2024 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 22–25, 2024, LONDON, UK

2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP) | 979-8-3503-7225-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/MLSP58920.2024.10734769

ENSEMBLES OF INFORMATIVE REPRESENTATIONS FOR SELF-SUPERVISED LEARNING

Konstantinos D. Polyzos*, *Panagiotis A. Traganitis*†, *Manish K. Singh*$^\#$, *and Georgios B. Giannakis**

* Dept. of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA
†Dept. of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA
# Dept. of Electrical and Computer Engineering, University of Wisconsin–Madison, Madison, WI, USA

ABSTRACT

The requirement of large-size labeled training datasets often prohibits the deployment of supervised learning models in several applications with high acquisition costs and privacy concerns. To alleviate the burden of obtaining labels, self-supervised learning aims to identify informative data representations using auxiliary tasks that do not require external labels. The representations serve as refined inputs for the main learning task aimed at improving sample efficiency. Nonetheless, selecting individual auxiliary tasks and combining the corresponding extracted representations constitutes a nontrivial design problem. Agnostic of the approach for extracting individual representations per auxiliary task, this paper develops a weighted ensemble approach for obtaining a unified representation. The weights signify the relative dominance of individual representations in informing predictions for the main task. The representation ensemble is further augmented with the input data to improve accuracy and avoid information loss concerns. Numerical tests on real datasets showcase the merits of the advocated approach.

Index Terms— Self-supervised learning, representation learning, ensemble learning, Gaussian processes

1. INTRODUCTION

Modern machine learning has shown impressive performance for a large variety of tasks and has found its way into our daily lives. This success is largely attributed to ever-growing models, such as deep neural networks, that are trained on millions of data points. While unlabeled data are relatively easy to find, training large models typically requires significant amounts of *labeled* data. However, curating and labeling large datasets is a labor-intensive, time-consuming, and expensive process. Thus, methods that reduce the need for labeled data while simultaneously achieving good performance are well-motivated.

One way to reduce the requirement for labeled data is through a class of methods termed *transfer learning* (TL) [20]. These methods alleviate the need for a massive labeled dataset for a learning task of interest, by transferring knowledge acquired from a different, yet related, “source” or “auxiliary” task. TL has proven successful in a gamut of applications, such as speech recognition, imaging, automated medical diagnosis, activity recognition, audio transcription, and power networks [12, 27, 8, 15, 16, 24] to name a few.

Recently, a new form of TL, termed self-supervised learning (SSL) has gained traction [13, 1, 5]. The key difference between SSL and conventional TL is that in SSL the auxiliary task, also known as the pretext task, is created using the unlabeled data of the task of interest. Typically, knowledge transfer is achieved by extracting an informative representation from the auxiliary task, that can potentially simplify the main task [11, 21, 17, 18]. The success of SSL hinges on designing a “good” auxiliary task, that can improve the performance of the main learning task [9]. The importance of the auxiliary task is further elucidated in [28, 25], where it has been shown that if the extracted representation from the auxiliary task fails to capture useful information, then SSL can have adverse effects on the overall learning performance. Existing works seek to quantify the conditions under which the auxiliary task is beneficial. Inspired by multi-view learning, [11, 25] exhibit significant performance gain when the auxiliary and main tasks are conditionally independent given the labels of the main task. In [2, 14] the informativeness of an auxiliary task is correlated with the alignment between the losses of the auxiliary task and the main task. The work in [3] shows that if the dimensionality of the extracted representations is sufficiently large and when augmented inputs map to the same representations, then the extracted representations can benefit the main task. Nonetheless, designing an effective auxiliary task is nontrivial, often requiring domain expertise and/or taxing numerical trials.

Contibutions. To circumvent the challenge of designing an effective auxiliary task, this work puts forth a novel SSL framework that fuses information from multiple auxiliary tasks. The advocated method systematically combines ex-

Part of this work was supported by NSF grants 2128593, 2212318, 2220292, 2312546 and 2312547. The work of Konstantinos D. Polyzos was also supported by the Onassis Foundation Scholarship.

tracted representations from different auxiliary tasks into a unified representation, without any prior knowledge about the effectiveness of individual auxiliary tasks. The algorithm design enables the automatic identification of effective auxiliary tasks from a pre-selected suite of tasks and emphasizes the high-quality representations obtained from such tasks. The extracted unified representation is then utilized for the main task by means of a sample-efficient Gaussian process model, that enjoys quantifiable uncertainty. Finally, the proposed approach is benchmarked on several real datasets, showcasing the benefits of combining multiple auxiliary tasks.

