This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Achievement-based Training Progress Balancing for Multi-Task Learning

Hayoung Yun and Hanjoo Cho*
Samsung Research
{hayoung.yun, hanjoo.cho}@samsung.com

Abstract

Multi-task learning faces two challenging issues: (1) the high cost of annotating labels for all tasks and (2) balancing the training progress of various tasks with different natures. To resolve the label annotation issue, we construct a large-scale "partially annotated" multi-task dataset by combining task-specific datasets. However, the numbers of annotations for individual tasks are imbalanced, which may escalate an imbalance in training progress. To balance the training progress, we propose an achievement-based multi-task loss to modulate training speed based on the "achievement," defined as the ratio of current accuracy to single-task accuracy. Then, we formulate the multi-task loss as a weighted geometric mean of individual task losses instead of a weighted sum to prevent any task from dominating the loss. In experiments, we evaluated the accuracy and training speed of the proposed multi-task loss on the large-scale multi-task dataset against recent multi-task losses. The proposed loss achieved the best multi-task accuracy without incurring training time overhead. Compared to single-task models, the proposed one achieved 1.28%, 1.65%, and 1.18% accuracy improvement in object detection, semantic segmentation, and depth estimation, respectively, while reducing computations to 33.73%. Source code is available at `https://github.com/samsung/Achievement-based-MTL`.

1. Introduction

Cooperation of various vision tasks is often required for high-level vision applications for autonomous driving and surveillance cameras [6, 16, 41, 8, 12, 38]. The vision task models typically consist of two parts: a feature extractor and a prediction head, and most computations are concentrated on the feature extractor. Hence, sharing the feature extractor among different tasks, multi-task learning, can significantly expedite inference and enhance the feature extractor to produce more general representations [25, 18, 3, 12, 38]. However, multi-task learning faces two major challenges: balancing the training progress of various tasks with different natures and the cost of annotating the labels of all tasks for plenty of images.

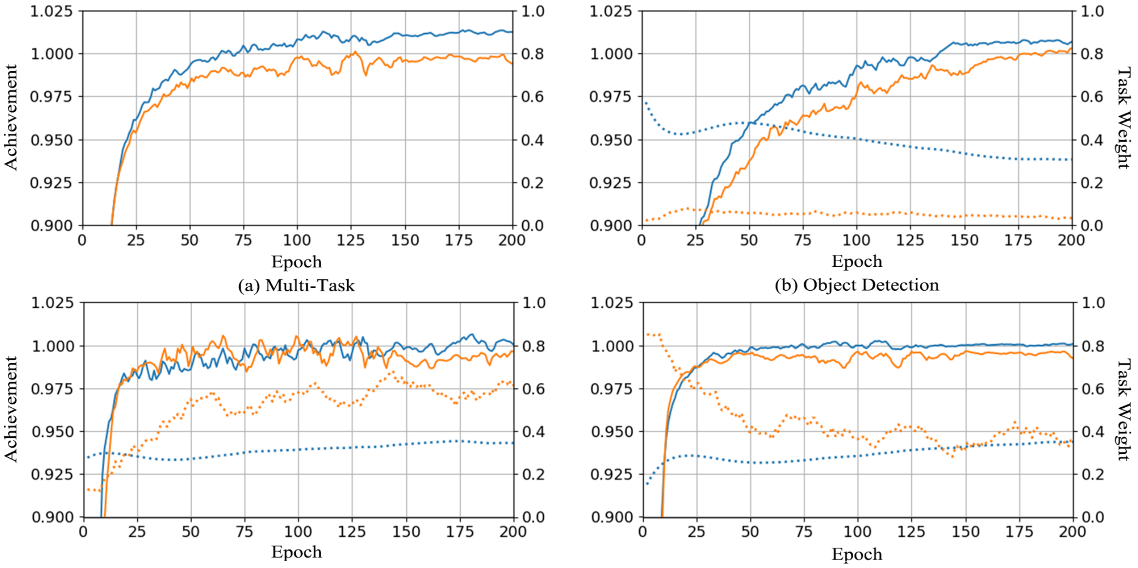
There are two major approaches to balancing the training progress: loss scale-based [25, 5, 20] and gradients-based [4, 31, 40, 24]. Primitive multi-task losses [25, 5] address the difference in loss scale among individual tasks due to their distinct loss functions (e.g., cross entropy for classification and L1 loss for regression). However, simply matching the loss scales is insufficient to balance the gradients because the derivatives of distinct functions can differ.

Recent multi-task losses have directly adjusted back-propagated gradients [4, 31, 40, 24, 23]. The gradient-based methods seek to equalize the task gradients at the last shared layer [4, 24]. However, achieving balance in task gradients does not guarantee balance in the training progress because the difficulty of tasks may differ. Easy tasks quickly converge, while difficult ones are trained slowly [13]. Hence, it is insufficient to consider only gradients to balance the training progress, but task difficulty should also be regarded.

Annotating labels for all tasks on plenty of images is expensive and time-consuming. Thus, multi-task datasets [33, 7, 10] suffer from a lack of annotations, while task-specific datasets have become larger and larger [29, 22, 32]. Some previous works [18, 38] construct a union dataset composed of task-specific datasets to resolve this issue. Images of the union dataset are partially annotated. Because task losses are only produced for existing labels, multi-task models can be easily biased toward the dominant task if the numbers of labels for individual tasks differ significantly. Moreover, the gradient of each task is also heavily influenced by the number of task labels presented in a batch, and thus gradient-based multi-task losses are significantly disturbed on a partially annotated dataset.

