Boosting Multi-task Learning Through Combination of Task Labels - with Applications in ECG Phenotyping

Ming-En Hsieh,[1] **Vincent S. Tseng**[1,2]
[1] Institute of Data Science and Engineering, National Chiao Tung University
[2] Department of Computer Science, National Chiao Tung University
mingen@cs.nctu.edu.tw, vtseng@cs.nctu.edu.tw

Abstract

Multi-task learning has increased in importance due to its superior performance by learning multiple different tasks simultaneously and its ability to perform several different tasks using a single model. In medical phenotyping, task labels are costly to acquire and might contain a certain degree of label noise. This decreases the efficiency of using additional human labels as auxiliary tasks when applying multi-task learning to medical phenotyping. In this work, we proposed an effective multi-task learning framework, CO-TASK, to boost multi-task learning performance by generating auxiliary tasks through COmbination of TASK Labels. The proposed CO-TASK framework generates auxiliary tasks without additional labeling effort, is robust to a certain degree of label noise, and can be applied in parallel with various multi-task learning techniques.

We evaluated our performance using the CIFAR-MTL dataset and demonstrated its effectiveness in medical phenotyping using two large-scale ECG phenotyping datasets, an 18 diseases multi-label ECG-P18 dataset and an echocardiogram diagnostic from electrocardiogram dataset ECG-EchoLVH. On the CIFAR-MTL dataset, we doubled the average per-task performance gain of the multi-task learning model from 4.38% to 9.78%. With the proposed task-aware imbalance data sampler, the CO-TASK framework can effectively deal with the different imbalance ratios for the different tasks in electrocardiogram phenotyping datasets. The proposed framework combined with noisy annotations as minor tasks increased the sensitivity by 7.1% compared to the single-task model while maintaining the same specificity as the doctor annotations on the ECG-EchoLVH dataset.

Introduction

In the rise of deep learning and large-scale datasets, pre-training models and transferring their knowledge to downstream tasks have become the go-to method to utilize knowledge learned from those large-scale datasets for improved results in downstream tasks. With the great success of transfer learning methods, multi-task learning tries to take a step further and learn multiple different tasks simultaneously instead of the traditional setting of learning one task at a time. Multi-task learning methods aim to improve individual tasks by learning all the tasks together, enabling the tasks to transfer knowledge between each other concurrently and help regularize the model to prevent overfitting on a single task, which leads to more generalized and robust models.

Multi-task learning methods has shown benefits in a variety of different domains, including computer vision (Xu et al. 2018; Liang et al. 2019), natural language processing (Liu et al. 2019), time series analysis (Cirstea et al. 2018), recommendation systems (Zhao et al. 2019) and medical data analysis (Harutyunyan et al. 2019). Multi-task learning in the deep learning era appeals to a variety of different application domains for three main reasons. First, in many application scenarios, we are not only interested in a single task, but we are also interested in a set of related tasks. Second, multi-task learning can improve the performance of individual tasks by sharing the knowledge learned between different tasks. This knowledge-sharing process helps harder tasks or tasks with fewer training data to learn faster and achieve better results. Learning multiple tasks together also regularizes the model to prevent overfitting on individual tasks. Third, by sharing part of the network to form shared representations, the multi-task model results in reduced memory footprint and faster inference speed, essential to edge-devices and large-scale deployment of the models.

Unlike research that focuses on achieving better results under a predefined set of tasks, various studies have shown that adding auxiliary tasks to train in parallel can help improve the performance of targeted tasks (Liebel and Körner 2018; Standley et al. 2020). These tasks act as hints to what shall be learned and regularize the model to prevent it from learning incorrect relationships between the input and output. Most research still requires labeled data to generate auxiliary tasks, which limits the usefulness of this method in practical scenarios. Automatic methods to generate useful auxiliary tasks for a targeted task (Liu, Davison, and Johns 2019) has become a challenging new direction.

In recent years, there is an increased interest in applying multi-task learning methods in the medical domain. A patient can be diagnosed with multiple diseases, so phenotyping can be treated as a multi-task learning problem, with each disease label being a unique task (Razavian, Marcus, and Sontag 2016). Combining multiple phenotyping tasks from different datasets (Tellez et al. 2020) and learning phenotyping tasks with auxiliary tasks (Ding et al. 2019) using

multi-task learning has shown performance gains.

Multi-task learning in the medical domain has its own set of challenges. First, the cost is much higher to obtain clean, supervised labels. Although cheaper labels can be generated from existing electronic health records (EHR) using natural language processing, they usually contain a higher degree of label noise. Second, most datasets in the medical domain are imbalanced, most patients are healthy, and each disease has different prevalence rates. These problems make it trickier to enhance the model's performance with auxiliary tasks through multi-task learning.

In this work, we ask the following question: Is it possible to generate useful auxiliary tasks without the additional cost of acquiring more labels? With the challenges in applying multi-task learning to the medical domain, extracting the most knowledge from existing labels and improving regularization to prevent overfitting is crucial. Improvements through multi-task learning that does not require additional task labels or methodologies that can utilize noisy auxiliary task labels can squeeze more performance from the same amount of labeled data. We aim to provide a general multi-task learning methodology that can be applied on top of existing multi-task learning methods to improve multi-task learning performance in various domains further.

