ASM2TV: An Adaptive Semi-supervised Multi-Task Multi-View Learning Framework for Human Activity Recognition

Zekai Chen[1], **Xiao Zhang**[2*], **Xiuzhen Cheng**[2]
[1]George Washington University
[2]School of Computer Science and Technology, Shandong University
zech_chan@gwu.edu, xiaozhang@sdu.edu.cn, xzcheng@sdu.edu.cn

Abstract

Many real-world scenarios, such as human activity recognition (HAR) in IoT, can be formalized as a multi-task multi-view learning problem. Each specific task consists of multiple shared feature views collected from multiple sources, either homogeneous or heterogeneous. Common among recent approaches is to employ a typical hard/soft sharing strategy at the initial phase separately for each view across tasks to uncover common knowledge, underlying the assumption that all views are conditionally independent. On the one hand, multiple views across tasks possibly relate to each other under practical situations. On the other hand, supervised methods might be insufficient when labeled data is scarce. To tackle these challenges, we introduce a novel framework *ASM2TV* for semi-supervised multi-task multi-view learning. We present a new perspective named gating control policy, a learnable task-view-interacted sharing policy that adaptively selects the most desirable candidate shared block for any view across any task, which uncovers more fine-grained task-view-interacted relatedness and improves inference efficiency. Significantly, our proposed gathering consistency adaption procedure takes full advantage of large amounts of unlabeled fragmented time-series, making it a general framework that accommodates a wide range of applications. Experiments on two diverse real-world HAR benchmark datasets collected from various subjects and sources demonstrate our framework's superiority over other state-of-the-arts. The detailed codes are available at https://github.com/zachstarkk/ASM2TV.

Introduction

In IoT mobile sensing world, with the rapidly growing volume of Internet-connected sensory devices, the IoT generates massive data characterized by its velocity in terms of spatial and temporal dependency (Mahdavinejad et al. 2018). The sensing data collected from heterogeneous devices are multiple modalities, and multi-task multi-view learning (M2TVL) provides a useful paradigm. For instance, in human activity recognition (HAR), sensor monitoring data from various on-body positions can be multi-view. M2TVL aims to improve accuracy by learning multiple objectives of tasks with multiple shared view features collected from diverse sources simultaneously (Caruana 1998; He and Lawrence 2011). Compared with single-task multi-view learning, the M2TVL paradigm can further improve the training efficiency and reduce inference cost while promoting the generalization effect (Baxter 1997; Ruder 2017; Wu, Zhan, and Jiang 2018; Sun et al. 2020) by learning shared representations across related tasks and views.

When addressing M2TVL in mobile sensing problems, we encounter several challenges. (1) Since both views and tasks can be either heterogeneous or homogeneous, one instinctive question is *which views should share across which tasks under what circumstances* to avoid harmful interference or negative transfer (Kang, Grauman, and Sha 2011; Standley et al. 2020) and optimally reinforce the positive learning. Most prior works (Zhang and Huan 2012; Jin et al. 2013, 2014; Lu et al. 2017) utilized soft-sharing constraints and considered each view separately by dividing the M2TVL problem into multiple multi-task learning problems under different views while ignoring the more fine-grained *task-view-interacted relatedness*. (2) Also, as the total number of views and tasks grows, the computation cost proliferates, *making the soft-sharing scheme more computationally limited*. (3) Moreover, *obtaining labeled data in the real mobile sensing world is costly*, while the unlabeled data is prevalent and easily accessible (Mahdavinejad et al. 2018). Thus, a semi-supervised learning approach is strongly preferred to utilize these unlabeled data adequately.

Along this line, we propose a novel semi-supervised multi-task multi-view learning framework *ASM2TV* to address the above problems (see Figure 1). Considering computation efficiency, we inherit the spirit of the *hard-parameter sharing*. Unlike previous M2TVL hard-sharing schemes, we pre-define a set of commonly shared bottoms composed of hidden layers for tasks across all views instead of designing specific shared layers for each view. The main idea is to learn an *adaptive gating control policy* that determines which shared block should be open-gated and others are closed for any view across any task. In other words, the model can adaptively learn what views across which tasks should share common knowledge via the same block to gain the maximum positive transfer benefits. We also design a regularization term that encourages each view-specific function to agree with the view-fusion function as much as possible. Considering most sensor devices generate data con-