In this paper, we propose a novel multi-task loss that can balance the training progress of different tasks effectively, without using task gradients. The proposed loss controls the training progress based on accuracy achievement, defined as the ratio of current accuracy to single-task accuracy.

*Corresponding author



(a) Multi-Task

(b) Object Detection

(c) Semantic Segmentation

(d) Depth Estimation

Figure 1. Achievement and task weight curves for (a) multi-task, (b) object detection, (c) semantic segmentation, and (d) depth estimation on the PASCAL VOC [9] + NYU v2 [33] dataset. The blue and orange lines are the proposed method and IMTL-G [24], respectively. The solid lines mean achievements (left y-axis), and the dotted lines denote task weights (right y-axis). Delivering the same amount of task gradients to the shared feature extractor, IMTL-G learned easy tasks (segmentation and depth) quickly while the difficult one (detection) suffered from under-fitting. The proposed method much focused on the challenging task (detection) having lower achievement than others, and as a result, demonstrated better multi-task accuracy than IMTL-G.

Furthermore, refraining from using a general weighted sum based loss, the proposed loss is composed of a weighted geometric mean to exploit its scale-invariant property.

The main contributions of this paper are as follows:

1. We propose an achievement-based multi-task loss that employs a weighted geometric mean in multi-task learning. The proposed loss effectively balances the training progress and prevents any task from dominating the loss. Moreover, the proposed weights and weighted geometric mean also dramatically improve the accuracy of other multi-task losses, respectively.

2. We conduct a robust evaluation for multi-task losses on a large-scale partially annotated multi-task dataset.

3. We empirically validate that multi-task learning on a partially annotated dataset can achieve better accuracy than filling in absent labels using single-task models.

2. Related Works

2.1. Multi-Task Loss

Recent research on multi-task learning has focused on developing effective multi-task losses to train all tasks in balance and improve the accuracy of each task as much as possible. Most multi-task losses are generally represented as the weighted sum of task losses as follows:

$$L_{total} = \sum_{t=1}^{N_T} w_t L_t \ , \tag{1}$$

where w_t and L_t mean the task weight and task loss of t-th task, N_T is the number of tasks, and L_{total} denotes the total multi-task loss. The task weight directly affects the accuracy of the corresponding task [17]. Hence, finding optimal weights is crucial for achieving good accuracy, but manual tuning of task weights is prohibitively expensive. Thus, extensive research has been conducted to determine the task weights automatically.

The first approach is the learning-based method [17], which defines task weights as learnable parameters based on the task-agnostic homoscedastic uncertainty of task losses. This method can be easily applied by simply adding the learnable parameters and regularization. However, it is only applicable if the uncertainty of output distribution can be derived to the task loss [24].

Loss scale-based methods [25, 20, 5] address the scale difference in task losses. RLW [20] chooses random task weights, while DWA [25] modulates task weights to decrease task losses evenly. Simply defining multi-task losses as the geometric mean of task losses, GLS [5] effectively addresses the scale variance. However, matching loss scales does not guarantee balance in task gradients because the derivatives of different functions are distinct, even if their scales are similar.

Gradient-based methods adjust task weights to control task gradients directly [31, 40, 23, 4, 24]. MGDA [31] employs an iterative optimization process to find the task weights so that the gradient vector of shared parameters is aligned toward the minimum norm points of the convex hull. PCGrad [40] and CAGrad [23] directly modulate task gradients, without using task weights, to avoid conflicts in their directions, which prevents destructive interference. In contrast, GradNorm [4] and IMTL [24] focus on task gradients at the last shared layer. While GradNorm controls task weights to make the magnitudes of the task gradients close to each other, IMTL [24] adjusts the task weights so that the gradient has an identical length when projected onto each task gradient. However, all gradient-based methods are sensitive to popular regularization modules that drop out units or layers during training, such as dropout or stochastic depth [15]. In addition, selecting shared and task-specific parameters incorrectly can significantly degrade accuracy. Furthermore, even if the same amount of gradient is delivered for all tasks, the training speed may vary depending on the difficulty of the tasks.

DTP [13], an accuracy-based method, introduces task difficulty to multi-task learning. DTP estimates task difficulty based on current task accuracy. Regarding tasks with low accuracy as difficult, DTP increases task weights of ones with low accuracy to expedite their training and vice versa. However, estimating training progress based on current accuracy alone is insufficient. If easy and difficult tasks have the same accuracy, DTP assumes their training progress is the same, regardless of how much task accuracy can be improved further. Moreover, DTP does not address the imbalance scales of individual task losses.

In this paper, we rediscover and improve GLS [5] and DTP [13], which were developed earlier, but have not received much attention. Based on DTP using current accuracy alone, we elaborate accuracy-based task weights by introducing the ”achievement,” defined as the ratio of current and single-task accuracy. Employing the proposed achievement-based task weight, we propose a novel multi-task loss that consists of a weighted geometric mean of individual task losses to effectively address the training imbalance caused by different derivatives and scales of distinct loss functions.