The main contributions of this paper are summarized as follows:

- We proposed a novel framework, CO-TASK, to boost multi-task learning performance by generating useful auxiliary tasks through COmbination of TASK Labels, which improves multi-task learning performance on major tasks without the need for additional labeling effort.
- We proposed a novel task-aware imbalance data sampler that is effective in dealing with different data imbalance ratios for different tasks.
- The proposed framework demonstrated significant improvements on the multi-task image classification benchmark dataset, CIFAR-MTL, and showed that it could provide additional performance gains in parallel with other multi-task learning techniques.
- The proposed framework is applied to two real-world electrocardiogram phenotyping datasets. The experimental results demonstrated that the proposed framework could improve the performance of current state-of-the-art models for electrocardiogram phenotyping.

Related Work

Multi-task Learning

Most applications that utilize multi-task learning in deep neural networks implement a variant of the basic shared-bottom model, a hard-parameter sharing model that shares all network layers before the final fully-connected networks. The full model's overall loss is the weighted summation of the loss for each task, with individual task weights searched through grid search and fixed during training.

Vandenhende et al. (2020) divided existing methods using multi-task learning in deep neural networks into deep multi-task architectures and optimization strategy methods. Deep multi-task architectures focus on better architectures for multi-task learning models, while a majority of optimization strategy methods focus on better methodologies to obtain individual task weights that could dynamically change through the training process.

Model architecture innovations in multi-task learning aim to find better ways to share layers or sub-spaces in the neural network. There are currently four main categories in multi-task learning models. Column-based models such as cross-stitch networks (Misra et al. 2016) or sluice networks (Ruder et al. 2019). Models focused on the attention of sub-module outputs, such as MMoE (Ma et al. 2018), SNR (Ma et al. 2019), and TRL (Strezoski, Noord, and Worring 2019). Models focused on dynamic linking or routing of sub-modules, such as soft layer ordering (Meyerson and Miikkulainen 2018) and routing networks (Rosenbaum, Klinger, and Riemer 2018). Models originated from PAD-NET (Xu et al. 2018) that utilized auxiliary multi-modal model output as multiple tasks during training.

Dynamic task weighting techniques can be categorized by the three different reference metrics it uses to adjust the task weighting: individual task loss, parameter gradients, and evaluation metrics. Kendall, Gal, and Cipolla (2018) proposed to train separate noise parameters for each task to tackle different homoscedastic uncertainty between tasks. Chen et al. (2018) proposed GradNorm and sets the common scale for all gradients to be the average l2 norm of individual task gradient. Liu, Johns, and Davison (2019) proposed Dynamic Weight Averaging (DWA) to balance the pace each task is learning according to the relative speed the task's loss is decreasing. Guo et al. (2018) proposed Dynamic Task Prioritization (DTP) that balance the tasks according to the current task difficulty by calculating the moving average of the targeted performance metrics.

Auxiliary Tasks in Multi-task Learning

Auxiliary tasks are tasks that are added to the multi-task learning model with the sole purpose of improving the performance of original tasks. These tasks are usually related to the major tasks which guide the multi-task model to learn important data features and act as a regularization to prevent the model from overfitting a particular task. Liebel and Körner (2018) created a synthetic dataset, synMT, and demonstrated that auxiliary tasks that are not directly related could be used to regularize and improve the original targeted tasks' performance. Standley et al. (2020) showed that the best separation of tasks into task groups to learn with multi-task learning under a fixed budget is achieved when some tasks are added as auxiliary tasks in specific tasks groups. Liu, Davison, and Johns (2019) proposed the Meta AuXiliary Learning (MAXL) framework by using meta-learning to generate beneficial auxiliary tasks to a multi-class learning problem. Lee, Hwang, and Shin (2019) generates self-supervised learning tasks as auxiliary tasks and train supervised learning models to learn them jointly.

Multi-task Learning in Medical Domain

Multi-task learning has been again and again proven to provide performance gains in the medical domain. Various stud-

ies demonstrated that using multi-task learning with auxiliary tasks can increase performance on targeted tasks. Ding et al. (2019) employs a shared-bottom multi-task neural network for the task of phenotyping and uses groups of ICD-9 codes as auxiliary tasks. Guendel et al. (2019) combines segmentation, spatial classification, and multiple abnormality classification tasks for each disease in a multi-task learning setting to increase the performance for Chest X-ray abnormality classification. Harutyunyan et al. (2019) and Song et al. (2018) extracted four tasks: in-hospital mortality, decompensation, length of stay, and phenotyping from the MIMIC-III dataset (Johnson et al. 2016) and demonstrated performance improvements for the four extracted tasks.

Electrocardiogram Phenotyping

Electrocardiogram (ECG) is a recording of electrical signals from the heart containing 12 separate leads. These signal relates to the cardiology system and is useful to diagnose common heart diseases. Deep learning methods have been shown to be effective in electrocardiogram phenotyping tasks. Hannun et al. (2019) developed a ResNet based deep neural network that can diagnose ten different arrhythmias from single-lead ECGs with diagnostic performance similar to those from experienced cardiologists.