2. PRELIMINARIES

In supervised learning tasks, given a labeled training dataset $\mathcal{D}_{\text{train}} := \{(\mathbf{x}_\tau, y_\tau)\}_{\tau=1}^{T}$, we hypothesize that there exists a function $f(\cdot)$ capturing the input-output mapping such that $y = f(\mathbf{x})$ for all $(\mathbf{x}, y)$ belonging to the underlying distribution of $\mathcal{D}_{\text{train}}$. The goal is then to find a function $\hat{f}(\cdot)$ that closely approximates $f(\cdot)$. The estimation quality is evaluated based on the prediction accuracy of $\hat{f}(\cdot)$ for unseen instances $\mathbf{x}^*$ belonging to a test or evaluation set $\mathcal{T}^e := \{(\mathbf{x}_\tau^e, y_\tau^e)\}_{\tau=1}^{T^e}$, where the superscript ^{e} stands for *evaluation*, and the datasets $\mathcal{T}^e$ and $\mathcal{D}_{\text{train}}$ follow identical distributions. The function estimation task becomes particularly challenging when the number of training samples T is severely limited by the concerns of privacy or high sampling cost. In medicine for instance, where y_τ could represent the existence of brain cancer of patient τ, obtaining the labels may require costly medical examinations or may be reluctantly revealed due to medical confidentiality. Nevertheless, while obtaining additional labels may be difficult in such settings, one can readily sample unlabeled instances of $\mathbf{x}$ from the underlying distribution of inputs. This motivates the self-supervised learning (SSL) paradigm which aims at utilizing readily available unlabeled input instances to facilitate accurate estimation of input-output mapping $f(\cdot)$. Typically, SSL involves an *auxiliary* task that does not require external labels while providing representations that simplify the *main* task of learning an accurate estimator $\hat{f}(\cdot)$ [11].

SSL requires building an *auxiliary* set $\mathcal{D}_{\text{train}}^{\text{aux}}$ defined as $\{(\mathbf{x}_t^{\text{aux}}, y_t^{\text{aux}})\}_{t=1}^{T^{\text{aux}}}$ where each $\mathbf{x}_t^{\text{aux}}$ is drawn from the underlying input distribution and y_t^{aux} is a label that emanates solely from the input data and does not involve any additional labeling process. For example, if the main task is that of image classification, an informative auxiliary task could be constructed by setting the auxiliary labels y_t^{aux} as the rotation or a masked part of the image [6]. Relying on $\mathcal{D}_{\text{train}}^{\text{aux}}$, the auxiliary task learns a mapping $g(\cdot) : \mathbf{x}_t^{\text{aux}} \to y_t^{\text{aux}}$. Typically, this mapping is expressed as the composition of two functions $h(\cdot)$ and $\phi(\cdot)$ such that $g(\cdot) = h(\phi(\cdot))$. The function $h(\cdot)$ pertains solely to the auxiliary task, while $\phi(\cdot)$ is supposed to extract a representation vector $\mathbf{z}_\tau = \phi(\mathbf{x}_\tau), \forall \tau = 1, \dots, T$ that can serve as an informative input for the main task. Specifically, if the auxiliary task is suitably designed, estimating the mapping $\mathbf{z}_\tau \to y_\tau$ for the main task is easier than estimating $f(\cdot) : \mathbf{x}_\tau \to y_\tau$, and thus can be accomplished even with small T. When the function $g(\cdot)$ is captured by a neural network (NN) for instance, the representation $\mathbf{z}_\tau$ could be the output of one of the intermediate layers of the NN. To summarize, the typical SSL paradigm involves: 1) Learn a composite mapping $g : \mathbf{x}_t^{\text{aux}} \to y_t^{\text{aux}}$ using $\mathcal{D}_{\text{train}}^{\text{aux}}$, and obtain $\phi(\cdot)$ from $g(\cdot) = h(\phi(\cdot))$; 2) Transform the input data in $\mathcal{D}_{\text{train}}$ to the new feature space via $\mathbf{z} = \phi(\mathbf{x})$, and; 3) estimate the mapping $\mathbf{z} \to y_\tau$ for the main task using the transformed $\mathcal{D}_{\text{train}}$.

