Diverse and Consistent Multi-view Networks for Semi-supervised Regression

Cuong Nguyen[1†], Arun Raja[1,2†], Le Zhang[3], Xun Xu[1], Balagopal Unnikrishnan[4], Mohamed Ragab[1], Kangkang Lu[1] and Chuan-Sheng Foo[1,2*]

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

*Corresponding author(s). E-mail(s): foo_chuan_sheng@i2r.a-star.edu.sg;
Contributing authors: nguyen_manh_cuong@i2r.a-star.edu.sg; arun@arunraja.dev; zhangleuestc@gmail.com; xu_xun@i2r.a-star.edu.sg; balagopal.unnikrishnan@mail.utoronto.ca; mohamed_ragab_mohamed_adam@i2r.a-star.edu.sg; lu_kangkang@i2r.a-star.edu.sg;
†These authors contributed equally to this work.

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 [1]. 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 learning methods [2] 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 [3], co-training [4] and entropy minimization [5]. Consistency-based approaches that have been popular in the classification setting, such as Mean Teacher [6] and Virtual Adversarial Training [7], which reinforce the output consistency of the network under input perturbations, have also been adapted to the regression setting [5]. However, enforcing consistency alone may not be sufficient for good performance, and may lead to model collapsing [8] or confirmation bias issues [9].

Another issue is that while the vast majority of semi-supervised learning research has been focused on semi-supervised classification [2], 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 [2]. This is because while some assumptions of semi-supervised learning such as the smoothness assumption or the manifold assumption [10] may hold for both classification and regression, other assumptions such as the cluster assumption [11] or the low-density separation assumption [12] do not apply to regression. Another issue with state-of-the-art classification methods such as FixMatch [13], UDA [14] or

ReMixMatch [15] 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 [16]. 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 [17, 18].

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 multi-view 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 [19]; or explored commonality and individuality in multi-modal curriculum learning [20], 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 semi-supervised regression. For regression tasks, graph-based methods are among the first to be developed. One example is Label Propagation (LP) [3] 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 Semi-supervised Deep Kernel Learning (SSDKL) [5]. 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) [21] 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) [22], Semi-supervised Predictive Clustering Trees (SSL-PCT) [23] and Semi-supervised Oblique Predictive Clustering Trees (SSL-SPYCT) [24]. 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) [6] 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) [7]. These methods were originally developed for classification, and were subsequently adapted to regression tasks [5]. However, both MT and VAT maintain only a single trainable network, which may lead to problems such as confirmation bias [9] 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 [25, 26]. For regression tasks, a commonly-used ensemble technique is Negative Correlation Learning (NCL) [17, 27], which formulates a diversity-promoting loss using an *ambiguity decomposition* of the squared ensemble loss [28]. 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 [25] that the strategy employed by NCL is equivalent to leveraging a *bias-variance-covariance trade-off* [29] of the ensemble error.

Recently, NCL has been extended to semi-supervised learning [30], 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) [18, 31], 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 [19]. To address this problem, multi-view learning assigns a modeling function to each view and jointly optimize these functions to improve overall generalization performance [32]. By analyzing the development of various techniques, Xu *et al.* [19] 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. Fig. 1 illustrates the intuition behind DiCoM in comparison with DNCL [33] and Co-Regression [34].

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 ground-truth 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 $\mathrm{F}_1, \mathrm{F}_2, \ldots, \mathrm{F}_M$, 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 inputs $x^m = \eta_m(x) \;\forall\, m = 1, \ldots, M$. With each augmented input, the corresponding neural network produces a regression output $f_m(x) = \mathrm{F}_m(x^m, \theta_m) \;\forall\, m = 1, \ldots, M$. Due to the different augmentations and network parameters, each output f_m can be

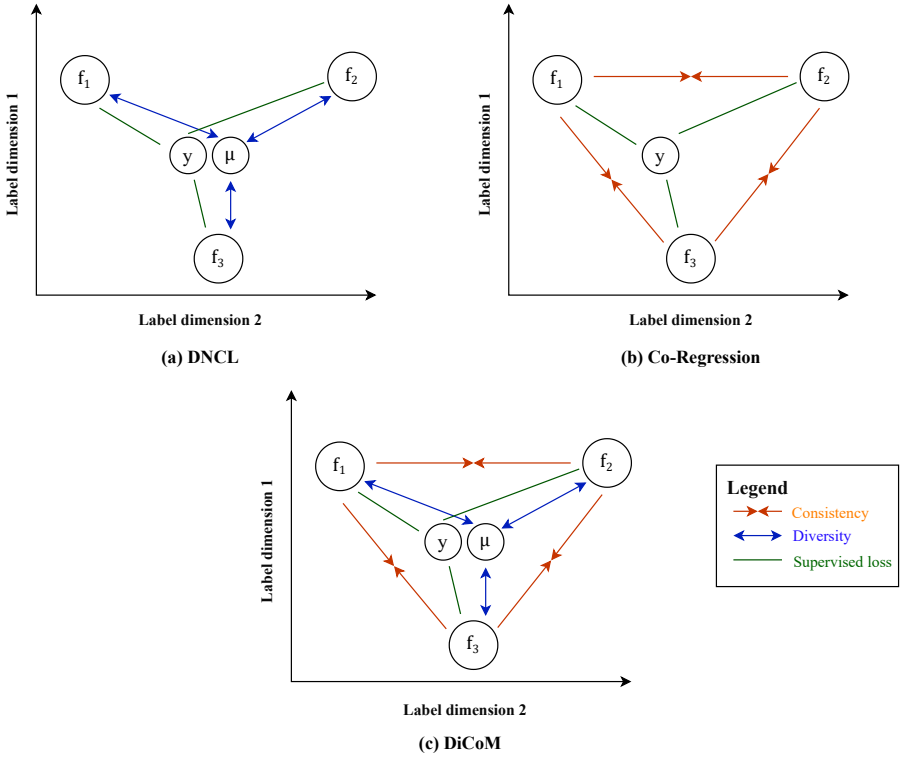

Fig. 1 The concepts of (a) DNCL, (b) Co-Regression and (c) DiCoM, visualized on a 2-D label space. While DNCL focuses on promoting ensemble diversity and Co-Regression enforces mutual agreement among the views, DiCoM enables both diversity and consistency simultaneously.

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. 2 (a)).

The second approach is to utilize a single network with a shared backbone B and multiple regressor branches $\mathrm{R}_1, \mathrm{R}_2, \ldots, \mathrm{R}_M$. We use θ_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 η_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. 2 (b)). Multi-branch technique was widely adopted for supervised classification [35]. 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 loss function $\mathfrak{L}_{\mathrm{DiCoM}}$. During training phase,

$\mathfrak{L}_{\text{DiCoM}}$ is back-propagated simultaneously through the deep views to optimize network parameters $\theta_1, \theta_2, \ldots, \theta_M$ (including θ_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 $\mathfrak{L}_{\text{DiCoM}}$ based on a probabilistic graphical assumption.