The electrocardiogram is also considered a cheap and non-invasive method compared to other methods such as echocardiograms that are used to generate detailed diagnostics of a person's cardiology system. Since an electrocardiogram can also detect some indication of heart problems similar to an echocardiogram, several studies try to obtain echocardiogram diagnostic from an electrocardiogram. Attia et al. (2019) proposed a convolution neural network to identify patients with asymptomatic left ventricular dysfunction (ALVD). Patients identify with some risk by the model but not diagnosed with ALVD by the echocardiogram turns out to have a much higher probability of being diagnosed with ALVD by echocardiogram in the future. Kwon et al. (2020) proposed a convolution neural network to detect left ventricular hypertrophy (LVH) from electrocardiogram that outperforms cardiologists and conventional methods.

Methodology

In this section, we will first provide the intuition of the proposed COmbination of TASK labels (CO-TASK) multi-task learning framework. Subsequently, we will describe the three main components of the proposed CO-TASK framework, auxiliary task generation component, multi-task learning (MTL) model training component, and task-aware imbalance data sampler in detail. An overview of the CO-TASK framework is shown in Figure 1.

Intuition for CO-TASK Framework

Auxiliary tasks are known to improve the performance of existing tasks through knowledge transfer and regularization. The auxiliary tasks are usually alternative labels from the original labeled dataset to prevent negative transfer from auxiliary tasks. Existing methods in unsupervised pre-training try to generate pseudo labels by clustering data points into clusters with cluster counts much larger than the original class count and set those cluster labels as pseudo labels (Yan et al. 2020). Under the multi-task learning setting, we can utilize the abundant knowledge of task labels from the data point of different tasks. Instead of forming finer labels through clustering, we can generate coarse labels from the combination of these task labels.

The intuition for using the combination of task labels is the following, if the representations for classes α, β, and γ are separable, a new class composed of data from class α and β shall be separable from class γ. We can use this characteristic and combine separate classes from different tasks to form a new class to train as in Figure 2.

Training on the combination of task labels as auxiliary tasks makes it easier for the model to learn the harder tasks and regularize the dominant tasks to prevent them from over-influencing the final model. Mapping the classes between different tasks together could also encourage the model to map the representation of different tasks onto a much similar representation space, which might help prevent overfitting and improve the representations' generalizability. For phenotyping tasks, the combination of task labels can be seen as forming more balanced sub-groups in the overall patient population with a specific subset of disease labels.

Auxiliary Task Generation

Consider we have a task-set $TS = \{T_1, T_2, \ldots, T_N\}$ with N tasks. To create a task-set of M auxiliary tasks $TS_A = \{T_{A1}, T_{A2}, \ldots, T_{AM}\}$, we randomly select M distinct combinations of 2 tasks $T_{S1} = T_i$ and $T_{S2} = T_j$ from task-set TS. Next, we randomly map one class from each task $C_x^{T_i}$ and $C_y^{T_j}$ into a new class in the auxiliary task $C_z^{T_{Al}}$. We choose to combine combinations of 2 tasks to limit the combination space of resulting tasks, which equals to the Bell Number B_K and grows much faster than 2^K. Due to the limitations of current multi-task learning models, we do not generate all possible auxiliary tasks but randomly sample a subset of class mappings as auxiliary tasks. An example of a possible mapping result is shown in Figure 3.

In some application settings, we might have major tasks in the task-set that are more important than the other minor tasks. In these scenarios, since it is impracticable to train with all possible combinations, we can increase the probability of sampling a major task. This increases the possibility of an auxiliary task being composed of a major task, which would make the model prioritize on those important tasks.

It is common for medical phenotyping datasets to be highly imbalanced as most patients are healthy. There are also much fewer patients having more than one disease. To prevent the class mapping from mapping all patients to a single class, we remove the class mapping of patients with both disease labels as a class and the other patients as another class from the four possible class mappings.

To reduce the training data for the joined auxiliary task, we randomly down-sample the combined training data with a ratio of $1/n$ when we combine n tasks into a single auxiliary task. This process has two benefits. First, this helps remove the imbalance issue between different data count of

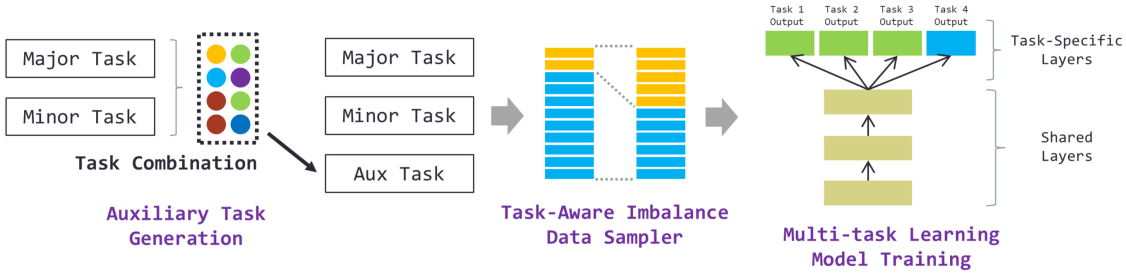

Figure 1: CO-TASK Framework Overview.

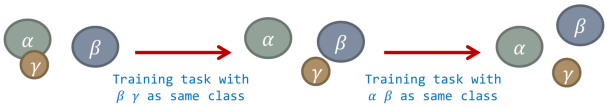

Figure 2: Data representation through the training process.