[*]Corresponding author

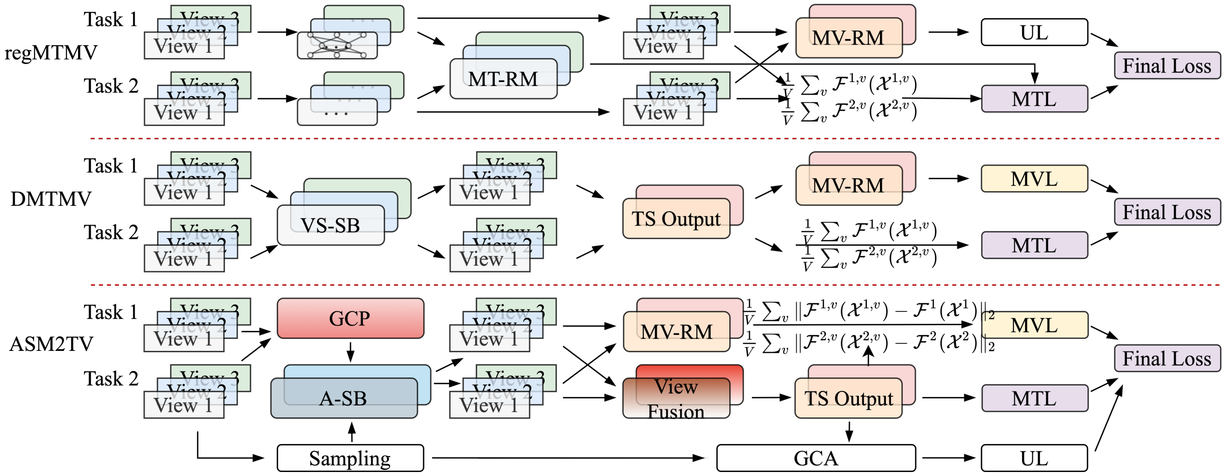

Task 1
regMTMV
Task 2
View 3
View 2
View 1
MT-RM
MV-RM
UL
Final Loss
$\frac{1}{V}\sum_v \mathcal{F}^{1,v}(\mathcal{X}^{1,v})$
$\frac{1}{V}\sum_v \mathcal{F}^{2,v}(\mathcal{X}^{2,v})$
MTL
DMTMV
VS-SB
TS Output
MVL
ASM2TV
GCP
A-SB
MV-RM
View Fusion
$\frac{1}{V}\sum_v \|\mathcal{F}^{1,v}(\mathcal{X}^{1,v}) - \mathcal{F}^{1}(\mathcal{X}^{1})\|_2$
$\frac{1}{V}\sum_v \|\mathcal{F}^{2,v}(\mathcal{X}^{2,v}) - \mathcal{F}^{2}(\mathcal{X}^{2})\|_2$
Sampling
GCA

Figure 1: An overview of our framework by comparing with other approaches. For simplicity, we consider an M2TVL scenario with two tasks consisted of three views each. Early M2TV methods use the soft-sharing (e.g., regMTMV) via controlling the similarity between functions built on top of each view across each task. Recent methods such as DMTMV applies the hard-sharing mechanism to share initial bottom layers across tasks under every single view. However, these typical M2TV methods focus modeling relationship on either tasks or views separately while ignoring the task-view-specific relatedness. In our proposed ASM2TV, we design an adaptive gating control policy that learns to determine which views across which tasks should share information to maximize the positive transfer gain combined with a new multi-view regularization based on the view-fusion block. Also, we devise a novel gathering consistency adaption algorithm for semi-supervised learning to take full advantage of unlabeled fragmented time series.

tinuously and intermittently, we devise a gathering consistency adaption algorithm for semi-supervised learning on fragmented time series. Based on a particular sampling strategy concerning the temporal data characteristics, we further combine it with the consistency training (Bachman, Alsharif, and Precup 2014; Rasmus et al. 2015; Laine and Aila 2017; Xie et al. 2020) for unsupervised learning. Additionally, a task-dependent uncertainty modeling strategy is applied to prevent unsupervised loss from being trivial. The main contributions of our work are as follows:

- We propose a novel multi-task multi-view learning framework *ASM2TV* that automatically learns the sharing schemes across different views and tasks to strengthen the positive learning.
- We design a novel regularization term to enforce models built on every view to agreeing with the view-fusion function, which further benefits the global optimization.
- We devise the gathering consistency adaption algorithm combined with a task-dependent uncertainty modeling strategy for semi-supervised learning on fragmented time series. It flexibly takes advantage of a large amount of unlabeled data, making our *ASM2TV* a general framework for a wide range of applications.

Related Work

Multi-Task Learning. MTL aims to improve the generalization and performance by weighing the training knowledge among multiple tasks. The existing methods of MTL have often been partitioned into two groups with a familiar dichotomy: hard parameter sharing vs. soft parameter sharing. Hard parameter sharing is the practice of sharing model weights between multiple tasks, so that each weight is trained to jointly minimize multiple loss functions (Huang et al. 2015; Kokkinos 2017; Ranjan, Patel, and Chellappa 2019; Bilen and Vedaldi 2016; Chen et al. 2020, 2021). Under soft parameter sharing, different tasks have individual task-specific models with separate weights, but the distance between the model parameters of different tasks is added to the joint objective function, such as Cross-stitch (Misra et al. 2016), Sluice (Ruder et al. 2019) and NDDR (Gao et al. 2019), consist of a network column for each task, and define a mechanism for feature sharing between columns.

Multi-Task Multi-View Learning. The research on multi-task multi-view learning has attracted wide attention in recent years. GraM2 (He and Lawrence 2011) proposed a graph-based framework, in which an effective algorithm was proposed to optimize the framework. However, GraM2 can only deal with the non-negative feature values. The reg-MVMT (Zhang and Huan 2012) algorithm was proposed based on the idea of co-regularization, assuming that different view of prediction models should be consistent. Along this line, a more generalized algorithm CSL-MTMV (Jin et al. 2013) was proposed based on the assumption that multiple related tasks with the same view should be shared in the low-dimensional common subspace. MAMUDA (Jin et al. 2014) was proposed for heterogeneous tasks, in which the shared structure and task-specific structures can be combined into a unified formulation. DMTMV (Wu, Zhan, and Jiang 2018) was a multi-task multi-view learning framework including shared feature network, specific feature network, and task network. Nevertheless, these methods assume independence among views and tasks, which is improper in many real-life scenarios. Therefore, we aim to propose a new framework to address these problems.

Human Activity Recognition. The existing approaches for sensor based HAR usually adopted kinds of deep learning techniques, for instance, LSTM learners (Guan and Plötz 2017) or combinations of recurrent models and CNN networks (Yao et al. 2017a; Chen et al. 2018). However, none of the previous methods addressed HAR from a semi-supervised multi-task multi-view learning perspective, which is the focus of our work.

ASM2TV: Adaptive Semi-Supervised Multi-Task Multi-View Learning

Given a set of T tasks with each task containing V different views, we define $\mathcal{X}^{t,v}$ as the original inputs from v-th view in t-th task. We aim to seek an adaptive sharing scheme that best describes the internal task-view-interacted relatedness (Jin et al. 2014; Lu et al. 2017) instead of investigating task-task or view-view association. From the perspective of hard-sharing, we seek to know which views should be shared across which tasks to optimally augment the positive learning while avoiding the negative transfer (Kang, Grauman, and Sha 2011; Standley et al. 2020). The computation efficiency for scalable M2TVL is also considered as we mainly target IoT mobile computing scenarios.

