# Multi-Task Cooperative Learning via Searching for Flat Minima

Fuping Wu[1], Le Zhang[1], Yang Sun[2], Yuanhan Mo[2], Thomas Nichols[1,2], and
Bartłomiej W. Papież[1,2]

[1] Nuffield Department of Population Health, University of Oxford, Oxford, UK
`{Fuping.Wu, Le.Zhang}@ndph.ox.ac.uk`
[2] Big Data Institute, University of Oxford, Oxford, UK
`{yang.sun, thomas.nichols, bartlomiej.papiez}@bdi.ox.ac.uk`
`yuanhan.mo@ndm.ox.ac.uk`

**Abstract.** Multi-task learning (MTL) has shown great potential in medical image analysis, improving the generalizability of the learned features and the performance in individual tasks. However, most of the work on MTL focuses on either architecture design or gradient manipulation, while in both scenarios, features are learned in a competitive manner. In this work, we propose to formulate MTL as a multi/bi-level optimization problem, and therefore force features to learn from each task in a cooperative approach. Specifically, we update the sub-model for each task alternatively taking advantage of the learned sub-models of the other tasks. To alleviate the negative transfer problem during the optimization, we search for flat minima for the current objective function with regard to features from other tasks. To demonstrate the effectiveness of the proposed approach, we validate our method on three publicly available datasets. The proposed method shows the advantage of cooperative learning, and yields promising results when compared with the state-of-the-art MTL approaches. *The code will be available online.*

**Keywords:** Multi-Task · Cooperative Learning · Optimization.

## 1 Introduction

With the development of deep learning, multi-task learning (MTL) has shown great potential to improve performance for individual tasks and to learn more transferable features (better generalizability), whilst reducing the number of the network parameters [16]. MTL has been widely studied in many domains including image classification [14] or image segmentation [9]. The core assumption behind MTL is that tasks could be correlated and thus provide complementary features for each other [4]. MTL is also applied in medical image analysis tasks [11, 6, 20, 5], where strong associations between multiple tasks commonly exist. For example, the diagnosis of cancer may indicate the extent of disease severity, which can be correlated with the patient's survival, thus diagnosis and prognosis of cancer could be learned simultaneously [18]. In clinical diagnosis, annotations of organs or tissues could support radiologists to grade disease, to mimic this process, Zhou *et.al* [24] studied to simultaneously segment and classify (grade)

tumors into benign or malignant class using 3D breast ultrasound images. Similarly, to improve the prediction of lymph node (LN) metastasis [21], Zhang *et.al* proposed a 3D multi-attention guided multi-task learning network for joint gastric tumor segmentation and LN classification [23].

Typically, MTL methods can be broadly categorized into hard and soft parameter-sharing paradigms [16]. The former adopts one backbone as the encoder to extract common features for all tasks, and the latter designs encoders for each task while constraining their associated parameters. To exploit the correlation between tasks, a large amount of work focuses on the architecture design of the network to enable the cross-task interaction [23]. For example, Misra *et.al* designed a cross-stitch model to combine features from multiple networks [12]. Besides network design, many researchers pay more attention to the neural network optimization process to counter the *negative transfer* issue [16]. As tasks could compete with each other for shared resources, the overall performance might be even poorer than those of solving individual tasks. To address this issue, previous works either change the weights of each task objective adaptively using heuristics [2], or manipulate the gradient to be descending direction for each task [10]. However, as those methods formulate MTL in a competitive manner, it is difficult to guarantee that the complementary information is fully utilized by each task. Moreover, most of them are designed for or evaluated on a simple scenario, where only one domain is involved and the tasks are homogeneous, namely all tasks are either dense prediction or image-level classification.

In this work, we propose a novel cooperative MTL framework (MT-COOL), which manages to update the features of one task while taking into account the current state of other features. Specifically, we adopt the soft parameter-sharing strategy and update each sub-model conditioning on the information learned by other tasks in an alternative manner. To avoid the *negative transfer* problem during the training, we further propose to search for flat minima of the current task with regard to others at each iteration. As a proof of concept, we first validate this method on the simple MNIST dataset for classification tasks. To show the advantage of the proposed approach in the medical domain, we use REFUGE2018 dataset for optic cup/disc segmentation and glaucoma classification, and HRF-AV dataset for artery and vein segmentation tasks. The results show a promising perspective of the proposed multi-task cooperative approach, compared to the state-of-the-art methods.