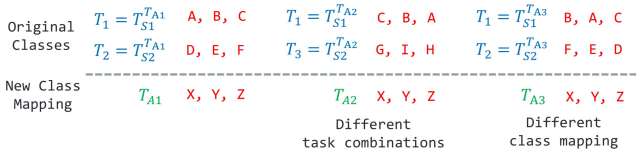

Figure 3: Example of possible mappings for auxiliary tasks.

the original tasks and the auxiliary tasks. Second, this process is similar to bagging as we let the auxiliary task look at only a subset of all the data.

Multi-task Learning Model Training

After the generation of auxiliary tasks through the combination of task labels, we follow the standard procedure for multi-task learning model training. For each batch in an epoch, we first pick a task we want to train randomly or weighted toward major tasks. We then generate a batch of training samples from the selected task and calculate the loss related to the task we pick.

The baseline models are trained on standard shared-bottom neural networks. The shared-bottom model is chosen as it does not add significant parameters compared to its original single-task backbone. Although the task-specific branches of auxiliary tasks on the shared-bottom model will increase the overall model size during training, we can remove those task-specific branches for auxiliary tasks before we deploy the model for inference. The final model for major tasks shall have the same capacity as the multi-task model trained without the CO-TASK Framework.

The benefit of applying standard procedure for multi-task learning model training is that we are able to adopt most frameworks targeted for improving multi-task learning performance. Our proposed method can be applied on top of other multi-task learning models easily and can also use dynamic task weighting techniques when training. Our proposed framework provides an add-on benefit for a variety of current state-of-the-art methods.

Task-Aware Imbalance Data Sampler

It is common for phenotyping tasks in medical data to have a high imbalance ratio between the two classes due to the fact that most patients are healthy. An important problem seldom mentioned in multi-task learning with real-world imbalanced datasets is that there will be different imbalance ratios for different tasks. For phenotyping tasks, we will have imbalance data ratios that are different by more than ten times between tasks and the same data as input having different labels for different tasks. This kind of relationship means that sampling each data according to any specific task label is sub-optimal.

To deal with this problem, we proposed the task-aware imbalance data sampler. The task-aware imbalance data sampler generates a batch of training samples for the selected task according to the designated weight function. To generate a batch of training samples for the selected tasks, we first calculated the weight of each class for the selected task by passing the class's data count for the selected task through the weight function. Then, all the data points have a probability of being sampled according to the corresponding weight. During training, the task to train for each batch is first picked then data is sampled according to the targeted task. This allows balanced training for task-specific classifier branches while allowing imbalanceness for representation learning in the shared layers.

The weight function $log(x)/x$ balances the small and large classes and allows larger classes to have a slightly higher total sample probability. This weight function is the default for the task-aware imbalance data sampler, which performs well in most imbalance data scenarios.

Experiments

Dataset Description

We use three datasets, an image classification benchmark dataset CIFAR-MTL and two real-world electrocardiogram phenotyping datasets ECG-P18 and ECG-EchoLVH.

Benchmark Dataset: CIFAR-MTL CIFAR-MTL is a multi-task learning dataset proposed by Rosenbaum, Klinger, and Riemer (2018) that is constructed from the well known CIFAR-100 dataset. The 100 classes in CIFAR-100 are grouped into 20 coarse super-classes with 5 related finer sub-classes each. CIFAR-MTL used the 20 coarse super-classes as 20 image classification tasks.

Disease	Imbalance Ratio	Disease	Imbalance Ratio
LVH	1 : 6.73	RVH	1 : 167.57
AFIB	1 : 20.68	AFL	1 : 171.24
IVCD	1 : 49.94	NIVCD	1 : 71.46
RBBB	1 : 19.34	LBBB	1 : 138.35
IRBBB	1 : 105.16	ILBBB	1 : 1843.32
1AVB	1 : 18.16	2AVB	1 : 526.66
3AVB	1 : 2321.22	EAR	1 : 123.86
VBIG	1 : 124.85	VT	1 : 1843.32
VPC	1 : 23.61	APC	1 : 39.17

Table 1: Tasks for the ECG-P18 dataset.

Disease Name	Imbalance Ratio
Echocardiogram LVH	1 : 7.28
Electrocardiogram LVH	1 : 5.75
AFIB	1 : 16.07
RBBB	1 : 21.77
1AVB	1 : 18.59

Table 2: Tasks for the ECG-EchoLVH dataset.

Real-World Dataset: ECG-P18 and ECG-EchoLVH An electrocardiogram phenotyping dataset, ECG-P18, and an echocardiogram diagnostic from electrocardiogram dataset, ECG-EchoLVH, are obtained by parsing electronic health records (EHR) from the database of a large national medical center in Taiwan. Each record consists of a 10 second, 12 lead electrocardiogram recording at 500 Hz and its diagnostic statements typed by the doctor in charge. The left ventricular hypertrophy (LVH) disease labels for the ECG-EchoLVH dataset are acquired from the echocardiogram recordings within 30 days of electrocardiogram recording.

The ECG-P18 dataset consists of 312,888 valid recordings with 687,092 diagnostic statements between 2013 and 2017. We have chosen 18 important diseases in electrocardiogram phenotyping to form the multi-label dataset ECG-P18. The average amount of labels per data instance, label cardinality, is 0.4536 for the full ECG-P18 dataset and 1.3283 if we only consider patients with diseases. Those metrics and Table 1 showed that the ECG-P18 dataset is highly imbalanced, with each task having a different imbalance ratio. The labels are sparse, with a majority of healthy patients having no labels at all. The ECG-EchoLVH dataset consists of 61,422 valid electrocardiogram and echocardiogram pairs with the LVH disease label acquired from echocardiogram recordings as major task and four selected diagnostic statements shown in Table 2 as minor tasks. Both datasets are split into the training set, validation set, and testing set without patient overlap with a ratio of 7:1:2.