Algorithm 1

```
Input: Training set $\mathcal{D}_{\text{train}} := \{(\mathbf{x}_\tau, y_\tau)\}_{\tau=1}^{T}$, validation set $\mathcal{V} := \{(\mathbf{x}_\tau^v, y_\tau^v)\}_{\tau=1}^{V}$, set of unlabeled input data $\{\mathbf{x}_t\}_{t=1}^{T^{\text{aux}}}$
Partition $\mathbf{x}_t$ as $\mathbf{x}_t = [\mathbf{x}_t^{\text{aux}}, \check{x}_t^{\text{aux,1}}, \dots \check{x}_t^{\text{aux,S}}], \forall t$ and define auxiliary sets $\mathcal{D}_{\text{train}}^{\text{aux,s}} := \{(\mathbf{x}_t^{\text{aux}}, \check{x}_t^{\text{aux,s}})\}_{t=1}^{T^{\text{aux}}}$ for $s = 1, \dots S$
for $s = 1, 2, \dots, S$ do
    Estimate functions $h^s(\cdot)$ and $\phi^s(\cdot)$ such that the composition $g^s(\cdot) := h^s(\phi^s(\cdot))$ maps $\mathbf{x}_t^{\text{aux}} \to \check{x}_t^{\text{aux,s}}$ in $\mathcal{D}_{\text{train}}^{\text{aux,s}}$.
    For data points in $\mathcal{D}_{\text{train}}$, collect the first $d_x - S$ entries of the vector $\mathbf{x}_\tau \in \mathbb{R}^{d_x}$ in the vector $\mathbf{x}_\tau^{\text{aux}}$.
    Obtain representation $\mathbf{z}_\tau^s = \phi^s(\mathbf{x}_\tau^{\text{aux}}),\ \forall \tau$.
    Estimate $f^s(\cdot) : [\mathbf{x}_\tau, \mathbf{z}_\tau^s] \to y_\tau, \forall \tau$ for the main task using $\mathcal{D}_{\text{train}}$.
    Compute error $\varepsilon^{v,s}$ on the validation set according to (3).
end for
for $s = 1, 2, \dots, S$ do
    Compute weight $\lambda^s$ according to (4).
end for
Estimate $\phi^{\text{ens}}(\cdot)$ according to (2) and compute $\mathbf{z}_\tau^{\text{ens}} = \phi^{\text{ens}}(\mathbf{x}_\tau), \forall \tau$.
Estimate $f(\cdot) : \mathbf{c}_\tau = [\mathbf{x}_\tau, \mathbf{z}_\tau^{\text{ens}}] \to y_\tau, \forall \tau$ using $\mathcal{D}_{\text{train}}$.
Evaluate $f(\cdot)$ on test data $\mathcal{T}^e := \{(\mathbf{x}_\tau^e, y_\tau^e)\}_{\tau=1}^{T^e}$.
```

From the above discussion, it is evident that the auxiliary task is a critical component of SSL as it determines the extracted representation $\mathbf{z}$, and hence the overall quality of function estimation for the main task. Instead of focusing on the design or selection of an effective auxiliary task, the next section will outline an approach for combining extracted representations from multiple candidate auxiliary tasks.

3. PROPOSED APPROACH

Suppose that S auxiliary datasets of the form $\mathcal{D}_{\text{train}}^{\text{aux,s}} := \{(\mathbf{x}_t^{\text{aux}}, y_t^{\text{aux,s}})\}_{t=1}^{T^{\text{aux,s}}}$ for $s = 1, \dots S$, are available. Here, the auxiliary labels $y_t^{\text{aux,s}}$ are simply chosen to be an arbitrary entry of the input feature vector $\mathbf{x}_t \in \mathbb{R}^{d_x}$. Specifically, if the input vector is partitioned as

$$\mathbf{x}_t = [\mathbf{x}_t^{\text{aux}}, \check{x}_t^{\text{aux,1}}, \dots \check{x}_t^{\text{aux,S}}], \tag{1}$$

