Split to Learn: Gradient Split for Multi-Task Human Image Analysis

Weijian Deng[1] Yumin Suh[2] Xiang Yu[2] Masoud Faraki[2]
Liang Zheng[1] Manmohan Chandraker[2,3]
[1]Australian National University [2]NEC Labs America [3]University of California, San Diego

Abstract

This paper presents an approach to train a unified deep network that simultaneously solves multiple human-related tasks. A multi-task framework is favorable for sharing information across tasks under restricted computational resources. However, tasks not only share information but may also compete for resources and conflict with each other, making the optimization of shared parameters difficult and leading to suboptimal performance. We propose a simple but effective training scheme called GradSplit that alleviates this issue by utilizing asymmetric inter-task relations. Specifically, at each convolution module, it splits features into T groups for T tasks and trains each group only using the gradient back-propagated from the task losses with which it does not have conflicts. During training, we apply GradSplit to a series of convolution modules. As a result, each module is trained to generate a set of task-specific features using the shared features from the previous module. This enables a network to use complementary information across tasks while circumventing gradient conflicts. Experimental results show that GradSplit achieves a better accuracy-efficiency trade-off than existing methods. It minimizes accuracy drop caused by task conflicts while significantly saving compute resources in terms of both FLOPs and memory at inference. We further show that GradSplit achieves higher cross-dataset accuracy compared to single-task and other multi-task networks.

1. Introduction

Comprehensive understanding of human appearances is critical for various vision applications. In recent few years, impressive progress has been made for various human-related tasks, such as person re-identification [50], pedestrian detection [7, 47], and pose estimation [1, 45]. However, most existing works focus on individual tasks only and lack the ability to jointly investigate multiple tasks. Similarly, most existing datasets are annotated for individual tasks and do not provide complete annotations for various tasks.

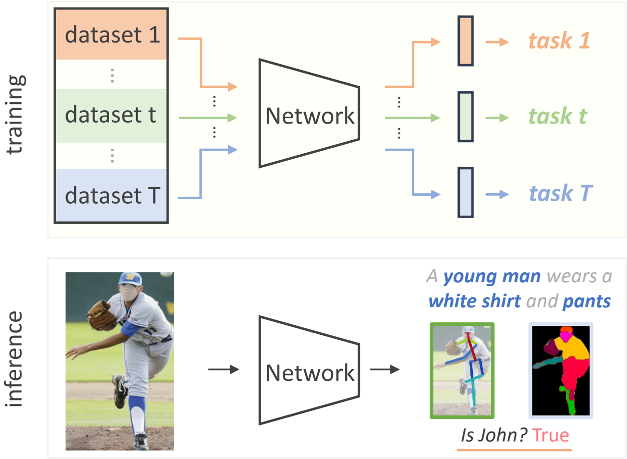

Figure 1. Traditional multi-task learning can encounter task conflicts, *e.g.*, when jointly training identity-variant (body attributes) and identity-invariant (body pose) tasks. Our goal is to train a unified network that solves multiple human-related tasks while avoiding such task conflicts. Given an image, the proposed network provides a rich explanation of the person, including attributes, pose, part masks, and identity. We assume a practical condition where the multi-task networks are trained across datasets and each dataset does not necessarily have exhaustive annotations for all tasks.

In this paper, we study a unified framework that solves multiple human-related tasks simultaneously. We consider a practical condition where each training dataset contains annotations for a single task only, as illustrated in Figure 1. The desired framework would utilize the mutual information across tasks and save the memory and computation cost via the shared network architecture. However, gradient signals useful for one task may negatively affect other tasks, causing conflicts when training a multi-task model with a shared backbone. For example, pose estimation needs pose-variant features, whereas person re-identification demands pose-invariant features. This results in training difficulty and thus leads to sub-optimal overall performance.

To address this issue, existing methods [15, 26, 30, 21, 31] integrate task-specific modules into the shared backbone to generate task-specific features. In this work, we also encourage the shared network to learn task-specific features for human-related tasks. However, instead of using additional modules, we achieve this using a simple training scheme.