Evaluation Metrics

Benchmark Dataset: CIFAR-MTL For evaluating the performance of different methods on CIFAR-MTL dataset, we use accuracy as our major metric for each task. To aggregate the metrics of each task, we use two different metrics, average accuracy (Rosenbaum, Klinger, and Riemer 2018) and average improvement over single-task Δm (Maninis, Radosavovic, and Kokkinos 2019; Vandenhende et al. 2020).

$$AverageAccuracy = \frac{1}{N}\sum_{i=1}^{N} Accuracy_i \tag{1}$$

$$\Delta m = \frac{1}{N}\sum_{i=1}^{N} \frac{M_{m,i} - M_{s,i}}{M_{s,i}} \tag{2}$$

where $M_{m,i}$ = Multi-Task Performance
$M_{s,i}$ = Single-Task Performance

Real-World Dataset: ECG-P18 For the ECG-P18 dataset, we evaluate the overall performance through four multi-label metrics, Macro-AUROC, Multi-label recall, Jaccard, and Exact.

We calculate the macro-average of the area under receiver operating characteristics curve (Macro-AUROC) as an indicator of each label's overall binary classification performance.

$$Macro_{AUROC} = \frac{1}{N}\sum_{i=1}^{N} AUROC_i \tag{3}$$

We define p_i as the set of prediction and t_i as the set of truth labels. The multi-label recall is the average percentage of labels caught for each user. The Jaccard metric is the average Jaccard similarity for each user, which indicates how similar the real disease labels are compared to the predictions. Finally, the Exact metric is the percentage of users with labels (user with diseases) having all the labels correct.

$$Recall_{Multi} = \frac{1}{n}\sum_{i=1}^{n} \frac{|p_i \cap t_i|}{t_i} \tag{4}$$

$$Jaccard = \frac{1}{n}\sum_{i=1}^{n} \frac{|p_i \cap t_i|}{|p_i \cup t_i|} \tag{5}$$

$$Exact = \frac{1}{n}\sum_{i=1}^{n} I[p_i = t_i] \quad with \quad t_i \neq \emptyset \tag{6}$$

Real-World Dataset: ECG-EchoLVH For the ECG-EchoLVH dataset, we only care about the performance of the major task, which is the ground truth disease label of echocardiogram left ventricular hypertrophy (Echo LVH). Since this is a binary classification task, we choose to use the area under receiver operating characteristics curve (AUROC) of the major task as the evaluation metric. We will also compare the sensitivity of the major task with the same specificity of current doctor annotations, so we can observe the increase in sensitivity when used as a screening tool.

Baseline Models and Training Procedure

The baseline models for the CIFAR-MTL experiments are shared-bottom models using ResNet-34 (He et al. 2016) as the backbone. We generate 50 auxiliary tasks using the proposed framework for the baseline performance experiment and 40 auxiliary tasks for other experiments.

For the ECG-P18 and ECG-EchoLVH dataset, we used a ResNet-based model proposed by Hannun et al. (2019) with 8 residual blocks as the backbone model. We generate 30 auxiliary tasks for the ECG-P18 dataset and 8 auxiliary tasks for the ECG-EchoLVH dataset.

Hyper-parameter search is done through Bayesian optimization using a Gaussian process to model the relation between the parameters and validation metrics, then choosing the parameters with the highest probability of improvement. All experiments are performed on a Ubuntu 18.04.2 server using RTX 2080ti with CUDA 10.0. We implemented all our algorithms using Pytorch 1.4.0 (Paszke et al. 2019) and torchvision 0.5.0 with Python 3.7. The signal processing for electrocardiogram signals is done using BioSPPy 0.6.1 (Carreiras et al. 2015–). Details of the training procedure are described in the appendix.

Performance on CIFAR-MTL

Baseline Performance of the Proposed Method We evaluate our proposed method on CIFAR-MTL and compare the single-task and multi-task model performance on the same CIFAR-MTL test set. The overall performance is shown in Table 3 and the per-task results are shown in Table 4. From Table 3, we can see that the proposed CO-TASK framework improved the average accuracy by an additional 3.82% and doubled the average improvement over single-task Δm. From Table 4, we can see that the proposed CO-TASK framework achieves the best performance on most tasks.

Combination of Proposed Method with Other Multi-task Learning Methods Given the flexibility of our proposed CO-TASK framework, we can apply various multi-task learning techniques in parallel with the CO-TASK framework. These include task weight balancing methods such as dynamic weight averaging (DWA) or dynamic task prioritization (DTP) and model architecture innovations such as task routing layers (TRL). From Table 5, we can see that the proposed CO-TASK framework can provide additional gains to multiple different task weight balancing methods and multi-task learning models. The DWA method and TRL model can boost the proposed CO-TASK framework to even higher performances. We can also observe from the table that the performance gains from the CO-TASK framework are much larger than those from other methods.