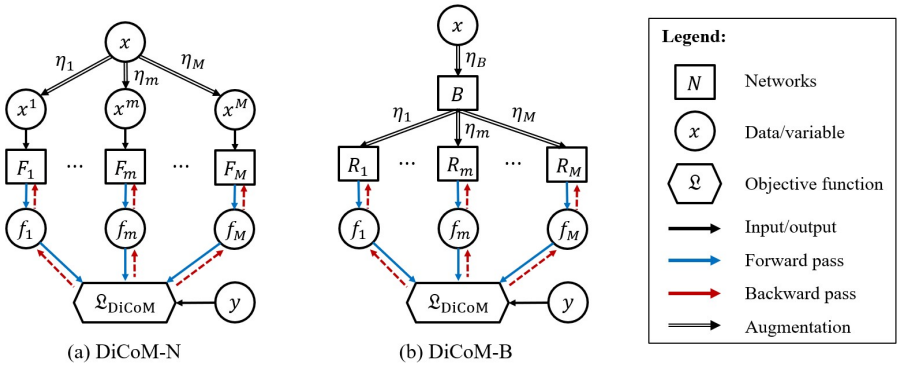

Fig. 2 The DiCoM framework with two variants: (a) DiCoM-N (multi-network) and (b) DiCoM-B (multi-branch).

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 [36] and linear regression [37], 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

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

This probabilistic relation is known as the *consensus potential* [36]. Considering the whole graph, this potential implies that all views are random Gaussian variables with a shared mean f_c and variance σ_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. 3 (a), is assumed for each unlabeled sample. The joint density associated with the graph is given by

$$p\left(f_c, f_1, \ldots, f_M\right) = \frac{1}{\mathcal{Z}} \prod_{m=1}^{M} \mathbf{\Psi}(f_c, f_m) \tag{2}$$

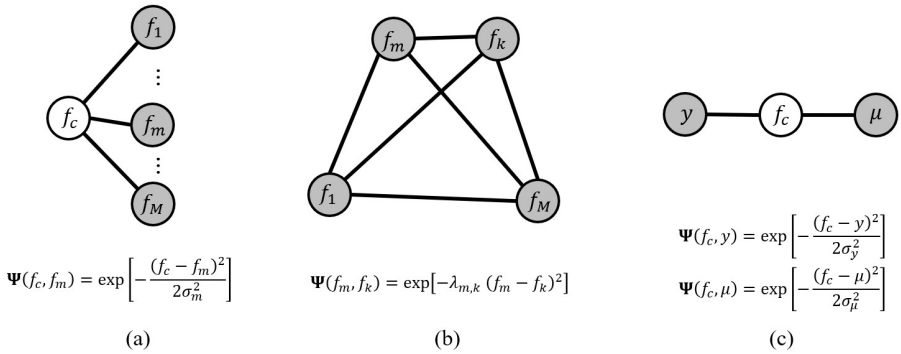

Fig. 3 Undirected probabilistic graphical models of DiCoM: (a) for an unlabeled sample, (b) after marginalization of the views and (c) for a labeled sample.

where $\mathcal{Z}$ is a normalizing constant and $\mathbf{\Psi}(f_c, f_m) = \exp\left[-\frac{(f_c - f_m)^2}{2\sigma_m^2}\right]$ 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.

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

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

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 $\binom{M}{2}$ 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. 3(b).

(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

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

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., $\sigma_m^2 = \sigma_{\mathrm{v}}^2 \quad \forall m$. For an unlabeled sample (x_n), we directly apply the first result **(I)** to obtain the following negative log

likelihood function

$$\mathrm{L}_{\mathrm{unl}} = \sum_{m=1}^{M} \sum_{k>m} \frac{1}{2M\sigma_{\mathrm{v}}^2} \left[f_m(x_n) - f_k(x_n)\right]^2 \tag{5}$$

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 μ. This graph is shown in Fig. 3(c). Since we assume a shared variance σ_{v}^2, the weighted output now reduces to an equal-weight average, following from result **(II)**

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

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

$$\mathrm{L}_{\mathrm{lab}} = \frac{M}{2M\sigma_y^2 + 2\sigma_{\mathrm{v}}^2} \left[y_n - \mu(x_n)\right]^2 \tag{7}$$

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

$$\approx \frac{1}{2\sigma_{\mathrm{v}}^2} \sum_{m=1}^{M} \left\{ \left[f_m(x_n) - y_n\right]^2 - \left[f_m(x_n) - \mu(x_n)\right]^2 \right\} \tag{9}$$

where in equation (8), we have applied the *ambiguity decomposition* [28] and in equation (9), we have assumed that the label is accurate, i.e., $\sigma_y^2 \ll \sigma_{\mathrm{v}}^2$.

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 equations (5) and (9)

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

where we introduce two hyperparameters κ_{div} and κ_{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 [30], 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.

4.1 Regression on Tabular Data

Datasets: We evaluate DiCoM on eight datasets from the UCI repository [38]: 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 [39] 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 [5, 40], 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 $(\kappa_{\text{div}}, \kappa_{\text{csc}})$ 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 [23], the formula for RRMSE is as follows:

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

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 [5], COREG [21], LP [3], VAE [5], MT [6], VAT [7], ST-Tree [22], SSL-PCT [23] 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: Fig. 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 $(\kappa_{\text{div}}, \kappa_{\text{csc}}) = (1, 0.01)$, while for ctslice, the most frequently selected values are $(\kappa_{\text{div}}, \kappa_{\text{csc}}) = (0.1, 1)$. 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.

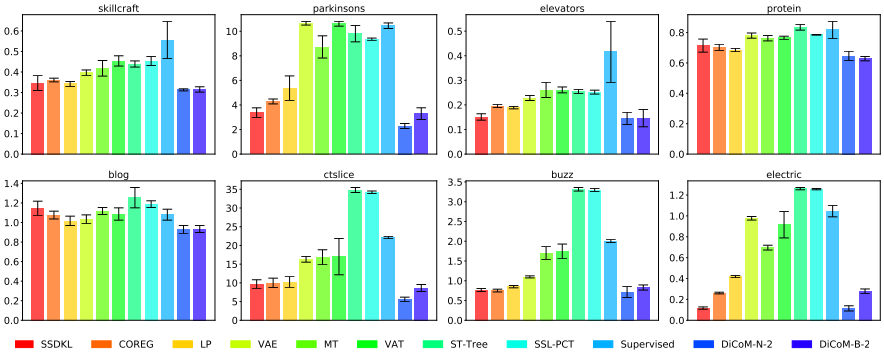

Fig. 4 Test RMSE on UCI datasets: each subplot shows the results for one dataset. Lower value indicates better performance. Bold face indicates the best performance for each setup.