The main contributions of this work are as follows:

– We propose a novel MTL framework, which learns features for each task in a cooperative manner.

– We propose an effective optimization strategy to alleviate convergence issues.

– We validate the proposed method on three MTL scenarios with different task settings. The proposed method delivers promising results in all settings, compared with the state-of-the-art MTL approaches.

## 2   Method

For a better explanation, here we take two-task learning as an example, which can be generalized to n-task problems easily.

### 2.1   Bi-Level Optimization for Cooperative Two-Task Learning

Formally, let $x_i \in \mathbb{R}^{W \times H \times C}$ denotes an image with the width $W$, height $H$ and channel $C$, $y_i \in \mathbb{R}^{C_0}$ is a label for classification, (or $y_i \in \mathbb{R}^{W \times H \times C_0}$ for segmentation) and $C_0$ is the number of classes, $F_i(\cdot; \theta_i)$ is a feature extractor, $G_i(\cdot; \phi_i)$ is a prediction function for task $i = 1, \ldots, T$ where $T$ is a number of tasks, and here $T = 2$. $\theta_i$ and $\phi_i$ are corresponding parameters to be learned. Our task is to predict label $\widehat{y}_i = G_i(F_i(x_i))$.

For MTL, instead of using shared backbone, *i.e.*, $F_1 = F_2$, and updating them simultaneously with a single loss $\ell$, we propose to optimize them in a cooperative manner, that is learning $(F_1, G_1)$ conditioned on a fixed and informative $F_2$, and versa vice. Generally, it can be formulated as a bi-level optimization problem:

$$(U) \min_{\theta_1, \phi_1} \mathcal{L}_1(\theta_1, \phi_1, \theta_2) = \ell_1(G_1(\mathcal{M}(F_1(x_1; \theta_1), F_2(x_1; \theta_2)); \phi_1), \widehat{y}_1), \quad (1)$$

$$(L) \min_{\theta_2, \phi_2} \mathcal{L}_2(\theta_2, \phi_2, \theta_1) = \ell_2(G_2(\mathcal{M}(F_1(x_2; \theta_1), F_2(x_2; \theta_2)); \phi_2), \widehat{y}_2), \quad (2)$$

where $\ell_i$ is the loss function, e.g. cross-entropy loss for classification. $\mathcal{M}$ denotes a feature fusion to facilitate the current task learning by incorporating useful information from other tasks. A common choice for $\mathcal{M}$ is to use a linear combination of features, also known as *cross-stitch* [12] or concatenation operation in multi-layers (which is used in this work due to its simplicity).

To solve the problem Eq.(1)-(2), we propose to update $(\theta_1, \phi_1)$ and $(\theta_2, \phi_2)$ alternatively, as other traditional methods for bi-level optimization problem could be inefficient [1] due to the complexity of deep neural networks. However, without any constraint, this alternative optimization strategy could fail to achieve convergence to an optimal solution. For example, at the $t$-th iteration, we first optimize $\mathcal{L}_1(\theta_1, \phi_1, \theta_2^{(t-1)})$ to obtain an optimum $(\theta_1^{(t)}, \phi_1^{(t)})$. It is possible that for the second task, $\mathcal{L}_2(\theta_2^{(t-1)}, \phi_2^{(t-1)}, \theta_1^{(t-1)}) < \mathcal{L}_2(\theta_2^{(t-1)}, \phi_2^{(t-1)}, \theta_1^{(t)})$, which means that the update for the first task could increase the prediction risk of the second one, and cancel the gain from optimization of $\mathcal{L}_2$. Here, we also term this issue as *negative transfer*. To alleviate this effect, we propose to search for flat minima for one task with regard to the features from the other task in each iteration.

### 2.2   Finding Flat minima via Injecting Noise

As mentioned above, the network optimized for one task could be sensitive to the change of parameters for other tasks, which may cause non-convergent solutions. Hence, at each iteration, for each task, we search for an optimum that is non-sensitive to the update of other parameters within a fixed neighborhood. We term this kind of optima as *flat minima*.