2.2. Annotating Multi-Task Labels

The biggest challenge of multi-task learning for practical usages is data collection and annotation [11]. Especially, the effort to annotate labels for all tasks is linearly proportional to the number of tasks. Thus, representative multi-task datasets [7, 33, 10] are order-of-magnitude smaller than conventional single-task datasets, and most research on multi-task learning [4, 31, 24, 25, 40] has used these small datasets for training and evaluation.

UberNet [18] attempts to construct a union dataset by combining different single-task datasets. To handle the issue of imbalanced data sizes across tasks, it also proposes a training method that delays parameter updates until sufficient data is accumulated. However, it focuses on the task-specific part of multi-task models, so the label imbalance issue still exists for the shared one.

MuST [11] applies self-training [39] to relieve the efforts for annotating multi-task labels, which constructing a fully-annotated multi-task dataset from partially annotated images by creating pseudo labels for label-absent tasks using pre-trained single-task teachers.

KD-MTL [19] adopts pre-trained single-task teachers in the training phase. Balancing the training progress of the tasks with different difficulties, KD-MTL trains a multi-task model to generate shared features similar to what the task-specific teachers produce.

In this paper, we empirically validate that multi-task learning on a partially annotated dataset can provide superior multi-task accuracy than methods leveraging single-task models by learning general representation for multiple tasks. Moreover, we also provide robust accuracy comparisons for various multi-task losses on a large-scale partially annotated dataset.

3. Achievement-based Multi-Task Loss

The proposed multi-task loss is inspired by focal loss [21], which was introduced to resolve the class imbalance in object detection. Generally, numerous background samples are in images, while foreground objects are only a few. As a result, most detection losses are from the easily-detected background, even though hard-to-detect foreground objects are critical. To focus on objects, focal loss modulates cross-entropy with focal weighting term, $(1 - p_c)^\gamma$:

$$FL(p_c;\gamma) = (1 - p_c)^\gamma CE = -(1 - p_c)^\gamma \log(p_c) \ , \quad (2)$$

where γ means the focusing factor and p_c denotes the probability that the prediction is correct (i.e., p_c is p for foreground samples and $1 - p$ for background). Through focal weighting, focal loss diminishes the contribution of easy samples while enhancing the influence of difficult ones.

We introduce focal weighting, $(1 - p_c)^\gamma$, as task weights for multi-task learning to address the imbalance of training progress across tasks. We define the achievement of each task as the ratio of current and single-task accuracy, and use it instead of p_c as follows:

$$w_t\left(Acc_t;\gamma\right) = \left(1 - Acc_t/p_t\right)^\gamma \ , \quad (3)$$

where Acc_t denotes current accuracy of task t, and p_t means task potential, defined as single-task accuracy. Like the focal loss, the achievement-based task weight encourages tasks with low achievement to expedite their training while slowing down the early converged ones.

Learning multiple tasks can enhance feature extractors to learn more general representations than single-task learning. Hence, a multi-task model often outperforms its single-task counterparts. During training, the achievement-based task weights decrease as the task accuracy of the multi-task model approaches the single-task accuracy. However, the task weights can unintentionally increase if the accuracy of the multi-task model surpasses the single-task ones. To prevent the unintended increase, we introduce a slight margin, $m > 1$, to the potential:

$$w_t = \left(1 - \frac{\overline{Acc_t}}{m \cdot p_t}\right)^{\gamma}. \tag{4}$$

As the accuracy of a task improves during training, its task weight is decreased. Decreasing task weights is theoretically identical to reducing the learning rate, inducing the under-fitting of the corresponding task. To avoid under-fitting, we normalize the task weights using softmax.

Finally, to resolve the scale imbalance of individual task losses, the proposed achievement-based multi-task loss employs the weighted geometric mean instead of the conventional weighted sum as follows:

$$L_{total} = \prod_{t=1}^{N_T} L_t^{w_t}. \tag{5}$$

4. Experimental Results

4.1. Experimental Setup

4.1.1 Preprocessing

We applied both geometric and photometric augmentations to improve accuracy. We conducted random scaling, resize, and random horizontal flip as geometric augmentation, and then performed SSD's photometric distortions [26] and random adjust sharpness as photometric augmentation.

4.1.2 Evaluation Metrics

A popular metric for multi-task accuracy is the average per-task accuracy drop [36]:

$$\Delta_{MTL} = \frac{1}{N_T}\sum_{t=1}^{N_T} S_t \frac{M_{m,t} - M_{b,t}}{M_{b,t}}, \tag{6}$$

where m and b denote the multi-task model and single-task baseline, respectively. M_t means the accuracy metric for task t. S_t is 1 if M_t is higher is better, otherwise -1. We slightly modified this metric for multiple metrics for a task:

$$\Delta_{MTL} = \frac{1}{N_T}\sum_{t=1}^{N_T} \frac{1}{N_t}\sum_{i=1}^{N_t} S_{t,i} \frac{M_{m.t,i} - M_{b,t,i}}{M_{b,t,i}}, \tag{7}$$

where N_t denotes the number of metrics for task t.

