Supplementary Material: Structured Sparse Multi-Task Learning with Generalized Group Lasso

Supplement of Table 1 in the Main Paper

The following table summarizes more structured sparse MTL methods with different settings of $\mathbf{d}_{g_i}$, which is a supplement of Table 1 in the main paper.

Table A1: A summary of detailed settings of GenGL for selected MTL methods. For decomposition methods, suppose there are h components and the lth component is associated with an operator $\boldsymbol{\pi}_{l,i}$. For clarity, when $h \in \mathrm{N}_+$, we show $\Omega(\mathbf{w}_l)$ instead.

Architecture	Method	$\Omega(\mathbf{w})$	$\boldsymbol{\pi}$	h
Shallow	Lasso [Tibshirani, 1996]	$\gamma\lVert\mathbf{w}\rVert_1$	$\boldsymbol{\pi}_1 \in \mathcal{P}_F, \boldsymbol{\pi}_2 \in \mathcal{P}_F$	$h=1$
	Group Lasso (GL) [Yuan and Lin, 2006]	$\gamma\lVert\mathbf{W}\rVert_{2,1}$	$\boldsymbol{\pi}_1 \in \mathcal{P}_F, \boldsymbol{\pi}_2 \in \mathcal{P}_C$	$h=1$
	rMTFL [Gong et al., 2012]	$\gamma_1\lVert\mathbf{W}_1\rVert_{2,1} + \gamma_2\lVert\mathbf{W}_2^T\rVert_{2,1}$ $s.t. \quad \mathbf{W} = \mathbf{W}_1 + \mathbf{W}_2$	$\boldsymbol{\pi}_{1,1} \in \mathcal{P}_F, \boldsymbol{\pi}_{1,2} \in \mathcal{P}_C$ $\boldsymbol{\pi}_{2,1} \in \mathcal{P}_C, \boldsymbol{\pi}_{2,2} \in \mathcal{P}_F$	$h=2$
	MeTaG [Han and Zhang, 2015] CCMTL [He et al., 2019]	$\frac{\gamma}{\phi^l}\sum_{j<k}\lVert\mathbf{w}_{l,:j} - \mathbf{w}_{l,:k}\rVert_2$	$\boldsymbol{\pi}_{l,1} \in \mathcal{P}_C, \boldsymbol{\pi}_{l,2} \in \mathcal{P}_{F^2}$	$h \in \mathrm{N}_+$
	Sparse Network Lasso [Okazaki and Kawano, 2022] [Wang and Sun, 2022]	$\sum_{j<k} r_{j,k}\lVert\mathbf{w}_{:j} - \mathbf{w}_{:k}\rVert_2 + \lVert\mathbf{w}\rVert_1$	$\boldsymbol{\pi}_1^1 \in \mathcal{P}_C, \boldsymbol{\pi}_2^1 \in \mathcal{P}_{F^2}$ $\boldsymbol{\pi}_1^2 \in \mathcal{P}_F, \boldsymbol{\pi}_2^2 \in \mathcal{P}_F$	$h=1$
	SSMTL$_m$ (The proposed model)	$\frac{\gamma_1}{\phi^l}\sum_{j<k}\lVert\mathbf{w}_{l,:j} - \mathbf{w}_{l,:k}\rVert_2 + \frac{\gamma_2}{\phi^{-l}}\lVert\mathbf{W}_l\rVert_{2,1}$ $+\frac{\gamma_1}{\phi^{-l}}\sum_i\sum_{j<k}\lvert w_{l,ij} - w_{l,ik}\rvert^2 + \frac{\gamma_2}{\phi^l}\lVert\mathbf{w}_l\rVert_1$	$\boldsymbol{\pi}_{l,1}^{(1)} \in \mathcal{P}_C, \boldsymbol{\pi}_{l,2}^{(1)} \in \mathcal{P}_{F^2}$ $\boldsymbol{\pi}_{l,1}^{(2)} \in \mathcal{P}_F, \boldsymbol{\pi}_{l,2}^{(2)} \in \mathcal{P}_C$ $\boldsymbol{\pi}_{l,1}^{(3)} \in \mathcal{P}_F, \boldsymbol{\pi}_{l,2}^{(3)} \in \mathcal{P}_{F^2}$ $\boldsymbol{\pi}_{l,1}^{(4)} \in \mathcal{P}_F, \boldsymbol{\pi}_{l,2}^{(4)} \in \mathcal{P}_F$	$h \in \mathrm{N}_+$