To formally state this idea, assume that noise $\epsilon_i \sim \{\mathcal{U}(-b, b)\}^{d_{\epsilon_i}}$ with $b > 0$, $d_\epsilon = d_{\theta_i}$ and $d_{\theta_i}$ the dimension of $\theta_i$. Then for *task 1*, at $t$-th iteration our target

is to minimize the expected loss function with regard to the parameters $(\theta_1, \phi_1)$ and noise $\epsilon_2$, *i.e.*,

$$(U)\ \mathcal{R}_1^{[t]}(\theta_1, \phi_1) = \int_{\mathbb{R}^{d_{\epsilon_2}}} \mathcal{L}_1(\theta_1, \phi_1, \theta_2^{[t-1]} + \epsilon_2) dP(\epsilon_2) = \mathbb{E}[\mathcal{L}_1(\theta_1, \phi_1, \theta_2^{[t-1]} + \epsilon_2)],$$

$$(3)$$

$$s.t.\ |\theta_1 - \theta_1^{[t-1]}| < b,$$

where $P(\epsilon_2)$ is the noise distribution, and the solution is denoted as $(\theta_1^{[t]}, \phi_1^{[t]})$. Similarly, for *task 2*, the loss function is as follows,

$$(L)\ \mathcal{R}_2^{[t]}(\theta_2, \phi_2) = \int_{\mathbb{R}^{d_{\epsilon_1}}} \mathcal{L}_2(\theta_2, \phi_2, \theta_1^{[t]} + \epsilon_1) dP(\epsilon_1) = \mathbb{E}[\mathcal{L}_2(\theta_2, \phi_2, \theta_1^{[t]} + \epsilon_1)],\ (4)$$

$$s.t.\ |\theta_2 - \theta_2^{[t-1]}| < b.$$

Note that it is hard to find an ideal flat minimum $(\theta_1^{[t]}, \phi_1^{[t]})$ for Eq. (3), such that $\mathcal{L}_1(\theta_1^{[t]}, \phi_1^{[t]}, \theta_2^{[t-1]} + \epsilon_2^{(j_1)}) = \mathcal{L}_1(\theta_1^{[t]}, \phi_1^{[t]}, \theta_2^{[t-1]} + \epsilon_2^{(j_2)}),\ \forall \epsilon_2^{(j_1)}, \epsilon_2^{(j_2)} \sim P(\epsilon_2)$, and $\mathcal{L}_1(\theta_1^{[t]}, \phi_1^{[t]}, \theta_2^{[t-1]}) < \mathcal{L}_1(\theta_1^{[t-1]}, \phi_1^{[t-1]}, \theta_2^{[t-1]})$, which satisfies the requirement to avoid the optimization issue (see Sect. 2.1). Hence, our goal is to find an approximately flat minimum to alleviate this issue. A similar idea has been proposed for continual learning [19]. However, our method differs as follows: (1) the flat minimum in [19] is searched for the current task, while in our work, it is searched with regard to other tasks; (2) Once the flat minimum is found for the first task in a continual learning problem, search region for the remaining tasks is fixed, while in our work, the parameters for each task are only constrained in a single iteration, and search region could change during the optimization.

In practice, it is difficult to minimize the expected loss, we instead minimize its empirical loss for Eq. (3) and Eq. (4) as follows,

$$(U)\ L_1^{[t]}(\theta_1, \phi_1) = \frac{1}{M} \sum_{j=1}^{M} \mathcal{L}_1(\theta_1, \phi_1, \theta_2^{[t-1]} + \epsilon_2^{(j)}) + \lambda \cdot KL(\widehat{y}_1^{(j)}, \frac{1}{M} \sum_{n=1}^{M} \widehat{y}_1^{(n)}),\ (5)$$

$$(L)\ L_2^{[t]}(\theta_2, \phi_2) = \frac{1}{M} \sum_{j=1}^{M} \mathcal{L}_2(\theta_2, \phi_2, \theta_1^{[t]} + \epsilon_1^{(j)}) + \lambda \cdot KL(\widehat{y}_2^{(j)}, \frac{1}{M} \sum_{n=1}^{M} \widehat{y}_2^{(n)}),\ (6)$$

where $\epsilon_i^{(j)}$ is a noise vector sampled from $P(\epsilon_i)$, $M$ is the sampling times, and $KL$ is the Kullback-Leibler Divergence. The first term in Eq. (5) or Eq. (6) is designed to find a satisfying minimum for the current task, and the second term enforces this minimum to be flat as desired.