Table 1 Relative Test Errors (RRMSE, in %) on UCI Datasets. 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

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 model [30], 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

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

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.

Table 2 Test RMSE from Ablation Study on UCI Datasets.

	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	**0.313 ± 0.005**	0.330 ± 0.017	-5.559	0.327 ± 0.008	-4.610	0.333 ± 0.012	-6.446	0.330 ± 0.011	-5.326
parkinsons	**2.285 ± 0.208**	2.437 ± 0.280	-6.666	2.390 ± 0.278	-4.593	2.563 ± 0.299	-12.159	2.438 ± 0.286	-6.692
elevators	0.145 ± 0.025	**0.142 ± 0.031**	2.082	0.149 ± 0.022	-2.328	0.155 ± 0.027	-6.225	0.157 ± 0.028	-7.936
protein	**0.646 ± 0.031**	0.679 ± 0.029	-5.153	0.654 ± 0.025	-1.313	0.669 ± 0.034	-3.546	0.661 ± 0.030	-2.327
blog	**0.930 ± 0.040**	1.014 ± 0.039	-8.998	1.021 ± 0.051	-9.752	1.027 ± 0.036	-10.455	1.007 ± 0.045	-8.261
ctslice	**5.575 ± 0.606**	6.976 ± 0.984	-25.120	7.154 ± 1.176	-28.306	7.461 ± 1.014	-33.813	7.398 ± 0.662	-32.699
buzz	**0.715 ± 0.136**	0.757 ± 0.064	-5.825	0.938 ± 0.489	-31.184	0.830 ± 0.131	-16.045	1.038 ± 0.550	-45.085
electric	0.114 ± 0.025	**0.097 ± 0.026**	14.658	0.153 ± 0.124	-33.833	0.107 ± 0.011	6.327	0.150 ± 0.117	-31.543
Average			-5.073		-14.490		-10.295		-17.484

Varying the number of views: We further evaluate the impact of the number of views M. Fig. 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 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 $\log_{10}(\kappa_{\text{div}}/\kappa_{\text{csc}})$. For most datasets, we see that this log ratio tends to increase for larger number of views. This is because in $\mathfrak{L}_{\text{DiCoM}}$, 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 $(\kappa_{\text{div}}/\kappa_{\text{csc}})$ ratio is required to keep those terms balanced.

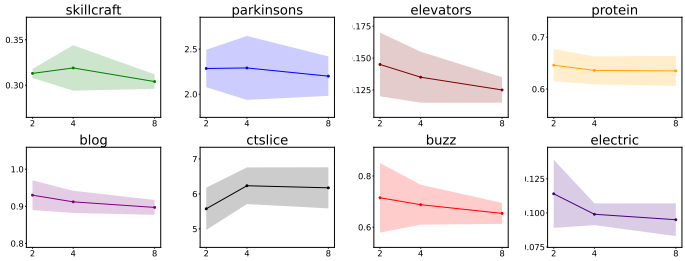

Fig. 5 Test RMSE of DiCoM-N on UCI datasets with varying number of views $M \in \{2, 4, 8\}$. The x-axis shows number of views M, the y-axis shows test RMSE.

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

	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

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 equation (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 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.

Table 4 From DiCoM-N to DiCoM-B: Percentage Reduction in Test RMSE and Execution Time.

Dataset	Test RMSE			Execution Time		
	$M = 2$	$M = 4$	$M = 8$	$M = 2$	$M = 4$	$M = 8$
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

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 [18]; for this task, counting by regression has been perceived as the state-of-the-art approach.

Dataset: We study the ShanghaiTech Part-A dataset [41]. 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 [42] 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 [18]; and (iii) the Co-Regression model [34]. 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 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.

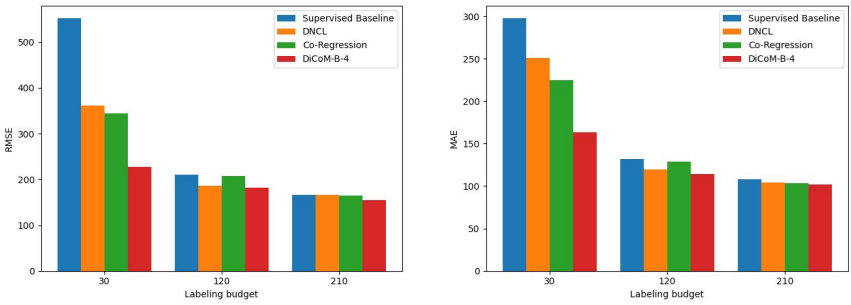

Fig. 6 Test results on ShanghaiTech dataset: (a) RMSE and (b) MAE.

Table 5 Relative Test Errors (RRMSE and RMAE, in %) on ShanghaiTech. Lower value indicates better performance. Bold face indicates the best performance for each setup.

	Supervised Baseline		DNCL		Co-Regression		DiCoM-B-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**
$N = 210$	26.14	30.54	25.23	30.55	25.03	30.38	**24.72**	**28.62**

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 [43]. Several deep learning architectures have been adopted for RUL task such as deep belief networks [43], convolutional neural networks [44, 45], and long-short-term memory (LSTM) [46, 47]. However, labeled data for RUL task can be only obtained after machine failure, which is costly and time-consuming, especially for complex machines [48]. 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 [49], 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 [48], 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 [50]. 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

$$\text{RUL-Score}_i = \begin{cases} e^{-\frac{\Delta\text{RUL}_i}{13}} - 1 & \text{for } \Delta\text{RUL}_i < 0 \\ e^{\frac{\Delta\text{RUL}_i}{10}} - 1 & \text{for } \Delta\text{RUL}_i \geq 0 \end{cases} \tag{13}$$

where ΔRUL_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 ΔRUL_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 [48]. 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) [51], Variational Autoencoder (VAE) [52] and Restricted Boltzmann Machines (RBM) [53].

Table 6 Test Results on C-MAPSS Datasets. Lower value indicates better performance. Bold face indicates the best performance for each setup.

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+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	**1.55E+04**	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

Table 7 Relative Test Errors (RRMSE, in %) on C-MAPSS Datasets. Lower value indicates better performance. Bold face indicates the best performance for each setup.