Task-View-Interacted Gating Control Policy

Following the spirit of *hard-parameter sharing*, we predefine a set of initially shared blocks (or shared bottoms) for any input from each view and task to potentially execute or share across (see Figure 2). We use a random categorical variable $\mathbf{z}^{t,v}$ that determinates whether blocks are open-gated to any inputs $\mathcal{X}^{t,v}$ from the v-th view of the t-th task. It can also be viewed as a soft-clustering process since similar representations attempt to share across the same block (see Figure 5). As long as the learned pattern produce **positive** feedback, different views across all tasks would be encouraged to share with each other, making it no longer limited to knowledge sharing under the same view. Inspired by (Maddison, Tarlow, and Minka 2014; Jang, Gu, and Poole 2017), we adopt the Gumbel-Softmax Sampling technique to optimize our gating control policy jointly with the model parameters θ through standard backpropagation. Instead of constructing a policy network to form a specific policy for each input mini-batch, we employ a universal learnable policy to make structural decisions to evade the explosion of parameter complexity restricted by the computation resources.

Gumbel-Softmax Sampling Trick. The sampling process of discrete data from a categorical distribution is originally non-differentiable, where typical backpropagation in deep neural networks cannot be conducted. (Maddison, Tarlow, and Minka 2014; Jang, Gu, and Poole 2017) proposed a differentiable substitution of discrete random variables in stochastic computations by introducing Gumbel-Softmax distribution, a continuous distribution over the simplex that can approximate samples from a categorical distribution. In our gating control policy with a total number of N candidate shared blocks, we let $\mathbf{z}^{t,v}$ be the gating control variable for inputs $\mathcal{X}^{t,v}$ with open-gate probabilities for each block as $\pi_1^{t,v}, \cdots, \pi_N^{t,v}$, where $\mathcal{X}^{t,v}$ are inputs from the v-th view of the t-th task and $\pi_i^{t,v}, \forall i \in \{1, \cdots, N\}$ represents the probability that the i-th shared block would be open to $\mathcal{X}^{t,v}$. Similarly, by Gumbel-Max trick, we can sample any block's open-or-not gating strategy $z^{t,v}$ for inputs $\mathcal{X}^{t,v}$ with:

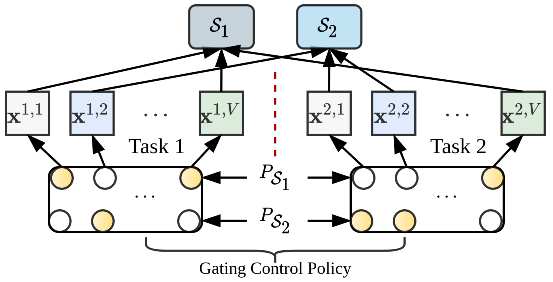


Figure 2: Suppose we have two tasks consisted of V views each and two shared blocks $\mathcal{S}_1$ and $\mathcal{S}_2$. Our gating control policy's main idea is to use the Gumbel-Softmax Sampling strategy to sample a random categorical vector for determining which views across which tasks to share through the same shared block. For $\mathbf{x}^{1,v}$, since $\mathcal{P}_{\mathcal{S}_1} > \mathcal{P}_{\mathcal{S}_2}$ (yellow circles), the first shared block will be chosen to execute.

$$z^{t,v} = \arg\max_i (\log \pi_i^{t,v} + g_i^{t,v}) \tag{1}$$

where $g_i, \cdots, g_N$ are i.i.d samples drawn from a standard Gumbel distribution which can be easily sampled using inverse transform sampling by drawing $u \sim \text{Uniform}(0,1)$ and computing $g = -\log(-\log u)$. We further substitute this $\arg\max$ operation, since it is not differentiable, with a SOFTMAX reparameterization trick, also known as Gumbel-Softmax trick, as:

$$z_i^{t,v} = \frac{\exp((\log \pi_i^{t,v} + g_i^{t,v})/\tau)}{\sum_{j=1}^{N} \exp((\log \pi_j^{t,v} + g_j^{t,v})/\tau)} \tag{2}$$

where $i \in \{1, \cdots, N\}$ and τ is the temperature parameter to control Gumbel-Softmax distribution's smoothness, as the temperature τ approaches 0, the Gumbel-Softmax distribution becomes identical to the one-hot categorical distribution. As the randomness of g is independent of π, we can now directly optimize our gating control policy using standard gradient descent algorithms.

View-Fusion for Co-Regularization. Existing approaches attempt to tackle the M2TV problem as multiple multi-task learning problems from different views, underlying the assumption that all views are conditionally independent (Zhang and Huan 2012; Jin et al. 2013, 2014; Wu, Zhan, and Jiang 2018). If we let $\mathcal{F}^v$ represent the view function built on view v, a typical final model obtained on one task t is the average of prediction results from all views (also see Figure 1) as $\mathcal{F}(\mathcal{X}) = \frac{1}{V}\sum_{v=1}^{V} \mathcal{F}^v(\mathcal{X}^v)$,

$$\mathcal{L}_f = \sum_t \sum_v \frac{1}{V} \|\mathcal{F}^{t,v}(\mathcal{X}^{t,v}) - \mathcal{F}^t(\mathcal{X}^t)\|_2 \tag{3}$$

where $\mathcal{F}^{t,v}$ is the task-view-specific model function for inputs $\mathcal{X}^{t,v}$ from the v-th view of t-th task, $\mathcal{F}^{t}$ is the view-fusion function for inputs $\mathcal{X}^{t}$ from task t with all views merged, where V is the total number of views, $\mathcal{X}$ is the original multi-task multi-view inputs, and $\mathcal{X}^{v}$ is the set of features containing data from all tasks under view v. Unlike the above, we encourage the function built on each view to agreeing with the view-fusion module compiled on merged views. In our work, we use a specific feedforward linear layer as the view-fusion block to merge multiple views by taking all views concatenated. Thus, we design a regularization term as:

Algorithm 1: Semi Supervised Multi-Task Multi-View Learning for Fragmented Time Series with Gathering Consistency Adaption