**Warm Up the Network.** To initialize the parameters for Eq.(3)) and Eq.(4) with non-sensitive $(\theta_1^{[0]}, \theta_2^{[0]})$, we minimize the following loss function,

$$\mathcal{L}_{total} = \frac{1}{M} \sum_{j=1}^{M} (\mathcal{L}_1(\theta_1 + \epsilon_1^{(j)}, \phi_1, \theta_2 + \epsilon_2^{(j)}) + \mathcal{L}_2(\theta_2 + \epsilon_2^{(j)}, \phi_2, \theta_1 + \epsilon_1^{(j)})).\quad (7)$$

**Algorithm.** We term the proposed **m**ulti-**t**ask **coo**perative **l**earning method as MT-COOL. The algorithm is described in Algorithm 1. Note that to alleviate the optimization issue discussed in Section 2.1, after the update for each task, we clamp the parameters to ensure that they fall within the flat region, as described in Line 17 in Algorithm 1.

---

**Algorithm 1:** Cooperative Learning via Searching Flat Minima

---

**Input:** Images and labels $(x_i, y_i)$ for task $i \in \mathcal{T} = \{1, 2\}$. Network for both tasks with randomly initialized parameters $\psi_i = (\theta_i, \phi_i)$, $\psi = (\psi_1, \psi_2)$. Sampling times $M$, inner iteration number $L$, the flat region bound $b$. The step sizes $\alpha$, $\beta$.

/* Warm up the network to obtain initialized parameters $\psi^{[0]}$                                */

1  **for** *iteration* $t = 1, 2, \cdots, T_w$ **do**
2       Sampling $\epsilon_i \sim \{\mathcal{U}(-b, b)\}^{d_{\epsilon_i}}$ with $M$ times for $i = 1, 2$, respectively;
3       Compute $\mathcal{L}_{total}$ in Eq. (7);
4       Update $\psi^{[t]} = \psi^{[t-1]} - \alpha \nabla \mathcal{L}_{total}(\psi)$;
5  **end**
6  Start cooperative learning with $\psi^{[0]} = \psi^{[T_w]}$;

/* Alternative Update $\psi_i$ for task $i = 1, 2$.                                                */

7  **for** *Outer iteration* $t = 1, 2, \cdots$ **do**
8       **for** *task* $i = 1, 2$ **do**
9           **for** *inner iteration* $l = 1, 2, \cdots, L$ **do**
10              Sampling $\epsilon_i \sim \{\mathcal{U}(-b, b)\}^{d_{\epsilon_i}}$ with $M$ times for task $i$ ;
11              Compute $L_i^{[t]}(\theta_i, \phi_i)$ in Eq. (5) (or Eq. (6)) with fixed $\theta_{\mathcal{T} \setminus \{i\}}^{[t-1]}$;
12              **if** *l=1* **then**
13                  Update $\psi_i^{[t]} = \psi_i^{[t-1]} - \beta \nabla L_i^{[t]}(\psi_i)$ ;
14              **else**
15                  Update $\psi_i^{[t]} = \psi_i^{[t]} - \beta \nabla L_i^{[t]}(\psi_i)$ ;
16              **end**
17              Clamp $\theta_i^{[t]}$ into $[\theta_i^{[t-1]} - b, \theta_i^{[t-1]} + b]$;
18          **end**
19      **end**
20 **end**

**Output:** Model parameters $(\theta_1, \phi_1, \theta_2, \phi_2)$.

---

**Network Configuration** Fig. 1 illustrates the framework for two-task cooperative learning. Our framework consists of an encoder and task-specific decoders. The parameters at each layer of the encoder are evenly allocated to each task, and the learned features are then concatenated as the input of the next layer.
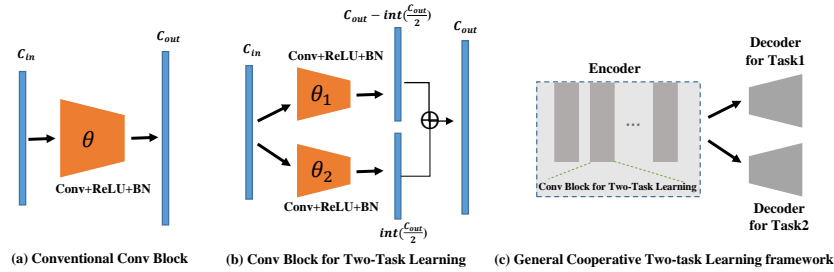