Deep	Sparse GL (SGL) [Scardapane et al., 2017]	$\frac{\gamma_1}{\phi^l}\sum_k^m\sum_j^{p_2}\lVert\mathbf{w}_{l,:jk}\rVert_2 + \frac{\gamma_2}{\phi^l}\sum_{i,j,k}\lvert\mathbf{w}_{l,ijk}\rvert$	$\boldsymbol{\pi}_{l,1}^{(1)} \in \mathcal{P}_C, \boldsymbol{\pi}_{l,2}^{(1)} \in \mathcal{P}_F, \boldsymbol{\pi}_{l,3}^{(1)} \in \mathcal{P}_F$ $\boldsymbol{\pi}_{l,1}^{(2)} \in \mathcal{P}_F, \boldsymbol{\pi}_{l,2}^{(2)} \in \mathcal{P}_F, \boldsymbol{\pi}_{l,3}^{(2)} \in \mathcal{P}_F$	$h \in \mathrm{N}_+$
	Adaptive GL (AGL) [Dinh and Ho, 2020a]	$\sum_k^m\sum_j^{p_2}\frac{\gamma_j}{\phi^l}\lVert\mathbf{w}_{l,:jk}\rVert_2$	$\boldsymbol{\pi}_{l,1} \in \mathcal{P}_C, \boldsymbol{\pi}_{l,2} \in \mathcal{P}_F, \boldsymbol{\pi}_{l,3} \in \mathcal{P}_F$	$h \in \mathrm{N}_+$
	GL + Adaptive GL (GLAGL) [Dinh and Ho, 2020b]	$\frac{\gamma}{\phi^l}\sum_k^m\sum_j^{p_2}\lVert\mathbf{w}_{l,:jk}\rVert_2 + \sum_k^m\sum_j^{p_2}\frac{\gamma_j}{\phi^l}\lVert\mathbf{w}_{l,:jk}\rVert_2$	$\boldsymbol{\pi}_{l,1}^{(1)} \in \mathcal{P}_C, \boldsymbol{\pi}_{l,2}^{(1)} \in \mathcal{P}_F, \boldsymbol{\pi}_{l,3}^{(1)} \in \mathcal{P}_F$ $\boldsymbol{\pi}_{l,1}^{(2)} \in \mathcal{P}_F, \boldsymbol{\pi}_{l,2}^{(2)} \in \mathcal{P}_C, \boldsymbol{\pi}_{l,3}^{(2)} \in \mathcal{P}_F$	$h \in \mathrm{N}_+$
	SSMTL$_t$ (The proposed model)	$\frac{\gamma_1}{\phi^l}\sum_{j<k}\lVert\mathbf{W}_{l,::j} - \mathbf{W}_{l,::k}\rVert_F + \frac{\gamma_2}{\phi^{-l}}\sum_k^m\sum_j^{p_2}\lVert\mathbf{w}_{l,:jk}\rVert_2$ $+\frac{\gamma_1}{\phi^{-l}}\sum_i^{p_2}\sum_{j<k}\lVert\mathbf{w}_{l,:ij} - \mathbf{w}_{l,:ik}\rVert_2 + \frac{\gamma_2}{\phi^l}\sum_{i,j,k}\lvert\mathbf{w}_{l,ijk}\rvert$	$\boldsymbol{\pi}_{l,1}^{(1)} \in \mathcal{P}_C, \boldsymbol{\pi}_{l,2}^{(1)} \in \mathcal{P}_C, \boldsymbol{\pi}_{l,3}^{(1)} \in \mathcal{P}_{F^2}$ $\boldsymbol{\pi}_{l,1}^{(2)} \in \mathcal{P}_C, \boldsymbol{\pi}_{l,2}^{(2)} \in \mathcal{P}_F, \boldsymbol{\pi}_{l,3}^{(2)} \in \mathcal{P}_F$ $\boldsymbol{\pi}_{l,1}^{(3)} \in \mathcal{P}_C, \boldsymbol{\pi}_{l,2}^{(3)} \in \mathcal{P}_F, \boldsymbol{\pi}_{l,3}^{(3)} \in \mathcal{P}_{F^2}$ $\boldsymbol{\pi}_{l,1}^{(4)} \in \mathcal{P}_F, \boldsymbol{\pi}_{l,2}^{(4)} \in \mathcal{P}_F, \boldsymbol{\pi}_{l,3}^{(4)} \in \mathcal{P}_F$	$h \in \mathrm{N}_+$

