Highly Scalable Task Grouping for Deep Multi-Task Learning in Prediction of Epigenetic Events

Mohammad Shiri
Department of Computer Science
Old Dominion University
Norfolk, VA, USA
mshir001@odu.edu

Jiangwen Sun
Department of Computer Science
Old Dominion University
Norfolk, VA, USA
jsun@odu.edu

Abstract—DNNs trained for predicting cellular events from DNA sequence have become emerging tools to help elucidate biological mechanisms underlying associations identified in genome-wide association studies. To enhance the training, multi-task learning (MTL) has been commonly exploited in previous works where trained networks were needed for multiple profiles differing in either event modality or cell type. All existing works adopted a simple MTL framework where all tasks share a single feature extraction network. Such a strategy even though effective to a certain extent leads to substantial negative transfer, meaning the existence of a large portion of tasks for which models obtained through MTL perform worse than those by single-task learning. There have been methods developed to address such negative transfer in other domains, such as computer vision. However, these methods are generally with limited scalability. In this paper, we propose a highly scalable task grouping framework to address negative transfer by only jointly training tasks that are potentially beneficial to each other. The proposed method exploits the network weights associated with task-specific classification heads that can be cheaply obtained by one-time joint training of all tasks. Our results using a dataset consisting of 367 epigenetic profiles demonstrate the effectiveness of the proposed approach and its superiority over baseline methods.

Index Terms—Deep learning, Multi-task learning, Task grouping, Negative transfer, Epigenetic events prediction, Genetic variant prioritization

I. INTRODUCTION

With a large number of genome-wide profiles of varying cellular activities accumulated through both large consortia and projects, such as ENCODE [1], Roadmap [2], and GTEx\(^1\), and individual research laboratories\(^2\), training machine learning models to predict cellular activites from DNA sequence holds the premises of elucidating links between genome and phenotype. Several such attempts have been recently made to predict varying activities, such as chromatin accessibility and interaction, DNA methylation, histone modification, and transcription factor binding [3]–[7]. Due to their highly expressive power and effectiveness in feature learning from sequence data, the majority of existing works exploited deep neural networks (DNNs).

On one hand, cellular activities are cell type specific, indicating the same needs to be true in models trained for predicting these activities. On the other hand, cellular constituents closely interact with each other in a unified system; therefore, cellular activities of different kinds are likely related and may be co-regulated [8]. Such shared regulations can be leveraged through multi-task learning (MTL) to enhance training, which is important due to the data-hungry nature of DNNs. Indeed, in a few studies [3], [6], models jointly trained for predicting events of different modalities showed improved performance on average. However, all existing studies that exploited MTL adopted a simple joint training framework, having all models sharing the same feature extraction component, known as hard parameter sharing [9]. As indicated in our experimental results, such a strategy, despite enhancing the learning on average, leads to models with decreased performance for many tasks due to conflicting training objectives, known as task interference [10].

Many methods have been recently proposed to address negative transfer due to task interference in MTL with hard parameter sharing. These methods generally follow four orthogonal research directions: task-specific routing through a network shared across all tasks [11]–[14], branching out from shared networks [15]–[18], task grouping [19]–[21], and training objective manipulation [22]–[25]. Task grouping based methods explicitly study task relationships and only jointly train tasks that are potentially beneficial to each other, leading to better applicability and easier interpretability compared to others. However, all existing task grouping based methods were originally developed for learning problems either in computer vision [19], [20] or natural language processing [21] where the number of jointly trained tasks are in relatively small amounts (up to 107 tasks in [21]). Therefore, these methods generally lack scalability and are not readily applicable for handling epigenetic events predictions considered in the present study, where the number of tasks can easily reach several hundreds [3] and even many thousands [7].

