Learning with Privileged Tasks

Yuru Song[1] Zan Lou[2] Shan You[2,3*] Erkun Yang[4*]
Fei Wang[5] Chen Qian[2] Changshui Zhang[3] Xiaogang Wang[2,6]
[1]University of California San Diego [2]SenseTime Research
[3]Department of Automation, Tsinghua University
Institute for Artificial Intelligence, Tsinghua University (THUAI)
Beijing National Research Center for Information Science and Technology (BNRist)
[4]Xidian University [5]University of Science and Technology of China
[6]Chinese University of Hong Kong
yus027@ucsd.edu, {louzan,youshan,qianchen}@sensetime.com, erkunyang@gmail.com
wangfei91@mail.ustc.edu.cn, zcs@mail.tsinghua.edu.cn, xgwang@ee.cuhk.edu.hk

Abstract

Multi-objective multi-task learning aims to boost the performance of all tasks by leveraging their correlation and conflict appropriately. Nevertheless, in real practice, users may have preference for certain tasks, and other tasks simply serve as privileged or auxiliary tasks to assist the training of target tasks. The privileged tasks thus possess less or even no priority in the final task assessment by users. Motivated by this, we propose a privileged multiple descent algorithm to arbitrate the learning of target tasks and privileged tasks. Concretely, we introduce a privileged parameter so that the optimization direction does not necessarily follow the gradient from the privileged tasks, but concentrates more on the target tasks. Besides, we also encourage a priority parameter for the target tasks to control the potential distraction of optimization direction from the privileged tasks. In this way, the optimization direction can be more aggressively determined by weighting the gradients among target and privileged tasks, and thus highlight more the performance of target tasks under the unified multi-task learning context. Extensive experiments on synthetic and real-world datasets indicate that our method can achieve versatile Pareto solutions under varying preference for the target tasks.

1. Introduction

Besides designing strong model structures [32, 46, 35, 13, 34] and informative task losses [19, 48, 5, 44, 52], multi-task learning (MTL) [50, 38] is apt to enhance performance and efficiency by seeking more appropriate ways to combine multiple tasks, and increasingly gains much research interest. The paradigm of MTL has been shown to outperform single-task learning (STL) on numerous computer vision problems, such as attribute recognition [51], scene understanding [24] and autonomous driving [9]. To exploit the task correlations, current MTL approaches mainly follow a soft-parameter or hard-parameter sharing principle. In soft-parameter sharing, tasks are aggregated separately and cross-talks [28] among these tasks are usually used to encourage shared knowledge. Nevertheless, the intricacy of designing the cross-talks is specific to particular problem sets, and does not scale well to many tasks. In contrast, hard-parameter sharing leverages a unique backbone to pursue direct shared representation of tasks [1, 2], along with task-specific sub-networks. Therefore, hard-parameter sharing is able to reduce parameter size in proportion to the task number, and to promote inference speed during testing.

Though parameters among tasks are shared in a hard manner, how to balance all tasks still matters for MTL. Task balancing to circumvent this difficulty goes beyond naive uniform weighting of tasks [51]. Existing heuristics to find appropriate weighting of multiple tasks include grid searching, exploring task uncertainties [10], and gradient normalization [6]. Recent avant-garde method is to treat the MTL as multi-objective optimization (MOO-MTL) [33]. It proposes to find the Pareto front that only allows the common improvement of tasks rather than sacrificing any individual. Task weights are evaluated dynamically during learning.

Current MTL methods treat all tasks equally, and focus on the average performance of all tasks during inference. In practice, however, users may only need to scrutinize the performance of some target tasks rather than that of

*Corresponding authors.

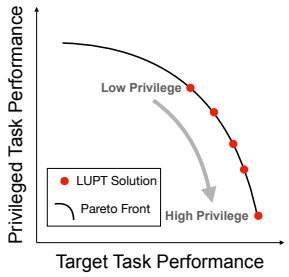


Figure 1: **Learning with Privileged Tasks (PTL)** endows some tasks with adjustable privilege to assist the target tasks. When the privileged parameter is low or zero, the trade-off is less or not biased towards the target tasks. When the privileged parameter is high, the target task performance is prioritized during training.

all tasks. One natural solution for such scenario is to learn only the target tasks solely. However, this practice neglects the potential benefits from other tasks. In contrast, our aim is to include all tasks under MTL context, but non-target tasks only serve as *privileged* tasks[1] to assist the training of target tasks, and are irrelevant for the performance evaluation by users. Some literature also refers *auxiliary* tasks as the privileged tasks. However, they either focus on design auxiliary tasks for target tasks [29], or does not consider the intrinsic conflicts and competitions among tasks [45]. Conflicts may exist naturally between target and privileged tasks, or even within the target tasks themselves. Existing variants of MOOMTL also model this problem with *preference* [22, 27, 26]. Nevertheless, the preference vectors are usually defined in the loss space. Designing these vectors requires *a priori* information of the individual loss magnitudes on the Pareto front. In practice, users often fail to provide an approximation of the Pareto front.

In this paper, we cast the learning of target and privileged tasks (PTL) in a unified multi-objective optimization problem [33] to manage the correlation and conflict between target and privileged tasks simultaneously. Instead of considering the loss space, we examine the task priority in the gradient space, and proposes a privileged multiple gradient descent algorithm (P-MGDA) to amplify the performance of target tasks. Concretely, for each mini-batch we introduce a privileged parameter so that the descent direction does not necessarily follow the gradients of privileged tasks, but concentrates more on the target tasks. In this way, we can flexibly moderate the conflict or competition between target and privileged tasks. Moreover, we also encourage a priority term to regulate direction correction towards the target tasks. In this way, when the target task and privileged tasks only cooperate, we can still ensure the consistent exploitation of the privileged tasks.

Our approach, privileged task learning (PTL), introduces parameters working on the gradient space, so that we can control them to achieve versatile Pareto critical points to cater user preference as shown in Figure 1. We optimize our proposed P-MGDA by an efficient hybrid-block coordinate descent (CD) algorithm. Extensive experiments on both synthetic and real datasets validate the effectiveness of our PTL. Results show that PTL is able to find the solution that not only has overall satisfactory performance across all tasks but especially good for the target task.