Summary of different types of linear operators in Sec. 4.1

Table A2: Summary of different types of operators for the ith dimension. (F: feature-level, T: task-level, E: element-wise, V: vector-wise, Net: net-wise, Neu: neuron-wise, W: weight-wise)

		$\boldsymbol{\pi}_i \in \mathcal{P}_F$		$\boldsymbol{\pi}_i \in \mathcal{P}_{F^2}$	
		$1 \leqslant i \leqslant N-1$	$i = N$	$1 \leqslant i \leqslant N-1$	$i = N$
$N=2$	$\forall j \neq i, \boldsymbol{\pi}_j \notin \mathcal{P}_C$	FE-selection	TE-selection	FE-clustering	TE-clustering
	$\exists! j \neq i, \boldsymbol{\pi}_j \in \mathcal{P}_C$	FV-selection	TV-selection	FV-clustering	TV-clustering
$N=3$	$\forall j \neq i, \boldsymbol{\pi}_j \notin \mathcal{P}_C$	FW-selection	TW-selection	FW-clustering	TW-clustering
	$\exists! j \neq i, \boldsymbol{\pi}_j \in \mathcal{P}_C$	FNeu-selection	TNeu-selection	FNeu-clustering	TNeu-clustering
	$\exists j, k \neq i, \boldsymbol{\pi}_j, \boldsymbol{\pi}_k \in \mathcal{P}_C$	FNet-selection	TNet-selection	FNet-clustering	TNet-clustering

Derivations for the Gradient of the Function in (14)

To compute the gradient $\nabla_{\mathbf{w}} f_\mu(\mathbf{w})$, we introduce the following lemma and provide its proof.

Lemma 1. *For any $\mu > 0$, $f_\mu(\mathbf{w})$ is convex and differentiable in $\mathbf{w}$, and the gradient of $f_\mu(\mathbf{w})$ w.r.t. $\mathbf{w}$ is*

$$\nabla_{\mathbf{w}} f_\mu(\mathbf{w}) = \gamma\mathbf{D}^T\boldsymbol{\beta}^*, \tag{1}$$

where $\boldsymbol{\beta}^*$ is the optimal solution to (11) in the main paper. The optimal $\boldsymbol{\beta}^*$ is got by a projection operator $S(\cdot)$, which projects any vector $\mathbf{u}$ to the l_2 ball:

$$\boldsymbol{\beta}^*_{g_1,\ldots,g_N} = S(\frac{\gamma \mathbf{D}_{g_1,\ldots,g_N}\mathbf{w}}{\mu}). \tag{2}$$

Proof. We first introduce that the *Fenchel conjugate* $\phi^*(\boldsymbol{\alpha})$ of a function $\phi(\boldsymbol{\beta})$ is defined as:

$$\phi^*(\boldsymbol{\alpha}) = \sup_{\boldsymbol{\beta}\in \mathrm{dom}(\phi)} (\boldsymbol{\beta}\boldsymbol{\alpha}^T - \phi(\boldsymbol{\beta})). \tag{3}$$