Depending on the single-task baseline, the average per-task accuracy drop influenced by the accuracy of single-task models. Hence, we propose a new multi-task accuracy metric, independent of the quality of single-task models, based on the geometric mean:

$$Acc_{MTL} = \prod_{t=1}^{N_T}\prod_{i=1}^{N_t} \sqrt[N_T N_t]{M_{m,t}^{S_{t,i}}}. \tag{8}$$

For example, the multi-task accuracy metric for segmentation, depth estimation, and surface normal is as follows:

$$Acc_{MTL} = \sqrt[3]{mIoU \cdot \sqrt{\frac{\delta_1}{rmse}} \cdot \sqrt[3]{\frac{11.25}{mean \cdot median}}}. \tag{9}$$

4.2. Comparison to Recent Multi-Task Losses

In this subsection, the accuracy and training speed of the proposed multi-task loss was compared to recent multi-task losses. As multi-task baseline, we simply added all task losses (uniform task weight). Then, as benchmark methods, we used loss-scale based method (RLW [20], DWA [25] and GLS [5]), gradient-based methods (MGDA [31], PCGrad [40], CAGrad [23], GradNorm [4], IMTL-G, and IMTL [24]), and an accuracy-based method (DTP [13]). All benchmark methods were implemented on the same code base for a fair comparison; the same augmentation, optimizer, search range of learning rates, and LR scheduler were applied to all benchmark and proposed methods. No manual scalers were used for all task losses.

4.2.1 Comparison on the NYU v2 Dataset

We evaluated the performance of the proposed and benchmark multi-task losses on the NYU v2 dataset for semantic segmentation, depth estimation, and surface normal. We used DeepLabV3 [2] as the baseline architecture and ResNet50 [14] as a feature extractor. The single-task and multi-task models were trained 10 times for learning rates of 8e-4, 4e-4, 2e-4, 1e-4, and 8e-5. The representative metric values were obtained by averaging the results of all trials, excluding the maximum and minimum of Acc_{MTL} (average of 8 trials). The metric values for the learning rate with the best average Acc_{MTL} were presented in Table 1. More training details are available in Appendix A.

Although multi-task models learned general features, none of the multi-task losses surpassed the single-task baseline because surface normal requires significantly different features, causing conflicts in training. Learning segmentation and depth estimation results improved accuracy than the single-task baseline as presented in Appendix B.

		segmentation	depth estimation		surface normal			total		
methods		$mIoU \uparrow$	$\delta_1 \uparrow$	$rmse \downarrow$	$mean \downarrow$	$median \downarrow$	$11.25 \uparrow$	$Acc_{MTL} \uparrow$	$\Delta_{MTL} \uparrow$	$time$
Single-Task		0.4437	0.8087	0.5814	19.3462	13.2045	0.4553	0.3989	0.00%	-
Constant	Uniform	0.4446 (0.20%)	0.8091 (0.05%)	0.5776 (0.66%)	22.8531 (-18.13%)	17.7271 (-34.25%)	0.3322 (-27.05%)	0.3666 (-8.10%)	-8.64% -	31.07 -
Scale -based	RLW [20]	0.4447 (0.23%)	0.8082 (-0.06%)	0.5759 (0.94%)	22.8410 (-18.06%)	17.6180 (-33.42%)	0.3350 (-26.43%)	0.3673 (-7.91%)	-8.43% -	30.78 -
	DWA [25]	**0.4465** **(0.62%)**	0.8093 (0.07%)	0.5751 (1.08%)	22.7934 (-17.82%)	17.6902 (-33.97%)	0.3330 (-26.85%)	0.3676 (-7.82%)	-8.34% -	30.84 -
	GLS [5]	0.4321 (-2.61%)	**0.8221** **(1.65%)**	**0.5665** **(2.56%)**	20.7032 (-7.01%)	15.0512 (-13.99%)	0.3982 (-12.54%)	0.3837 (-3.80%)	-3.90% -	30.07 -
Gradient -based	MGDA [31]	0.2511 (-43.41%)	0.7636 (-5.58%)	0.6266 (-7.77%)	**19.2796** **(0.34%)**	**13.1962** **(0.06%)**	**0.4553** **(-0.01%)**	0.3229 (-19.05%)	-16.65% -	76.23 -
	PCGrad [40]	0.4435 (-0.06%)	0.8017 (-0.87%)	0.5825 (-0.19%)	24.2444 (-25.32%)	19.3005 (-46.17%)	0.3038 (-33.27%)	0.3558 (-10.79%)	-11.83% -	58.06 -
	CAGrad [23]	0.4448 (0.24%)	0.8001 (-1.07%)	0.5854 (-0.68%)	24.2759 (-25.48%)	19.3395 (-46.46%)	0.3033 (-33.38%)	0.3556 (-10.85%)	-11.91% -	59.08 -
	GradNorm [4]	0.4458 (0.46%)	0.7888 (-2.46%)	0.5928 (-1.96%)	22.3488 (-15.52%)	16.9259 (-28.18%)	0.3524 (-22.60%)	0.3690 (-7.50%)	-7.95% -	35.56 -
	IMTL-G [24]	0.4361 (-1.72%)	0.8021 (-0.82%)	0.5788 (0.45%)	20.5248 (-6.09%)	14.6814 (-11.19%)	0.4097 (-10.01%)	0.3846 (-3.58%)	-3.67% -	35.84 -
	IMTL [24]	0.4162 (-6.20%)	0.7876 (-2.61%)	0.5930 (-2.00%)	20.8134 (-7.58%)	14.8333 (-12.34%)	0.4064 (-10.74%)	0.3746 (-6.08%)	-6.24% -	58.98 -
Accuracy -based	DTP [13]	0.4458 (0.46%)	0.7648 (-5.42%)	0.6140 (-5.61%)	22.2556 (-15.04%)	16.7507 (-26.86%)	0.3568 (-21.63%)	0.3660 (-8.23%)	-8.74% -	31.07 -
	AMTL (proposed)	0.4377 (-1.35%)	0.8205 (1.46%)	0.5667 (2.54%)	20.7974 (-7.50%)	15.1003 (-14.36%)	0.3969 (-12.82%)	**0.3847** **(-3.54%)**	**-3.64%** -	31.05 -