---

\(^1\)https://www.gtexportal.org
\(^2\)Data are available through public repositories, such as NCBI GEO: https://www.ncbi.nlm.nih.gov/geo
This paper presents a highly scalable task grouping framework for deep MTL that jointly trains DNNs for a massive number of tasks. Our method exploits the network weights associated with task-specific classification heads in a simple MTL framework where all tasks share the same feature extraction component. Due to the shared feature extraction component, the representation (or embedding) of any input is identical across tasks. This, together with the assumption of classification heads of tasks having the same network architecture (which is the case in all previous works on functional genomic events prediction where MTL is utilized [4], [7], [26]) mean that weights in the classification head completely identify individual tasks and importantly indicate how they exploit the shared representation of input data. Therefore, they are good sources of information to use for studying the relationship among tasks to determine their grouping. In summary, we propose to identify task groups by performing cluster analysis using the task-specific classification weights obtained from one-time joint training of all tasks, followed by separate in-group joint training to address negative transfer.

To evaluate the proposed framework, we composed a dataset obtained from ENCODE and NCBI GEO that consists of a total of 367 profiles of cellular events of three modalities in a number of cell types related to the human central nervous system. The experimental results demonstrate the effectiveness of the proposed method in addressing negative transfer, with 3.20% improved F1-score compared to simple MTL. In addition, the comparison to the task grouping by either event modalities or cell types indicates the superiority of the proposed grouping over the two with improved F1-scores of 1.32% and 2.55%, respectively.

II. Methodology and Materials

A. Highly scalable task grouping framework

The key in task grouping to promote effective joint training while minimizing negative transfer is to find appropriate numerical representations of tasks followed by a proper definition of affinity among tasks. It has been shown previously that just considering the distribution of the class label often does not lead to a useful similarity measure, because similar (or different) class label distribution among tasks does not necessarily imply positive (or negative) transfer in joint training [10]. Therefore, the challenge here is to find a similarity measure that encourages tasks that can learn from each other to be grouped together and those that compete with each other to be grouped separately. Another challenge is that the computation cost of the similarity needs to be reasonably low. So, it can be applicable to learning problems where the number of tasks can be in several hundreds or even many thousands, such as the epigenetic event prediction concerned in this study.

Here, we propose a task grouping framework that is highly scalable and effective in reducing negative transfer without sacrificing positive transfer among tasks in joint training. All existing works that train DNNs by MTL to predict functional genomic events adopted a simple hard parameter sharing framework [3], [4], [7], [26]. As illustrated in the leftmost component of Figure 1, this framework includes a feature extractor shared by all tasks followed by task-specific classification heads. The feature extractor is designed to learn sequence patterns in the input DNA fragment that are predictive of the targeted functional genomic events. Even though the detailed implementation of the feature extractor varies across studies, it is typically a convolution neural network or its variants, for example with incorporating self-attention mechanisms [7]. The task-specific classification head is often a subnetwork that consists of one or two fully connected layers optionally (if classification) followed by a softmax layer to produce predicted probabilities.

Due to the shared feature extractor, the inputs that task-specific heads receive are identical across all tasks. Intuitively, the way of such inputs being exploited in each classification head for task-specific prediction can be used to determine the task relationship. More specifically, tasks that exploit the extracted features in ways that are more similar have a higher potential to benefit each other’s training. Given that all classification heads have identical network architecture, what determining the way of inputs being used is the set of network parameters, represented by \( \mathcal{W}_i \) for task \( i \). Therefore, we propose to exploit \( \mathcal{W}_i \)'s to construct the numerical representation of tasks for task grouping, denoted by column vectors \( \mathbf{e}_i \)'s, \( \mathbf{e}_i \in \mathbb{R}^d \). Depending on the specific network architecture of the classification head, \( \mathcal{W}_i \) may be a vector (i.e., weights associated with the last fully connected layer) with/without additional weight matrices. A set of vectors can be composed for each task \( i \) by including the vector and vectorized additional weight matrices if any. Then, the \( \mathbf{e}_i \) can be constructed by the concatenation of either all vectors in the set or one of its subsets. In our experiments, the weight vector associated with the last fully connected layer was used.