CIFAR-MTL with Label Noise To simulate the label noise we might encounter in application datasets, we apply symmetric noise to the CIFAR-MTL dataset. When we have a noise ratio of σ, we have a probability of σ for the data in the training dataset to corrupt into another class, with each class having the same probability. The testing set is not corrupted to evaluate how well our model performs when encountering label noise in training data. In Table 6, we can observe that as the noise ratio increases, the performance difference between the multi-task model and the same multi-task model using CO-TASK framework increases.

Method	Average Accuracy	Absolute Improvement	Δm
Single-Task	0.7321	Baseline	Baseline
Multi-task	0.7547	↑ 2.26%	↑ 4.38%
CO-TASK	**0.7929**	↑ **6.08%**	↑ **9.78%**

Table 3: Overall performance on CIFAR-MTL.

Task Name	Single Task	Multi-task	CO-TASK
Reptiles	0.634	**0.708**	0.706
Fish	0.710	0.790	**0.802**
Aquatic Mammal	0.592	0.576	**0.716**
Small Mammal	0.594	**0.718**	0.706
Medium Mammal	0.864	0.804	**0.894**
Carnivore	0.836	0.790	**0.854**
Omnivore/Herbivore	0.692	**0.824**	0.820
Insect	0.780	0.816	**0.830**
Non-insect	0.772	0.802	**0.826**
People	0.384	0.518	**0.564**
Tree	0.646	0.644	**0.708**
Flower	0.738	0.726	**0.772**
Fruit and Vegetable	**0.790**	0.758	0.762
Food Container	0.760	0.802	**0.804**
Electrical Device	0.772	0.832	**0.840**
Furniture	0.726	0.746	**0.764**
Building	0.832	0.814	**0.874**
Outdoor Scene	0.802	0.708	**0.842**
Vehicles Common	0.810	0.826	**0.832**
Vehicles Uncommon	0.908	0.892	**0.942**

Table 4: Per-Task performance on CIFAR-MTL.

Method	Average Accuracy (Baseline)	Average Accuracy (CO-TASK)
Single-Task	0.7321	
Multi-task	0.7547	0.7850 (↑ 3.03%)
DWA	**0.7589**	**0.7943 (↑ 3.54%)**
DTP	0.7373	0.7640 (↑ 2.67%)
TRL	**0.7528**	**0.7981 (↑ 4.53%)**

Table 5: Performance on CIFAR-MTL when combined with other multi-task learning methods.

Performance of Electrocardiogram Phenotyping

Table 7 shows the different multi-label metrics of the baseline methods and the proposed method on the ECG-P18 dataset. We can see that multi-task learning increased the overall performance compared to the single-task models, and the proposed CO-TASK framework increases all other metrics while achieving a similar $Recall_{Multi}$ metric. De-

Noise Ratio	Single Task	Multi-task	CO-TASK	Difference with Multi-task
0.0	0.7321	0.7547	0.7850	3.03%
0.1	0.6831	0.6831	0.7239	4.08%
0.3	0.5701	0.5204	0.5843	6.39%
0.5	0.4523	0.3549	0.4335	7.86%
0.7	0.3518	0.2548	0.3306	7.58%

Table 6: Robustness to label noise in CIFAR-MTL.

Metric	Single-Task	Multi-task	CO-TASK
Macro AUROC	0.9640	0.9720	**0.9750**
$Recall_{Multi}$	0.9268	0.9744	0.9745
Jaccard	0.4374	0.5643	**0.5868**
Exact	0.1925	0.2744	**0.3094**

Table 7: Multi-label performance on ECG-P18 dataset.

Model	Sampler	AUROC	Absolute Improvement
Single-Task	Major-Task	0.8240	Base
Multi-task	None	0.7399	↓ 8.41%
Multi-task	Major-Task	0.6585	↓ 16.55%
Multi-task	Task-Aware	0.8390	↑ 1.50%
Multi-task+DWA	Task-Aware	0.8414	↑ 1.74%
CO-TASK	Task-Aware	**0.8462**	↑ 2.22%

Table 8: Performance on ECG-EchoLVH dataset.

spite the extremely different imbalance ratios for different tasks, as shown in Table 1, the CO-TASK framework with task-aware imbalance data sampler can still manage to perform multi-task learning training and achieve better performance.

Performance of Predicting Echocardiogram Diagnostic from Electrocardiogram

An echocardiogram is much more expensive compare to an electrocardiogram, so it is of great interest to predict echocardiogram diagnostic from electrocardiogram signals. We used minor tasks from the raw doctor annotations that could contain a certain degree of label noise and apply the CO-TASK framework to train the multi-task learning model.

From Table 8, we can first observe the impact of the task-aware imbalance data sampler. Re-sampling the data according to the major task or train the model without any re-sampling will make the multi-task learning model perform poorly. With the proposed task-aware imbalance data sampler, the multi-task model can improve significantly compared to the single-task model using the same backbone. Table 8 also shows that the proposed CO-TASK framework improves the original multi-task learning model's performance.

Figure 4 shows the performance of the doctor's annotation compared to the ROC curve of different models. It shows that by using a deep learning model, we can achieve significant improvements over the doctor diagnostic from the electrocardiogram signal. When combined with other noisy annotations for the electrocardiogram signal to form a multi-task learning model, we can further improve the performance. Looking at the sensitivity of different models while maintaining the same specificity as the doctor annotations, the proposed CO-TASK framework increased the sensitivity by 3.0% compared to the original multi-task model and 7.1% compared to the single-task model.