Table 1. Comparison to recent multi-task losses on the NYU v2 dataset. $mIoU$, δ_1, 11.25, Acc_{MTL}, and Δ_{MTL} are better when higher while $rmse$, $mean$, and $median$ are better when lower. $time$ denotes the average training time for epoch in seconds. The best and runner-up results for each metric are highlighted by **bold** and underline, respectively.

RLW [20], randomly choosing task weights, showed similar accuracy to the multi-task baseline. Modulating task weights to decrease task losses evenly, DWA [25] provided slightly better multi-task accuracy than the baseline. GLS [5], based on a scale-invariant geometric mean, demonstrated the best accuracy among scale-based methods even though it did not use task weights. All scale-based losses did not incur additional training time.

There are three types of gradient-based multi-task losses: using optimization (MGDA [31]), resolving gradient conflict (PCGrad [40] and CAGrad [23]), and modulating the task gradients on the last shared layer (GradNorm [4], IMTL-G, and IMTL [24]). When task gradients conflict, MGDA [31] excessively enhanced the task of which gradient magnitude was the least. As a result, while MGDA provided the best accuracy in surface normal, its multi-task accuracy suffered seriously. PCGrad [40] and CAGrad [23] directly modified task gradients to resolve gradient conflict, easily affected by dropout or stochastic depth. Thus, they did not improve accuracy compared to the multi-task baseline. We discussed the influence of droupout to gradient-based losses in Appendix B.5. GradNorm [4] controlled task weights so that all task losses evenly decreased, which promoted multi-task accuracy compared to the baseline. IMTL-G [24] achieved the best multi-task accuracy among gradient-based methods by adjusting task weights so that the gradient at the last shared layer has the same length when projected to each gradients.

All gradient-based losses incurred overhead in training time for computing task gradients. GradNorm and IMTL-G have the smallest overhead because using task gradients only for determining task weights. Resolving the conflict (PCGrad and CAGrad) and additional back-propagation for task-specific parameters (IMTL) further increased training time. The iterative optimization process of MGDA induced the most significant overhead in training time.

The proposed multi-task loss effectively addressed scale difference in task losses by employing a weighted geometric mean that is invariant to loss scale. The proposed one also balanced training progress of various tasks by modulating the achievement-based task weights. As a result, it achieved the best multi-task accuracy without impeding training.

		Shared ASPP						Individual ASPP					
		DeepLabV3 (197 GMAC)			DeepLabV3+ (207 GMAC)			DeepLabV3 (334 GMAC)			DeepLabV3+ (343 GMAC)		
Methods		Acc_{MTL}	Δ_{MTL}	$time$	Acc_{MTL}	Δ_{MTL}	$time$	Acc_{MTL}	Δ_{MTL}	$time$	Acc_{MTL}	Δ_{MTL}	$time$
Single-Task		0.3989	-	-	0.3986	-	-	0.3989	-	-	0.3986	-	-
Constant	Uniform	0.3666	-8.64%	30.24	0.3683	-8.07%	30.98	0.3763	-6.00%	43.31	0.3762	-5.96%	45.61
Scale -based	RLW [20]	0.3673	-8.43%	30.50	0.3665	-8.55%	31.55	0.3774	-5.71%	43.22	0.3764	-5.88%	43.37
	DWA [25]	0.3676	-8.34%	30.65	0.3677	-8.25%	30.77	0.3768	-5.87%	43.01	0.3761	-5.97%	43.83
	GLS [5]	0.3784	-5.33%	31.08	0.3761	-5.83%	31.27	0.3786	-5.25%	44.16	0.3798	-4.80%	43.70
Gradient -based	MGDA [31]	0.3229	-16.65%	74.99	0.3394	-13.41%	76.78	0.3770	-5.34%	88.97	0.3793	-4.75%	90.15
	PCGrad [40]	0.3558	-11.83%	58.05	0.3581	-11.04%	58.02	0.3697	-7.95%	63.91	0.3710	-7.44%	64.25
	CAGrad [23]	0.3556	-11.91%	57.63	0.3584	-10.94%	58.99	0.3689	-8.15%	63.01	0.3708	-7.51%	63.75
	GradNorm [4]	0.3690	-7.95%	35.12	0.3677	-8.21%	36.19	0.3773	-5.74%	57.90	0.3760	-6.02%	59.13
	IMTL-G [24]	0.3846	-3.67%	35.22	**0.3838**	**-3.78%**	35.33	**0.3917**	**-1.79%**	57.89	**0.3910**	**-1.93%**	58.67
	IMTL [24]	0.3746	-6.24%	57.31	0.3746	-6.18%	57.85	0.3815	-4.43%	99.27	0.3807	-4.57%	103.50
Accuracy -based	DTP [13]	0.3660	-8.74%	29.88	0.3625	-9.63%	31.46	0.3739	-6.63%	43.43	0.3751	-6.24%	43.10
	AMTL	**0.3847**	**-3.64%**	32.08	0.3831	-3.98%	30.65	0.3899	-2.29%	44.08	0.3883	-2.59%	43.39

Table 2. Comparison of multi-task accuracy and training time for various DeepLab prediction heads.