Input: Labeled multi-task multi-view time series data $\mathcal{X}_s$, unlabeled data $\mathcal{X}_u$, number of tasks T, model $\mathcal{F}$ with parameters θ, adaption steps K, supervised loss $\mathcal{L}_s$, unsupervised loss $\mathcal{L}_u$, unsupervised loss coefficient λ, uncertainty weights parameter α_t, β_t for any task t, learning rate η

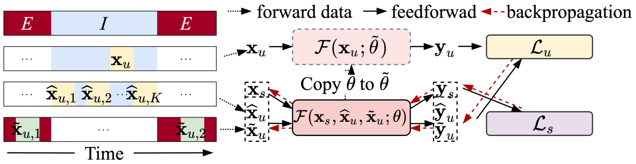
Output: Model parameters $\theta^{'}$

```
Let $t \in \{1, \cdots, T\}$ represents the specific task code, the current
Initialization $\theta$ and $\{\beta_t\}$
while Training do
    Let $\mathbf{x}_s = \{\mathbf{x}_s^t\}$ be a mini-batch of $\mathcal{X}_s$
    For all tasks, randomly select subseries of $\mathcal{X}_u$ from the internal time slot as original unlabeled inputs $\mathbf{x}_u = \{\mathbf{x}_u^t\}_{t\in\{1,\cdots,T\}}$ for reference
    For all tasks, select other subseries of $\mathcal{X}_u$ uniformly at random from the internal time slot for $K$ times as $\widehat{\mathbf{x}}_u = \{\widehat{\mathbf{x}}_{u,1}, \cdots, \widehat{\mathbf{x}}_{u,K}\}$, in which for $\forall k \in \{1, \cdots, K\}, \widehat{\mathbf{x}}_{u,k} = \{\widehat{\mathbf{x}}_{u,k}^t\}_{t\in\{1,\cdots,T\}}$
    For all tasks, select two subseries from the external time slot, one before the internal and one after, as $\tilde{\mathbf{x}}_u = \{\tilde{\mathbf{x}}_{u,1}, \tilde{\mathbf{x}}_{u,2}\}$, where $\tilde{\mathbf{x}}_{u,i} = \{\tilde{\mathbf{x}}_{u,1}^t\}_{t\in\{1,\cdots,T\}}, \forall i \in [1, 2]$
    $[\mathbf{y}_s, \widehat{\mathbf{y}}_u, \tilde{\mathbf{y}}_u] \leftarrow \mathcal{F}(\mathbf{x}_s, \widehat{\mathbf{x}}_u, \tilde{\mathbf{x}}_u; \theta)$
    $\mathbf{y}_u \leftarrow \mathcal{F}(\mathbf{x}_u; \tilde{\theta})$, $\tilde{\theta}$ is a hard copy of $\theta$
    Update $\mathcal{L}_u$ based on Eq. 6
    $\mathcal{J}(\theta) \leftarrow \mathcal{L}_s(\mathcal{X}_s; \theta) + \lambda\mathcal{L}_u(\mathcal{X}_u; \theta, \tilde{\theta})$
    $\theta^{'} \leftarrow \theta - \eta\nabla_\theta\mathcal{J}(\theta)$
end while
```

Semi-Supervised Learning for Time Series

Considering temporal data characteristics from mobile sensors, we argue that the time series should always reflect an individual's coherent and unified physical status within a relatively short interval. Thus, we expect adjacent sub time series within the same time interval (aka. from internal time interval) to obtain similar representations. In contrast, time series distant from each other (aka. from external time interval) may obtain distinctive representations.

Gathering Consistency Adaption. To accommodate this principle to mobile sensing time series, we proposed this gathering consistency adaption algorithm, as described in Algorithm 1 and Figure 3, for semi-supervised learning. We utilize an original sampling strategy combined with the consistency training (Bachman, Alsharif, and Precup 2014; Laine and Aila 2017; Franceschi, Dieuleveut, and Jaggi 2019; Xie et al. 2020) on numerous unlabeled time series to constrain model predictions to be invariant to similar inputs with inevitable noise while sensitive to essential differentiation. Let $\mathcal{X}_s$ and $\mathcal{X}_u$ represents the overall labeled data and unlabeled data, respectively. We split any given long time series from $\mathcal{X}_u$ into multiple fragmented time series for convenience. Intuitively, for every training step, the chosen fragmented sequence is viewed as an internal time slot while other sequences are external time slots. We first select a random subseries $\mathbf{x}_u$ within the internal time slot as the original reference. We then uniformly select multiple random subseries $\{\widehat{\mathbf{x}}_{u,1}, \cdots, \widehat{\mathbf{x}}_{u,K}\}$ as a set of similar but naturally noisy samples compared to $\mathbf{x}_u$ within the same internal time interval for consistency regularization. Finally, we also randomly select two other subseries $\{\tilde{\mathbf{x}}_{u,1}, \tilde{\mathbf{x}}_{u,2}\}$ from the external slots, respectively before and after the internal slot in time for inconsistency differentiation. Therefore, we name this approach as gathering consistency adaption since it is a cumulative loss adaption process, including consistency and inconsistency training. In this case, we apply KL-DIVERGENCE as the divergence metric between any pair of probability distributions $\mathcal{D}(P_\theta(y|x_u)\|P_\theta(y|\widehat{x}_u, \tilde{x}_u))$. Figure 3 is a visualization of the semi-supervised learning architecture.



forward data
feedforwad
backpropagation
E
I
E
$\mathbf{x}_u$
$\widehat{\mathbf{x}}_{u,1}$ $\widehat{\mathbf{x}}_{u,2}$ $\widehat{\mathbf{x}}_{u,K}$
$\tilde{\mathbf{x}}_{u,1}$
$\tilde{\mathbf{x}}_{u,2}$
Time
$\mathbf{x}_u$
$\mathcal{F}(\mathbf{x}_u; \tilde{\theta})$
$\mathbf{y}_u$
$\mathcal{L}_u$
Copy θ to $\tilde{\theta}$
$\mathbf{x}_s$
$\widehat{\mathbf{x}}_u$
$\tilde{\mathbf{x}}_u$
$\mathcal{F}(\mathbf{x}_s, \widehat{\mathbf{x}}_u, \tilde{\mathbf{x}}_u; \theta)$
$\mathbf{y}_s$
$\widehat{\mathbf{y}}_u$
$\tilde{\mathbf{y}}_u$
$\mathcal{L}_s$

Figure 3: The semi-supervised learning framework shown in Algorithm 1

We calculate a cumulative loss through multiple adaption steps, yet the KL-DIVERGENCE is still too trivial to use directly. Also, the loss between different representations can effortlessly dominate the loss between similar ones. To tackle this challenge, we extend the task-dependent uncertainty modeling approach inspired by (Kendall, Gal, and Cipolla 2018).