Budget	Subset	DiCoM-N-2	VAE	RBM	SelfSL	Supervised
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

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

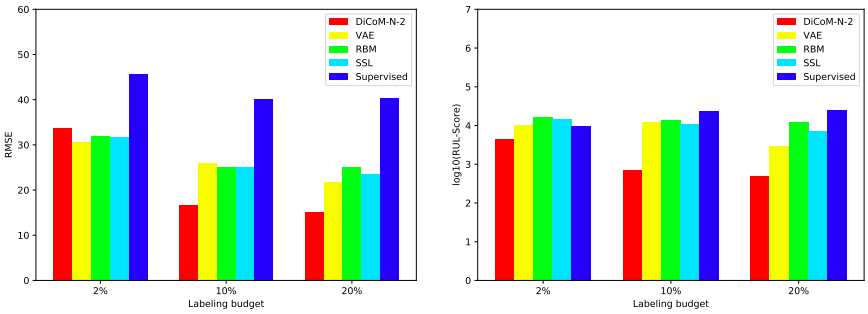

Fig. 7 Test results on C-MAPSS FD001 dataset: (a) RMSE and (b) RUL-Score on log scale. Lower value indicates better performance.

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 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%. Fig. 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 Semi-supervised 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 μ 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.

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

Declarations

- Funding: This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funds (Grant No. A20H6b0151).
- Conflict of interest/Competing interests: 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.
- Availability of data and materials: The data used in this work are publicly available.
- Code availability: Yes. We will seek approval from our organization to release the code.
- Authors' contributions: Conceptualization: Cuong Nguyen, Xun Xu, Chuan-Sheng Foo; Methodology: Cuong Nguyen, Arun Raja, Le Zhang; Experimentation and analysis: Cuong Nguyen, Arun Raja, Le Zhang, Balagopal Unnikrishnan, Mohamed Ragab, Kangkang Lu; Writing: Cuong Nguyen, Arun Raja, Zhang Le, Xun Xu, Mohamed Ragab, Chuan-Sheng Foo; Funding acquisition: Chuan-Sheng Foo.

References

[1] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)

[2] Van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Machine Learning **109**(2), 373–440 (2020)

[3] Zhur, X., Ghahramanirh, Z.: Learning from labeled and unlabeled data with label propagation (2002)

[4] Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pp. 92–100 (1998). ACM

[5] Jean, N., Xie, S.M., Ermon, S.: 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 (2018)

[6] Tarvainen, A., Valpola, H.: 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 (2017)

[7] Miyato, T., Maeda, S.-i., Koyama, M., Ishii, S.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE transactions on pattern analysis and machine intelligence (2018)

[8] Qiao, S., Shen, W., Zhang, Z., Wang, B., Yuille, A.: Deep co-training for semi-supervised image recognition. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 135–152 (2018)

[9] Ke, Z., Wang, D., Yan, Q., Ren, J., Lau, R.W.: 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 (2019)

[10] Chapelle, O., Scholkopf, B., Zien, A.: Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. IEEE Transactions on Neural Networks **20**(3), 542–542 (2009)

[11] Chapelle, O., Chi, M., Zien, A.: A continuation method for semi-supervised svms. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 185–192 (2006)

[12] Chapelle, O., Zien, A.: Semi-supervised classification by low density separation. In: International Workshop on Artificial Intelligence and Statistics, pp. 57–64 (2005). PMLR

[13] Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.-L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Advances in Neural Information Processing Systems **33**, 596–608 (2020)

[14] Xie, Q., Dai, Z., Hovy, E., Luong, T., Le, Q.: Unsupervised data augmentation for consistency training. Advances in Neural Information Processing Systems **33**, 6256–6268 (2020)

[15] Berthelot, D., Carlini, N., Cubuk, E.D., Kurakin, A., Sohn, K., Zhang, H., Raffel, C.: Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. arXiv preprint arXiv:1911.09785 (2019)

[16] Dietterich, T.G.: Ensemble methods in machine learning. In: International Workshop on Multiple Classifier Systems, pp. 1–15 (2000). Springer

[17] Liu, Y., Yao, X.: Ensemble learning via negative correlation. Neural networks **12**(10), 1399–1404 (1999)

[18] Zhang, L., Shi, Z., Cheng, M.-M., Liu, Y., Bian, J.-W., Zhou, J.T., Zheng, G., Zeng, Z.: Nonlinear regression via deep negative correlation learning.

IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)

[19] Xu, C., Tao, D., Xu, C.: A survey on multi-view learning. arXiv preprint arXiv:1304.5634 (2013)

[20] Gong, C.: Exploring commonality and individuality for multi-modal curriculum learning. In: Thirty-first AAAI Conference on Artificial Intelligence (2017)

[21] Zhou, Z.-H., Li, M.: Semi-supervised regression with co-training. In: IJCAI, vol. 5, pp. 908–913 (2005)

[22] Levatić, J., Ceci, M., Kocev, D., Džeroski, S.: Self-training for multi-target regression with tree ensembles. Knowledge-based systems **123**, 41–60 (2017)

[23] Levatić, J., Kocev, D., Ceci, M., Džeroski, S.: Semi-supervised trees for multi-target regression. Information Sciences **450**, 109–127 (2018)

[24] Stepišnik, T., Kocev, D.: Semi-supervised oblique predictive clustering trees. PeerJ Computer Science **7**, 506 (2021)

[25] Brown, G., Wyatt, J.L., Tino, P., Bengio, Y.: Managing diversity in regression ensembles. Journal of machine learning research **6**(9) (2005)

[26] Tang, E.K., Suganthan, P.N., Yao, X.: An analysis of diversity measures. Machine learning **65**(1), 247–271 (2006)

[27] Liu, Y., Yao, X., Higuchi, T.: Evolutionary ensembles with negative correlation learning. IEEE Transactions on Evolutionary Computation **4**(4), 380–387 (2000)

[28] Krogh, A., Vedelsby, J., *et al.*: Neural network ensembles, cross validation, and active learning. Advances in neural information processing systems **7**, 231–238 (1995)

[29] Ueda, N., Nakano, R.: Generalization error of ensemble estimators. In: Proceedings of International Conference on Neural Networks (ICNN'96), vol. 1, pp. 90–95 (1996). IEEE

[30] Chen, H., Jiang, B., Yao, X.: Semisupervised negative correlation learning. IEEE transactions on neural networks and learning systems **29**(11), 5366–5379 (2018)

[31] Shi, Z., Zhang, L., Liu, Y., Cao, X., Ye, Y., Cheng, M.-M., Zheng, G.: Crowd counting with deep negative correlation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5382–5390 (2018)

[32] Zhao, J., Xie, X., Xu, X., Sun, S.: Multi-view learning overview: Recent progress and new challenges. Information Fusion **38**, 43–54 (2017)

[33] Zhang, Y., Wen, J., Wang, X., Jiang, Z.: Semi-supervised learning combining co-training with active learning. Expert Systems with Applications **41**(5), 2372–2378 (2014)

[34] Brefeld, U., Gärtner, T., Scheffer, T., Wrobel, S.: Efficient co-regularised least squares regression. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 137–144 (2006). ACM

[35] Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated Residual Transformations for Deep Neural Networks (2017)