	MobileNetV2 [30]		EfficientNetV2-S [34]	
Methods	Acc_{MTL}	Δ_{MTL}	Acc_{MTL}	Δ_{MTL}
single-task	0.3581	-	0.3877	-
Uniform	0.3313	-7.91%	0.3868	-0.20%
RLW [20]	0.328	-8.92%	0.383	-1.20%
DWA [25]	0.3311	-7.94%	0.387	-0.14%
GLS [5]	0.3464	-3.30%	0.3958	2.06%
MGDA [31]	0.3109	-11.93%	0.3268	-14.08%
PCGrad [40]	0.3204	-11.44%	0.3743	-3.51%
CAGrad [23]	0.3202	-11.48%	0.3742	-3.52%
GradNorm [4]	0.3346	-6.87%	0.3873	-0.06%
IMTL-G [24]	**0.3513**	**-1.88%**	**0.3991**	**2.90%**
IMTL [24]	0.3445	-3.82%	0.3936	1.54%
DTP [13]	0.3289	-8.58%	0.3837	-0.97%
AMTL	0.3476	-2.95%	0.3989	2.85%

Table 3. Comparison of multi-task accuracy for MobileNetV2 and EfficientNetV2-S backbones.

Robustness We evaluated the robustness of the proposed method for various prediction heads and backbones. First, we estimated the performance of the proposed and benchmark methods for various DeepLab heads (Table 2). The details of the architectures are described in appendix B. No remarkable accuracy improvement was achieved by exploiting high resolution features (DeepLabV3+) since we adopted the dilated ResNet50 [1] like MTI-Net [37]. However, it significantly escalated GMAC to use individual ASPP for each task. The proposed multi-task loss provided stable and excellent accuracy across all prediction heads.

Next, we evaluated the accuracy of the benchmark and proposed losses with other backbones: MobileNet-V2 [30] and EfficientNetV2-S [34] (Table 3). We used shared ASPP and DeepLabV3+ architecture in this comparison. The results showed similar patterns when using the ResNet50 backbone. The proposed method achieved runner-up accuracy for the MobileNetV2 and EfficientNetV2-S backbones.

	Acc_{MTL}	Δ_{MTL}
DTP [13]	0.3660	-8.74%
+ achievement-based weight	0.3745	-6.11%
+ weighted geometric mean	0.3847	-3.64%

Table 4. Ablation study for the proposed multi-task loss.

	Uniform		Achievement-based	
	Acc_{MTL}	Δ_{MTL}	Acc_{MTL}	Δ_{MTL}
PCGrad [40]	0.3558	-11.83%	0.3662	-8.73%
CAGrad [23]	0.3556	-11.91%	0.3653	-8.98%

Table 5. Comparison of multi-task accuracy for the uniform and achievement-base weights.

	Arithmetic		Geometric	
	Acc_{MTL}	Δ_{MTL}	Acc_{MTL}	Δ_{MTL}
RLW [20]	0.3673	-8.43%	0.3774	-5.59%
DWA [25]	0.3676	-8.34%	0.3811	-4.60%
DTP [13]	0.3660	-8.74%	0.3800	-4.81%

Table 6. Comparison of multi-task accuracy for the weighted arithmetic and geometric means.

methods		detection	segmentation	depth estimation		total		
		mAP@50:95 ↑	$mIoU$ ↑	δ_1 ↑	$rmse$ ↓	Acc_{MTL} ↑	Δ_{MTL} ↑	$time$
Single-Task		0.5795	0.7895	0.8882	0.4393	0.8665	-	-
Constant	Uniform	**0.5922** **(2.19%)**	0.7823 (-0.91%)	0.8731 (-1.69%)	0.4498 (-2.40%)	0.8642 (-0.26%)	-0.25% -	713.65 -
Scale -based	RLW [20]	0.5900 (1.81%)	0.7835 (-0.76%)	0.8716 (-1.87%)	0.4587 (-4.42%)	0.8605 (-0.69%)	-0.70% -	707.60 -
	DWA [25]	0.5853 (1.01%)	0.7835 (-0.75%)	0.8621 (-2.93%)	0.4565 (-3.92%)	0.8574 (-1.05%)	-1.06% -	737.70 -
	GLS [5]	0.5833 (0.65%)	0.8007 (1.42%)	**0.8917** **(0.39%)**	0.4329 (1.45%)	0.8752 (1.00%)	1.00% -	733.23 -
Gradient -based	MGDA [31]	0.4064 (-29.88%)	0.7714 (-2.29%)	0.8880 (-0.02%)	0.4453 (-1.36%)	0.7621 (-12.04%)	-10.95% -	1475.13 -
	PCGrad [40]	0.5898 (1.78%)	0.7799 (-1.22%)	0.8428 (-5.11%)	0.4829 (-9.92%)	0.8470 (-2.25%)	-2.32% -	1120.39 -
	CAGrad [23]	0.5877 (1.41%)	0.7785 (-1.39%)	0.8461 (-4.73%)	0.4781 (-8.84%)	0.8474 (-2.20%)	-2.26% -	1081.03 -
	GradNorm [4]	0.5881 (1.47%)	0.7884 (-0.13%)	0.8722 (-1.80%)	0.4462 (-1.57%)	0.8654 (-0.12%)	-0.12% -	839.57 -
	IMTL-G [24]	0.5740 (-0.95%)	**0.8080** **(2.35%)**	0.8916 (0.39%)	**0.4295** (2.24%)	0.8743 (0.90%)	0.90% -	820.15 -
	IMTL [24]	0.5891 (1.65%)	0.8005 (1.39%)	0.8908 (0.30%)	0.4392 (0.02%)	0.8757 (1.07%)	1.07% -	1271.22 -
Accuracy -based	DTP [13]	0.5853 (1.00%)	0.7666 (-2.90%)	0.8265 (-6.94%)	0.5025 (-14.38%)	0.8318 (-4.01%)	-4.19% -	705.87 -
	AMTL (proposed)	0.5870 (1.28%)	0.8025 (1.65%)	0.8903 (0.24%)	0.4300 (2.11%)	**0.8784** **(1.37%)**	**1.37%** -	707.61 -