Recall that $q(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{\beta}\|_2^2$ with $\mathrm{dom}(q) = \mathcal{B}$ in the main paper, the conjugate of $q(\cdot)$ at $\frac{\mathbf{Dw}}{\mu}$ is derived as $q^*(\frac{\mathbf{Dw}}{\mu}) = \sup_{\boldsymbol{\beta}\in\mathcal{B}}(\boldsymbol{\beta}^T\frac{\mathbf{Dw}}{\mu} - q(\boldsymbol{\beta}))$. Hence, $f_\mu(\mathbf{w})$ is reformulated as:

$$f_\mu(\mathbf{w}) = \max_{\boldsymbol{\beta}\in\mathcal{B}}(\gamma\boldsymbol{\beta}^T\mathbf{Dw} - \mu q(\boldsymbol{\beta})) = \mu q^*(\frac{\mathbf{Dw}}{\mu}). \tag{4}$$

Since $q(\boldsymbol{\beta})$ is strictly convex, its conjugate is smooth. Therefore, $f_\mu(\boldsymbol{\beta})$ is a smooth function. Let $\phi(\boldsymbol{\beta}, \mathbf{w}) = \boldsymbol{\beta}^T\mathbf{Dw} - \mu q(\boldsymbol{\beta})$. Since $q(\cdot)$ is strongly convex, $\mathrm{argmax}_{\boldsymbol{\beta}\in\mathcal{B}}\,\phi(\boldsymbol{\beta},\mathbf{w})$ has a unique optimal solution denoted as $\boldsymbol{\beta}^*$. According to Danskin's theorem [Mangasarian, 1994],

$$\nabla_{\mathbf{w}} f_\mu(\mathbf{w}) = \nabla_{\mathbf{w}}\phi(\boldsymbol{\beta}^*, \mathbf{w}) = \gamma\mathbf{D}^T\boldsymbol{\beta}^*. \tag{5}$$

Then we calculate the optimal $\boldsymbol{\beta}^*$.

$$\begin{aligned}
\boldsymbol{\beta}^* &= \underset{\boldsymbol{\beta}\in\mathcal{B}}{\mathrm{argmax}}(\gamma\boldsymbol{\beta}^T\mathbf{Dw} - \mu q(\boldsymbol{\beta})) \\
&= \underset{\boldsymbol{\beta}\in\mathcal{B}}{\mathrm{argmax}} \sum_{g_1,\ldots,g_N} (\gamma\boldsymbol{\beta}^T_{g_1,\ldots,g_N}\mathbf{D}_{g_1,\ldots,g_N}\mathbf{w} - \frac{\mu}{2}\|\boldsymbol{\beta}_{g_1,\ldots,g_N}\|_2^2) \\
&= \underset{\boldsymbol{\beta}\in\mathcal{B}}{\mathrm{argmin}} \sum_{g_1,\ldots,g_N} \|\boldsymbol{\beta}_{g_1,\ldots,g_N} - \frac{\gamma\mathbf{D}_{g_1,\ldots,g_N}\mathbf{w}}{\mu}\|_2^2.
\end{aligned}$$

We can see that $\boldsymbol{\beta}^*$ is a column concatenation of $\boldsymbol{\beta}^*_{g_1,\ldots,g_N}$, and each $\boldsymbol{\beta}^*_{g_1,\ldots,g_N}$ can be calculated by:

$$\boldsymbol{\beta}^*_{g_1,\ldots,g_N} = \underset{\boldsymbol{\beta}_{g_1,\ldots,g_N}:\|\boldsymbol{\beta}_{g_1,\ldots,g_N}\|_2\leqslant 1}{\mathrm{argmin}} \|\boldsymbol{\beta}_{g_1,\ldots,g_N} - \frac{\gamma\mathbf{D}_{g_1,\ldots,g_N}\mathbf{w}}{\mu}\|_2^2. \tag{6}$$

According to the property of the l_2 ball, it can be shown that:

$$\boldsymbol{\beta}^*_{g_1,\ldots,g_N} = S(\frac{\gamma\mathbf{D}_{g_1,\ldots,g_N}\mathbf{w}}{\mu}), \tag{7}$$

where

$$S(\mathbf{u}) = \begin{cases} \frac{\mathbf{u}}{\|\mathbf{u}\|_2} & \|\mathbf{u}\|_2 > 1, \\ \mathbf{u} & \|\mathbf{u}\|_2 \leqslant 1. \end{cases}$$

□

Optimization Algorithm

In Algorithm 1, we provide the pseudocode of the optimization algorithm discussed in Sec. 5 of the main paper.