2. Related work

Auxiliary learning in multi-task learning Recent review such as [37] offers an extensive and detailed survey of the auxiliary tasks in the MTL setting. There are diverse ways to form the auxiliary tasks. In PAD-Net [43], intermediate auxiliary outputs from the network are trained to help the main tasks. The feature space of the main task is directly built upon that of the auxiliary tasks. Du et al. proposed that there should be a similarity between the main tasks and the constructed auxiliary tasks for the MTL to be effective [15]. Their proposed method measures the similarities between the main and auxiliary tasks using cosine similarity, based on task gradients. Weight balancing for auxiliary tasks proposed by [45] drives positive transfer and suppresses negative transfer by leveraging class-wise weights in the learning process, and exploits the helpful information from the auxiliary tasks. Whereas in our setting, privileged tasks and auxiliary tasks have identical placements.

Multi-objective optimization with preference. Following the pivotal framework brought by [33], extensions of the MOOMTL mostly try to expand the Pareto front. In Pareto-MTL [22], reference vectors are used to guide the MOO searching, which leads to a set of solutions with different trade-off among all tasks. [27] improves Pareto-MTL and strictly solves the preference-specific Pareto solutions, which sit exactly along the reference vectors. In [26], continuous solutions are found in the vicinity of a Pareto solution. Our framework can also cater user preference by tuning the priority and privileged parameters.

Learning using privileged information (LUPI). LUPI [40, 25, 39] assumes each example corresponds to a regular feature and an additional privileged feature, so the privileged features can be used to boost the performance of trained models [47, 36, 49, 41]. However, privileged features are not involved or even available during inference. We name our method *privileged tasks* inspired by this analogy, but there are significant differences in the problem setting. In LUPI, both regular and privileged features are subject to the same single task; in our PTL, examples correspond to multiple and maybe different tasks, and privileged tasks are to boost the target tasks.

[1] We name our method *privileged tasks* motivated by the prior work [40, 25, 39] which investigates Learning Using Privileged Information (LUPI) to boost the training and is not involved during inference as well.

3. Revisiting MGDA in Multi-objective MTL

We formalize the multi-objective optimization problem of MTL as follows [33]. Assume there are M tasks with the index set as $\mathcal{I}$. Denote the shared parameters of the network as $\boldsymbol{\theta}$, and the updated $\boldsymbol{\theta}$ at training step τ as $\boldsymbol{\theta}^{(\tau)}$. But we will dismiss τ for short in the following. Each task, t, has its own task-specific network, with parameters $\boldsymbol{\theta}^t, t \in \mathcal{I}$. And the whole parameter set for the MTL network is $\{\boldsymbol{\theta}, \boldsymbol{\theta}^t, t \in \mathcal{I}\}$. The input space is $\mathcal{X}$, and the output space of the task t is $\mathcal{Y}^t, t \in \mathcal{I}$. The dataset is therefore $\{x_i, y_i^1, y_i^2, \ldots, y_i^T\}_{i\in[N]}$, where N is the size of the dataset. Each task has its loss function as $l^t(\boldsymbol{\theta}^t, \boldsymbol{\theta})$. When working with the shared parameter $\boldsymbol{\theta}$, the task-specific parameter $\boldsymbol{\theta}^t$ will be dismissed in notation for short.

Our goal is to find appropriate weighting vector, c^t, to formulate the scalar loss, $\sum_{t\in\mathcal{I}} c^t \ell^t$, such that all of its components, task-specific losses, are jointly optimized to the most during learning. Towards a moderation of the task conflicts, we take a worthy detour by considering the vectorized loss, $(\ell^1, \ell^2, \ldots, \ell^T)$, leveraging the existing framework of multi-objective optimization. The essential conviction is to find the *Pareto optimal* solution of all the objectives [12], defined below.

Definition 1 (Pareto optimality for MTL)**.** *(a). A solution $\boldsymbol{\theta}$ dominates a solution $\bar{\boldsymbol{\theta}}$ if $\ell^t(\boldsymbol{\theta}, \boldsymbol{\theta}^t) \leq \ell^t(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}^t)$ for all tasks t and $\big(\ell^1(\boldsymbol{\theta}, \boldsymbol{\theta}^t), \ell^2(\boldsymbol{\theta}, \boldsymbol{\theta}^t), \ldots, \ell^T(\boldsymbol{\theta}, \boldsymbol{\theta}^t)\big) \neq \big(\ell^1(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}^t), \ell^2(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}^t), \ldots, \ell^T(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}^t)\big)$. (b). A solution $\boldsymbol{\theta}^\star$ is called Pareto optimal if there exists no solution $\boldsymbol{\theta}$ that dominates $\boldsymbol{\theta}^\star$.*

Instead of solving the Pareto optimal point, the multiple gradient descent algorithm (MGDA) [12, 17] turns to a necessary Pareto critical point by leveraging the Karush-Kuhn-Tucker (KKT) condition. As in [17, 33, 22], the minimization problem

$$\min_{v,\boldsymbol{d}} \Big(v + \frac{1}{2}\|\boldsymbol{d}\|^2\Big), s.t. \langle \nabla \ell^m_{\boldsymbol{\theta}}, \boldsymbol{d}\rangle \leq v, m \in \mathcal{I}. \quad (1)$$

satisfy the following Lemma 1.

Lemma 1. *[17, 33, 22] Let $(\boldsymbol{d}, v)$ be the solution of problem (1),*

1. *If $\boldsymbol{\theta}$ is Pareto critical, then $\boldsymbol{d} = \mathbf{0}$ and $v = 0$.*
2. *If $\boldsymbol{\theta}$ is not Pareto critical, then*

$$\begin{aligned} &v \leq -(1/2)\|\boldsymbol{d}\|^2 < 0 \\ &\nabla \ell^m_{\boldsymbol{\theta}} \leq v, \forall\, m \in \mathcal{I} \end{aligned} \quad (2)$$

In other words, either the solution of problem (1) is 0 thus there is no direction to improve all tasks at the same time, or the solution leads to a descent direction that improves all tasks. Then recent multi-objective MTL methods (*e.g.*, MOO-MTL [33] and Pareto-MTL [22]) adopt MGDA to solve the MTL problem by considering its dual problem