Impact of Auxiliary Task Count

To understand the impact of different amounts of auxiliary tasks sampled, we did experiments on the CIFAR-MTL

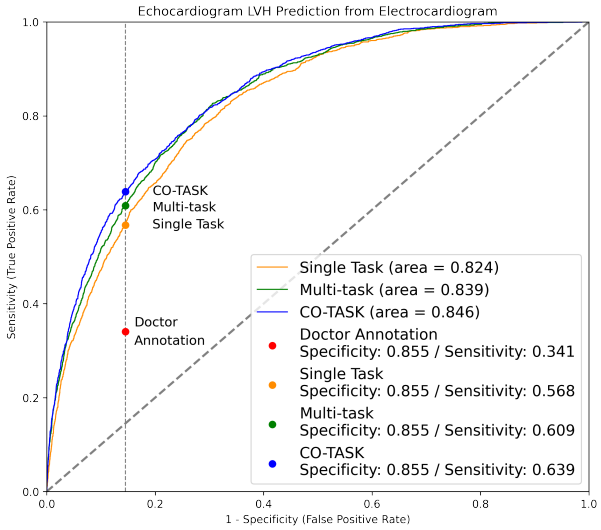

Figure 4: Receiver operating characteristic curve on the ECG-EchoLVH dataset.

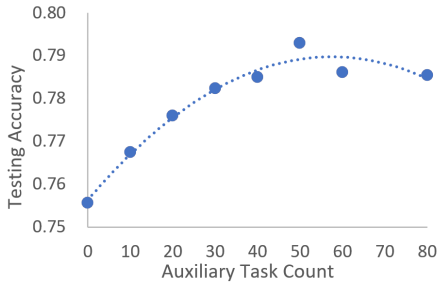

Figure 5: Impact of different auxiliary task count.

dataset. As seen in Figure 5, the overall performance increases as the auxiliary task count increase and starts to flatten out when there are more than 50 auxiliary tasks. This might be due to the current limitations of multi-task learning models on really large task counts. In the future, we could look into better ways of dealing with large task counts.

Conclusions

In this work, we provide an effective multi-task learning framework, CO-TASK, to improve performance on targeted tasks without the need for additional labeling effort and is robust to a certain degree of label noise. The proposed framework generates useful auxiliary tasks through the combination of existing task labels, which utilize the labels more effectively. The CO-TASK framework can provide additional performance gains when applied in parallel with other multi-task learning techniques. By incorporating the proposed task-aware imbalance data sampler, we can effectively deal with the different imbalance ratios for the different tasks in electrocardiogram phenotyping datasets. We demonstrated the effectiveness of the CO-TASK framework on both a benchmark multi-task image classification dataset and two real-world electrocardiogram phenotyping datasets.

Acknowledgments

This work was supported in part by the Taiwan Ministry of Science and Technology under grant no. MOST 109-2218-E-009-014 and MOST 109-2321-B-009-007. The authors would also like to thank Dr. Yu-Feng Hu and Dr. Chih-Min Liu, who provided valuable insight and expertise from the medical perspective that assisted this research.

References

Attia, Z. I.; Kapa, S.; Lopez-Jimenez, F.; McKie, P. M.; Ladewig, D. J.; Satam, G.; Pellikka, P. A.; Enriquez-Sarano, M.; Noseworthy, P. A.; Munger, T. M.; et al. 2019. Screening for cardiac contractile dysfunction using an artificial intelligence–enabled electrocardiogram. *Nature medicine* 25(1): 70–74.

Carreiras, C.; Alves, A. P.; Lourenço, A.; Canento, F.; Silva, H.; Fred, A.; et al. 2015–. BioSPPy: Biosignal Processing in Python. URL https://github.com/PIA-Group/BioSPPy/. [Online; Version 0.6.1; accessed Feb 1, 2020].

Chen, Z.; Badrinarayanan, V.; Lee, C.-Y.; and Rabinovich, A. 2018. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 794–803. PMLR.

Cirstea, R.-G.; Micu, D.-V.; Muresan, G.-M.; Guo, C.; and Yang, B. 2018. Correlated Time Series Forecasting using Multi-Task Deep Neural Networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018*, 1527–1530.

Ding, D. Y.; Simpson, C.; Pfohl, S.; Kale, D. C.; Jung, K.; and Shah, N. H. 2019. The Effectiveness of Multitask Learning for Phenotyping with Electronic Health Records Data. In *PSB*, 18–29. World Scientific.

Guendel, S.; Ghesu, F. C.; Grbic, S.; Gibson, E.; Georgescu, B.; Maier, A.; and Comaniciu, D. 2019. Multi-task Learning for Chest X-ray Abnormality Classification on Noisy Labels. *arXiv preprint arXiv:1905.06362* .

Guo, M.; Haque, A.; Huang, D.-A.; Yeung, S.; and Fei-Fei, L. 2018. Dynamic Task Prioritization for Multitask Learning. In *Proceedings of the European Conference on Computer Vision, ECCV 2018*, 270–287.

Hannun, A. Y.; Rajpurkar, P.; Haghpanahi, M.; Tison, G. H.; Bourn, C.; Turakhia, M. P.; and Ng, A. Y. 2019. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine* 25(1): 65–69.

