-----------------------------|-------|-------|-------|-------|
| No. of training engines     | 100   | 260   | 100   | 249   |
| No. of training samples     | 17731 | 48558 | 21220 | 56815 |
| No. of test engines         | 100   | 259   | 100   | 248   |
| No. of test samples         | 100   | 259   | 100   | 248   |
| Longest lifespan (cycles)   | 362   | 378   | 512   | 128   |
| No. of operating conditions | 1     | 6     | 1     | 6     |
| No. of failure modes        | 1     | 1     | 2     | 2     |
|                             |       |       |       |       |

![](_page_32_Figure_1.jpeg)

Fig. 9 Test results on C-MAPSS FD002 dataset: a RMSE and b RUL-Score on log scale

![](_page_32_Figure_3.jpeg)

Fig. 10 Test results on C-MAPSS FD003 dataset: a RMSE and b RUL-Score on log scale

![](_page_32_Figure_5.jpeg)

Fig. 11 Test results on C-MAPSS FD004 dataset: a RMSE and b RUL-Score on log scale

![](_page_33_Figure_1.jpeg)

**Fig. 12** Experiment results on toy data. The top row shows training losses in symmetric log scale. The bottom row shows model predictions on eight random test samples. In the legend, next to the model name, we report the MSE scores evaluated on 1000 test samples

Acknowledgements This research is supported by the Agency for Science, Technology and Research (A\*STAR) under its AME Programmatic Funds (Grant No. A20H6b0151).

Author contributions Conceptualization: CN, XX, C-SF; Methodology: CN, AR, LZ; Experimentation and analysis: CN, AR, LZ, BU, MR, KL; Writing: CN, AR, ZL, XX, MR, C-S F; Funding acquisition: C-SF.

**Funding** This research is supported by the Agency for Science, Technology and Research (A\*STAR) under its AME Programmatic Funds (Grant No. A20H6b0151).

Data availability The data used in this work are publicly available.

Code availability Yes. We will seek approval from our organization to release the code.

#### Declarations

Conflict of interest Not applicable.

Ethics approval Not applicable.

**Consent to participate** All authors whose names appear on this manuscript agree with the content and give consent for submission.

Consent to publication All authors whose names appear on this manuscript give consent for publication.