Let \( \mathbf{E} \in \mathbb{R}^{T \times d} \) represent the matrix composed by stacking all obtained \( \mathbf{e}_i \)'s in rows, where \( T \) is the total number of tasks. Cluster analysis is performed to put tasks into groups by taking \( \mathbf{E} \) as input. A variety of clustering algorithms can be potentially used here, such as hierarchical clustering and DBSCAN. Following a previous study [27], in this work the use of k-means was explored. K-means identifies clusters that minimize the sum of squared distance between individual data points to the centroid (i.e., mean) of their assigned clusters, performing well when the desired clusters are in globular shape. K-means is an iterative algorithm, repeatedly assigning data points to clusters according to centroids from the previous iteration followed by updating the centroids. It is fast since each iteration in the algorithm has a linear time complexity, \( O(T) \), which adds to the scalability of the
In summary, as illustrated in Figure 1, the proposed framework for grouped multi-task learning consists of four consecutive steps: (1) joint training all tasks with hard parameter sharing to obtain trained task-specific classification heads; (2) composing task representation matrix $E$ based on weights in obtained classification heads; (3) performing cluster analysis on $E$ to identify task groups; and (4) joint training tasks within each group from scratch. The high scalability of the proposed framework is mainly due to that only one-time joint training of all tasks is needed for the task grouping and also importantly there is no need for extra storage space.

**B. Study data**

To validate the proposed framework for grouped MTL and obtain models for future elucidation of functional impacts of genetic variants identified in genome-wide association studies (GWASs) of psychiatric disorders, we compiled a dataset that consists of a total of 367 profiles of epigenetic events of three modalities: chromatin accessibility (CA), histone modification (HM), and transcription factor binding (TFB) in varying cell types related to human central nervous system and tissues in different brain locations. The majority of the profiles were obtained from ENCODE project [1], with the remaining from NCBI GEO with accession numbers: GSE113483, GSE96615, and GSE96949. For profiles from ENCODE, we directly downloaded files that contain narrow peaks; while for those from NCBI GEO, we downloaded respective FastQ files from SRA and followed the procedures described previously by us [28] to obtain peaks.

This dataset includes chromatin accessibility profiles by two different techniques: DNase-seq and ATAC-seq. According to the cell type, we assigned each of the profiles into six groups: (1) neuron group including profiles with labels of neuron, bipolar neuron, and excitatory neuron; (2) glial cell group including profiles with labels of astrocytes and glial cells; (3) progenitor group including profiles with labels of neural progenitor cell, neuronal stem cell, radial glial cell, neurosphere, and ecto neural progenitor cell; (4) (cerebral) circulation group including profiles with labels of epithelial cell, endothelial cell, and pericyte; (5) cancer group including profiles of cancer cells: SK-N-SH, PFSK-1, SK-N-MC, BE2C, D721Med, D341Med, Daoy, M059J, SK-N-DZ, H4, and A172; and (6) mix group including profiles of the bulk of brain tissue that can contain multiple cell types.

The distribution of profiles across the three event modalities and the six cell groups is provided in Table I. HM has the largest number of profiles ($N = 191$) among the three event modalities followed by TFB ($N = 90$) and CA ($N = 86$). Among the CA profiles, there are 41 obtained with ATAC-seq and 45 with DNase-seq. In the cell group end, cancer is the largest group containing 107 profiles followed by mix, neuron, progenitor, glial cell, and circulation in sequence.

**III. Experiment setup**

**A. Network architecture**

We adopted a network architecture (Figure 2) that has been used previously for predicting functional genomic events with success [3]. The feature extractor consists of three stacked convolution blocks and the classification head includes two consecutive fully connected layers. All convolution operations were done with filters of a uniform size of 8 and step size of 1. The number of filters

![Fig. 1. Overview of the proposed framework for grouped multi-task learning. CA: chromatin accessibility; HM: histone modification; TFB: transcription factor binding.](image-url)
We excluded bins that have no positive label in the dataset size while minimizing the loss of discriminative information in the respective profile or otherwise, negative. To reduce peaks in a profile for at least 50 bps was labeled positive according to the called peaks in each profile. More specifically, a bin that contains an overlap with any of the peaks in a profile for at least 50 bps was labeled positive in the respective profile or otherwise, negative. To reduce dataset size while minimizing the loss of discriminative signals, we excluded bins that have no positive label in any of the profiles. These lead to a dataset consisting of a total of 10,386,311 examples, with 367 sets of binary labels (corresponding to the 367 profiles). Networks were trained to take the 1001bp DNA fragment centered around a respective 200bp bin to predict the corresponding 367 binary labels. The dataset was partitioned into three disjoint subsets for training, validation, and testing, respectively. Specifically, all the 947,056 examples from chromosomes 8 and 9 were reserved for testing, with the 6,5531 examples between coordinates: 16,059,401 and 32,570,401 on chromosome 7 for validation and all remaining for training.