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\sum_{m\in\mathcal{I}} \alpha_m \nabla_{\boldsymbol{\theta}} \ell^m\|^2, \text{s.t.} \sum_{m\in\mathcal{I}} \alpha_m = 1, \alpha_m \geq 0. \quad (3)$$

with the optimal solution $\boldsymbol{d}^*$ of Eq.(1) and optimal solution $\boldsymbol{\alpha}^*$ of Eq.(3) satisfying

$$\boldsymbol{d}^* = \sum_{m\in\mathcal{I}} \alpha^*_m \nabla_{\boldsymbol{\theta}} \ell^m, \text{s.t.} \sum_{m\in\mathcal{I}} \alpha^*_m = 1, \alpha^*_m \geq 0. \quad (4)$$

To elude the time-consuming operation of calculating the gradient over parameter $\nabla_{\boldsymbol{\theta}}\ell$, [33, 22, 14] found an upper bound (MGDA-UB) of $\|\sum_{m\in\mathcal{I}} \alpha_m \nabla_{\boldsymbol{\theta}} \ell^m\|$, by computing $\|\sum_{m\in\mathcal{I}} \alpha_m \nabla_{\boldsymbol{Z}} \ell^m\|$. $\nabla_{\boldsymbol{Z}}\ell^m$ is the gradient over the shared intermediate representations.

4. Privileged MGDA with Task Priority

We expand the MGDA based on the following consideration. Despite the effectiveness of finding Pareto solutions, the MTL problem can have numerous optimal tradeoffs among the tasks, but the single solution obtained by MOOMTL might not serve or even contravene the specific preference from the user. To alleviate this scenario, we further consider the existence of privileged tasks, which serves to assist the training of the target task and are of less concern at inference stage. Therefore the privileged tasks are allowed to recess moderately during learning. These potentially deteriorated privileged tasks are denoted by index set $\mathcal{I}_p$. And the target task is denoted by index set $\mathcal{I}_t$. Assume there are M_t target tasks and M_p privileged tasks. The total number of both target and privileged tasks is $M_t + M_p$, which will be interchangeably written as M. Consider the inequality constraint in Problem (1),

$$\langle \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}), \boldsymbol{d}\rangle \leq v. \quad (5)$$

It guarantees that the amount of descent is no less than $|v|$. This applies to both target tasks and privileged tasks. But when task conflict or competition arises, this constraint can be relaxed for the privileged tasks, which will give priority to the target tasks. We solidify and justify this idea as follows.

4.1. Slack descent for privileged tasks

We introduce slack variables $\xi_i \geq 0$ in the Eq. (5) for the privileged tasks, which can be learned during training and flexibly moderate the task conflicts. The resulted descent direction might increase the loss function for the privileged tasks, as ξ_j could be large enough and $v + \xi_j$ is positive. To avoid unnecessary sacrifice of the privileged task, we set an upper bound of $\boldsymbol{\xi}$ with the regularization parameter $C_1 \geq 0$.

And the slack descent model is concreted in Eq. (6)

$$\langle \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}), \boldsymbol{d} \rangle \leq v + \xi_j, \forall j \in \mathcal{I}_p. \tag{6}$$

In the corresponding dual problem, we have new constraint for the coefficient of the privileged tasks as

$$0 \leq \alpha_j \leq C_1, \forall j \in \mathcal{I}_p. \tag{7}$$

C_1 controls the scope of privilege and is decided by the user. Under the simple case with two task learning, there is only one target task l_t and one privileged task l_p. $\sum_{m=1}^{M} \alpha_m \nabla_{\boldsymbol{\theta}} \ell_m(\boldsymbol{\theta})$ is reduced to the summation of only two scaled gradients, $\alpha_1 \nabla l_t + \alpha_2 \nabla l_p$. C_1 sets the upper bound of α_2, i.e. the largest possible consideration for the privileged task during training. Therefore C_1 determines how close the descent direction locates to the gradient of the target task, shown by the upper panel of Figure 2. Large C_1 gives space for α_2, therefore promotes the proportion of privileged task during training while small C_1 limits the participation of privileged tasks and prioritizes more the target tasks accordingly.

4.2. Direction correction with priority

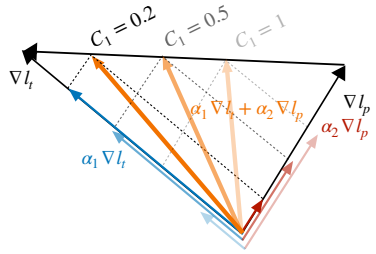
Degeneracy of Eq. (6) happens when there is no conflict between target and privileged tasks, as loss ascent for privileged task will only be redundant. To consistently exploit the privileged tasks, we further demand larger improvement only for the target tasks, disregarding the privileged tasks. Such request is established by additional inequality constraints. In other words, $\langle \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}), \boldsymbol{d} \rangle \leq \langle \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}), \boldsymbol{d} \rangle, \ \forall i \in \mathcal{I}_t, j \in \mathcal{I}_p$. This will introduce additional term , $\sum_{i \in \mathcal{I}_t} \sum_{j \in \mathcal{I}_p} \beta_{ij} (\nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}))$, in the dual problem, where $\beta_{ij} \geq 0$ stands for the multiplier for each pair of the inequality constraint in Eq. (8). We can interpret this term as further correction for the descent direction, which bends the optimization even towards the target tasks

$$\langle \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}), \boldsymbol{d} \rangle \leq \langle \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}), \boldsymbol{d} \rangle + \eta_{ij}, \forall i \in \mathcal{I}_t, j \in \mathcal{I}_p. \tag{8}$$

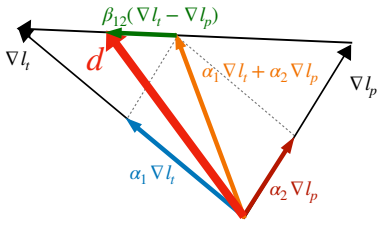
However, prioritizing each target task over the privileged task may shrink the feasible set excessively. To relax this constraint appropriately, we also introduce learnable slack variable $\eta_{ij} \geq 0$, which can be regularized by parameter $C_2 \geq 0$. Similarly, additional constraints on β_{ij} will be introduced in the dual problem as

$$\beta_{ij} \leq C_2, \forall i \in \mathcal{I}_t, j \in \mathcal{I}_p. \tag{9}$$

There could be the case that for $i \in \mathcal{I}_p, \alpha_i < \sum_{j \in \mathcal{I}_p} \beta_{ij}$. And the coefficient of $\nabla_{\boldsymbol{\theta}} \ell_i$ in $\boldsymbol{d}$ is negative. In this case, $\boldsymbol{d}$ will not be a convex combination of the task gradients. To avoid the collapse of the MGDA approach, we further exert



(a) Slack descent for privileged tasks.