Harutyunyan, H.; Khachatrian, H.; Kale, D. C.; Ver Steeg, G.; and Galstyan, A. 2019. Multitask learning and benchmarking with clinical time series data. *Scientific data* 6(1): 1–18.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, 770–778.

Johnson, A. E.; Pollard, T. J.; Shen, L.; Li-Wei, H. L.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L. A.; and Mark, R. G. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3(1): 1–9.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, 7482–7491.

Kwon, J.-M.; Jeon, K.-H.; Kim, H. M.; Kim, M. J.; Lim, S. M.; Kim, K.-H.; Song, P. S.; Park, J.; Choi, R. K.; and Oh, B.-H. 2020. Comparing the performance of artificial intelligence and conventional diagnosis criteria for detecting left ventricular hypertrophy using electrocardiography. *EP Europace* 22(3): 412–419.

Lee, H.; Hwang, S. J.; and Shin, J. 2019. Rethinking Data Augmentation: Self-Supervision and Self-Distillation. *arXiv preprint arXiv:1910.05872* .

Liang, M.; Yang, B.; Chen, Y.; Hu, R.; and Urtasun, R. 2019. Multi-Task Multi-Sensor Fusion for 3D Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, 7345–7353.

Liebel, L.; and Körner, M. 2018. Auxiliary Tasks in Multitask Learning. *arXiv preprint arXiv:1805.06334* .

Liu, S.; Davison, A.; and Johns, E. 2019. Self-Supervised Generalisation with Meta Auxiliary Learning. In *Advances in Neural Information Processing Systems, NeurIPS 2019*, 1677–1687.

Liu, S.; Johns, E.; and Davison, A. J. 2019. End-To-End Multi-Task Learning With Attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, 1871–1880.

Liu, X.; He, P.; Chen, W.; and Gao, J. 2019. Multi-Task Deep Neural Networks for Natural Language Understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, ACL 2019*, 4487–4496.

Ma, J.; Zhao, Z.; Chen, J.; Li, A.; Hong, L.; and Chi, E. H. 2019. SNR: Sub-Network Routing for Flexible Parameter Sharing in Multi-task Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI 2019*, volume 33, 216–223.

Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; and Chi, E. H. 2018. Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018*, 1930–1939.

Maninis, K.-K.; Radosavovic, I.; and Kokkinos, I. 2019. Attentive Single-Tasking of Multiple Tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, 1851–1860.

Meyerson, E.; and Miikkulainen, R. 2018. Beyond Shared Hierarchies: Deep Multitask Learning through Soft Layer Ordering. In *6th International Conference on Learning Representations, ICLR 2018*. URL https://openreview.net/forum?id=BkXmYfbAZ.

Misra, I.; Shrivastava, A.; Gupta, A.; and Hebert, M. 2016. Cross-Stitch Networks for Multi-task Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, 3994–4003.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d' Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems, NeurIPS 2019*, 8024–8035. Curran Associates, Inc. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Razavian, N.; Marcus, J.; and Sontag, D. 2016. Multi-task prediction of disease onsets from longitudinal laboratory tests. In *Machine Learning for Healthcare Conference*, 73–100.

Rosenbaum, C.; Klinger, T.; and Riemer, M. 2018. Routing Networks: Adaptive Selection of Non-Linear Functions for Multi-Task Learning. In *6th International Conference on Learning Representations, ICLR 2018*. URL https://openreview.net/forum?id=ry8dvM-R-.

Ruder, S.; Bingel, J.; Augenstein, I.; and Søgaard, A. 2019. Latent Multi-Task Architecture Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI 2019*, volume 33, 4822–4829.

Song, H.; Rajan, D.; Thiagarajan, J.; and Spanias, A. 2018. Attend and Diagnose: Clinical Time Series Analysis Using Attention Models. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI 2018*, volume 32, 4091–4098.

Standley, T.; Zamir, A.; Chen, D.; Guibas, L.; Malik, J.; and Savarese, S. 2020. Which Tasks Should Be Learned Together in Multi-task Learning? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, 9120–9132. PMLR.

Strezoski, G.; Noord, N. v.; and Worring, M. 2019. Many Task Learning With Task Routing. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV 2019*, 1375–1384.

Tellez, D.; Höppener, D.; Verhoef, C.; Grünhagen, D.; Nierop, P.; Drozdzal, M.; Laak, J.; and Ciompi, F. 2020. Extending Unsupervised Neural Image Compression With Supervised Multitask Learning. In *Medical Imaging with Deep Learning*, 770–783. PMLR.

Vandenhende, S.; Georgoulis, S.; Proesmans, M.; Dai, D.; and Van Gool, L. 2020. Revisiting Multi-Task Learning in the Deep Learning Era. *arXiv preprint arXiv:2004.13379* .

Xu, D.; Ouyang, W.; Wang, X.; and Sebe, N. 2018. PAD-Net: Multi-Tasks Guided Prediction-and-Distillation Network for Simultaneous Depth Estimation and Scene Parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, 675–684.

Yan, X.; Misra, I.; Gupta, A.; Ghadiyaram, D.; and Mahajan, D. 2020. ClusterFit: Improving Generalization of Visual Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, 6509–6518.

Zhao, Z.; Hong, L.; Wei, L.; Chen, J.; Nath, A.; Andrews, S.; Kumthekar, A.; Sathiamoorthy, M.; Yi, X.; and Chi, E. 2019. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019*, 43–51.