All models were trained up to 1,920k steps with an early stopping mechanism in place to minimize the training time. One round of forward and backward passing of a mini-batch of training data was considered a step. We used mini-batch size of 64 and calculated the validation loss for every 16k steps. The early stopping works to terminate the training after completing a certain number of more validation steps (denoted by \( \delta \)) from the one where the minimum validation loss was observed. To determine a proper \( \delta \) to use, the full course of training was carried out for 15 representative profiles, five for each of the three event modalities with varying proportions of positive examples: small, medium, and large. The analysis of the collected validation loss values indicates that \( \delta = 25 \) was the best value to use for all profiles.

The network training was carried out using the Pytorch deep learning framework. Attempts were made to tune the learning rate with all other hyperparameters set to their respective default values. Due to the expensive training cost (average 16 hours per profile using NVIDIA GPU V100) and the large number of profiles, it is impractical to perform fine learning rate tuning for each profile in STL. Instead, fine-tuning was performed for the aforementioned representative profiles, which leads to a fixed learning rate of 0.01 to use for all single-task training. For all joint training, a tuning strategy of two phases was used: an initial search in the range from 0.05 to 0.35 with a step size of 0.05 followed by searching around the best value found in the previous phase with a step size of 0.025.

There is a very large class imbalance in our dataset, with the vast majority of the profiles having a proportion of positive examples well below 2%. It is known that accuracy is biased in the case of large class imbalance and cannot reflect the number of intrinsic patterns that a model learned from the data for the concerned prediction. Also, due to such a large class imbalance, inflation in the area under ROC (AUC) was observed in our results, failing to differentiate the performance of compared methods. As a result, in this paper, we used F1-score to evaluate the performance of all models.

IV. Results

A. Task grouping by proposed method

As illustrated in Figure 1, a network was trained jointly for all 367 tasks to obtain task representation for sub-
sequent cluster analysis. To visualize the distribution of tasks in space, we performed principal component analysis (PCA) on the obtained task representation matrix \( E \). The distribution of tasks in space spanned by the first two principal components is shown in Figure 3A. According to this figure, there is a clear clustering tendency among the tasks. Figures 3B and 3C indicate that neither the event modality of profiles nor cell type aligns very well with the clustering tendency and better aligning is seen with the former than the latter.

Most clustering algorithms including k-means do not automatically determine the number of clusters (\( K \)) in a dataset, which needs to be prespecified. In our experiment, k-means was run multiple times with different \( K \) values, ranging from 2 to 10 with a step size of one. For each obtained cluster solution, we trained models by grouping tasks accordingly and explored how the model performance varies along with different choices of \( K \). The significant improvement in the performance of models predicting chromatin accessibility and transcription factor binding when \( K \) goes from 2 to 4 (Figure 4A) suggests the existence of substantial negative transfer when \( K \) is below 4. Figure 4A also indicates when \( K \) goes above 4, the drop in cluster size (i.e., number of tasks in joint training) generally hurts the performance, especially among models for predicting transcription factor binding. Another important note is that among the three types of functional genomic events, chromatin accessibility is the easiest one to predict with an average F1-score above 0.42. This is in contrast to only around 0.34 for histone modification and transcription factor binding (Figure 4A).