Algorithm 1 Optimization algorithm for solving (13) in the main paper

Input: $\mathbf{X}, \mathbf{y}, \mathbf{D}, \gamma, \widehat{\mathbf{w}}^0$, desired accuracy ε, learning rate η
Output: $\mathbf{w}$

Initialize $\nabla F(\widehat{\mathbf{w}}^0)$, $\alpha = 0$, $\theta_0 = 1$, smoothness parameter $\mu = \frac{\varepsilon}{\prod_i |G_i|}$.
repeat
  Compute $\nabla F(\widehat{\mathbf{w}}^k)$ using Eq. (16) in the main paper.
  Solve the proximal step:

$$\mathbf{w}^{k+1} = \arg\min_{\mathbf{w}} F(\widehat{\mathbf{w}}^k) + < \mathbf{w} - \widehat{\mathbf{w}}^k, \nabla F(\widehat{\mathbf{w}}^k) > + \frac{2}{\eta}\|\mathbf{w} - \widehat{\mathbf{w}}^k\|_2^2. \tag{8}$$

  Set $\theta_{k+1} = \frac{2}{\alpha+3}$.
  Set $\widehat{\mathbf{w}}^{k+1} = \mathbf{w}^{k+1} + \frac{1-\theta_k}{\theta_k}\theta_{k+1}(\mathbf{w}^{k+1} - \widehat{\mathbf{w}}^k)$.
  Set $\alpha = \alpha + 1$.
until convergence

Real-World Datasets

To evaluate the proposed SSMTL_m, we conduct experiments on the following six regression datasets:

- **RF1**: The RF1 dataset is to predict the river network flows for 48 h in the future at 8 sites, and each site contributes 8 attribute variables to facilitate prediction, thus there are a total of 64 variables plus 8 target variables. We randomly select 1000 samples in the RF1 dataset.
- **Isolet**: The Isolet dataset is generated as follows. 150 subjects spoke the name of each letter of the alphabet twice, and they are divided into 5 groups, leading to 5 tasks. We randomly select 1000 samples in the Isolet dataset.
- **Energy**: The Energy dataset performs energy analysis based on different building shapes simulated in Ecotect, aiming to predict 2 real valued responses by using 8 features. We randomly select 600 samples in the Energy dataset.
- **Parkinsons**: The Parkinsons dataset is to predict the disease symptom scores of 42 patients by using 16 bio-medical features, which results in 42 tasks.
- **School**: The School dataset contains the examination information of 15,362 students from 139 schools, where each school is regarded as a task. We are aiming to predict the score of each student.
- **SARCOS**: The SARCOS dataset is based on a inverse dynamic problem, which maps from a input space with 21 dimensions to 7 torques. We randomly select 1000 samples in the SARCOS dataset.

To evaluate the proposed SSMTL_t, we conduct experiments on the following three classification datasets:

- **SSD**: The SSD dataset contains features extracted from electric current drive signals. The drive has intact and defective components, resulting in 11 different classes with different conditions. We randomly select 2000 samples in the SSD dataset.
- **MNIST**: The MNIST dataset is a large database of handwritten digits with images in 28x28 pixel boxes. We reduce the feature dimension to 154 by Principle Component Analysis (PCA) [Wold et al., 1987], and randomly select 1000 samples in the MNIST dataset.
- **COVER**: The COVER dataset predicts forest cover type in the Roosevelt National Forest of northern Colorado. There are 7 kinds of cover types, leading to 7 tasks. We randomly select 2000 samples in the COVER dataset.