(b) Direction correction prioritizes the target tasks.

Figure 2: **Geometric interpretation**. (a). Privileged parameter biases the descend direction towards target task to a controllable extent. (b). Direction correction further push the descent direction towards the target task. In whole, this illustrates the theoretical effectiveness of PTL.

the following constraint so that $\alpha_i, i \in \mathcal{I}_p$ is large enough, as

$$\sum_{i \in \mathcal{I}_t} \beta_{ij} \leq \alpha_j, \forall j \in \mathcal{I}_p. \tag{10}$$

Again we refer two task learning for illustrative analysis of the model, shown in Figure 2. In this case $\sum_{i \in \mathcal{I}_t} \sum_{j \in \mathcal{I}_p} \beta_{ij} (\nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}))$ is as simple as $\beta_{12}(\nabla l_t - \nabla l_p)$. Larger β_{12} corresponds to greater correction for the direction towards the target task. Such correction is bounded by C_2, and thus adjustable according to user preference.

4.3. Theoretical analysis

With the slack descent of the privileged tasks and direction correction towards the target tasks, the PTL model is summarized as follows

$$\begin{aligned} \min_{\boldsymbol{d}, v, \boldsymbol{\xi}, \boldsymbol{\eta}} \ & v + C_1 \cdot \sum_{j \in \mathcal{I}_a} \xi_j + C_2 \cdot \sum_{i \in \mathcal{I}_p} \sum_{j \in \mathcal{I}_a} \eta_{ij} + \frac{1}{2} \|\boldsymbol{d}\|^2, \\ \text{s.t.} \ & \langle \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}), \boldsymbol{d} \rangle \leq v, \ \forall i \in \mathcal{I}_t, \\ & \langle \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}), \boldsymbol{d} \rangle \leq v + \xi_j, \ \xi_j \geq 0, \ \forall j \in \mathcal{I}_p, \\ & \langle \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}), \boldsymbol{d} \rangle \leq \langle \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}), \boldsymbol{d} \rangle + \eta_{ij}, \\ & \eta_{ij} \geq 0, \forall i \in \mathcal{I}_t, \forall j \in \mathcal{I}_p. \end{aligned} \tag{11}$$

By means of the Lagrangian multiplier, the dual problem of the PTL problem is formalized as follows

$$\begin{aligned}\min_{\boldsymbol{\alpha},\boldsymbol{\beta}}\ & \frac{1}{2}\|\sum_{m=1}^{M}\alpha_m\nabla_{\boldsymbol{\theta}}\ell_m(\boldsymbol{\theta})+\\ & \sum_{i\in\mathcal{I}_t}\sum_{j\in\mathcal{I}_p}\beta_{ij}(\nabla_{\boldsymbol{\theta}}\ell_i(\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}\ell_j(\boldsymbol{\theta}))\|^2,\\ \text{s.t.}\ & \sum_{m\in\mathcal{I}}\alpha_m=1,\ 0\le\alpha_j\le C_1,\ \forall j\in\mathcal{I}_p,\\ & 0\le\beta_{ij}\le C_2,\forall i\in\mathcal{I}_t,\forall j\in\mathcal{I}_p,\\ & \sum_{i\in\mathcal{I}_t}\beta_{ij}\le\alpha_j,\forall j\in\mathcal{I}_p.\end{aligned}\tag{12}$$

where $\boldsymbol{\alpha}\in\mathbb{R}^M,\boldsymbol{\beta}\in\mathbb{R}^{M_t\times M_p}$. Following Lemma 1, we can also prove that the direction $\boldsymbol{d}$ can indeed improve the target tasks while keeping the privileged tasks controllably improved or declined, as Theorem 1.

Theorem 1. *Following Lemma 1, let $(\boldsymbol{d}^*,v^*,\boldsymbol{\xi}^*,\boldsymbol{\eta}^*)$ be the solution of Problem (11).*

1. *If $\boldsymbol{d}^*=\mathbf{0}$, then the solution is Pareto critical;*
2. *If $\boldsymbol{d}^*\neq\mathbf{0}$, then target tasks and the privileged tasks that satisfies $-\|\boldsymbol{d}\|^2-C_1\sum_{j\in\mathcal{I}_p}\xi_j^*-C_2\sum_{i\in\mathcal{I}_t}\sum_{j\in\mathcal{I}_p}\eta_{ij}^*+\xi_j^*<0$ will descend:*

$$\begin{aligned}&\forall i\in\mathcal{I}_t,\ \langle\nabla_{\boldsymbol{\theta}}\ell_i(\boldsymbol{\theta}),\boldsymbol{d}^*\rangle\le v^*\\ &=-\|\boldsymbol{d}\|^2-C_1\sum_{j\in\mathcal{I}_t}\xi_j^*-C_2\sum_{i\in\mathcal{I}_t}\sum_{j\in\mathcal{I}_p}\eta_{ij}^*,\\ &\forall j\in\mathcal{I}_p,\ \langle\nabla_{\boldsymbol{\theta}}\ell_j(\boldsymbol{\theta}),\boldsymbol{d}^*\rangle\le v^*+\xi_j^*\\ &=-\|\boldsymbol{d}\|^2-C_1\sum_{j\in\mathcal{I}_p}\xi_j^*-C_2\sum_{i\in\mathcal{I}_t}\sum_{j\in\mathcal{I}_p}\eta_{ij}^*+\xi_j^*.\end{aligned}\tag{13}$$

Theorem 1 can be proved similarly as Lemma 1. Detailed proofs refer to the supplementary materials. Note that instead of using the gradient over parameters, we also use gradient over shared intermediate representations in PTL for efficiency. In the sequel, we proceed to solve the dual Problem (12) with a novel hybrid coordinate descent method, or, the "HybridSolver". Our learning algorithm is summarized in Algorithm 1.