The four-cluster solution contains two relatively large groups: Group 1 (\( N = 130 \)) and 2 (\( N = 151 \)) and another two groups in smaller sizes: Group 3 (\( N = 34 \)) and 4 (\( N = 52 \)). The distribution of all profiles across the four task groups is illustrated in Figure 3D, which shows the task grouping generally follows the clustering tendency in the two-dimensional space spanned by the first two principal components of matrix \( E \). Table II provides the distribution of profiles by event modalities and cell types across the four task groups obtained by running k-means. As shown in this table, Group 1 is primarily a histone modification group with the majority of the tasks (71.54%) predicting histone modification. Interestingly, all other tasks in this group are those to predict chromatin accessibility profiled with ATAC-seq. Group 2 is the largest and contains tasks to predict events of diverse modalities, with over half (56.67%) of the tasks predicting transcription factor binding. This group contains all the 45 tasks that predict chromatin accessibility profiled with DNase-seq. Group 3 is the smallest group among the four, containing only histone modification tasks. Similar to Group 3, Group 4 is another histone modification group with all but five tasks predicting histone modification. It is interesting to note that tasks predicting chromatin accessibility were very cleanly split into two groups: Group 1 and 2 by profiling techniques being used, i.e., ATAC-seq vs DNase-seq. This may suggest the existence of systemic but distinct technical bias in the two profiling techniques.

Compared to event modalities, the concentration of cell types in identified task groups is much less clear (bottom half of Table II). There are two groups with a noticeable concentration of specific cell types. One is Group 2, over half (52.98%) of the tasks to predict varying epigenetic events in cancer cells. Group 4 is the other one, containing primarily tasks predicting events in neurons assuming a large portion of cells in bulk tissue are neurons. In terms of the distribution of cell types, Group 3 looks quite similar to Group 4, except that it is slightly less concentrated by tasks associated with neurons.

**Table II**

| Profile Category | Group 1 \((N = 130)\) | Group 2 \((N = 151)\) | Group 3 \((N = 34)\) | Group 4 \((N = 52)\) |
|------------------|------------------------|------------------------|------------------------|------------------------|
| **Event modality** |                        |                        |                        |                        |
| CA (ATAC)        | 37                     | 4                      | 0                      | 0                      |
| CA (DNase)       | 0                      | 45                     | 0                      | 0                      |
| HM               | 93                     | 17                     | 34                     | 47                     |
| TFB              | 0                      | 85                     | 0                      | 5                      |
| **Cell type**    |                        |                        |                        |                        |
| Cancer cell      | 12                     | 80                     | 6                      | 9                      |
| Cerebral circulation | 3                     | 5                      | 1                      | 1                      |
| Glial cell       | 25                     | 8                      | 3                      | 3                      |
| Mix              | 36                     | 19                     | 10                     | 20                     |
| Neuron           | 37                     | 23                     | 8                      | 10                     |
| Progenitor cell  | 17                     | 16                     | 6                      | 9                      |

**B. Comparison with baseline methods**

The performance of models resulting from baseline methods is illustrated in Figure 4B, which also includes the proposed method (specifically, from the four-cluster solution, KMTL) for easy comparison. To further facilitate the comparison, we computed its improvement in model performance over baselines and provided the results in Table III. KMTL has the best overall performance, with a high level of improvement compared to single-task learning (STL, 7.63%), simple multi-task learning (SMTL, 3.20%), and cell-grouped multi-task learning (CMTL, 2.55%). By event modality, KMTL helps the most in the tasks predicting transcriptional factor binding (TFB), having 11.66%, 12.05%, 3.82%, and 0.93% improvement over STL, SMTL, CMTL, and event-grouped multi-task learning (EMTL), respectively. It is interesting to note that even though CMTL has the worst overall performance among the grouped MTL methods, it has the best performance for predicting histone modification (HM). This result suggests there is still room to improve by introducing flexibility into the task grouping, for example factoring cell type information.

**V. Conclusion**

In this paper, we developed a highly scalable framework for grouped multi-task learning to address negative transfer when simultaneously training DNNs for a massive
number of epigenetic events. Our results demonstrate the effectiveness of the proposed framework and its superiority over baseline methods. Even though our framework was developed in the setting of predicting epigenetic events, it is a general multi-task learning approach and certainly applicable to other domains. One immediate future research would be to exploit obtained models in this study to help elucidate the functional impact of genetic variants identified in GWASs of psychiatric disorders.