the labels for S auxiliary tasks are the last S entries of input, i.e., $y_t^{\text{aux,s}} = \check{x}_t^{\text{aux,s}}$. Without loss of generality, here the labels of the auxiliary task are written as the last S elements of vector $\mathbf{x}_t$. Compared to the conventional SSL approach described in the previous section where the dimension of $\mathbf{x}_t^{\text{aux}}$ is the same as that of $\mathbf{x}_t$, here $\mathbf{x}_t^{\text{aux}}$ is a subvector of $\mathbf{x}_t$. For each auxiliary task, s, the function $g^s(\cdot) := h^s(\phi^s(\cdot))$ can be estimated using standard machine learning approaches, similar to conventional SSL. Upon learning the functions $\{g^s(\cdot) := h^s(\phi^s(\cdot))\}_{s=1}^S$ for all auxiliary tasks, the first $d_x - S$ entries of each instance $\mathbf{x}_\tau \in \mathbb{R}^{d_x}$ in $\mathcal{D}_{\text{train}}$ are collected in the vector $\mathbf{x}_\tau^{\text{aux}}$ [1]. Then, for all $\{\mathbf{x}_\tau^{\text{aux}}, \forall\tau\}$ the representations $\{\mathbf{z}_\tau^s = \phi^s(\mathbf{x}_\tau^{\text{aux}}),\ \forall\tau\}_{s=1}^S$ are extracted for the main task. Using these representations, S different functions $\{f^s(\cdot) : [\mathbf{x}_\tau, \mathbf{z}_\tau^s] \rightarrow f^s([\mathbf{x}_\tau, \mathbf{z}_\tau^s]) \rightarrow y_\tau\ , \forall\tau\}$ can be learned for the main task, in order to assess the quality of individual representations for the main task. Note here, that the input vector includes both the representation $\mathbf{z}_\tau^s$ and the original input vector $\mathbf{x}_\tau$. This is to ensure that no information loss could occur by relying solely on the representations, that are in turn obtained using $\mathbf{x}_\tau^{\text{aux}}$. Furthermore, any parametric or nonparametric model can be used to learn these functions.

At this stage, however, the generalization performance of these different representations and functions is still unknown. To combine the merits of all different representations, a weighted ensemble of representations is advocated, that is

$$\phi^{\text{ens}}(\cdot) = \sum_{s=1}^{S} \lambda^s \phi^s(\cdot)\ , \quad \sum_{m=1}^{M} \lambda^s = 1. \tag{2}$$

where the weight $\lambda^s \in [0, 1]$ captures the significance of the s^{th} auxiliary task representation in the ensemble.

To quantify the contribution of each representation, a validation set $\mathcal{V} := \{(\mathbf{x}_\tau^v, y_\tau^v)\}_{\tau=1}^V$ is utilized to assess the prediction performance of each representation on the main task. Using the learnt f^s functions, the prediction error on the validation set for each representation is computed as

$$\varepsilon^{v,s} = \frac{1}{V} \sum_{\tau=1}^{V} \mathcal{L}(y_\tau^v, \hat{y}_\tau^{v,s}) \tag{3}$$

where $\hat{y}_\tau^{v,s} = f^s([\mathbf{x}_\tau, \mathbf{z}_\tau^s])$ is an estimate of the label y_τ^v, and $\mathcal{L}(\cdot)$ is a predefined loss function. For instance, in the regression task, the mean square loss can be used, which is expressed as $\mathcal{L}(y_\tau^v, \hat{y}_\tau^{v,s}) = \sum_{\tau=1}^V (y_\tau^v - \hat{y}_\tau^{v,s})^2$ and in the classification task the cross-entropy loss can be adopted which can be written as $\mathcal{L}(y_\tau^v, \hat{y}_\tau^{v,s}) = -\frac{1}{V} \sum_{\tau=1}^V [y_\tau^v \log(\hat{y}_\tau^{v,s}) + (1 - y_\tau^v) \log(1 - \hat{y}_\tau^{v,s})]$.

The prediction error on the validation set can then be used to compute the representation weights $\{\lambda^s\}_{s=1}^S$ as

$$\lambda^s = \frac{\exp(-\eta\varepsilon^{v,s})}{\sum_{s'=1}^{S} \exp(-\eta\varepsilon^{v,s'})}. \tag{4}$$

Intuitively, if representation s leads to lower prediction error compared to the other representations, it should have a larger contribution in the ensemble ϕ^{ens}. The parameter $\eta > 0$ is used to control the magnitude of the prediction errors $\exp(-\eta\varepsilon^{v,s})$ for each $s \in \{1, \dots, S\}$.