[36] Yu, S., Krishnapuram, B., Rosales, R., Rao, R.B.: Bayesian co-training. Journal of Machine Learning Research **12**(Sep), 2649–2680 (2011)

[37] Nguyen, C.M., Li, X., Blanton, R.D.S., Li, X.: Partial Bayesian co-training for virtual metrology. IEEE Transactions on Industrial Informatics **16**(5), 2937–2945 (2019)

[38] Dua, D., Graff, C.: UCI Machine Learning Repository (2017). http://archive.ics.uci.edu/ml

[39] Oliver, A., Odena, A., Raffel, C.A., Cubuk, E.D., Goodfellow, I.: Realistic evaluation of deep semi-supervised learning algorithms. In: Advances in Neural Information Processing Systems, pp. 3235–3246 (2018)

[40] Wilson, A.G., Hu, Z., Salakhutdinov, R., Xing, E.P.: Deep kernel learning. In: Artificial Intelligence and Statistics, pp. 370–378 (2016). PMLR

[41] Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: 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 (2016)

[42] Li, Y., Zhang, X., Chen, D.: 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 (2018)

[43] Deutsch, J., He, D.: 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 (2017)

[44] Li, X., Ding, Q., Sun, J.-Q.: Remaining useful life estimation in prognostics using deep convolution neural networks. Reliability Engineering &

System Safety **172**, 1–11 (2018)

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

[46] Huang, C.-G., Huang, H.-Z., Li, Y.-F.: A bidirectional lstm prognostics method under multiple operational conditions. IEEE Transactions on Industrial Electronics **66**(11), 8792–8802 (2019)

[47] Chen, Z., Wu, M., Zhao, R., Guretno, F., Yan, R., Li, X.: Machine remaining useful life prediction via an attention-based deep learning approach. IEEE Transactions on Industrial Electronics **68**(3), 2521–2531 (2020)

[48] Ragab, M., Chen, Z., Wu, M., Foo, C.S., Kwoh, C.K., Yan, R., Li, X.: Contrastive adversarial domain adaptation for machine remaining useful life prediction. IEEE Transactions on Industrial Informatics **17**(8), 5239–5249 (2020)

[49] Saxena, A., Goebel, K., Simon, D., Eklund, N.: Damage propagation modeling for aircraft engine run-to-failure simulation. In: 2008 International Conference on Prognostics and Health Management, pp. 1–9 (2008). IEEE

[50] Heimes, F.O.: Recurrent neural networks for remaining useful life estimation. In: 2008 International Conference on Prognostics and Health Management, pp. 1–6 (2008). IEEE

[51] Krokotsch, T., Knaak, M., et al.: Improving semi-supervised learning for remaining useful lifetime estimation through self-supervision. International Journal of Prognostics and Health Management **13**(1) (2022)

[52] Yoon, A.S., Lee, T., Lim, Y., Jung, D., Kang, P., Kim, D., Park, K., Choi, Y.: Semi-supervised learning with deep generative models for asset failure prediction. arXiv preprint arXiv:1709.00845 (2017)

[53] Ellefsen, A.L., Bjørlykhaug, E., Æsøy, V., Ushakov, S., Zhang, H.: Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliability Engineering & System Safety **183**, 240–251 (2019)

[54] Petersen, K.B., Pedersen, M.S., *et al.*: The matrix cookbook. Technical University of Denmark **7**(15), 510 (2008)

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

$$\mathbf{f}_c - \mathbf{f}_m \propto \mathcal{N}\left(\mathbf{0}, \sigma_m^2 \mathbf{I}\right). \tag{A1}$$

We note that in this general case, each deep view is a vector (instead of a scalar) and is assigned a separate variance σ_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

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

$$= \frac{1}{\mathcal{Z}_1} \exp\left(-\sum_m \frac{\mathbf{f}_c^\top \mathbf{f}_c}{2\sigma_m^2} + \sum_m \frac{\mathbf{f}_c^\top \mathbf{f}_m}{\sigma_m^2} - \sum_m \frac{\mathbf{f}_m^\top \mathbf{f}_m}{2\sigma_m^2}\right) \tag{A3}$$

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

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

$$\psi = \sum_{m=1}^{M} \frac{1}{\sigma_m^2} \qquad \boldsymbol{\phi} = \sum_{m=1}^{M} \frac{\mathbf{f}_m}{\sigma_m^2} \qquad \chi = \sum_{m=1}^{M} -\frac{\mathbf{f}_m^\top \mathbf{f}_m}{2\sigma_m^2}. \tag{A5}$$

Notice that $\psi, \boldsymbol{\phi}, \chi$ are constants w.r.t. $\mathbf{f}_c$. By applying the following integration rule for a multivariate Gaussian variable $\mathbf{x}$ [54]

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

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

$$p\left(\mathbf{f}_1, \ldots, \mathbf{f}_M\right) = \int p\left(\mathbf{f}_c, \mathbf{f}_1, \ldots, \mathbf{f}_M\right) d\mathbf{f}_c \tag{A7}$$

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

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

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

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

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

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

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

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

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

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

$$= \frac{1}{\mathcal{Z}_3} \exp\left(\sum_{m=1}^M \frac{-\|\mathbf{f}_c - \mathbf{f}_m\|^2}{2\sigma_m^2} + \sum_{m=1}^M \sum_{k>m}^M \frac{\|\mathbf{f}_m - \mathbf{f}_k\|^2}{2\rho_{m,k}^2}\right) \tag{A16}$$

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

$$= \frac{1}{\mathcal{Z}_3} \exp\left(\frac{-\mathbf{f}_c^\top \mathbf{f}_c}{2\sigma_\mu^2} + \frac{\tilde{\mu}^\top \mathbf{f}_c}{\sigma_\mu^2} + \aleph\right), \tag{A18}$$

where the normalizing factor $\mathcal{Z}_3$ is a constant w.r.t. $\mathbf{f}_c, \mathbf{f}_1, \ldots, \mathbf{f}_M$ and

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

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

$$\aleph = \sum_{m=1}^{M} \frac{-\mathbf{f}_m^\top \mathbf{f}_m}{2\sigma_m^2} + \sum_{m=1}^{M} \sum_{k>m}^{M} \frac{\|\mathbf{f}_m - \mathbf{f}_k\|^2}{2\rho_{m,k}^2}. \tag{A21}$$