(a) Conventional Conv Block     (b) Conv Block for Two-Task Learning     (c) General Cooperative Two-task Learning framework

**Fig. 1.** A general framework for our MTL method. (a) is the conventional convolution block, (b) illustrates the structure of a convolution block for cooperative two-task learning, and (c) shows the general framework for MTL, which contains an encoder and task-specific decoders.

## 3  Experiments

We validate our MTL framework in three scenarios as follows: (1) classification tasks on different classes with the MNIST dataset [8], (2) one domain for simultaneous segmentation and classification tasks using the REFUGE2018 dataset [13], and (3) one domain for two segmentation tasks with HRF-AV dataset [7]. For our method, we adopt the stochastic gradient descent (SGD) optimizer, and empirically set the bound value $b = 0.05$, the learning rate $\alpha = \beta = 0.1$. To reduce the training time and the memory, we simply set the sampling number $M = 1$. All experiments are implemented using one GTX 1080Ti GPU.

### 3.1  Dataset

(1) **MNIST.** This dataset contains 50,000 training and 10,000 testing images. To simulate a multi-task learning setting, we divide both the training and test images into two subsets with either even numbers $\{0, 2, 4, 6, 8\}$ (denoted as *Task 1*) or odd numbers $\{1, 3, 5, 7, 9\}$ (denoted as *Task 2*). For the network, we adopt the widely used LeNet architecture for MNIST dataset [8], of which the last layer contains 50 hidden units, followed by a final prediction output. (2) **REFUGE2018.** The REFUGE2018 challenge [13] provides 1200 retinal color fundus photography. The target of this challenge is glaucoma detection and optic disc/cup segmentation. We divide this dataset into 800 samples for training and 400 test subset, where the ratio of the number of glaucomas to non-glaucoma images are both 1 : 9. As discussed in [13], glaucoma is mostly characterized by the optic nerve head area. Hence, we cropped all images around the optic disc into $512 \times 512$. We used the UNet [15] for the segmentation task, with the four down-sampling modules as the shared encoders. The output of segmentation and the features from the bottom layers are taken as the input of the decoder for classification. (3) **HRF-AV.** This dataset [7] contains 45 fundus images with a high resolution of $3504 \times 2336$. The tasks for this dataset are the binary vessel segmentation and the artery/vein (A/V) segmentation. We randomly split the dataset into 15 and 30 samples for training and testing. We adopt the U-Net as the backbone with the bottom feature channel being 256. During training, we crop patches with size of $2048 \times 2048$ randomly as input.

### 3.2  Results on MNIST Dataset

**Ablation Study** To validate the effectiveness of the two terms in Eq.(5) and Eq.(6), we conduct two experiments: (1) **Vanilla.** We simply optimize the objective of each task alternatively without any constraints or sampling operations. (2) **Ours (*w/o* Reg).** We sample noises during training, and optimize the losses with solely the first term in Eq.(5) and Eq.(6), *i.e.,* without the similarity regularization. We run 5 times for each method, and report their mean and standard deviation values.

As shown in the top four rows of Table 1, compared to the **Independent** approach, the proposed **Vanilla** bi-level optimization method can utilize the

**Table 1.** Performance of SOTA MTL methods on MNIST dataset. We set the number of parameters of **Joint** method as the base 1, and the values in the column 'Params' are the ratio of the parameter number of each method to the **Joint**.