For each convolution module in the shared backbone, we split its filters along the output channel into T groups for T tasks (Fig. 2). At each iteration, each group is updated by gradients only from the tasks that do not have conflict with its assigned task. Specifically, we define the asymmetric relation $t' \rightarrow t$ between tasks t and t' by comparing the validation accuracy of two models for task t. If the model trained solely with task t achieves higher accuracy than the one trained jointly with both tasks, we define the relation as "negative". When updating the t^{th} filter group, we mask gradients back-propagated from tasks that have negative relations with task t. We dub it *Gradient Split* (or *GradSplit*) as it divides gradients into groups during updates.

It is worth noting that GradSplit only applies to filters during back-propagation – the forward pass is the same as the baseline. This naturally brings three benefits. First, the task-specific filters can still use information from other tasks as it receives features produced from the other task-specific filters. In addition, there are no additional parameters or computational overhead at inference time. Lastly, it does not require comparisons of gradients from all task losses [46, 5] and thus simplifies the training procedure, especially for the case of dealing with multiple single annotation datasets. As a minor contribution, we provide a strong multi-task baseline by analyzing the normalization layers in the shared backbone. It effectively alleviates the domain gap issue when learning from multiple datasets of different domains.

We evaluate GradSplit on several combinations of tasks from four human-related tasks, *i.e.*, pose estimation, attribute recognition, person re-identification, and body part parsing. Experiments show that GradSplit minimizes accuracy drop from task conflicts while significantly saving compute resources in terms of both FLOPs and memory at inference. Compared with existing methods, GradSplit achieves a better accuracy-efficiency trade-off. Furthermore, GradSplit gains higher cross-dataset performance compared to both single-task and other multi-task networks.

2. Related Work

2.1. Human Analysis

Many task-specific datasets and solutions have been proposed for various human analysis tasks, such as pedestrian detection [7, 47], person re-identification [50, 35, 37], body pose estimation [1, 45], human body parsing [11], and body attribute recognition [22]. Several of the methods attempt to address multiple tasks together. However, most of them focus on enhancing the main task using auxiliary tasks, such as body attributes [20, 39], and human poses to enhance person re-identification by learning pose-invariant features [49, 34, 43] as well as guiding feature extraction [34, 43]. We aim at tackling every task equally important and developing a unified model that solves multiple tasks simultaneously.

2.2. Multi-Task Learning

Multi-task learning aims to train a unified model for multiple tasks [41, 17, 24]. It is desirable because it can reduce the computation resource (*e.g.*, memory and run time) by sharing modules across tasks and can utilize knowledge across tasks to learn richer representations. However, during training, gradients from different tasks may conflict with each other and cause sub-optimal solutions. We categorize the literature of multi-task learning according to each of the focus as below.

Optimization From optimization perspective, methods have been proposed to reduce the gradient conflict. Some methods compare directions of gradients and discard conflicting components to make them aligned [46, 5, 42]. Maninis *et al.* [26] enforced the task gradients to be statistically indistinguishable from each other through adversarial training. They typically require comparing gradients from the same image or domain, whereas task gradients may arise from different domains in our setting. In the above methods, each filter is updated using manipulated gradients that aggregate from all tasks, following a specific task-agnostic rule. In contrast, we explicitly assign a set of tasks to each filter and drops gradients from other tasks if they conflict to the assigned one. As a result, each filter is enforced to fit its dedicated task throughout the training, whereas dominant task of each filter can continuously vary during training in other works.

Loss Balancing Since task losses have different magnitudes and their relative magnitudes may vary during training, methods have been proposed to adjust task weights that minimize prediction uncertainty [16] and match gradient scales [4]. Other approaches formulate multi-task learning as a multi-objective optimization problem and seek for Pareto optimal solutions [32, 19, 25]. They are complementary to our method and could be potentially combined for better performance, but is out of the scope of this study.