Task-Dependent Uncertainty Modeling. (Kendall, Gal, and Cipolla 2018) derived a multi-task loss function based on maximizing the Gaussian likelihood with homoscedastic uncertainty for weighting losses in multi-task learning. Specifically, for a given multi-task model $\mathcal{F}$ with parameters θ, it leads to an overall minimization loss objective function as:

$$\mathcal{L}(\theta, \sigma_1, \sigma_2) \propto \frac{1}{2\sigma_1^2}\mathcal{L}_1(\theta) + \frac{1}{2\sigma_2^2}\mathcal{L}_2(\theta) + \log\sigma_1\sigma_2 \tag{4}$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ are the loss functions for task 1 and 2, underlying the assumption that the prediction probabilities of any given model denote to a Gaussian distribution with mean given by the model output and an observation noise scalar σ:

$$P_\theta(y|\mathcal{F}(x)) = \mathcal{N}(\mathcal{F}(x), \sigma^2) \tag{5}$$

Proband	Metrics	Supervised Models			Semi-Supervised Models				$\Delta \uparrow$ (%)
		DeepSense	IteMM	DMTMV	reg-MTMV	CSL-MTMV	MFMs	Ours	
$\mathcal{T}_1$	Acc	0.5667	0.2465	<u>0.6937</u>	0.5312	0.5337	0.6932	**0.7915**	14.10%
	M-F1	0.4949	0.2639	0.6762	0.5317	0.5121	<u>0.6816</u>	**0.7612**	11.68%
	W-F1	0.5361	0.2571	0.6678	0.5211	0.5313	<u>0.6994</u>	**0.7493**	7.13%
	Avg	0.5326	0.2558	0.6792	0.5280	0.5257	<u>0.6914</u>	**0.7673**	10.98%
$\mathcal{T}_2$	Acc	0.7131	0.3841	0.8361	0.7649	0.7745	<u>0.8237</u>	**0.8659**	5.12%
	M-F1	0.6353	0.4532	0.7031	0.7505	0.7614	<u>0.8127</u>	**0.8662**	6.58%
	W-F1	0.7076	0.4319	0.7851	0.7694	0.7794	<u>0.8263</u>	**0.8416**	1.85%
	Avg	0.6853	0.4231	0.7748	0.7616	0.7718	<u>0.8209</u>	**0.8579**	4.51%
$\mathcal{T}_3$	Acc	0.5829	0.2633	0.7374	0.6962	0.7362	<u>0.7945</u>	**0.8474**	6.66%
	M-F1	0.4659	0.2798	0.7381	0.6986	0.7276	<u>0.7545</u>	**0.8073**	7.00%
	W-F1	0.5412	0.2726	0.7343	0.6832	0.7306	<u>0.7497</u>	**0.7875**	5.04%
	Avg	0.5300	0.2719	0.7366	0.6927	0.7315	<u>0.7662</u>	**0.8141**	6.24%
$\mathcal{T}_4$	Acc	0.7046	0.2793	<u>0.7222</u>	0.5287	0.5291	0.6891	**0.7911**	9.54%
	M-F1	0.5631	0.3479	<u>0.6557</u>	0.5279	0.5193	0.6429	**0.7152**	9.07%
	W-F1	0.6743	0.3034	<u>0.6961</u>	0.5295	0.5279	0.6632	**0.7583**	8.94%
	Avg	0.6473	0.3102	<u>0.6913</u>	0.5287	0.5254	0.6651	**0.7549**	9.19%
$\mathcal{T}_5$	Acc	0.5020	0.2991	<u>0.7461</u>	0.5068	0.518	0.657	**0.8163**	9.41%
	M-F1	0.4867	0.3741	<u>0.7458</u>	0.4735	0.4742	0.6411	**0.7815**	4.79%
	W-F1	0.4714	0.3269	<u>0.7388</u>	0.4674	0.478	0.6429	**0.7665**	3.75%
	Avg	0.4867	0.3334	<u>0.7436</u>	0.4826	0.4901	0.6470	**0.7881**	5.99%

Table 1: Prediction results on **RealWorld-HAR** (5 out of 8 tasks 7 views) with best model performance in bold and second-best results with underlines. *ASM2TV* generally achieves the best performance and prediction results on all metrics across all exhibited tasks. $\Delta_{\mathcal{T}}$ represents the improvement percentage compared to the second-best approach for all tasks. Our framework achieves a significant improvement compared to other state-of-the-arts. Refer to Appendix for more details.

Intuitively, if one particular task's noise effect enlarges, the overall loss for that task would be balanced adaptively to prevent other tasks training from being dominated. In our consistency adaption procedure, the divergence distance between any pair of probability distributions genuinely denotes the Gaussian distribution following the fact that most of the time series are with Gaussian white noise (Mahdavinejad et al. 2018), especially for the IoT sensing data. As a result, the assumption above matches the facts, which makes the uncertainty modeling approach for multi-task semi-supervised loss suitable in this case. We let α_t and β_t (which are equivalent to $2 \log \sigma_t$) be the learnable noise parameter for consistency loss and discrimination loss of task t, respectively. Thus, the minimization objective function becomes:

$$\mathcal{L}_u(\theta, \alpha, \beta) = \sum_t^T \sum_k^K e^{-\alpha_t} \mathbb{E}[\log P_{\tilde{\theta}}(y_1 | \mathbf{x}_u^t) - \log P_\theta(\widehat{y}_2 | \widehat{\mathbf{x}}_{u,k}^t)] + \alpha_t$$
$$+ \sum_t^T \sum_{i=1}^2 -e^{-\beta_t} \mathbb{E}[\log P_{\tilde{\theta}}(y_1 | \mathbf{x}_u^t) - \log P_\theta(\tilde{y}_2 | \tilde{\mathbf{x}}_{u,i}^t)] + \beta_t \quad (6)$$

As the consistency loss is more trivial than the differentiation loss, the learnable variance scalar may also become smaller for the consistency piece. In contrast, the weights for the consistency part increase, which lowers the risk of being dominated by other significant task losses.