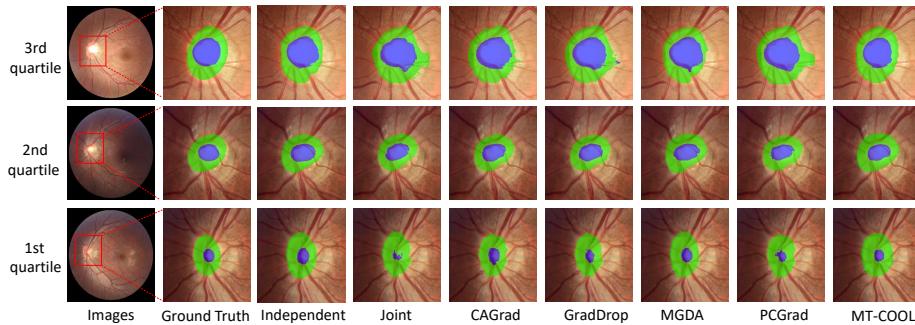| Methods | Params | Task 1 | Task 2 |
|---|---|---|---|
| Independent | $\approx 2$ | 99.41 $\pm$ 0.03492 | 98.77 $\pm$ 0.06029 |
| Ours (Vanilla) | 1 | 99.61$\pm$0.06210 | 99.37$\pm$0.04494 |
| Ours ($w/o$ Reg) | 1 | 99.66$\pm$0.03765 | 99.56$\pm$0.07203 |
| MT-COOL (Ours) | 1 | **99.72$\pm$0.03978** | **99.62$\pm$0.01576** |
| Joint | 1 | 99.60 $\pm$ 0.03765 | 99.51 $\pm$0.06281 |
| CAGrad [10] | 1 | 99.67$\pm$0.05293 | 99.51$\pm$0.05229 |
| GradDrop [3] | 1 | 99.65$\pm$ 0.03492 | 99.53$\pm$0.04245 |
| MGDA [17] | 1 | 99.63$\pm$ 0.05883 | 99.47$\pm$0.05078 |
| PCGrad [22] | 1 | 99.66$\pm$0.04180 | 99.51$\pm$0.09108 |



**Fig. 2.** Visualization results from MTL methods on REFUGE2018 dataset. The selected samples rank the 1st quartile, median and 3rd quartile in terms of the segmentation performance of **Independent**.

features from other tasks and boost the performance of the current one. By introducing noises to find flat minima during training, **Ours ($w/o$ Reg)** further achieves higher prediction, particularly for *Task 2*. Finally, by adding similarity regularization, our method obtains the best results.

**Comparison Study** We compare the proposed method with four state-of-the-art (SOTA) MTL approaches, including MGDA [17], PCGrad [22], GradDrop [3] and CAGrad [10]. We also implement the **Joint** method as a baseline, which simply sums the loss of each task as the total loss for training.

As shown in Table 1, all MTL methods improve the performance on each task, compared to **Independent**. Among all the compared methods, our technique performs the best on both tasks.

### 3.3   Comparison on REFUGE2018 Dataset

For REFUGE2018 dataset, we compare our method with CAGrad, GradDrop, MGDA, PCGrad, and Joint. We run each method three times, and report the

**Table 2.** Performance of SOTA MTL methods on REFUGE2018 dataset.

| Methods | Params | Segmentation | | Classification | | | |
|---|---|---|---|---|---|---|---|
| | | Cup (Dice%) | Disc (Dice%) | Acc | AUROC | Sen | Spe |
| Independent | ≈ 2 | 95.14±0.05110 | 86.87± 005644 | 0.900±0.00235 | 0.902±0.0106 | 0.658±0.0117 | 0.927±0.00392 |
| Joint | 1 | 91.19±0.7600 | 77.36±0.5236 | 0.907±0.0183 | 0.895±0.0221 | 0.658±0.0656 | 0.935±0.0264 |
| CAGrad [10] | 1 | 92.67±0.7702 | 81.71±0.2874 | 0.914±0.00513 | 0.904±0.00562 | 0.658±0.0235 | 0.942±0.00796 |
| GradDrop [3] | 1 | 91.70±0.6376 | 78.91±1.439 | 0.909±0.00424 | 0.922±0.0115 | 0.716±0.0471 | 0.930±0.00988 |
| MGDA [17] | 1 | 93.87±0.5017 | 83.87±0.9732 | 0.895±0.0154 | 0.914±0.00610 | 0.633±0.0824 | 0.924±0.0260 |
| PCGrad [22] | 1 | 91.74±0.5569 | 79.80±0.8748 | 0.911±0.00849 | 0.898±0.0136 | 0.675±0.0204 | 0.937±0.00796 |
| MT-COOL (Ours) | 1 | **94.37±0.1706** | **86.18±0.3046** | **0.937±0.0113** | **0.942±0.0149** | **0.750±0.000** | **0.958±0.0126** |