Detailed Settings of the Networks

For the three datasets used for deep models, we construct three soft-parameter sharing networks respectively. Each network is associated with a number of sub-nets corresponding to the tasks. Details are shown in Table A3.

Table A3: Detailed settings of the networks used in deep models.

Datasets	SSD	MNIST	COVER
Optimizer		Adam	
Learning rate		10^{-3}	
# of neurons	40/40/20/1	80/40/20/1	50/50/25/1
Batch size	128	64	128
# of sub-nets	11	10	7

Evaluation Metrics

For those regression tasks in experiments for MTL in shallow architectures, we adopt the following three metrics: normalized Mean Squared Error (nMSE) with the form of $\frac{1}{mn_t}\sum_{t=1}^{m}\frac{\|\mathbf{y}_t-\mathbf{X}_t\mathbf{w}_{:t}\|_2^2}{\|\mathbf{y}_t\|_2^2}$, Mean Absolute Error (MAE) with the form of $\frac{1}{mn_t}\sum_{t=1}^{m}\|\mathbf{y}_t-\mathbf{X}_t\mathbf{w}_{:t}\|_1$ and Explained Variance (EV) [Bakker and Heskes, 2003] with the form of $1-\frac{1}{mn_t}\sum_{t=1}^{m}\frac{var(\mathbf{y}_t-\mathbf{X}_t\mathbf{w}_{:t})}{var(\mathbf{y}_t)}$ to evaluate the performance. As for the classification tasks in deep MTL experiments, the two metrics we use are: Accuracy with the form of $\frac{1}{m}\sum_{t=1}^{m}\frac{\mathrm{TP}_t+\mathrm{TN}_t}{n_t}$, where TP_t and TN_t are the number of True Positive and True Negative samples of the t-th task, and Area Under ROC-Curve (AUC) [Hand and Till, 2001] with the form of $\frac{1}{m}\mathrm{AUC}_t$, where AUC_t is the AUC score of the t-th task.

Comparison Methods

In experiments for MTL in shallow architectures, we compare SSMTL$_m$ with the following six methods:

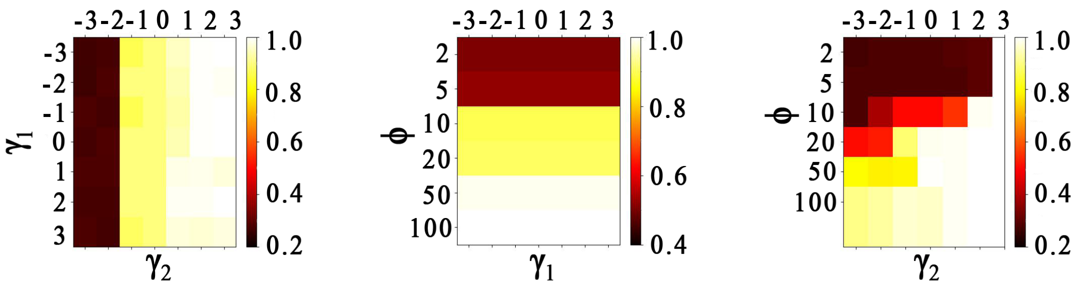
- **Lasso** [Tibshirani, 1996]: A single task learning method by penalizing each task with l_1-norm regularization independently, which is selected as the baseline method.
- **GL** [Yuan and Lin, 2006]: MTL with Group Lasso, which penalizes sum of the l_2-norms of task-common feature groups to learn global features that shared by multiple tasks.
- **rMTFL** [Gong et al., 2012]: Robust multi-task feature Learning method, which simultaneously captures features of related tasks and identifies outlier tasks.
- **MeTaG** [Han and Zhang, 2015]: Multi-Level task grouping method, which decomposes the weight into multiple levels and imposes the l_2-norm regularization on the pairwise difference among the tasks.
- **GBDSP** [Yang et al., 2019]: By assuming that tasks share a latent group structure, it learns a generalized block-diagonal structure for the latent basis of parameters.
- **KMSV** [Chang et al., 2021]: Based on the low-rank assumption, it minimizes exactly k minimal singular values to learn the latent group structure.