Experiments

In this section, we conduct experiments on two real-world human activity recognition datasets to show that our model outperforms many strong M2TV baselines, meanwhile maintaining a low-memory footprint for computation efficiency.

Model	$\Delta_{Acc} \uparrow$ (%)	$\Delta_{M-F1} \uparrow$ (%)	$\Delta_{W-F1} \uparrow$ (%)	$\Delta_{\#Params} \downarrow$ (%)
Supervised				
DeepSense	0.7216	0.7016	0.7116	10M
IteMM	0.3969	0.3859	0.3985	-
DMTMV	0.8443	0.8209	0.8397	<u>-20%</u>
Semi-Supervised				
regMVMT	0.7432	0.7297	0.7401	-0%
CSL-MTMV	0.7721	0.7507	0.7685	-0%
MFMs	<u>0.8515</u>	<u>0.8209</u>	<u>0.8468</u>	-0%
ASM2TV (ours)	**0.9742**	**0.9401**	**0.9464**	**-47%**

Table 2: Our ASM2TV framework averaged prediction results over all metrics across all tasks compared with other methods on **GLEAM** dataset.

Datasets and Descriptions

We evaluate our framework using two real-world human activity recognition (HAR) datasets, namely **RealWorld-HAR** (Sztyler and Stuckenschmidt 2016), this data set covers various mobile sensor level data of the activities (e.g., climbing stairs down and up, jumping, etc.) of fifteen probands (or subjects) each for 10 minutes roughly. We take each proband as a separate **task** with different on-body positions (e.g., chest, forearm, etc.) recognized as different **views**. **GLEAM** (Rahman et al. 2015), this dataset is a head-motion-tracking dataset collected with Google Glass with the labeled data from the head-mounted sensor that can be used to recognize eating and other activities, and ultimately to assist individuals with diabetes. We also take each subject as an individual **task**, and each sensor is recognized as a single **view**. For both datasets, we consecutively split the original multivariate time series into multiple fragmented time series. Each piece lasts only 5 seconds to meet the requirements of instant response under most mobile computing scenarios. For data split, we uniformly separate the time series in a consecutive manner for each specific activity of each subject, such that we can make sure to predict each category sufficiently.

Baselines and Metrics

We compare our method with the following baselines. First, we consider using **DeepSense** (Yao et al. 2017b), a deep learning model designed for time-series mobile sensing data, as a single-task baseline where we train each task separately. **IteM2** (He and Lawrence 2011) is a transductive algorithm, and it can only handle non-negative feature values. When applying the IteM2 algorithm to some of our datasets that have negative feature values, we add a positive constant to guarantee the non-negativity; **regMVMT** algorithm (Zhang and Huan 2012) is an inductive algorithm, which assumes all tasks should be similar to achieve a good performance; **CSL-MTMV** is an inductive M2TVL algorithm (Jin et al. 2013) that assumes the predictions of different views within a single task are consistent; **MFMs** is a multilinear factorization model proposed by (Lu et al. 2017) which can learn the task-specific feature map. **DMTMV** (Wu, Zhan, and Jiang 2018) is a deep multi-task multi-view method to learn nonlinear feature representations and classifier in a unified framework, which can also learn task relationships in nonlinear models.

Experiment Settings

We apply the basic feedforward multilayer perceptron (MLP) as the backbone for each task-specific and view-specific layer. The shared blocks are also composed of primary linear hidden layers. We use the CROSS ENTROPY as the supervised loss for human activity classification. The unsupervised loss can be viewed as an extension of KL-DIVERGENCE loss. We use Adam as the optimizer for all deep neural network-based models and set the initial learning rate to $3e^{-4}$ with a weight decay of $1e^{-6}$. Specifically, an upsampling strategy is applied to overcome the label imbalance problem. We utilize a dropout strategy for all deep neural networks with a dropout rate of 0.5 to prevent overfitting. For more setting details, readers may refer to the supplementary materials.

Experimental Results

Quantitative Results Table 1, 2, 3 and 4 show the quantitative results under three metrics for our framework and all the other competitive approaches on two datasets **RealWorld-HAR** and **GLEAM**. We report all metrics and relative performance of four tasks (eight tasks in total but we only exhibit four due to space limitation) in **RealWorld-HAR** learning scenario (see Table 1) and report a comparative performance improvements results with the single-task baseline (DeepSense) in percentage on both two datasets (see Table 3 and Table 4). For more details, one may refer to the supplementary material for a full version. Specifically, we have few following observations: 1) Our ASM2TV generally achieves the best prediction performance on all metrics across all tasks under two different learning scenarios or two human action recognition datasets. 2) IteM2 is an early transductive M2TV algorithm based on graphs, originally designed for binary classification problems. It can only handle positive feature values, making it difficult to predict complex real-life mobile sensing time series data. 3) We apply the DeepSense model as the single-task learning baseline for all tasks with 10M model parameters in total and compare all the other M2TV models with it. As we discussed in Section , M2TV learning methods such as regMTMV, CSL-MTMV, and MFMs all utilize the soft-sharing mechanisms by constraining either the similarity or parameter consistency between tasks or views by keeping every model

Model	$\Delta_{Acc} \uparrow$	$\Delta_{M-F1} \uparrow$	$\Delta_{W-F1} \uparrow$	$\Delta_{\#Params} \downarrow$
	(%)	(%)	(%)	(%)
Supervised				
DeepSense	0.5988 (-)	0.5760 (-)	0.5797 (-)	10M (-)
IteMM	-49%	-49%	-48%	-
DMTMV	25%	24%	25%	-20%
Semi-Supervised				
regMVMT	7%	6%	6%	-0%
CSL-MTMV	8%	11%	10%	-0%
MFMs	24%	25%	26%	-0%
ASM2TV (ours)	**46%**	**43%**	**43%**	**-47%**

Table 3: Overview of comparison on **RealWorld-HAR**. We additionally exhibit average prediction results of all tasks and display the overall improvement percentage through all quantitative metrics and model space complexity ($\Delta_{\#Params}$). Generally, our *ASM2TV* can achieve over 40% significant improvements on all metrics compared to a single-task model using 47% fewer model parameters.