#### References

- Berthelot, D., Carlini, N., Cubuk, E. D., Kurakin, A., Sohn, K., Zhang, H., & Raffel, C. (2019). Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. arXiv preprint arXiv:1911.09785
- Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In Proceedings of the eleventh annual conference on computational learning theory (pp. 92–100). ACM.
- Brefeld, U., Gärtner, T., Scheffer, T., & Wrobel, S. (2006). Efficient co-regularised least squares regression. In Proceedings of the 23rd international conference on machine learning (p. 137). ACM.
- Brown, G., Wyatt, J. L., Tino, P., & Bengio, Y. (2005). Managing diversity in regression ensembles. *Journal of Machine Learning Research*, 6(9), 1621–1650.
- Chapelle, O., & Zien, A. (2005). Semi-supervised classification by low density separation. In International workshop on artificial intelligence and statistics (pp. 57–64). PMLR
- Chapelle, O., Chi, M., & Zien, A. (2006). A continuation method for semi-supervised SVMs. In Proceedings of the 23rd international conference on machine learning (pp. 185–192).
- Chapelle, O., Scholkopf, B., & Zien, A. (2009). Semi-supervised learning (chapelle, O. et al., eds.; 2006) [book reviews]. *IEEE Transactions on Neural Networks*, 20(3), 542–542.
- Chen, H., Jiang, B., & Yao, X. (2018). Semisupervised negative correlation learning. *IEEE Transactions on Neural Networks and Learning Systems*, 29(11), 5366–5379.
- Chen, Z., Wu, M., Zhao, R., Guretno, F., Yan, R., & Li, X. (2020). Machine remaining useful life prediction via an attention-based deep learning approach. *IEEE Transactions on Industrial Electronics*, 68(3), 2521–2531.
- Deutsch, J., & He, D. (2017). Using deep learning-based approach to predict remaining useful life of rotating components. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48*(1), 11–20.
- Dietterich, T. G. (2000). Ensemble methods in machine learning. In International workshop on multiple classifier systems (pp. 1–15). Springer.
- Dua, D., Graff, C. (2017). UCI machine learning repository. http://archive.ics.uci.edu/ml
- Ellefsen, A. L., Bjørlykhaug, E., Æsøy, V., Ushakov, S., & Zhang, H. (2019). Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering & System Safety, 183,* 240–251.
- Gong, C. (2017). Exploring commonality and individuality for multi-modal curriculum learning. In Thirty-first AAAI conference on artificial intelligence.
- Heimes, F. O. (2008). Recurrent neural networks for remaining useful life estimation. In 2008 international conference on prognostics and health management (pp. 1–6). IEEE.
- Huang, C.-G., Huang, H.-Z., & Li, Y.-F. (2019). A bidirectional LSTM prognostics method under multiple operational conditions. *IEEE Transactions on Industrial Electronics*, 66(11), 8792–8802.
- Jean, N., Xie, S. M., & Ermon, S. (2018). Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance. In *Proceedings of the 32nd international conference* on neural information processing systems (pp. 5327–5338).
- Ke, Z., Wang, D., Yan, Q., Ren, J., & Lau, R. W. (2019). Dual student: Breaking the limits of the teacher in semi-supervised learning. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6728–6736).
- Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. Advances in Neural Information Processing Systems, 7, 231–238.
- Krokotsch, T., Knaak, M., & Gühmann, C. (2022). Improving semi-supervised learning for remaining useful lifetime estimation through self-supervision. *International Journal of Prognostics and Health Management*, 13(1).
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
- Levatić, J., Ceci, M., Kocev, D., & Džeroski, S. (2017). Self-training for multi-target regression with tree ensembles. *Knowledge-Based Systems*, 123, 41–60.
- Levatić, J., Kocev, D., Ceci, M., & Džeroski, S. (2018). Semi-supervised trees for multi-target regression. *Information Sciences*, 450, 109–127.
- Li, Y., Zhang, X., & Chen, D. (2018). Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1091–1100).
- Li, X., Ding, Q., & Sun, J.-Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172, 1–11.
- Liu, Y., & Yao, X. (1999). Ensemble learning via negative correlation. Neural Networks, 12(10), 1399–1404.