**Table 3.** Performance of SOTA MTL methods on HRF-AV dataset.

| Methods | Params | A/V Segmentation | | | | | | Binary Segmentation | |
|---|---|---|---|---|---|---|---|---|---|
| | | Acc (A) | F1 (A) | Acc (V) | F1 (V) | Acc (AV) | F1 (A/V) | Acc | F1 |
| Independent | ≈ 2 | 0.9814 | 0.6999 | 0.9821 | 0.7492 | 0.9692 | 0.7698 | 0.9691 | 0.7831 |
| Joint | 1 | 0.9622 | 0.3537 | 0.9661 | 0.5171 | 0.9664 | 0.7360 | 0.9691 | 0.7835 |
| CAGrad [10] | 1 | 0.9687 | 0.4754 | 0.9696 | 0.5520 | 0.9668 | 0.7364 | 0.9690 | 0.7790 |
| GradDrop [3] | 1 | 0.9708 | 0.5127 | 0.9716 | 0.5736 | 0.9666 | 0.7343 | 0.9686 | 0.7742 |
| MGDA [17] | 1 | 0.9636 | 0.2343 | 0.9632 | 0.5315 | 0.9660 | 0.7263 | 0.9691 | 0.7793 |
| PCGrad [22] | 1 | 0.9671 | 0.4262 | 0.9681 | 0.5387 | 0.9667 | 0.7357 | 0.9687 | 0.7763 |
| MT-COOL (Ours) | 1 | **0.9801** | **0.6671** | **0.9811** | **0.7135** | **0.9674** | **0.7424** | **0.9701** | **0.7912** |

$mean \pm std$ values of Dice score on optic cup and disc for the segmentation task, and accuracy (Acc), Area Under the Receiver Operating Characteristics (AUROC), sensitivity (Sen) and specificity (Spe) for the classification task.

As shown in Table 2, our method achieves comparable results on the segmentation task with the **Independent**, while other MTL methods degrade significantly, particularly on Disc. For the classification task, our method achieves the best performance in terms of all the metrics. Fig. 2 provides the visualization results for qualitative comparison. One can see that the proposed method obtains the best prediction shape among all MTL methods.

### 3.4   Comparison on HRF-AV Dataset

We also conduct a comparison study on HRF-AV dataset. Each method is repeated three times, and the mean results are presented in Table 3. One can see that compared to the **Independent**, all the other MTL methods perform poorly, especially on A/V segmentation task. For example, the best F1 scores on A/V segmentation among the five MTL methods are 0.5127 and 0.5736, respectively, obtained by GradDrop, which are much lower than those from **Independent**. On the contrary, our method performs comparably with the **Independent** on A/V segmentation, and even slightly better on binary segmentation. For qualitative comparison, please refer to Fig.1 in the Supplementary material.

## 4 Conclusion

In this work, we propose a novel MTL framework via bi-level optimization. Our method learns features for each task in a cooperative manner, instead of competing for resources with each other. We validate our model on three datasets, and the results prove its great potential in MTL. However, there are still some issues that need to be studied in the future. For example, we need to validate our method on large-scale tasks and find a more efficient learning strategy such as using distributed learning. Moreover, how to allocate the parameters to each task automatically and effectively is important for model generalization. For better interpretability, learning features specific to each task should also be studied.