Model	$\Delta_{Acc} \uparrow$	$\Delta_{M-F1} \uparrow$	$\Delta_{W-F1} \uparrow$	$\Delta_{\#Params} \downarrow$
	(%)	(%)	(%)	(%)
Supervised				
DeepSense	-	-	-	-
IteMM	-45%	-45%	-44%	-
DMTMV	17%	17%	18%	-20%
Semi-Supervised				
regMVMT	3%	4%	4%	-0%
CSL-MTMV	7%	7%	8%	-0%
MFMs	18%	17%	19%	-0%
ASM2TV (ours)	**35%**	**34%**	**33%**	**-47%**

Table 4: Overview of comparison on **GLEAM** (10 tasks 6 views). Generally, *ASM2TV* can achieve over 30% significant improvements on all metrics compared to a single-task model using 47% fewer model parameters.

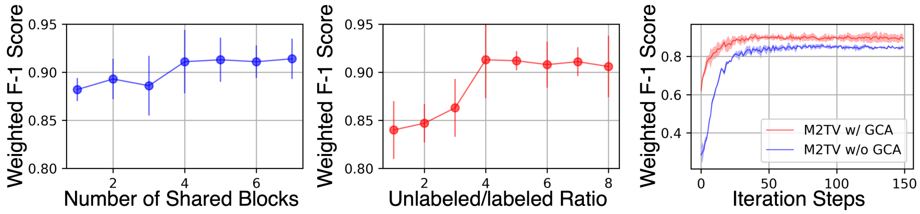


Figure 4: Ablation studies.

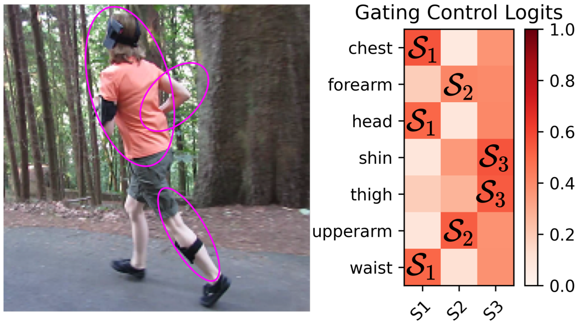


Figure 5: A visualization of gating control logits and soft-clustering.

or function built on each task and each view. Thus, these models have a relatively larger parameter space since they hold the same amount of parameters as needed for single-task learning. Astonishingly, our ASM2TV inherits the spirit of hard-sharing, which achieves over 40% overall improvements across all metrics while only using 47% fewer parameters than the single-task learning baseline. 4) The MFMs approach generally reaches the second-best prediction results other than our framework. It is mainly due to MFMs can model a high-dimension-interacted relationship instead of either task-specific only or view-specific only relationship, using the factorization machine. However, the huge computation cost limits its power in many mobile computing scenarios.

Soft-Clustering Process. We visualize the gating control policy logits (probabilities of open-gated for each candidate shared block) to demonstrate that this process can learn a task-view-interacted relatedness in terms of softly clustering different views from different tasks (see Figure 5). From Figure 5(a), different on-body positions (or views) have been automatically clustered into three groups learned by the policy. Among which, the head, chest, and waist have been clustered into $\mathcal{S}_1$, the upper arm and forearm have been clustered into $\mathcal{S}_2$ and the rest parts are assigned into the last group. under each specific view between two randomly selected tasks. We find that this distance pattern under the head, chest, and waist is quite similar, and two observations provide a resembling conclusion.

Ablation Studies. We further investigate the influence of how 1) the number of shared blocks, 2) the ratio of unlabeled data to labeled data, and 3) our gathering consistency adaption can potentially affect the model performance (see Figure 4). We have several interesting observations: 1) The overall model performance generally reaches the best when the number of shared blocks attend to 4, roughly half of the total number of views. We hypothesize that our gating control policy can find a smaller, trainable, and optimal sub-M2TV network that needs fewer shared blocks to achieve a relatively good performance rather than building a sharing scheme across tasks under every specific view. 2) For semi-supervised learning, as the proportion of unlabeled data enlarges, the model performance eventually reaches the maximum at around 4, indicating the amount of unlabeled data is four times more than labeled data. It further demonstrates our framework's effectiveness, especially under scenarios where unlabeled data is way more than labeled data such as mobile sensing in IoT. 3) Figure 4(c) illustrates how our framework performs with or without the gathering consistency adaption procedure. It can also be viewed as supervised learning against semi-supervised learning. Our GCA procedure dramatically improves the overall prediction results by combing the unsupervised loss from fragmented time series.

Conclusion

In this work, we presented a novel semi-supervised multi-task multi-view learning framework that adaptively decided which views across which tasks should share common knowledge using the gating control policy. We further showed that this gating control policy can also be viewed as a soft clustering function for different views across different tasks. Our experimental results demonstrated its superiority by achieving the best model performance with considerably fewer model parameters than the other state-of-the-arts in the M2TVL field. Moreover, we introduced a novel semi-supervised learning method, namely gathering consistency adaption, to assist the model training on fragmented time series and enhance the overall model performance by combining the unsupervised loss. Moving forward, we would like to extend our ASM2TV to a more fine-grained architecture-wise approach and explored other deep learning acceleration techniques to make our framework more efficient for mobile computing.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62072279, in part by

Shandong Provincial Natural Science Foundation of China under Grant ZR2021QF044.