ACKNOWLEDGMENT

This work was supported by NIH grant R01HL157519.

REFERENCES

[1] E. Feingold and L. Pachter, “The encode (encyclopedia of dna elements) project,” Science, vol. 306, pp. 636–640, 2004.
[2] A. Kundaje et al., “Integrative analysis of 111 reference human epigenomes,” Nature, vol. 518, no. 7539, pp. 317–330, 2015.
[3] J. Zhou and O. G. Troyanskaya, “Predicting effects of noncoding variants with deep learning–based sequence model,” Nature methods, 2015.
[4] J. Zhou et al., “Deep learning–based ab initio prediction of variant effects on expression and disease risk,” Nature genetics, 2018.
[5] D. R. Kelley et al., “Sequential regulatory activity prediction across chromosomes with convolutional neural networks,” Genome research, 2018.
[6] D. R. Kelley, “Cross-species regulatory sequence activity prediction,” PLoS computational biology, vol. 16, no. 7, 2020.
[7] Ž. Avsec et al., “Effective gene expression prediction from sequence by integrating long-range interactions,” Nature methods, 2021.
[8] J. Ernst et al., “Large-scale imputation of epigenomic datasets for systematic annotation of human epigenomes,” Nature Biotechnology, 2015.
[9] S. Ruder, “An overview of multi-task learning in deep neural networks,” arXiv preprint arXiv:1706.05098, 2017.
[10] Z. Kang, K. Grauman, and F. Sha, “Learning with whom to share in multi-task feature learning,” in ICML, 2011.
[11] C. Rosenbaum et al., “Routing networks: Adaptive selection of non-linear functions for multi-task learning,” in ICLR, 2018.
[12] F. Bragman et al., “Stochastic filter groups for multi-task cnns: Learning specialist and generalist convolution ...” in ICCV, 2019.
[13] G. Strezoski et al., “Many task learning with task routing,” ICCV, pp. 1375–1384, 2019.
[14] X. Sun et al., “Adashare: Learning what to share for efficient deep multi-task learning,” NeurIPS, 2020.
[15] M. Long et al., “Learning multiple tasks with multilinear relationship networks,” in NIPS, 2017, pp. 1595–1604.
[16] Y. Lu et al., “Fully-adaptive feature sharing in multi-task networks with applications in person attribute ...” in CVPR, 2017.
[17] P. Guo, C.-Y. Lee, and D. Ullrich, “Learning to Branch for Multi-Task Learning,” in ICML, 2020.
[18] J. Lu et al., “12-in-1: Multi-Task Vision and Language Representation Learning,” in CVPR, 2020, pp. 10434–10443.
[19] T. Standley et al., “Which tasks should be learned together in multi-task learning?” in ICML, 2020, pp. 9120–9132.
[20] C. Fifty et al., “Efficiently identifying task groupings for multi-task learning,” NeurIPS, vol. 34, pp. 27 503–27 516, 2021.
[21] V. Aribandi et al., “ExT5: Towards Extreme Multi-Task Scaling for Transfer Learning,” in NeurIPS, 2021.
[22] Z. Chen et al., “Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks,” in ICML, 2018.
[23] B. Liu et al., “Conflict-Averse Gradient Descent for Multi-task Learning,” in NeurIPS, 2021.
[24] T. Yu et al., “Gradient surgery for multi-task learning,” NeurIPS, 2020.
[25] A. Javaloy and I. Valera, “RotoGrad: Gradient Homogenization in Multitask Learning,” in ICLR, 2022.
[26] C. Sharma et al., “DeepCpG: accurate prediction of single-cell DNA methylation states using deep ...” in Genome Biology, 2017.
[27] L. Jacob, J.-p. Vert, and F. Bach, “Clustered multi-task learning: A convex formulation,” NIPS, 2008.
[28] C. Li et al., “The landscape of accessible chromatin in quiescent cardiac fibroblasts and cardiac fibroblasts activated after myocardial infarction,” Epigenetics, 2021.