Using the definitions (A13), (A19) and (A20), $\aleph$ can be rewritten as follows

$$\aleph = \sigma_\mu^2 \left(\sum_m \frac{-\mathbf{f}_m^\top \mathbf{f}_m}{2\sigma_m^2 \sigma_\mu^2} + \sum_m \sum_{k>m} \frac{\|\mathbf{f}_m - \mathbf{f}_k\|^2}{2\sigma_m^2 \sigma_k^2} \right) \tag{A22}$$

$$= \sigma_\mu^2 \left[\left(\sum_m \frac{-\mathbf{f}_m^\top \mathbf{f}_m}{2\sigma_m^2} \right) \left(\sum_k \frac{1}{\sigma_k^2} \right) + \sum_m \sum_{k>m} \frac{\|\mathbf{f}_m - \mathbf{f}_k\|^2}{2\sigma_m^2 \sigma_k^2} \right] \tag{A23}$$

$$= \sigma_\mu^2 \left(\sum_m \sum_k \frac{-\mathbf{f}_m^\top \mathbf{f}_m}{2\sigma_m^2 \sigma_k^2} + \sum_m \sum_{k \neq m} \frac{\mathbf{f}_m^\top \mathbf{f}_m}{2\sigma_m^2 \sigma_k^2} + \sum_m \sum_{k>m} \frac{-\mathbf{f}_m^\top \mathbf{f}_k}{\sigma_m^2 \sigma_k^2} \right) \tag{A24}$$

$$= \sigma_\mu^2 \left(\sum_m \frac{-\mathbf{f}_m^\top \mathbf{f}_m}{2\sigma_m^4} + \sum_m \sum_{k>m} \frac{-\mathbf{f}_m^\top \mathbf{f}_k}{\sigma_m^2 \sigma_k^2} \right) \tag{A25}$$

$$= -\frac{\sigma_\mu^2}{2} \left(\sum_m \frac{\mathbf{f}_m^\top}{\sigma_m^2} \right) \left(\sum_m \frac{\mathbf{f}_m}{\sigma_m^2} \right) \tag{A26}$$

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

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

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

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

Therefore,

$$\mathbf{f}_c \mid \mathbf{f}_1, \ldots, \mathbf{f}_M \sim \mathcal{N}\left(\tilde{\mu}, \sigma_\mu^2 \mathbf{I}\right). \tag{A30}$$

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: $20, 40, \ldots, 200$ (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^{-9}. Learning rate is 10^{-4} for ctslice and is 10^{-3} 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: $\kappa_{\text{div}} \in \{0.01, 0.05, 0.1, 0.5, 1\}$.
- Consistency hyperparameter search range: $\kappa_{\text{csc}} \in \{0.01, 0.05, 0.1, 0.5, 1\}$.

Experiment setup for DiCoM in ShanghaiTech experiment:

- Network architecture: CSRNet B [42].
- 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: $\kappa_{\text{div}} \in \{10^{-5}, 10^{-4}, 10^{-3}\}$.
- Consistency hyperparameter search range: $\kappa_{\text{csc}} \in \{10^{-5}, 10^{-4}, 10^{-3}\}$.

Experiment setup for DiCoM in C-MAPSS experiments:

- Network architecture: bi-directional LSTM [48].
- 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: $\kappa_{\text{div}} \in \{10^{-5}, 10^{-2}, 10^{-1}\}$.
- Consistency hyperparameter search range: $\kappa_{\text{csc}} \in \{10^{-5}, 10^{-2}, 10^{-1}\}$.

Experiment setup for competing methods in UCI experiments:

We follow the setup in [5] 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 [22, 23]. 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: $20, 40, \ldots, 200$ (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: $20, 40, \ldots, 200$ (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 σ^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: $20, 40, \ldots, 200$ (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: $20, 40, \ldots, 200$ (10 seeds in total).
  - Training: 2000 epochs.
  - Optimizer: Stochastic Gradient Descent with momentum 0.95. Learning rate is 10^{-3}.
  - 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: $20, 40, \ldots, 200$ (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 [18]. Publicly available at: https://github.com/shizenglin/Deep-NCL. The setup details for each method are as follows:

- For DNCL:
  - Network architecture: CSRNet B [42].
  - 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 [42].
  - 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 [51]. 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^{-4}.
- 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^{-4}.
- 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^{-4}.

Appendix C Additional Experiment Results

Information of UCI datasets: The detailed statistics of the eight UCI datasets are given in Table C1.

Table C1 UCI Regression Datasets

Dataset	No. of Samples	Input Dim.	No. of Unique Label Values	Prediction Target
skillcraft	3,325	18	7	Skill level of gamers (ordinal classification)
parkinsons	5,875	20	1,129	Unified Parkinson's Disease Rating Scale (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	8,123	Popularity of a topic in social media
electric	2,049,280	6	4,186	Power consumption in one household per minute

Additional experiment results on UCI datasets with labeling budget $N = 300$: Please see additional results from Section 4.1 in Table C2 and Table C3.

Table C2 Test RMSE on UCI Datasets with Labeling Budget $N = 300$.

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

Table C3 Test RMSE of DiCoM-N on UCI Datasets with Varying Number of Views, $N = 300$.

Dataset	DiCoM-N-2	DiCoM-N-4		DiCoM-N-8	
	RMSE	RMSE	% Redc. $2 \to 4$	RMSE	% Redc. $4 \to 8$
skillcraft	0.313 ± 0.005	0.319 ± 0.025	-1.917	**0.304 ± 0.008**	4.702
parkinsons	2.285 ± 0.208	2.291 ± 0.355	-0.263	**2.200 ± 0.219**	3.972
elevators	0.145 ± 0.025	0.135 ± 0.020	7.187	**0.125 ± 0.010**	7.407
protein	0.646 ± 0.031	0.636 ± 0.027	1.548	**0.635 ± 0.029**	0.157
blog	0.930 ± 0.040	0.912 ± 0.030	1.935	**0.897 ± 0.020**	1.645
ctslice	**5.575 ± 0.606**	6.233 ± 0.524	-11.795	6.174 ± 0.587	0.947
buzz	0.715 ± 0.136	0.688 ± 0.078	3.804	**0.654 ± 0.041**	4.942
electric	0.114 ± 0.025	0.099 ± 0.008	13.132	**0.095 ± 0.012**	4.058
Average			1.704		3.479

Additional experiment results on UCI datasets with labeling budget $N = 100$: Using the same UCI datasets, we conduct experiments similar to the ones in Section 4.1 with a smaller labeling budget of $N = 100$ samples. The results are provided in Fig. C1, Table C4, Table C5, Table C6, Table C7 and Table C8.