4.4. Optimization

In this section, we describe our approach for solving the dual problem (12). For succinct formulation, let $\nabla_t=[,\nabla_{\boldsymbol{\theta}}\ell_m(\boldsymbol{\theta}),]\in\mathbb{R}^{d\times M_t}$, $m\in\mathcal{I}_t$, and $\nabla_p=[,\nabla_{\boldsymbol{\theta}}\ell_m(\boldsymbol{\theta}),]\in\mathbb{R}^{d\times M_p}$, $m\in\mathcal{I}_p$. Then Eq.(11) can be rewritten as

$$\begin{aligned}\min_{\boldsymbol{\alpha},\boldsymbol{\beta}}\ & \frac{1}{2}\left\|[\nabla_t,\nabla_p]\boldsymbol{\alpha}+\nabla_t\boldsymbol{\beta}\mathbf{1}_p-\nabla_p\boldsymbol{\beta}^T\mathbf{1}_t\right\|^2,\\ \text{s.t.}\ & \boldsymbol{\alpha}^T\mathbf{1}=1,\quad\boldsymbol{\beta}^T\mathbf{1}_t\le\boldsymbol{\alpha}_p,\\ & 0\le\boldsymbol{\alpha}_p\le C_1,\quad 0\le\boldsymbol{\beta}\le C_2.\end{aligned}\tag{14}$$

Algorithm 1: Learning under privileged tasks

Initialization;
for $m\leftarrow 1$ **to** M **do**
  Compute task specific gradients: $\nabla_{\boldsymbol{\theta}^m}l_m(\boldsymbol{\theta}^m,\boldsymbol{\theta})$
  Update task specific parameters: $\boldsymbol{\theta}^m=\boldsymbol{\theta}^m-\nabla_{\boldsymbol{\theta}^m}l_m(\boldsymbol{\theta}^m,\boldsymbol{\theta})$
end
Compute shared gradients: $[\nabla_t,\nabla_p]$
Solve Eq. (14): $\boldsymbol{\alpha},\boldsymbol{\beta}=\text{HybridSolver}(\boldsymbol{\theta})$
Update shared parameters: $\boldsymbol{\theta}=\boldsymbol{\theta}-r\cdot([\nabla_t,\nabla_p]\boldsymbol{\alpha}+\nabla_t\boldsymbol{\beta}\mathbf{1}_p-\nabla_p\boldsymbol{\beta}^T\mathbf{1}_t)$

where $\boldsymbol{\alpha}\in\mathbb{R}^M,\boldsymbol{\beta}\in\mathbb{R}^{M_t\times M_p}$. We use $\boldsymbol{x}=[\boldsymbol{\alpha},\hat{\boldsymbol{\beta}}]$. $\hat{\boldsymbol{\beta}}$ is the vectorization of $\boldsymbol{\beta}$ that satisfies $\boldsymbol{\beta}\mathbf{1}_p=A\hat{\boldsymbol{\beta}}$, $\mathbf{1}_t^T\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}P$. Q is the positive semi-definite matrix calculated from the inner products of the task gradients and their subtractions. Refer to the supplementary materials for the explicit expression of Q.

This problem can relate with one class support vector machine's (OC-SVM) dual problem in their formats. One efficient method for this kind of optimization is the coordinate descent (CD) method (or decomposition method [7]). It iteratively optimizes along a few coordinates in a working set B each time. Each iteration solves one sub-problem of optimization. Due to the constraint $\boldsymbol{\alpha}^T\mathbf{1}=1$ on $\boldsymbol{\alpha}$, we apply the selection method of B in [8]. For $\boldsymbol{\beta}$, random selection of the coordinates has been shown to be effective [3]. In our case, the sub-problems can be formulated in (15) and (16) below.

$$\begin{aligned}\min_{\alpha_i,\alpha_j}\quad & Q_{ii}\alpha_i^2+Q_{jj}\alpha_j^2+2Q_{ij}\alpha_i\alpha_j\\ & +2\sum_{x_k\notin\{\alpha_i,\alpha_j\}}(Q_{ik}\alpha_i+Q_{jk}\alpha_j)x_k\\ \text{s.t.}\quad & \alpha_i+\alpha_j=1-\sum_{x_k\notin\{\alpha_i,\alpha_j\}}\alpha_k,\\ & 0\le\alpha_j\le C_1,\forall i\in\mathcal{I}_t,\\ & \sum_j\beta_{ij}\le\alpha_j\le C_1,\forall i\in\mathcal{I}_p.\end{aligned}\tag{15}$$

$$\begin{aligned}\min_{\hat{\beta}_i}\quad & Q_{ii}\hat{\beta}_i^2+2\sum_{x_k\notin\{\beta_i\}}Q_{ik}\hat{\beta}_i x_k\\ \text{s.t.}\quad & 0\le\hat{\beta}_i\le\min(C_2,\alpha_k-\sum_j\beta_{jk}),\hat{\beta}_i\in\{\beta_{jk}\}.\end{aligned}\tag{16}$$

Selecting working set B influences the speed of convergence. Some heuristics have been proposed for linear SVM. As has been discussed in [8], the inherent linear constraint

in the OC-SVM's dual problem will slow down the convergence in the CD procedure. Since a linear constraint may cause a sub-problem to be already optimal and thus the variables cannot be updated [3]. To alleviate this, the authors proposed two-level CD methods for OC-SVM. But in our case, the linear constraint only applies partially to $\boldsymbol{\alpha}$. We introduce the following hybrid CD algorithm to tackle the problem. The gist is that when optimizing along $\boldsymbol{\alpha}$, we apply the two-level CD in [8] to accelerate the optimization, and when optimizing along $\boldsymbol{\beta}$, we use the dual CD in [3] to exploit its success. The details of selecting method is in supplementary material. Solving the dual problem Eq. (14) is summarized in **Algorithm 2** in supplementary materials.

5. Experimental Results

In this section, we evaluate our algorithm's performance on synthetic as well as real-world tasks, including Multi-MNIST and its variants [31, 42], CelebA [23] and CIFAR100 [21]. We compare with the following algorithms:

- **STL**: single task learning where tasks are trained once at a time;
- **t-MOOMTL**: only learning the target tasks with MOOMTL;
- **MOOMTL**: finding one Pareto optimal solution for multi-objective optimization problem [33];
- **GradNorm**: using the normalization proposed by [6];
- **Uncertainty**: using the uncertainty weighting [10];
- **Uniform Scaling**: linear scalarization of tasks with equal weights;
- **Pareto-MTL** : decomposing the multi-objective optimization problem into a set of constrained subproblems with different trade-off preferences [22].