Architecture Design Beyond the basic architecture that consists of a common feature extractor and task-specific heads, multi-stream architectures [27, 10] consisting of multiple single-task streams and interaction modules show promising performance. In contrast, Liu *et al.* [21] proposed to share a common feature extractor with task-specific attention modules. Some works study a way to find combinations of tasks that benefits each other when trained together [33, 8]. Moreover, a controllable multi-task network is proposed in [29]. The network dynamically adjusts its architecture and weights to match the desired task preference and the resource constraints. Recently, network architecture search (NAS) techniques are developed to learn how to branch [12], select task-specific paths [36], and select relevant inter-connections between task-specific networks [9].

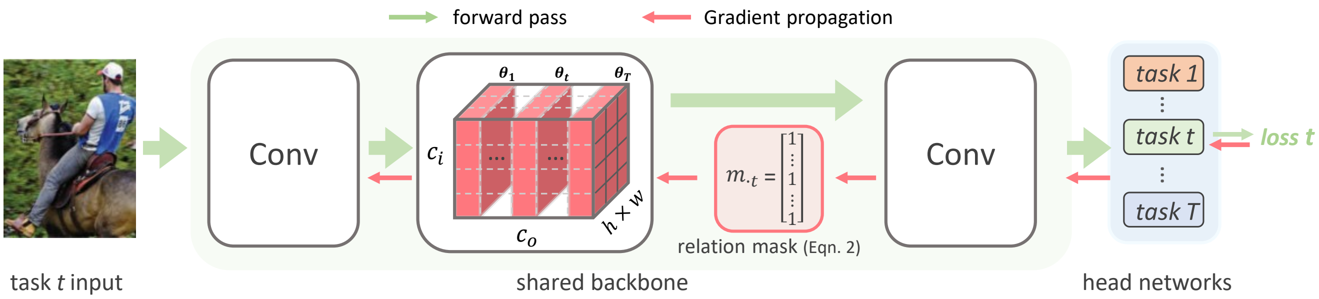

Figure 2. An overview of the proposed framework. The network consists of a shared backbone and task-specific head modules. For each convolution module in the shared backbone, we split its filters along the output channel into T groups for T tasks (*e.g.*, θ_t for t^{th} task). GradSplit manipulates the gradient of each task loss with respect to every group of filters according to the relation mask defined by the paired task relationship (Section 3.1). GradSplit ensures the gradients from the t^{th} task loss to affect the selected filters only (marked as red blocks in the middle). It encourages each filter group to be beneficial to the assigned task, while alleviating the potential conflict issue. Note that this training scheme changes the backward pass only, thus forward pass remains the same. Best viewed in color.

This paper uses a simple architecture: all tasks share a common backbone and have separate task-specific heads. We focus on mitigating task conflicts on the shared backbone.

Task-specific Feature Learning The study proposes to learn task-specific attention modules [26], task-specific feature scaling [15], or task-specific masks [28]. These methods need to pass the inputs through the network several times as task-specific intermediate features are not shared. In contrast, our method only needs one forward pass to predict all the tasks and saves computation cost in the inference, especially when a large backbone architecture is used. Bragman *et al*. [2] propose stochastic filter grouping (SFG) that learns to group filters into task-specific or task-agnostic ones in a stochastic manner. Compared with the above methods, which use task-specific features for the dedicated task only, our method enjoys richer representations by sharing the learned task-specific features across tasks.

3. Method

Overview Our aim is to train a unified model that solves multiple human body-related tasks simultaneously. We seek optimal parameters Θ that minimize the joint task loss L,

$$\min_{\Theta} L(\Theta) = \sum_{t=1}^{T} \lambda_t L_t(\Theta), \tag{1}$$

where T and L_t denote the number of tasks and t^{th} task loss, respectively. In this work, we assume a multi-head network which shares a common backbone across tasks and has task-specific heads as illustrated in Fig. 2.

When training a multi-head network, the shared backbone is updated using the gradients back-propagated from all the task losses. It is commonly observed that a multi-task model achieves sub-optimal accuracy compared to single-task models [46, 5, 15]. This is potentially due to conflicts across the tasks. To mitigate this issue, we propose a training scheme, *GradSplit*, that reduce interference among tasks during gradient propagation. Instead of using all the task gradients to update every filter of convolution in the shared backbone, GradSplit utilizes asymmetric pairwise relations between tasks to selectively mask gradients and avoid conflicts.