Finally, in the main task the sought function f is identified using as input the vector $\mathbf{c}_\tau := [\mathbf{x}_\tau, \mathbf{z}_\tau^{\text{ens}}]$ where $\mathbf{z}_\tau^{\text{ens}} = \phi^{\text{ens}}(\mathbf{x}_\tau), \forall\tau$. Algorithm 1 summarizes the steps of the advocated approach. With $\mathbf{c}_\tau$ as input for all τ, the present work leverages the so-termed Gaussian processes (GPs) as a nonparametric Bayesian approach to estimate the sought function f along with its probability density function (pdf) in a sample-efficient manner, as outlined next.

3.1. Learning with Gaussian processes

Unlike deterministic approaches, learning with GPs begins with the assumption that the unknown f is deemed random and a GP prior is considered over f; that is $f \sim \mathcal{GP}(0, \kappa(\mathbf{c}, \mathbf{c}'))$ where $\kappa(\cdot)$ is a positive-definite kernel function that captures the pairwise similarity between $\mathbf{c}$ and $\mathbf{c}'$. Equivalently, the random vector $\mathbf{f}_T := [f(\mathbf{c}_1) \dots f(\mathbf{c}_T)]^\top$ collecting all function evaluations at $\mathbf{C}_T := [\mathbf{c}_1 \dots \mathbf{c}_T]^\top$, is Gaussian distributed as

$$p(\mathbf{f}_T | \mathbf{C}_T) = \mathcal{N}(\mathbf{f}_T; \mathbf{0}_t, \mathbf{K}_T)$$

with $\mathbf{K}_T$ denoting the $T \times T$ kernel (covariance) matrix whose (n, n') entry is $[\mathbf{K}_T]_{n,n'} = \text{cov}(f(\mathbf{c}_n), f(\mathbf{c}_{n'})) := \kappa(\mathbf{c}_n, \mathbf{c}_{n'})$ [23].

The next assumption is that the batch likelihood relating $\mathbf{f}_T$ with the (possibly noisy) output vector $\mathbf{y}_T := [y_1, \dots, y_T]^\top$ is factored across individual labels as

$$p(\mathbf{y}_T | \mathbf{f}_T; \mathbf{C}_T) = \prod_{\tau=1}^{T} p(y_\tau | f(\mathbf{c}_\tau)).$$

This assumption could certainly pertain to the regression task where the output y_τ can be written as $y_\tau = f(\mathbf{c}_\tau) + n_\tau$ with $n_\tau \sim \mathcal{N}(n_\tau; 0, \sigma_n^2)$ denoting white Gaussian noise uncorrelated across τ, and hence the per-datum likelihood is $p(y_\tau | f(\mathbf{c}_\tau)) = \prod_{\tau=1}^T \mathcal{N}(y_\tau; f(\mathbf{c}_\tau), \sigma_n^2)$. Then for any (unlabeled) instance $\mathbf{c}$, one can write the joint pdf of $f(\mathbf{c})$ and $\mathbf{y}_T$ as

$$\begin{bmatrix} \mathbf{y}_T \\ f(\mathbf{c}) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}_{T+1}, \begin{bmatrix} \mathbf{K}_T + \sigma_n^2 \mathbf{I}_t & \mathbf{k}_T(\mathbf{c}) \\ \mathbf{k}_T^\top(\mathbf{c}) & \kappa(\mathbf{c}, \mathbf{c}) + \sigma_n^2 \end{bmatrix}\right)$$

[1] Note that $\mathbf{x}_\tau^{\text{aux}}$ refers to the main task and the superscript $^{\text{aux}}$ does not refer to the auxiliary task but is used for notational consistency with (1)

Table 1. NN parameters and characteristics for the auxiliary tasks

Dataset	Layers	Activation functions	Learning rates	Epochs
Malaria	$(20, 1), (20, 1), (20, 1)$	(tanh, linear), (tanh, linear), (tanh, linear)	$0.015, 0.015, 0.015$	$400, 400, 400$
Diabetes	$(15, 1), (15, 1), (15, 1)$	(tanh, linear), (tanh, linear), (tanh, linear)	$0.015, 0.015, 0.015$	$200, 100, 100$
California housing	$(15, 1), (15, 1), (15, 1)$	(tanh, linear), (tanh, linear), (tanh, linear)	$0.015, 0.015, 0.015$	$200, 100, 100$