In experiments for deep MTL, we adopt the same soft-sharing network with totally independent sub-nets for multiple tasks, and compare SSMTL$_t$ with the following five deep models:

- **DMTRL** [Yang and Hospedales, 2017]: Deep multi-task representation learning, which aims to obtain a low-rank structure by tensor factorization.
- **SGL** [Scardapane et al., 2017]: It imposes the l_1-norm and GL together to guarantee both weight-wise sparsity and neuron-wise sparsity of layer parameters.
- **GL+AGL** [Dinh and Ho, 2020b]: It combines GL and Adaptive GL [Dinh and Ho, 2020a], and builds neuron-wise sparse model by consistent feature selection.
- **STG** [Yamada et al., 2020]: It is a robust and deep feature selection method using stochastic gates, which is based on a continuous relaxation of the l_0-norm.

Hyperparameter Sensitivity Analysis of SSMTL$_m$

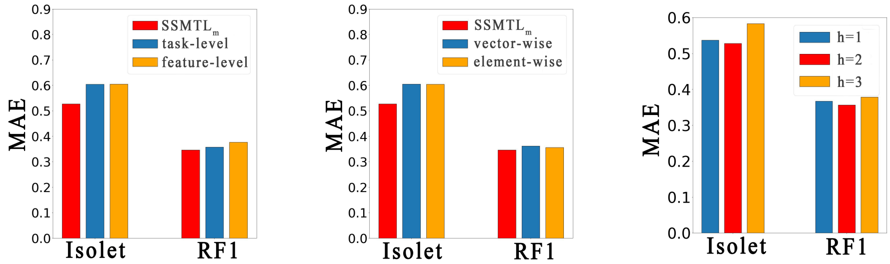
The sensitivity on γ_1, γ_2 and ϕ of SSMTL$_m$ is invested on the RF1 dataset for shallow MTL. In SSMTL$_m$, γ_1 controls the regularization degree of *task-level* operations while γ_2 controls the regularization degree of *feature-level* operations, which are selected from $\{10^{-3}, 10^{-2}, ..., 10^{2}, 10^{3}\}$, and ϕ alters the impact among components (layers), which is selected from $\{2, 5, 10, 20, 50, 100\}$. Fig. A1 shows the result in nMSE of three experiments. Specifically, Fig. A1(a), A1(b) and A1(c) are shown by fixing $\phi = 10$, $\gamma_2 = 1$ and $\gamma_1 = 1$ respectively. The result shows that: 1) it is recommended to set $\gamma_1 < 10^{-1}$ and $\phi < 10$ on the RF1 dataset; 2) γ_2 is not as sensitive as the other parameters.



(a) $\phi = 10$ (b) $\gamma_2 = 1$ (c) $\gamma_1 = 1$

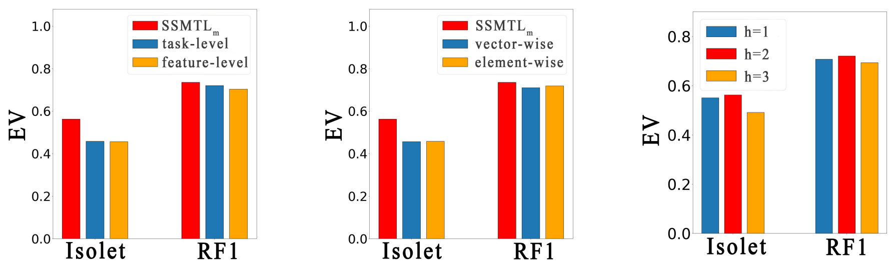
Figure A1: Hyperparameter Sensitivity Analysis on γ_1, γ_2 and ϕ of SSMTL$_m$ on the RF1 dataset in shallow MTL experiments. The values of γ_1 and γ_2 are shown in the logarithmic scale while the value of ϕ is selected from $\{2, 5, 10, 20, 50, 100\}$.