3.1. GradSplit

Asymmetric Inter-task Relationship The asymmetric inter-task relationship is defined by measuring the impact of tasks on each other. We first train two models with the same backbone architecture with different task heads, where a single-task network is trained for task t and a multi-task network is trained to jointly minimize losses of tasks t and t'. Then, we measure the impact of task t' on task t based on the relative accuracy change $\mathcal{A}_{t' \to t} = \frac{\text{acc}_t|\{t,t'\} - \text{acc}_t|\{t\}}{\text{acc}_t|\{t\}}$, where $\text{acc}_t|\{t, t'\}$ is the accuracy on task t of a multi-task network trained on tasks t and t', $\text{acc}_t|\{t\}$ is the accuracy of a single-task network trained on task t. A positive value of $\mathcal{A}_{t' \to t}$ indicates that the training along with task t' results in performance increase on task t, while a negative value indicates that performance decreases on task t. Based on the relative accuracy change, we determine the directive relation $t' \to t$. Concretely, if $\mathcal{A}_{t' \to t}$ is *smaller* than a threshold τ, the directive relation $t' \to t$ is defined as *negative*. Formally, we represent the relations using $\mathbf{m} \in \{0, 1\}^{T \times T}$:

$$\mathbf{m}_{tt'} = \begin{cases} 0 & \text{if } t \neq t' \text{and relation } t' \to t \text{ is negative} \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

Note that this relationship can be *asymmetric* if joint training affects each task performance in an opposite way, *e.g.*, Attribute → Pose is negative, while Pose → Attribute is not negative. In practice, we enable the network to tolerate a relatively small accuracy drop. Thus, we use τ = - 0.01, which works stably and effectively in the experiment.

Inter-task Relationship based Gradient Update Consider a convolution with c_i input channels and c_o output channels, parameterized by $\theta \in \mathbb{R}^{h\times w\times c_i\times c_o}$. It contains c_o filters and each filter produces one feature map, where h and w indicates height and width, respectively. Based on Eqn. 1, the standard stochastic gradient descent is formulated as:

$$\theta \leftarrow \theta - \alpha \sum_t \nabla_\theta L_t. \tag{3}$$

Since it averages gradients from different tasks, it may cancel out useful signals if the tasks conflict and thus potentially degrade the performance.

To alleviate this issue, GradSplit exploits inter-task relationship to manipulate gradient. Given T tasks, we divide filters into T groups and assign each group explicitly to one task. We denote the parameters assigned to the t^{th} task as $\theta_t \in \mathbb{R}^{h\times w\times c_i\times n_t}$, where $n_t = \lfloor c_o/T \rfloor$ is the number of output channels assigned to the task t. Then, one iteration of parameter update using GradSplit is formulated as:

$$\theta_t \leftarrow \theta_t - \alpha \nabla^{GS}_{\theta_t} L, \text{ where } \nabla^{GS}_{\theta_t} L = \sum_{t'} \mathbf{m}_{tt'} \nabla_{\theta_t} L_{t'}. \tag{4}$$

It updates parameter θ_t using the gradients from only a subset of tasks $\{t'\}$, where the relationship task $t' \rightarrow t$ is not negative, while discarding gradients from the other tasks.

GradSplit *does not* influence the forwarding procedure while affecting only the gradient updating procedure. As a result, it is easily applicable to any convolution layers without modifying the network structure. In the experiment, we apply GradSplit to the last layer (*e.g.*, Layer4 of ResNet-50) of the shared backbone which empirically leads to the best performance (shown in Table 4).

Intuitive Understanding of GradSplit as Regularization Consider manipulating gradients with respect to θ_t as weighted linear sum of task gradients, *i.e.*, $\sum_{t'} \mathbf{w}_{t'} \nabla_{\theta_t} L_{t'}$. When $\mathbf{w}_{t'}$ is a stochastic binary mask, it is equivalent to dropping-out gradients [40] with specifically designed dropout masks with the drop rate $p \in [0, 1]$. When $\mathbf{w}_{t'} = \mathbf{m}_{tt'}$, this becomes equivalent to GradSplit, which is an extreme case when $p \rightarrow 1$. Thus, GradSplit can be also interpreted as a regularizer, as it drops out gradients with specific masks and injects noise to gradients during training.