Table 2. NMSE performance

GP with input	Malaria	Diabetes	California housing
$\mathbf{x}$	1.5415 ± 0.3827	0.2954 ± 0.1538	0.2987 ± 0.0027
$[\mathbf{x}, \mathbf{z}^1]$	1.4864 ± 0.2806	0.2747 ± 0.0886	0.2853 ± 0.0029
$[\mathbf{x}, \mathbf{z}^2]$	1.5513 ± 0.2991	0.3299 ± 0.1591	0.2861 ± 0.0035
$[\mathbf{x}, \mathbf{z}^3]$	1.5542 ± 0.4421	0.4073 ± 0.1694	0.2860 ± 0.0033
$\mathbf{c}$ (ours)	$\mathbf{1.3705 \pm 0.2308}$	$\mathbf{0.1415 \pm 0.0028}$	$\mathbf{0.2846 \pm 0.0007}$

where $\mathbf{k}_T(\mathbf{c}) := [\kappa(\mathbf{c}_1, \mathbf{c}), \ldots, \kappa(\mathbf{c}_T, \mathbf{c})]^\top$. The latter yields the posterior pdf of $f(\mathbf{c})$, which in the regression task is Gaussian distributed as [23]

$$p(f(\mathbf{c})|\mathcal{D}_{\text{train}}) = \mathcal{N}(f(\mathbf{c}); \mu_T(\mathbf{c}), \sigma_T^2(\mathbf{c})) \tag{5}$$

with mean and variance given in closed form as

$$\mu_T(\mathbf{c}) = \mathbf{k}_T^\top(\mathbf{c})(\mathbf{K}_T + \sigma_n^2\mathbf{I}_T)^{-1}\mathbf{y}_T \tag{6a}$$

$$\sigma_T^2(\mathbf{c}) = \kappa(\mathbf{c}, \mathbf{c}) - \mathbf{k}_T^\top(\mathbf{c})(\mathbf{K}_T + \sigma_n^2\mathbf{I}_T)^{-1}\mathbf{k}_T(\mathbf{c}). \tag{6b}$$

Note that the mean in (6a) offers a point estimate for the label of instance $\mathbf{c}$ and the variance in (6b) quantifies the associated uncertainty. Although the posterior in (6) incurs complexity $\mathcal{O}(T^3)$ due to the inversion of a $T \times T$ matrix, the latter is affordable in settings with small T that are the focus of this work. Nevertheless, in the case of computational constraints, several works that reduce the cubic complexity can be used; see e.g [22, 10, 7].

4. NUMERICAL TESTS

In this section, the performance of our novel SSL approach is assessed on three real-world datasets: the malaria dataset, the diabetes dataset, and the California housing dataset. Detailed descriptions of the datasets are provided below.

Malaria dataset. In this dataset, the input vector $\mathbf{x}_\tau$ consists of 8 features for location τ including longitude, latitude, and some bioclimatic characteristics, and the target label y_τ represents the infection rate of Plasmodium falciparum, the parasite that causes malaria [26]. For this dataset we consider $|\mathcal{D}_{\text{train}}| = 100$ training data, $|\mathcal{V}| = 100$ validation data, $|\mathcal{T}^{\text{aux}}| = 1000$ data, $|\mathcal{T}^e| = 680$ test data and $S = 3$ auxiliary tasks, where three bioclimatic features of the input are randomly selected.

Diabetes dataset. In this dataset, given 10 characteristics of diabetes patients, including age, sex, body mass index, average blood pressure, and six blood-related measurements, the label to be predicted for each patient is a metric that quantifies the disease progression in a single year [4]. Here, we consider $|\mathcal{D}_{\text{train}}| = 100, |\mathcal{V}| = 100, |\mathcal{T}^{\text{aux}}| = 1000, |\mathcal{T}^e| = 300$ and $S = 3$.