5.1. Synthetic Data

We analyze our model with synthetic data from [22]. There are two non-convex objectives to be minimized, shown in Eq. (17), where $x \in \mathbb{R}^n$. Our algorithm can generate the subset of Pareto fronts with different preference towards the target task l_1 , as shown in Figure 3. Pareto-MTL [22] can also generate distributed solutions on the Pareto front. MOOMTL [33] fails to extend the solution set towards any desired region, whereas linear scalarization (LinScalar) only finds the extreme solutions.

$$\begin{aligned} l_1(x) &= 1 - e^{-\|x - \frac{1}{\sqrt{n}}\|_2^2}, \\ l_2(x) &= 1 - e^{-\|x + \frac{1}{\sqrt{n}}\|_2^2}. \end{aligned} \tag{17}$$

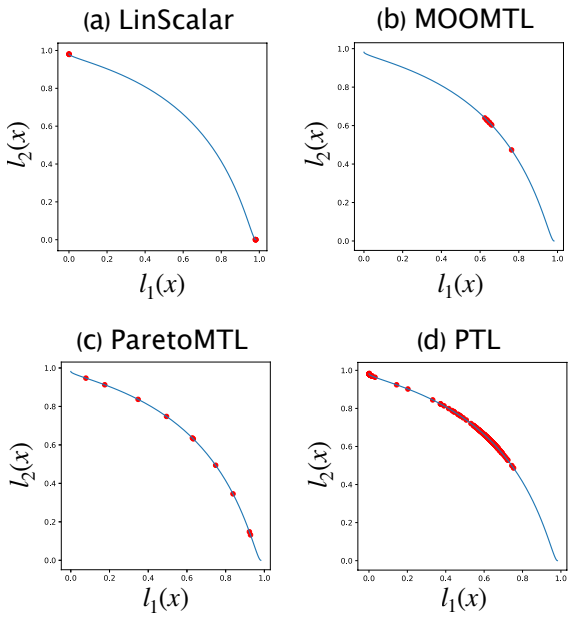


Figure 3: **Synthetic dataset performance.** (a). The obtained solutions of linear scalarization of the target and privileged tasks. (b). The obtained solutions of MOOMTL. (c). The obtained solutions of a Pareto-MTL. (d). The obtained solution from PTL in present work. The x-axis is the loss for the target task and the y-axis is the loss for the privileged task. The proposed PTL successfully generates a set of widely distributed Pareto solutions that can sacrifice the privileged task based on user preference. Details of the synthetic example can be found in section 5.

5.2. Multi-MNIST and Multi-Fashion

Dataset and Task Description In the Multi-MNIST dataset, each image has two digits. In the Multi-Fashion dataset, each image has two fashion icons. In the Multi-MNISTFashion dataset, each image has one digit on the left and one fashion icon on the right. We followed [31] to generate the three datasets. There are two tasks: 1) classifying the top- left image, and 2) classifying the bottom-right image. We use the first task as target task and the other as the privileged one. Each dataset has 60,000 training images and 10,000 test images. The objectives are the cross entropy loss.

Network Architecture The backbone network is a modified LeNet [42]. Our network starts from two convolutional layers with a 5×5 kernel and a stride of 1 pixel. The two layers have 10 and 20 channels respectively. A fully connected layer of 50 channels appends the convolutional layers, which is then followed by two 10-channel fully connected layers, one for each task. We add a 2×2 max pooling layer right after each convolutional layer and use ReLU as the nonlinear function. The performance of the target task

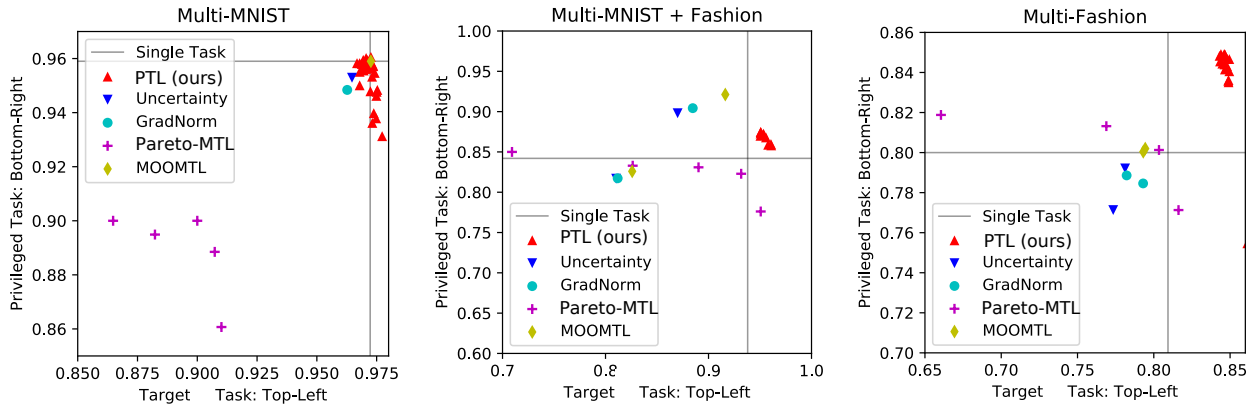


Figure 4: **Results on Multi-MNIST, Multi-Fashion and Multi-FashionMNIST.** The x-axis is the accuracy of the target task, and the y-axis is the accuracy of the privileged task.

are summarized in Table 1.

Table 1: Performance on the Multi-MNIST and its variants.

Method	MultiMNIST	Fashion	FashionMNIST
STL	97.23%	80.93%	93.80%
GradNorm [6]	96.27%	78.86%	90.43%
Uncertainty [10]	96.47%	79.26%	89.80%
MOOMTL [33]	97.26%	80.14%	92.26%
Pareto-MTL [22]	91.92%	82.75%	95.07%
PTL (ours)	**97.80%**	**86.71%**	**96.09%**

PTL improves the accuracy of the state-of-the-art results by 0.57%, 4% and 1% on Multi-MNIST, Multi-Fashion and Multi-MNISTFashion. As PTL can prioritize the target task, such improvements are as prediction by our theoretical analysis. Figure 4 shows the trade-offs between target and privileged tasks under current methods. PTL can generates multiple solutions under different C_1 and C_2 values specified by the user. Although Pareto-MTL [22] can also achieve multiple solutions, its overall performance fails to compete with ours. Our method also maintains the best performance for the privileged task on Multi-MNIST and Multi-Fashion, which might result from the mutually beneficial task setting.