## 5 Acknowledgments

## References

1. Biswas, A., Hoyle, C.: A literature review: solving constrained non-linear bi-level optimization problems with classical methods. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. vol. 59193, p. V02BT03A025. American Society of Mechanical Engineers (2019)
2. Chen, Z., Badrinarayanan, V., Lee, C.Y., Rabinovich, A.: Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In: International conference on machine learning. pp. 794–803. PMLR (2018)
3. Chen, Z., Ngiam, J., Huang, Y., Luong, T., Kretzschmar, H., Chai, Y., Anguelov, D.: Just pick a sign: Optimizing deep multitask models with gradient sign dropout. Advances in Neural Information Processing Systems **33**, 2039–2050 (2020)
4. Crawshaw, M.: Multi-task learning with deep neural networks: A survey. arXiv preprint arXiv:2009.09796 (2020)
5. He, K., Lian, C., Zhang, B., Zhang, X., Cao, X., Nie, D., Gao, Y., Zhang, J., Shen, D.: Hf-unet: learning hierarchically inter-task relevance in multi-task u-net for accurate prostate segmentation in ct images. IEEE Transactions on Medical Imaging **40**(8), 2118–2128 (2021)
6. He, T., Hu, J., Song, Y., Guo, J., Yi, Z.: Multi-task learning for the segmentation of organs at risk with label dependence. Medical Image Analysis **61**, 101666 (2020)
7. Hemelings, R., Elen, B., Stalmans, I., Van Keer, K., De Boever, P., Blaschko, M.B.: Artery–vein segmentation in fundus images using a fully convolutional network. Computerized Medical Imaging and Graphics **76**, 101636 (2019)
8. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
9. Li, W.H., Liu, X., Bilen, H.: Learning multiple dense prediction tasks from partially annotated data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18879–18889 (2022)
10. Liu, B., Liu, X., Jin, X., Stone, P., Liu, Q.: Conflict-averse gradient descent for multi-task learning. Advances in Neural Information Processing Systems **34**, 18878–18890 (2021)

11. Liu, L., Dou, Q., Chen, H., Qin, J., Heng, P.A.: Multi-task deep model with margin ranking loss for lung nodule analysis. IEEE transactions on medical imaging **39**(3), 718–728 (2019)
12. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3994–4003 (2016)
13. Orlando, J.I., Fu, H., Breda, J.B., Van Keer, K., Bathula, D.R., Diaz-Pinto, A., Fang, R., Heng, P.A., Kim, J., Lee, J., et al.: Refuge challenge: A unified framework for evaluating automated methods for glaucoma assessment from fundus photographs. Medical image analysis **59**, 101570 (2020)
14. Requeima, J., Gordon, J., Bronskill, J., Nowozin, S., Turner, R.E.: Fast and flexible multi-task classification using conditional neural adaptive processes. Advances in Neural Information Processing Systems **32** (2019)
15. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)
16. Ruder, S.: An overview of multi-task learning in deep neural networks. CoRR, abs/1706.05098 (2017)
17. Sener, O., Koltun, V.: Multi-task learning as multi-objective optimization. Advances in neural information processing systems **31** (2018)
18. Shao, W., Wang, T., Sun, L., Dong, T., Han, Z., Huang, Z., Zhang, J., Zhang, D., Huang, K.: Multi-task multi-modal learning for joint diagnosis and prognosis of human cancers. Medical Image Analysis **65**, 101795 (2020)
19. Shi, G., Chen, J., Zhang, W., Zhan, L.M., Wu, X.M.: Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima. Advances in Neural Information Processing Systems **34**, 6747–6761 (2021)
20. Uslu, F., Varela, M., Boniface, G., Mahenthran, T., Chubb, H., Bharath, A.A.: La-net: A multi-task deep network for the segmentation of the left atrium. IEEE Transactions on Medical Imaging **41**(2), 456–464 (2021)
21. Wang, Y., Liu, W., Yu, Y., Liu, J.j., Xue, H.d., Qi, Y.f., Lei, J., Yu, J.c., Jin, Z.y.: Ct radiomics nomogram for the preoperative prediction of lymph node metastasis in gastric cancer. European radiology **30**(2), 976–986 (2020)
22. Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., Finn, C.: Gradient surgery for multi-task learning. Advances in Neural Information Processing Systems **33**, 5824–5836 (2020)
23. Zhang, Y., Li, H., Du, J., Qin, J., Wang, T., Chen, Y., Liu, B., Gao, W., Ma, G., Lei, B.: 3d multi-attention guided multi-task learning network for automatic gastric tumor segmentation and lymph node classification. IEEE Transactions on Medical Imaging **40**(6), 1618–1631 (2021)
24. Zhou, Y., Chen, H., Li, Y., Liu, Q., Xu, X., Wang, S., Yap, P.T., Shen, D.: Multi-task learning for segmentation and classification of tumors in 3d automated breast ultrasound images. Medical Image Analysis **70**, 101918 (2021)