Results in MAE and EV of the Ablation Study for SSMTL$_m$



(a) task/featrue-level (b) vector/element-wise (c) decomposition

Figure A2: Results in MAE of different operations and model decomposition of SSMTL$_m$ on the Isolet and RF1 datasets.



(a) task/featrue-level (b) vector/element-wise (c) decomposition

Figure A3: Results in EV of different operations and model decomposition of SSMTL$_m$ on the Isolet and RF1 datasets.

Significant test results

We do paired Wilcoxon tests for statistical test. The results in Table A4 show that SSMTL$_m$ significantly outperforms MeTaG and GBDSP on the datasets.

Table A4: Results of paired samples Wilcoxon test at 5% significance level. We show p-values and •/○ indicates that SSMTL$_m$ is superior/inferior to the comparing method.

Method	Datasets			
	Parkinsons	RF1	Isolet	SARCOS
MeTaG vs SSMTL$_m$	0.0217 •	0.0724 ○	0.0439 •	0.0362 •
GBDSP vs SSMTL$_m$	0.0278 •	0.0320 •	0.0199 •	0.0225 •

References

[Bakker and Heskes, 2003] Bakker, B. and Heskes, T. M. (2003). Task clustering and gating for bayesian multitask learning. *J. Mach. Learn. Res.*, 4:83–99.

[Chang et al., 2021] Chang, W., Nie, F., Wang, R., and Li, X. (2021). New tight relaxations of rank minimization for multi-task learning. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2910–2914.

[Dinh and Ho, 2020a] Dinh, V. and Ho, L. S. T. (2020a). Consistent feature selection for neural networks via adaptive group lasso. *arXiv preprint arXiv:2006.00334*.

[Dinh and Ho, 2020b] Dinh, V. C. and Ho, L. S. (2020b). Consistent feature selection for analytic deep neural networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2420–2431.

[Gong et al., 2012] Gong, P., Ye, J., and Zhang, C. (2012). Robust multi-task feature learning. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, page 895–903, New York, NY, USA.

[Han and Zhang, 2015] Han, L. and Zhang, Y. (2015). Learning multi-level task groups in multi-task learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2638–2644. AAAI Press.

[Hand and Till, 2001] Hand, D. J. and Till, R. J. (2001). A simple generalisation of the area under the roc curve for multiple class classification problems. *Mach. Learn.*, 45(2):171–186.

[He et al., 2019] He, X., Alesiani, F., and Shaker, A. (2019). Efficient and scalable multi-task regression on massive number of tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:3763–3770.

[Mangasarian, 1994] Mangasarian, O. L. (1994). *Nonlinear programming*. SIAM.

[Okazaki and Kawano, 2022] Okazaki, A. and Kawano, S. (2022). Multi-task learning for compositional data via sparse network lasso. *Entropy*, 24(12).

[Scardapane et al., 2017] Scardapane, S., Comminiello, D., Hussain, A., and Uncini, A. (2017). Group sparse regularization for deep neural networks. *Neurocomput.*, 241(C):81–89.

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the royal statistical society series b-methodological*, 58:267–288.

[Wang and Sun, 2022] Wang, J. and Sun, L. (2022). Multi-task personalized learning with sparse network lasso. In Raedt, L. D., editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 3516–3522.

[Wold et al., 1987] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3):37–52.

[Yamada et al., 2020] Yamada, Y., Lindenbaum, O., Negahban, S., and Kluger, Y. (2020). Feature selection using stochastic gates. In *International Conference on Machine Learning*, pages 10648–10659. PMLR.

[Yang and Hospedales, 2017] Yang, Y. and Hospedales, T. M. (2017). Deep multi-task representation learning: A tensor factorisation approach. In *International Conference on Learning Representations*.

[Yang et al., 2019] Yang, Z., Xu, Q., Jiang, Y., Cao, X., and Huang, Q. (2019). Generalized block-diagonal structure pursuit: Learning soft latent task assignment against negative transfer. *Advances in Neural Information Processing Systems*, 32.

[Yuan and Lin, 2006] Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, 68:49–67.