5.3. Results of More Tasks

Now we investigate the comprehensive effectiveness of our proposed method PTL on more datasets with various task types and number of tasks. We randomly select half of tasks as target tasks and the rest as privileged tasks by default if not illustrated explicitly.

Cityscapes. Cityscapes [11] is a large dataset for road scene understanding, labelled with instance and semantic segmentation from 20 classes. The dataset consists of 2,975 training and 500 validation images. 1,525 images are withheld for testing on an online evaluation server. Our encoder is based on DeepLabV3 [4]. We use ResNet101 [18] as the base feature encoder, followed by an Atrous Spatial Pyramid Pooling (ASPP) module [4] to increase contextual awareness. And we take semantic segmentation as the target task.

CelebA. CelebA dataset [23] includes 200K face images annotated with 40 attributes. Each attribute is a binary classification task and therefore this can be modified into a 40-way MTL problem. We divide the target-privileged task sets by taking the hardest 23 tasks as the target ones, and the remaining tasks are privileged tasks. Following [33], we use ResNet-18 [18] without the final layer as a shared representation function. Since there are 40 attributes, we add 40 separate 2048 x 2 dimensional fully-connected layers as task-specific functions. The final two-dimensional output is passed through a 2-class softmax function to get binary attribute classification probabilities. We use cross-entropy as a task-specific loss.

CIFAR-100. Following [21], we we split the CIFAR-100 dataset [21] into 20 tasks, where each task is a 5-way classification problem. The shared architecture has four convolution layers that have 3x3 convolution and 32 filters, with batch normalization and one ReLU following them. There are 20 task-specific FC layers. We report the test accuracy for all 20 tasks.

PASCAL. PASCAL dataset [16] includes 20 classification labels for 11540 images. We modify the dataset into a 20-way MTL problem. The 10 target tasks are randomly selected from all the tasks, and the remaining 10 tasks are the privileged tasks. We use SENet-101 as the shared architecture [20] with a head for binary classification per task.

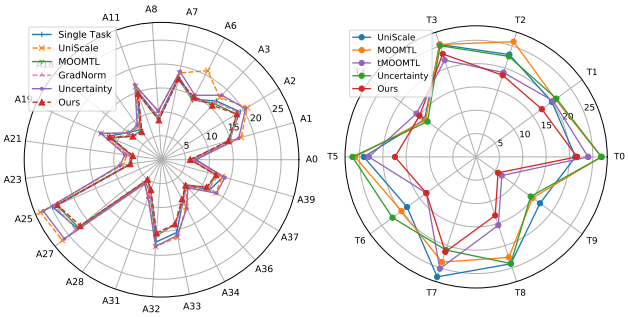
ImageNet-100. We randomly select 100 classes from the ImageNet [30] dataset to form a 100-way MTL classification problem. We randomly select 50% tasks as target tasks and the rest 50% tasks as privileged tasks. We use ResNet-50 [18] as the backbone with a head for binary clas-

Table 2: Average performance of target tasks for different task types (↑ for prediction accuracy and ↓ for error).

	Cityscapes ↑	CelebA ↓	CIFAR-100 ↓	PASCAL ↑	ImageNet-100 ↑
#tasks	2	40	20	20	100
task type	dense segmentation	binary attribute	multi-class	binary attribute	binary classification
tMOOMTL	64.35%	28.55%	17.76%	81.30%	78.31%
Uncertainty [10]	-	13.46%	20.65%	-	-
MOOMTL [33]	65.95%	13.94%	19.86%	77.13%	75.90%
PTL (ours)	**66.75%**	**11.94%**	**16.72%**	**83.56%**	**79.46%**

sification per task.

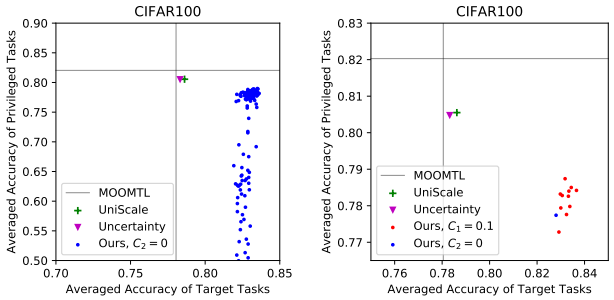
Results. As in Table 2 and Figure 5, our algorithm on average can improve the SOTA results by 1.4% for all datasets, implying the efficacy of including privileged learning in MTL. In our experiments, the number of tasks ranges from 2 to 100, suggesting the applicability of our algorithm at different scales. For CelebA dataset, we observe that tMOOMTL has extremely low performance. We infer that it is due to the fierce competition between the *hard* tasks within the target task set. But the PTL method can alleviate the competition and significantly improve the performance. In addition, our experiments include the shared-network that ranges from 4-layered convolutional network, ResNet-18, ResNet-50, ResNet-101 and SENet-101. This shows that our algorithm is effective under various neural architectures and sizes.



(a) CelebA (b) CIFAR-100

Figure 5: Prediction error of individual target tasks.

Ablation Study on the Effect of Parameters C_1 and C_2 in Eq.(14). To show the effect of parameters C_1 and C_2 in Eq.(14), we perform ablation study with CIFAR100 dataset. The 10 targeted tasks are randomly selected from the 20 5-way classification problems, and the remaining 10 tasks are used as the privileged tasks. For investigation of C_1, we set C_2 as 0, i.e. no direction correction against the privileged tasks. During the investigation of C_2, C_1 is set as 0.1, which highlights the impact from direction correction. We report the average error of the targeted tasks and the privileged tasks. The results are shown in Figure 6. When C_1 increases from 0 to 1, the performance peaks at around $C_1 = 0.17$ with error as 16.52%. When C_2 increases from



(a) Effect of C_1. (b) Effect of C_2.

Figure 6: Ablation study on CIFAR100 dataset.

$1e-4$ to 1, the performance peaks around $C_2 = 0.82$ with error as 16.18%. Altogether, it shows that appropriate amount of slack descent for the privileged tasks helps the performance of the targeted tasks, and that additional direction correction can further improve learning of the targeted tasks.