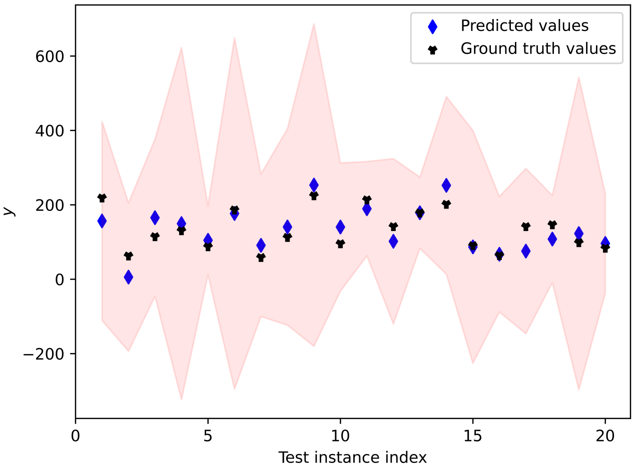

Fig. 1. Performance visualization across 20 test instances on the Diabetes dataset.

California housing dataset. In this dataset, the input vector $\mathbf{x}_\tau$ comprises 8 features of district τ in California including demographic and location data along with more general features such as the average number of rooms and bedrooms per household. The target variable is the median house price in these districts [19]. In this dataset, we consider $|\mathcal{D}_{\text{train}}| = 100, |\mathcal{V}| = 100, |\mathcal{T}^e| = 700, |\mathcal{T}^{\text{aux}}| = 1000$ and $S = 3$.

The advocated ensemble of representations-based approach is compared with a GP model that (i) uses as input the original input vector $\mathbf{x}$ and (ii) relies on each representation individually. For all S auxiliary tasks in all datasets, NN models are used to capture the underlying functions $g^s(\cdot)$. Details about the structure and training of each NN are given in Table 1. The extracted representation for each auxiliary task is the output of the first layer of the trained NN. For the GP model in the main task, a radial basis function (RBF) kernel is adopted and all hyperparameters including the characteristic lengthscale of the kernel along with σ_n^2 are obtained maximizing the marginal log-likelihood using the *sklearn* package.

The performance of all competing methods is evaluated on the test set $\mathcal{T}^e$ using the normalized mean square error (NMSE) which is expressed as

$$\text{NMSE} := \frac{1}{T^e} \sum_{\tau=1}^{T^e} (\hat{y}_\tau^e - y_\tau^e)^2 / \sigma_y^2$$

where $\hat{y}_{\tau|t}^e$ denotes a point estimate of test instance τ, and $\sigma_y^2 := \mathbb{E}\|\mathbf{y}_{T^e}^e - \mathbb{E}\{\mathbf{y}_{T^e}^e\}\|^2$, with $\mathbf{y}_{T^e}^e := [y_1^e \dots y_{T^e}^e]^\top$. Table 2 demonstrates the average NMSE performance along with the corresponding standard deviation of 10 independent runs, where it is evident that the advocated approach achieves substantially improved performance compared to the baseline methods in almost all datasets. It is worth noting that some representations, when used individually, may fail to lower NMSE compared to the GP model that uses as input the original input vector $\mathbf{x}$; see e.g the Diabetes dataset. This showcases the importance of prudently combining the merits of all individual representations to markedly improve prediction performance.

Finally, to evaluate the uncertainty quantification performance of the advocated approach, Fig. 1 illustrates the predicted values of 20 randomly selected test instances on the Diabetes dataset, along with the corresponding standard deviation σ-confidence intervals. It is evident that the ground truth labels fall inside these confidence intervals.

5. CONCLUSION

To alleviate the challenges in designing an informative auxiliary task for self-supervised learning, this work introduced a novel approach that combines distilled knowledge from multiple arbitrary candidate auxiliary tasks and yields a unified representation. The unified representation, when included in the input of the main tasks, is empirically shown to provide improved prediction capability. The proposed approach shows promising results in three real-world datasets. Future research will involve rigorous performance analysis, extensive tests on additional setups, as well as alternative methods for combining auxiliary tasks.

6. REFERENCES

[1] S. Albelwi, "Survey on self-supervised learning: Auxiliary pretext tasks and contrastive learning methods in imaging," *Entropy*, vol. 24, no. 4, p. 551, 2022.

[2] L. Chen, Y. Zhang, Y. Song, Y. Shan, and L. Liu, "Improved test-time adaptation for domain generalization," in *Proc.IEEE/CVF Conf. Comput. Vision Pat. Recognition*, 2023, pp. 24 172–24 182.