References

Bachman, P.; Alsharif, O.; and Precup, D. 2014. Learning with Pseudo-Ensembles. In *NeurIPS*.

Baxter, J. 1997. A Bayesian/Information Theoretic Model of Learning to Learn via Multiple Task Sampling. *Mach. Learn.*

Bilen, H.; and Vedaldi, A. 2016. Integrated perception with recurrent multi-task neural networks. In *NeurIPS*.

Caruana, R. 1998. Multitask Learning. In *Learning to Learn*.

Chen, K.; Yao, L.; Wang, X.; Zhang, D.; Gu, T.; Yu, Z.; and Yang, Z. 2018. Interpretable parallel recurrent neural networks with convolutional attentions for multi-modality activity modeling. In *IJCNN*.

Chen, Z.; Chen, D.; Zhang, X.; Yuan, Z.; and Cheng, X. 2021. Learning Graph Structures with Transformer for Multivariate Time Series Anomaly Detection in IoT. In *IEEE IoTJ*.

Chen, Z.; E, J.; Zhang, X.; Sheng, H.; and Cheng, X. 2020. Multi-Task Time Series Forecasting With Shared Attention. In *ICDMW*.

Franceschi, J.; Dieuleveut, A.; and Jaggi, M. 2019. Unsupervised Scalable Representation Learning for Multivariate Time Series. In *NeurIPS*.

Gao, Y.; Ma, J.; Zhao, M.; Liu, W.; and Yuille, A. L. 2019. NDDR-CNN: Layerwise Feature Fusing in Multi-Task CNNs by Neural Discriminative Dimensionality Reduction. In *CVPR*.

Guan, Y.; and Plötz, T. 2017. Ensembles of deep lstm learners for activity recognition using wearables. *IMWUT*.

He, J.; and Lawrence, R. 2011. A Graphbased Framework for Multi-Task Multi-View Learning. In *ICML*.

Huang, J.; Feris, R. S.; Chen, Q.; and Yan, S. 2015. Cross-Domain Image Retrieval with a Dual Attribute-Aware Ranking Network. In *ICCV*.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparameterization with Gumbel-Softmax. In *ICLR*.

Jin, X.; Zhuang, F.; Wang, S.; He, Q.; and Shi, Z. 2013. Shared Structure Learning for Multiple Tasks with Multiple Views. In *ECML/PKDD*.

Jin, X.; Zhuang, F.; Xiong, H.; Du, C.; Luo, P.; and He, Q. 2014. Multi-task Multi-view Learning for Heterogeneous Tasks. In *CIKM*.

Kang, Z.; Grauman, K.; and Sha, F. 2011. Learning with Whom to Share in Multi-task Feature Learning. In *ICML*.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *CVPR*.

Kokkinos, I. 2017. UberNet: Training a Universal Convolutional Neural Network for Low-, Mid-, and High-Level Vision Using Diverse Datasets and Limited Memory. In *CVPR*.

Laine, S.; and Aila, T. 2017. Temporal Ensembling for Semi-Supervised Learning. In *ICLR*.

Lu, C.; He, L.; Shao, W.; Cao, B.; and Yu, P. S. 2017. Multilinear Factorization Machines for Multi-Task Multi-View Learning. In *WSDM*.

Maddison, C. J.; Tarlow, D.; and Minka, T. 2014. A* Sampling. In *NeurIPS*.

Mahdavinejad, M. S.; Rezvan, M.; Barekatain, M.; Adibi, P.; Barnaghi, P. M.; and Sheth, A. P. 2018. Machine learning for Internet of Things data analysis: A survey. *CoRR*.

Misra, I.; Shrivastava, A.; Gupta, A.; and Hebert, M. 2016. Cross-Stitch Networks for Multi-task Learning. In *CVPR*.

Rahman, S. A.; Merck, C. A.; Huang, Y.; and Kleinberg, S. 2015. Unintrusive eating recognition using Google Glass. In *ICPCTH*.

Ranjan, R.; Patel, V. M.; and Chellappa, R. 2019. HyperFace: A Deep Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition. *TPAMI*.

Rasmus, A.; Berglund, M.; Honkala, M.; Valpola, H.; and Raiko, T. 2015. Semi-supervised Learning with Ladder Networks. In *NeurIPS*.

Ruder, S. 2017. An Overview of Multi-Task Learning in Deep Neural Networks. *CoRR*.

Ruder, S.; Bingel, J.; Augenstein, I.; and Søgaard, A. 2019. Latent Multi-Task Architecture Learning. In *AAAI*.

Standley, T.; Zamir, A. R.; Chen, D.; Guibas, L. J.; Malik, J.; and Savarese, S. 2020. Which Tasks Should Be Learned Together in Multi-task Learning? In *ICML*.

Sun, X.; Panda, R.; Feris, R.; and Saenko, K. 2020. AdaShare: Learning What To Share For Efficient Deep Multi-Task Learning. In *NeurIPS*.

Sztyler, T.; and Stuckenschmidt, H. 2016. On-body localization of wearable devices: An investigation of position-aware activity recognition. In *ICPCC*.

Wu, Y.; Zhan, D.; and Jiang, Y. 2018. DMTMV: A Unified Learning Framework for Deep Multi-task Multi-view Learning. In *ICBK*.

Xie, Q.; Dai, Z.; Hovy, E. H.; Luong, T.; and Le, Q. 2020. Unsupervised Data Augmentation for Consistency Training. In *NeurIPS*.

Yao, S.; Hu, S.; Zhao, Y.; Zhang, A.; and Abdelzaher, T. 2017a. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *WWW*.

Yao, S.; Hu, S.; Zhao, Y.; Zhang, A.; and Abdelzaher, T. F. 2017b. DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing. In *WWW*.

Zhang, J.; and Huan, J. 2012. Inductive multi-task learning with multiple view data. In *SIGKDD*.