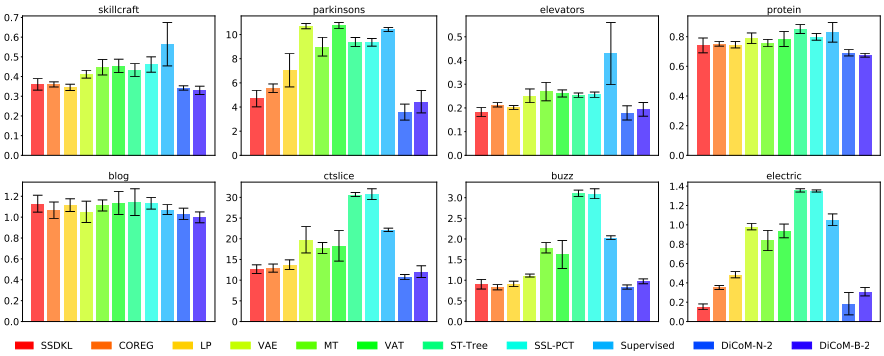

Fig. C1 Test RMSE on UCI datasets: each subplot shows the results for one dataset, $N = 100$.

Additional experiment results on ShanghaiTech dataset: The numerical results accompanying Fig. 6 are provided in Table C9.

Table C4 Test RMSE on UCI Datasets with Labeling Budget $N = 100$.

Dataset	skillcraft	parkinsons	elevators	protein	blog	ctslice	buzz	electric
SSDKL	0.360 ± 0.029	4.701 ± 0.687	0.182 ± 0.019	0.741 ± 0.050	1.129 ± 0.081	12.646 ± 1.059	0.904 ± 0.117	**0.152 ± 0.030**
COREG	0.360 ± 0.013	5.552 ± 0.358	0.213 ± 0.009	0.751 ± 0.015	1.067 ± 0.078	12.911 ± 0.983	**0.830 ± 0.063**	0.352 ± 0.020
LP	0.345 ± 0.016	7.037 ± 1.373	0.202 ± 0.008	0.746 ± 0.022	1.115 ± 0.061	13.726 ± 1.178	0.911 ± 0.066	0.484 ± 0.032
VAE	0.412 ± 0.020	10.694 ± 0.216	0.251 ± 0.029	0.789 ± 0.036	1.051 ± 0.102	19.750 ± 3.178	1.114 ± 0.036	0.982 ± 0.034
MT	0.447 ± 0.039	8.998 ± 0.778	0.269 ± 0.039	0.757 ± 0.025	1.112 ± 0.053	17.752 ± 1.352	1.786 ± 0.125	0.838 ± 0.105
VAT	0.454 ± 0.034	10.738 ± 0.254	0.261 ± 0.015	0.784 ± 0.050	1.136 ± 0.111	18.318 ± 3.720	1.625 ± 0.339	0.937 ± 0.071
ST-Tree	0.433 ± 0.032	9.387 ± 0.368	0.254 ± 0.009	0.851 ± 0.030	1.144 ± 0.128	30.681 ± 0.487	3.104 ± 0.079	1.356 ± 0.017
SSL-PCT	0.461 ± 0.039	9.357 ± 0.319	0.256 ± 0.011	0.799 ± 0.023	1.132 ± 0.055	30.774 ± 1.309	3.095 ± 0.121	1.350 ± 0.011
Supervised	0.564 ± 0.110	10.419 ± 0.158	0.429 ± 0.131	0.829 ± 0.066	1.072 ± 0.047	22.167 ± 0.396	2.034 ± 0.051	1.052 ± 0.061
DiCoM-N-2	0.342 ± 0.011	**3.580 ± 0.662**	**0.179 ± 0.030**	0.691 ± 0.021	1.031 ± 0.055	**10.742 ± 0.629**	0.836 ± 0.093	0.185 ± 0.116
DiCoM-B-2	**0.330 ± 0.021**	4.442 ± 0.929	0.194 ± 0.029	**0.675 ± 0.013**	**0.998 ± 0.052**	12.062 ± 1.413	0.974 ± 0.059	0.308 ± 0.043

Table C5 Relative Test Errors (RRMSE, in %) on UCI Datasets with Labeling Budget $N = 100$. Lower value indicates better performance.

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
Supervised	125.17	99.10	183.65	106.22	99.46	98.73	101.13	109.17
DiCoM-N-2	75.95	**34.05**	**76.76**	88.53	95.66	**47.84**	41.56	19.20
DiCoM-B-2	**73.22**	42.25	83.07	**86.47**	**92.60**	53.72	48.42	31.97

Table C6 Test RMSE from ablation study on UCI datasets, $N = 100$.

	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 ± 0.011**	0.354 ± 0.030	-3.404	0.361 ± 0.032	-5.372	0.355 ± 0.031	-3.756	0.363 ± 0.032	-6.124
parkinsons	**3.580 ± 0.662**	4.044 ± 0.623	-12.969	3.688 ± 0.674	-3.039	3.674 ± 0.326	-2.643	3.516 ± 0.374	1.783
elevators	**0.179 ± 0.030**	0.186 ± 0.033	-3.844	0.192 ± 0.032	-6.864	0.193 ± 0.029	-7.574	0.197 ± 0.026	-9.775
protein	**0.691 ± 0.021**	0.705 ± 0.023	-1.959	0.711 ± 0.013	-2.818	0.723 ± 0.040	-4.603	0.714 ± 0.023	-3.268
blog	**1.031 ± 0.055**	1.074 ± 0.052	-4.214	1.090 ± 0.043	-5.721	1.124 ± 0.072	-9.030	1.117 ± 0.081	-8.295
ctslice	10.742 ± 0.629	10.822 ± 0.564	-0.747	11.341 ± 1.114	-5.575	**10.290 ± 0.911**	4.209	10.673 ± 1.070	0.642
buzz	**0.836 ± 0.093**	0.918 ± 0.085	-9.760	0.933 ± 0.263	-11.599	1.019 ± 0.183	-21.893	1.017 ± 0.242	-21.613
electric	0.185 ± 0.116	0.176 ± 0.103	4.602	0.198 ± 0.115	-6.798	**0.158 ± 0.023**	14.693	0.193 ± 0.104	-4.460
Average			-4.037		-5.973		-3.825		-6.389

Table C7 Test RMSE of DiCoM-N on UCI Datasets with Varying Number of Views, $N = 100$.

Dataset	DiCoM-N-2	DiCoM-N-4		DiCoM-N-8	
	RMSE	RMSE	% Redc. $2 \to 4$	RMSE	% Redc. $4 \to 8$
skillcraft	0.342 ± 0.011	0.349 ± 0.022	-1.977	**0.334 ± 0.018**	4.250
parkinsons	3.580 ± 0.662	3.343 ± 0.394	6.610	**3.250 ± 0.323**	2.782
elevators	0.179 ± 0.030	0.171 ± 0.026	4.605	**0.169 ± 0.028**	1.185
protein	0.691 ± 0.021	0.681 ± 0.021	1.456	**0.679 ± 0.024**	0.294
blog	1.031 ± 0.055	0.975 ± 0.038	5.432	**0.961 ± 0.053**	1.436
ctslice	10.742 ± 0.629	11.164 ± 1.972	-3.924	**10.457 ± 1.109**	6.333
buzz	0.836 ± 0.093	0.832 ± 0.083	0.447	**0.796 ± 0.052**	4.350
electric	0.185 ± 0.116	0.171 ± 0.059	7.650	**0.137 ± 0.029**	20.096
Average			2.537		5.091

Table C8 From DiCoM-N to DiCoM-B: Percentage Reduction in Test RMSE and Execution Time, $N = 100$.