[3] Y. Dubois, S. Ermon, T. B. Hashimoto, and P. S. Liang, "Improving self-supervised learning by characterizing idealized representations," vol. 35, pp. 11 279–11 296, 2022.

[4] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[5] L. Ericsson, H. Gouk, C. C. Loy, and T. M. Hospedales, "Self-supervised representation learning: Introduction, advances, and challenges," *IEEE Sig. Proc. Magazine*, vol. 39, no. 3, pp. 42–62, 2022.

[6] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[7] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian processes for big data," in *Proc. of Uncertainty in Artificial Intelligence*. Citeseer, 2013, p. 282.

[8] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131.e9, 2018.

[9] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1920–1929.

[10] M. Lázaro-Gredilla, J. Quiñonero Candela, C. E. Rasmussen, and A. Figueiras-Vidal, "Sparse spectrum Gaussian process regression," *J. Mach. Learn. Res.*, vol. 11, no. Jun, pp. 1865–1881, 2010.

[11] J. D. Lee, Q. Lei, N. Saunshi, and J. Zhuo, "Predicting what you already know helps: Provable self-supervised learning," *Proc. Advances Neural Inf. Process. Syst.*, vol. 34, pp. 309–323, 2021.

[12] T. Liu, S. Xie, J. Yu, L. Niu, and W. Sun, "Classification of thyroid nodules in ultrasound images using deep model based transfer learning and hybrid features," in *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Process.*, Mar. 2017, pp. 919–923.

[13] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2021.

[14] Y. Liu, P. Kothari, B. Van Delft, B. Bellot-Gurlet, T. Mordan, and A. Alahi, “Ttt++: When does self-supervised test-time training fail or thrive?” *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 808–21 820, 2021.

[15] G. Lu, Y. Yan, L. Ren, P. Saponaro, N. Sebe, and C. Kambhamettu, “Where am i in the dark: Exploring active transfer learning on the use of indoor localization based on thermal imaging,” *Neurocomputing*, vol. 173, pp. 83–92, 2016.

[16] M. Matassoni, R. Gretter, D. Falavigna, and D. Giuliani, “Non-native children speech recognition through transfer learning,” in *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Process.*, Apr. 2018, pp. 6229–6233.

[17] I. Misra and L. v. d. Maaten, “Self-supervised learning of pretext-invariant representations,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6707–6717.

[18] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash, “Boosting self-supervised learning via knowledge transfer,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9359–9367.

[19] R. K. Pace and R. Barry, “Sparse spatial autoregressions,” *Statistics and Probability Letters*, vol. 33, no. 3, pp. 291–297, 1997.

[20] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[21] K. D. Polyzos, A. Sadeghi, and G. B. Giannakis, “Bayesian self-supervised learning using local and global graph information,” in *Proc. IEEE Int. Workshop Comput. Advances Multi-Sensor Adaptive Process.* IEEE, 2023, pp. 256–260.

[22] A. Rahimi and B. Recht, “Random features for large-scale kernel machines,” *Proc. Advances Neural Inf. Process. Syst.*, pp. 1177–1184, 2008.

[23] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

[24] M. K. Singh, K. D. Polyzos, P. A. Traganitis, S. V. Dhople, and G. B. Giannakis, “Physics-informed transfer learning for voltage stability margin prediction,” *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Process.*, 2023.

[25] J. Teng, W. Huang, and H. He, “Can pretext-based self-supervised learning be boosted by downstream data? a theoretical analysis,” in *Proc. Int. Conf. Artificial Intel. and Stats.* PMLR, 2022, pp. 4198–4216.

[26] D. J. Weiss, T. C. Lucas, M. Nguyen, A. K. Nandi, D. Bisanzio, K. E. Battle, E. Cameron, K. A. Twohig, D. A. Pfeffer, J. A. Rozier *et al.*, “Mapping the global prevalence, incidence, and mortality of plasmodium falciparum, 2000–17: a spatial and temporal modelling study,” *The Lancet*, vol. 394, no. 10195, pp. 322–331, 2019.

[27] X. Xu, H. Dinkel, M. Wu, Z. Xie, and K. Yu, “Investigating local and global information for automated audio captioning with transfer learning,” in *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Process.*, Jun. 2021, pp. 905–909.

[28] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. Le, “Rethinking pre-training and self-training,” *Proc. Advances Neural Inf. Process. Syst.*, vol. 33, pp. 3833–3845, 2020.