3.2. Training with Multiple Task-Specific Datasets

We assume a practical setting where each dataset contains annotations for a single task. Under this condition, a model is trained using multiple datasets whose images from different datasets present unique visual characteristics for background, lighting and resolutions. Eqn.(1) is further specified as:

$$\min_\Theta \sum_{t=1}^{T} \lambda_t \mathbb{E}_{\mathcal{D}_t}[\ell_t(f_\Theta(X_t), Y_t)], \tag{5}$$

where ℓ_t and f_Θ denote task t loss function and prediction function, respectively.

Round-Robin Batch-Level Update We adopt round-robin batch-level update regime [24] for optimization. One multi-task iteration consists of a sequence of forwarding each task batch and updating parameters. It is flexible enough to allow different input sizes for different tasks and also scales to the number of tasks with constrained GPU memory.

Domain Gaps between Training Datasets With round-robin batch construction, a mini-batch for task t consists of images sampled from the distribution $\mathcal{D}_t$. The empirical loss is computed as:

$$\sum_{t=1}^{T} \sum_{\mathcal{B}_t} \frac{1}{|\mathcal{B}_t|} \sum_{x_t \in \mathcal{B}_t} \ell_t(f_\Theta(x_t), y_t), \tag{6}$$

where $\mathcal{B}_t$ denotes a mini-batch sampled for task t. Meanwhile, batch normalization (BN) is widely adopted for state-of-the-art network architectures such as EfficientNet [38] and ResNet [13]. Note that BN uses running batch statistics during training and accumulated statistics during inference, with i.i.d. mini-batch assumption. Due to domain gaps between datasets, running BN statistics used to compute task t loss for mini-batch $\mathcal{B}_t$ follow different distributions across tasks during training, whereas common BN statistics are accumulated over tasks and used in the testing stage. We find that such BN statistics mismatch between training and testing stage degrades performance significantly.

As one candidate solution, task-specific BN [3] mitigates this issue by using separate BN modules for different tasks while sharing the remaining convolutions. However, features following the first task-specific BN cannot be shared across tasks and require N forward passes for N tasks, which increases the computational cost. Another solution is to fix BN statistics during training, however, we empirically find this practice is also not suitable. Instead, we use group normalization (GN) [44] in the shared backbone, which can circumvent the above issue. In our experiments, we observed dramatic gains from this choice as shown in the large accuracy gap between two backbone choices (ResNet-50-BN vs. ResNet-50-GN) for the multi-head baseline in Table 2.

4. Experiments

4.1. Experimental Settings

Task setup In this work, we seek to learn a single network that solves several human-related tasks. In the experiment, we choose four representative tasks including both *dense prediction* tasks (human pose estimation and parsing) and *image-level* tasks (person re-identification and attribute recognition). In all experiments, each task is trained and tested on the different splits of the same dataset. Under this setting, there exist domain gaps among different tasks, but there is no domain gap between training and testing set for each task.

Datasets and metric: (I) PA-100k dataset [22] contains 100,000 pedestrian images in total. Each image is annotated with 26 commonly used attributes. We report results for *Attribute* recognition on this dataset. We use mean accuracy (MA) as the evaluation measure.

(II) Market-1501 [50] contains 12,936 training images and 19,732 gallery images. 751 identities are for training and 750 identities for testing. There are 3,368 images from the 750 identities used as queries. We apply this dataset for person re-identification (***ReID***). Rank-1 accuracy and mAP score are reported as the evaluation metrics.

(III) MPII dataset [1] is used for ***Pose*** estimation. This dataset contains 24,920 scene images and 40,522 annotated persons (28,821 for training and 11,701 for the test). Each person has 16 labeled body joints. We adopt the standard Percentage of Correct Keypoints (PCK) measurement as the evaluation metric. Specifically, we measure PCKh@0.5 for individual joints and report the average of them.