Table 7. Comparison to recent multi-task losses on the partially annotated VOC+NYU dataset. mAP, $mIoU$, δ_1, Acc_{MTL}, and Δ_{MTL} are better when higher while $rmse$ is better when lower. $time$ denotes the average training time for epoch in seconds. The best and runner-up results for each metric are highlighted by **bold** and underline, respectively.

Effectiveness We evaluated the effectiveness of the proposed achievement-base task weights and weighted geometric mean (Table 4). Compared to DTP that not consider task potential, the achievement-based weight improved multi-task accuracy from 0.3660 to 0.3745. The weighted geometric mean further improved accuracy to 0.3847.

We also adopted the proposed weight and weighted geometric mean to compatible benchmark methods to validate their effectiveness. We applied the proposed achievement-based weight to PCGrad and CAGrad that did not use task weights. As described in Table 5, the proposed weights greatly promoted multi-task accuracy of both losses.

Furthermore, we employed a weighted geometric mean to RLW, DWA, and DTP that used task weight but not used task gradients (Table 6). The weighted geometric mean significantly improved multi-task accuracy of all of them.

4.2.2 Comparison on the VOC + NYU Dataset

In the following experiments, the multi-task accuracy and training time of the proposed and benchmark multi-task losses were evaluated on the large-scale *partially annotated* multi-task dataset that consists of task-specific datasets (PASCAL VOC [9] and NYU depth [33]). The dataset has abundant training images, compared to the existing fully-annotated multi-task datasets such as NYU v2 [33] (795 training images), Cityscapes [7] (2,975 training images), and KITTI [10] (200 training images for multi-task). Its total number of training images is 39,446, which contains 15,215 (38.57%) for object detection, 10,477 (26.56%) for semantic segmentation, and 24,231 (64.43%) for depth estimation. Some images from PASCAL VOC have labels for both detection and segmentation. More details for the partially-annotated dataset were described in Appendix C.

method		$Acc_{MTL} \uparrow$	$\Delta_{MTL} \uparrow$	$time$
Single-Task		0.8665	-	-
MuST [11]	Uniform	0.8332	-3.78%	717.28
	IMTL-G [24]	0.8365	-3.46%	869.29
	AMTL	0.8431	-2.60%	728.85
NoisyStudent[39]	Uniform	0.8596	-0.68%	939.07
	IMTL-G [24]	0.8714	0.69%	1081.99
	AMTL	0.8726	0.83%	940.03
KD-MTL [19]	Uniform	0.8673	0.22%	946.63
	IMTL-G [24]	0.8736	0.94%	1055.44
	AMTL	0.8742	1.02%	940.03
w/o Teachers (partially annotated)	Uniform	0.8642	-0.25%	713.65
	IMTL-G [24]	0.8743	0.90%	820.15
	AMTL	0.8784	1.37%	707.61

Table 8. Comparison to methods leveraging single-task teachers.

In previous works using VOC [43, 44, 27], mAP@50 was used to evaluate detection accuracy. However, it is loose to catch the improvement of regression quality, achieved by recent regression losses such as cIoU and gIoU losses [28, 42]. Hence, we adopted mAP@50:95, the standard MS COCO metric, instead of mAP@50.

We used EfficientDet [35] as the baseline architecture to address object detection, and EfficientNet-V2-small [34] as a feature extractor. More description for network architecture and training details is presented in Appendix C.

The accuracy of the benchmark and proposed losses on the partially annotated dataset is presented in Table 7. When using the partially annotated dataset, the task loss was only produced for existing labels. Hence, the gradient of each task was heavily influenced by the number of labels present in each batch, which seriously degraded accuracy of methods that directly use task gradients (MGDA, PCGrad, and CAGrad). However, IMTL and IMTL-G performed well on the partially annotated dataset also because they can compensate for the absence of task labels while balancing the effective task gradients at the last shared layer. Remarkably, despite its simplicity, GLS demonstrated superior multi-task accuracy. Employing the achievement-based task weights in addition, the proposed multi-task loss further improved and achieved the best multi-task accuracy, without task gradients requiring additional computations.