6. Conclusion

We present the *learning with privileged task* framework that generalizes the MOO-MTL algorithm, which is adaptable to the user's preference for the certain tasks. The model consists of slack descent of the privileged tasks and the direction correction towards the target tasks. We strictly prove the effectiveness of PTL using the KKT condition and provides illustrative analysis of the model. The novel hybrid block coordinate descent method can solve the dual problem efficiently. PTL can achieve state-of-the-art performance on both synthetic and real-world datasets. Further analysis of the effect of task correlation and conflict under this model would be helpful. Theoretical improvement on the relationship between our model and the ϵ-Pareto optimality could be future direction.

Acknowledgment

This work is funded by the National Key Research and Development Program of China (No. 2018AAA0100701) and the NSFC 61876095. Shan You is supported by Beijing Postdoctoral Research Foundation.

References

[1] Jonathan Baxter. A model of inductive bias learning. *J. Artif. Int. Res.*, 12(1):149–198, Mar. 2000. 1

[2] Rich Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, 1997. 1

[3] Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. Coordinate descent method for large-scale l2-loss linear support vector machines. *Journal of Machine Learning Research*, 9(45):1369–1398, 2008. 5, 6

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. 7

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1

[6] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. volume 80 of *Proceedings of Machine Learning Research*, pages 794–803, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 1, 6, 7

[7] Chih-Chung Chang, Chih-Wei Hsu, and Chih-Jen Lin. The analysis of decomposition methods for support vector machines. *IEEE Transactions on Neural Networks*, 11(4):1003–1008, jul 2000. 5

[8] Hung-Yi Chou, Pin-Yen Lin, and Chih-Jen Lin. Dual coordinate-descent methods for linear one-class SVM and SVDD. In *Proceedings of SIAM International Conference on Data Mining (SDM)*, 2020. 5, 6

[9] S. Chowdhuri, T. Pankaj, and K. Zipser. Multinet: Multi-modal multi-task learning for autonomous driving. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1496–1504, 2019. 1

[10] R. Cipolla, Y. Gal, and A. Kendall. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018. 1, 6, 7, 8

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2016. 7

[12] Jean Antoine Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, mar 2012. 3

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[14] Shangchen Du, Shan You, Xiaojie Li, Jianlong Wu, Fei Wang, Chen Qian, and Changshui Zhang. Agree to disagree: Adaptive ensemble knowledge distillation in gradient space. *Advances in Neural Information Processing Systems*, 33, 2020. 3

[15] Yunshu Du, Wojciech M. Czarnecki, Siddhant M. Jayakumar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity, 2018. 2

[16] Mark Everingham, Luc Van Gool, Christopher K I Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 7

[17] Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000. 3

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 7

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1

[20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *CoRR*, abs/1709.01507, 2017. 7

[21] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 6, 7

[22] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 12060–12070. Curran Associates, Inc., 2019. 2, 3, 6, 7

[23] Z Liu, P Luo, X Wang, and X Tang. Deep Learning Face Attributes in the Wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015. 6, 7

[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 1

[25] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *4th International Conference on Learning Representations, ICLR 2016*, 2016. 2

[26] Pingchuan Ma, Tao Du, and W. Matusik. Efficient continuous pareto exploration in multi-task learning. *ArXiv*, abs/2006.16434, 2020. 2

[27] Debabrata Mahapatra and Vaibhav Rajan. Multi-Task Learning with User Preferences: Gradient Descent with Controlled Ascent in Pareto Optimization. 2020. 2

[28] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3994–4003, 2016. 1

[29] Taylor Mordan, Nicolas THOME, Gilles Henaff, and Matthieu Cord. Revisiting multi-task learning with rock: a deep residual auxiliary block for visual detection. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 1310–1322. Curran Associates, Inc., 2018. 2

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. 7

[31] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3859–3869, Red Hook, NY, USA, 2017. Curran Associates Inc. 6

[32] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 1

[33] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, pages 525–536, 2018. 1, 2, 3, 6, 7, 8

[34] Xiu Su, Shan You, Jiyang Xie, Mingkai Zheng, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Vision transformer architecture search. *arXiv preprint arXiv:2106.13700*, 2021. 1

[35] Xiu Su, Shan You, Mingkai Zheng, Fei Wang, Chen Qian, Changshui Zhang, and Chang Xu. K-shot nas: Learnable weight-sharing for nas with k-shot supernets. *arXiv preprint arXiv:2106.06442*, 2021. 1

[36] Fengyi Tang, Cao Xiao, Fei Wang, Jiayu Zhou, and Li-wei H Lehman. Retaining privileged information for multi-task learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1369–1377, 2019. 2

[37] Partoo Vafaeikia, Khashayar Namdar, and Farzad Khalvati. A brief review of deep multi-task learning and auxiliary task learning, 2020. 2

[38] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey, 2020. 1

[39] Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: similarity control and knowledge transfer. *J. Mach. Learn. Res.*, 16(1):2023–2049, 2015. 2

[40] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009. 2

[41] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. Adversarial distillation for learning with privileged provisions. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 2

[42] H. Xiao, K. Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv*, abs/1708.07747, 2017. 6

[43] D. Xu, W. Ouyang, X. Wang, and N. Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018. 2

[44] Erkun Yang, Cheng Deng, Wei Liu, Xianglong Liu, Dacheng Tao, and Xinbo Gao. Pairwise relationship guided deep hashing for cross-modal retrieval. In *proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. 1

[45] Jonghwa Yim and Sang Hwan Kim. Learning boost by exploiting the auxiliary task in multi-task domain, 2020. 2

[46] Shan You, Tao Huang, Mingmin Yang, Fei Wang, Chen Qian, and Changshui Zhang. Greedynas: Towards fast one-shot nas with greedy supernet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1999–2008, 2020. 1

[47] Shan You, Chang Xu, Yunhe Wang, Chao Xu, and Dacheng Tao. Privileged multi-label learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3336–3342, 2017. 2

[48] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1285–1294, 2017. 1

[49] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning with single-teacher multi-student. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[50] Yu Zhang and Qiang Yang. A survey on multi-task learning, 2018. 1

[51] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 94–108, Cham, 2014. Springer International Publishing. 1

[52] Mingkai Zheng, Shan You, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Ressl: Relational self-supervised learning with weak augmentation. *arXiv preprint arXiv:2107.09282*, 2021. 1