- Liu, Y., Yao, X., & Higuchi, T. (2000). Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation*, 4(4), 380–387.
- Miyato, T., Maeda, S.-I., Koyama, M., & Ishii, S. (2018). Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8), 1979–1993.
- Nguyen, C. M., Li, X., Blanton, R. D. S., & Li, X. (2019). Partial Bayesian co-training for virtual metrology. *IEEE Transactions on Industrial Informatics*, 16(5), 2937–2945.
- Oliver, A., Odena, A., Raffel, C. A., Cubuk, E. D., & Goodfellow, I. (2018). Realistic evaluation of deep semi-supervised learning algorithms. In Advances in neural information processing systems (pp. 3235–3246).
- Petersen, K. B., & Pedersen, M. S. (2008). The matrix cookbook. *Technical University of Denmark*, 7(15), 510.
- Qiao, S., Shen, W., Zhang, Z., Wang, B., & Yuille, A. (2018). Deep co-training for semi-supervised image recognition. In Proceedings of the European conference on computer vision (ECCV) (pp. 135–152).
- Ragab, M., Chen, Z., Wu, M., Foo, C. S., Kwoh, C. K., Yan, R., & Li, X. (2020). Contrastive adversarial domain adaptation for machine remaining useful life prediction. *IEEE Transactions on Industrial Informatics*, 17(8), 5239–5249.
- Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In 2008 international conference on prognostics and health management (pp. 1–9). IEEE
- Shi, Z., Zhang, L., Liu, Y., Cao, X., Ye, Y., Cheng, M.-M., & Zheng, G. (2018). Crowd counting with deep negative correlation learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5382–5390).
- Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., et al. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Advances in Neural Information Processing Systems, 33, 596–608.
- Stepišnik, T., & Kocev, D. (2021). Semi-supervised oblique predictive clustering trees. PeerJ Computer Science, 7, 506.
- Tang, E. K., Suganthan, P. N., & Yao, X. (2006). An analysis of diversity measures. *Machine Learning*, 65(1), 247–271.
- Tarvainen, A., & Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Advances in neural information processing systems (pp. 1195–1204).
- Ueda, N., & Nakano, R. (1996). Generalization error of ensemble estimators. In Proceedings of international conference on neural networks (ICNN'96) (Vol. 1, pp. 90–95). IEEE
- Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, 109(2), 373–440.
- Wilson, A. G., Hu, Z., Salakhutdinov, R., & Xing, E. P. (2016). Deep kernel learning. In Artificial intelligence and statistics (pp. 370–378). PMLR.
- Xie, S., Girshick, R., Dollar, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks.
- Xie, Q., Dai, Z., Hovy, E., Luong, T., & Le, Q. (2020). Unsupervised data augmentation for consistency training. Advances in Neural Information Processing Systems, 33, 6256–6268.
- Xu, C., Tao, D., & Xu, C. (2013). A survey on multi-view learning. arXiv preprint arXiv:1304.5634
- Yoon, A. S., Lee, T., Lim, Y., Jung, D., Kang, P., Kim, D., Park, K., & Choi, Y. (2017). Semi-supervised learning with deep generative models for asset failure prediction. arXiv preprint arXiv:1709.00845
- Yu, S., Krishnapuram, B., Rosales, R., & Rao, R. B. (2011). Bayesian co-training. Journal of Machine Learning Research, 12(Sep), 2649–2680.
- Zhang, Y., Zhou, D., Chen, S., Gao, S., & Ma, Y. (2016). Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 589–597).
- Zhang, L., Shi, Z., Cheng, M.-M., Liu, Y., Bian, J.-W., Zhou, J. T., et al. (2019). Nonlinear regression via deep negative correlation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3), 982–998.
- Zhang, Y., Wen, J., Wang, X., & Jiang, Z. (2014). Semi-supervised learning combining co-training with active learning. *Expert Systems with Applications*, 41(5), 2372–2378.
- Zhao, J., Xie, X., Xu, X., & Sun, S. (2017). Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, 38, 43–54.
- Zhou, Z.-H., & Li, M. (2005). Semi-supervised regression with co-training. In IJCAI (Vol. 5, pp. 908–913).

Zhu, J., Chen, N., & Peng, W. (2018). Estimation of bearing remaining useful life based on multiscale convolutional neural network. *IEEE Transactions on Industrial Electronics*, 66(4), 3208–3216.

Zhur, X., & Ghahramanirh, Z. (2002). Learning from labeled and unlabeled data with label propagation

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

# **Authors and Affiliations**

# Cuong Nguyen<sup>1</sup> · Arun Raja<sup>1,2</sup> · Le Zhang<sup>3</sup> · Xun Xu<sup>1</sup> · Balagopal Unnikrishnan<sup>4</sup> · Mohamed Ragab<sup>1</sup> · Kangkang Lu<sup>1</sup> · Chuan-Sheng Foo<sup>1,2</sup>

Cuong Nguyen nguyen\_manh\_cuong@i2r.a-star.edu.sg

Arun Raja arun@arunraja.dev

Le Zhang zhangleuestc@gmail.com

Xun Xu xu\_xun@i2r.a-star.edu.sg

Balagopal Unnikrishnan balagopal.unnikrishnan@mail.utoronto.ca

Mohamed Ragab mohamed\_adam@i2r.a-star.edu.sg

Kangkang Lu lu\_kangkang@i2r.a-star.edu.sg

- Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A\*STAR), 1 Fusionopolis Way, Connexis North Tower #20-10, Singapore 138632, Singapore
- <sup>2</sup> Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A\*STAR), 1 Fusionopolis Way, Connexis North Tower #16-16, Singapore 138632, Singapore
- <sup>3</sup> University of Electronic Science and Technology of China, 4 1st Ring Rd East 2 Section, Chengdu 610056, Sichuan, China
- <sup>4</sup> University of Toronto, 27 King's College Cir, Toronto, Ontario ON M5S 1A1, Canada