Finally, to verify the effectiveness of multi-task learning on partially annotated datasets, we compared multi-task accuracy using various methods that leverage single-task models: hard pseudo labels (MuST [11]), soft-pseudo labels (NoisyStudent [39]), and knowledge distillation (KD-MTL [19]) (Table 8).

Multi-task self-training (MuST) [11] constructs a complete multi-task dataset by producing hard-pseudo labels for label-absent tasks before conducting multi-task learning. However, this method suffers from out-of-distribution and false-positive issues, and as a result, demonstrated lower accuracy when compared to the partially annotated dataset.

NoisyStudent [39] encourages a student model to learn beyond its teacher. The teacher produces predictions on images without augmentation, and then the student is trained to generate identical predictions to its teacher's on difficult images (augmented images). Using soft-pseudo labels, NoisyStudent successfully relieved accuracy degradation caused by unreliable teacher predictions. As a result, NoisyStudent greatly improved multi-task accuracy than MuST, but still lower than using the partially annotated one.

KD-MTL [19] minimizes the difference between the shared features of the multi-task model and the projected features of single-task teachers. By learning from features instead of pseudo labels, KD-MTL does not suffer from out-of-distribution or false-positive issues. However, KD-MTL showed lower multi-task accuracy than using the partially annotated dataset. As trained for multiple tasks, a multi-task model learned more general and powerful representations than its single-task counterparts. Hence, imitating single-task teachers rather hindered multi-task learning.

5. Conclusion

In this paper, we proposed a novel achievement-based multi-task loss (AMTL) to balance the training progress of various tasks with different natures. To focus on how much accuracy can be improved further, we assessed the potential of task accuracy using the single-task model in advance. Then, we estimated the training progress as the ratio of current accuracy to its potential. Furthermore, to prevent any task from dominating the loss, we formulated the proposed multi-task loss as the weighted geometric mean of task losses instead of the conventional weight sum.

In experiments, we conducted comprehensive evaluations for the proposed loss with various recent benchmark multi-task losses. We demonstrated that the proposed loss achieved excellent multi-task accuracy regardless of backbones and prediction heads. Moreover, to validate the effectiveness of the proposed achievement-base task weights and the weighted geometric mean, we applied them to compatible benchmark methods, respectively, and observed significant improvements in accuracy for each.

Further, we constructed a large-scale partially annotated multi-task dataset composed of task-specific datasets and performed an accuracy comparison. Not using task gradients, the proposed loss outperformed benchmark losses, including sophisticated gradient-based losses, on the partially annotated dataset without incurring training time overheads.

Finally, as learning more general representations, multi-task learning on the partially annotated dataset can produce higher accuracy than methods leveraging single-task teachers. We hope that experiments using such large-scale partially annotated datasets become a new experimental baseline for further multi-task learning research.

References

[1] Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark A Hasegawa-Johnson, and Thomas S Huang. Dilated recurrent neural networks. *Advances in neural information processing systems*, 30, 2017.

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[3] Po-Yi Chen, Alexander H Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2624–2632, 2019.

[4] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR, 2018.

[5] Sumanth Chennupati, Ganesh Sistu, Senthil Yogamani, and Samir A Rawashdeh. Multinet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) Workshops*, pages 0–0, 2019.

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

[7] 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2016.

[8] Keval Doshi and Yasin Yilmaz. Multi-task learning for video surveillance with limited data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3889–3899, 2022.

[9] Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2012.

[11] Golnaz Ghiasi, Barret Zoph, Ekin D Cubuk, Quoc V Le, and Tsung-Yi Lin. Multi-task self-training for learning general representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8856–8865, 2021.

[12] Kratarth Goel, Praveen Srinivasan, Sarah Tariq, and James Philbin. Quadronet: Multi-task learning for real-time semantic depth aware instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 315–324, 2021.

[13] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 270–287, 2018.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 646–661. Springer, 2016.

[16] Keishi Ishihara, Anssi Kanervisto, Jun Miura, and Ville Hautamaki. Multi-task learning with attention for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2902–2911, 2021.

[17] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[18] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6129–6138, 2017.

[19] Wei-Hong Li and Hakan Bilen. Knowledge distillation for multi-task learning. In *European Conference on Computer Vision*, pages 163–176. Springer, 2020.

[20] Baijiong Lin, YE Feiyang, Yu Zhang, and Ivor Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Transactions on Machine Learning Research*, 2022.

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2980–2988, 2017.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[23] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021.

[24] Liyang Liu, Yi Li, Zhanghui Kuang, J Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2021.

[25] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of*

the IEEE/CVF conference on computer vision and pattern recognition, pages 1871–1880, 2019.

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 21–37. Springer, 2016.

[27] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[28] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

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

[31] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[32] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019.

[33] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 746–760. Springer, 2012.

[34] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021.

[35] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[36] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[37] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning supplementary materials.

[38] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018.

[39] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[40] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

[41] Mingliang Zhai, Xuezhi Xiang, Ning Lv, and Abdulmotaleb El Saddik. Multi-task learning in autonomous driving scenarios via adaptive feature refinement networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2323–2327. IEEE, 2020.

[42] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12993–13000, 2020.

[43] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[44] Yousong Zhu, Chaoyang Zhao, Jinqiao Wang, Xu Zhao, Yi Wu, and Hanqing Lu. Couplenet: Coupling global structure with local parts for object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 4126–4134, 2017.