Dataset	Test RMSE			Execution Time		
	$M = 2$	$M = 4$	$M = 8$	$M = 2$	$M = 4$	$M = 8$
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 C9 Test Results on ShanghaiTech.

	Supervised Baseline		DNCL		Co-Regression		DiCoM-B-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**

Information of C-MAPSS dataset: The detailed statistics of the four C-MAPSS subsets are given in Table C10.

Table C10 C-MAPSS Remaining Useful Life 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

Additional experiment results on C-MAPSS datasets: For subsets FD002, FD003 and FD004, the plots of test results are provided in Fig.C2, Fig. C3 and Fig. C4.

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

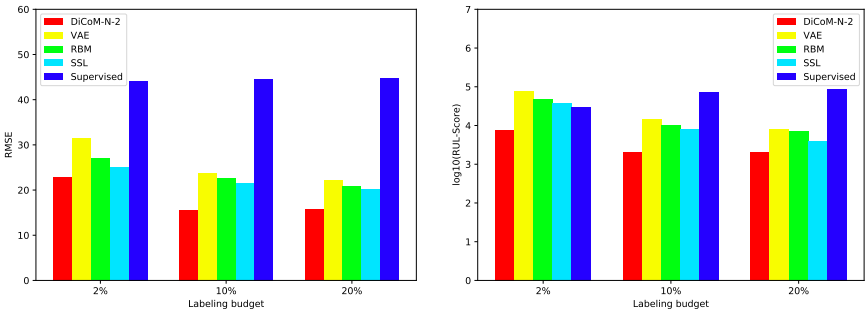

Fig. C2 Test results on C-MAPSS FD002 dataset: (a) RMSE and (b) RUL-Score on log scale.

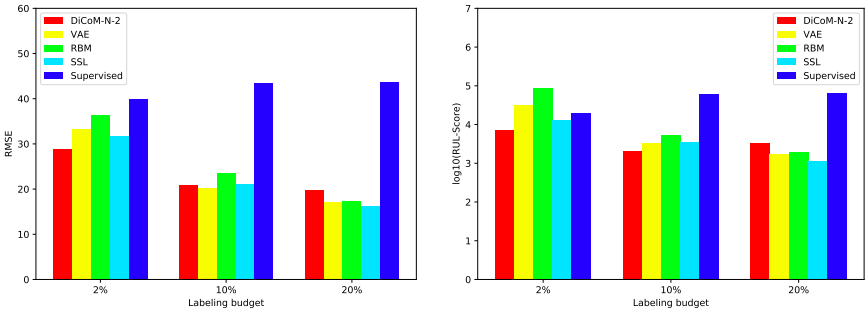

Fig. C3 Test results on C-MAPSS FD003 dataset: (a) RMSE and (b) RUL-Score on log scale.

is a fixed 2×30 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 5×10^{-2}, then evaluate the mean-squared-error (MSE) of the model on the test set.

Results: We keep $\kappa_{\text{div}} = 1$ and vary κ_{csc} on a log scale: $\kappa_{\text{csc}} \in \{0.01, 0.1, 1\}$. Both quantitative and qualitative results are shown in Fig. D5. 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. D5(a)) when $\kappa_{\text{div}} \gg \kappa_{\text{csc}}$, the diversity loss dominates the consistency component. Even though the total loss converges on the training set, the 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. D5(c)). The individual views and the averaged output seem to have all collapsed

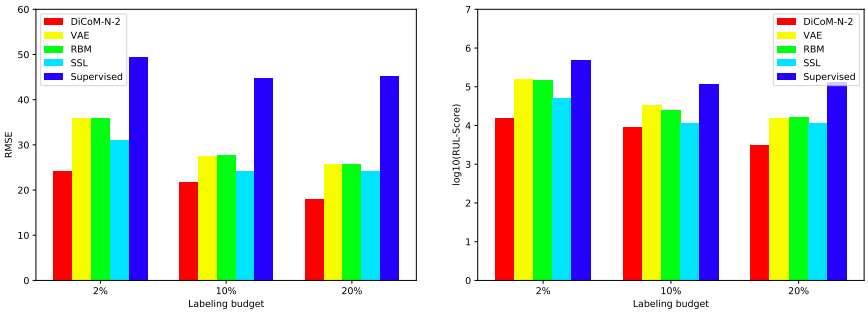

Fig. C4 Test results on C-MAPSS FD004 dataset: (a) RMSE and (b) RUL-Score on log scale.

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. D5(b)), 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.

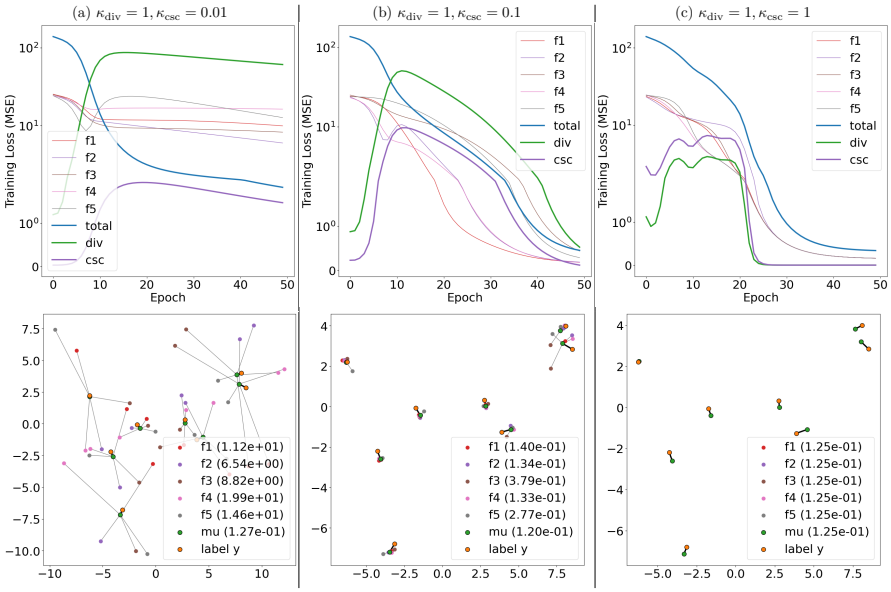

Fig. D5 Experiment results on toy data. The top row shows training losses in symmetric log scale. The bottom row shows model predictions on eight random test samples. In the legend, next to the model name, we report the MSE scores evaluated on 1000 test samples.