(IV) LIP [11] is used for training the human ***Parsing*** task, which consists of pixel-level annotations of 20 semantic human parts (including one background label) from 50,462 images. We follow the standard training/validation split (30,462/10,000) setting and report mIoU score and mean accuracy (Mean Acc.) to evaluate the parsing performance.

Overall multi-task performance We report the summary of multi-task performance following [15, 26]. We measure the performance gain obtained by a multi-task network m compared to the reference single-task networks s: $\Delta_m = \frac{1}{T}\sum_{i=1}^{T}(-1)^{l_i}\frac{M_{m,i}-M_{s,i}}{M_{s,i}}$, where $l_i = 1$ if a lower value means better performance for metric M_i of task i, and 0 otherwise. T is the total number of tasks.

Implementation details We use Pytorch for all the experiments. The ResNet-18-GN and ResNet-50-GN [44] pretrained on ImageNet [6] are used as the shared backbone networks. The PSP structure [48] is used as a head network for Parsing task. For Pose, we follow the encoder-decoder structure proposed in Xiao *et al*. [45]. Finally, we use head structure of "*Conv-BN-FC*" and "*Conv-FC*" for ReID and Attribute, respectively. The Pose head module is initialized from normal distribution followed the practice in [45], and the remaining head modules use kaiming initialization. We used separate optimizers for tasks following [18]. We train the network using a linear warm-up for the initial 3 epochs and then switch to cosine learning rate decay with 10 annealing cycles [23] for the remaining 100 epochs. Each epoch includes 1,000 major iterations, in which we learn each task sequentially. We tried different update frequencies for tasks to handle the different difficulties but empirically observed only slight difference in performance. Thus we used uniform weights in all the experiments for simplicity. We set the initial learning rate to 0.001 and batch size to 64. We use threshold τ= -0.01 to define the pairwise relations among four human-related tasks and summarize them in Table 1.

Table 1. Asymmetric pairwise task relations across four human body-related tasks. Each entry (t', t) corresponds to the relation $t' \rightarrow t$. Negative relation is indicated using ↓. Empty entries denote that performance drop of task t is smaller than a threshold τ=-0.01 when trained together with task t'.

		Performance On			
		Attribute	ReID	Pose	Parsing
Trained With	Attribute	–	↓	↓	↓
	ReID	↓	–	↓	↓
	Pose	–	–	–	–
	Parsing	–	–	–	–

Single-task networks Multi-task models have a accuracy-efficiency trade-off. A naïve solution for multi-task learning is to employ a set of networks where each network is dedicated to one task. We compare with two sets of single-task networks with different network capacities. (***I***) using ResNet-18 models as a **baseline**: each for one of the four tasks. It has 63M parameters which is larger than the proposed model of 52M on the four-task setting. (***II***) using ResNet-50 models as an **an upper bound** for calculating the multi-task performance Δ_m as it shares the same backbone architecture with our multi-task model.

4.2. Domain Gaps between Training Datasets

Discussion on the backbone choice of multi-task models Training using multiple datasets causes a subtle normalization problem as discussed in Section 3.2. This motivates us to use choose a backbone with appropriate normalization. Specifically, the backbone is expected to 1) does not introduce extra parameters or computational cost and 2) mitigate the influence caused by domain gap and achieve reasonably good multi-task accuracy.

In Table 2, we report the results of networks with BN, TBN, and GN. **First**, we observe that BN is relatively suitable for single-task networks, achieving higher or comparable performance compared to its GN-based counterparts (row 1-4). **Second**, due to the reason discussed in Section 3.2, using BN in a multi-task network significantly degrades performance amongst all tasks. For example, using BN in the multi-head baseline is 16.1% lower than using GN in mAP on ReID. **Third**, using TBN [3] as backbone (*i.e.*, ResNet-50-TBN) can mitigate the domain gaps and achieves comparable accuracy with GN on all tasks. However, TBN uses separate BN modules for different tasks, which increases the computation cost for inference: it needs 41G FLOPs under the four-task setting, while using GN only requires 18G FLOPs. Based on the above analysis, we use the backbone with GN to build up solid multi-head baselines.

Table 2. Method comparison under **four**-task setting. We report MA score (%) for **Attribute** on PA-100K [22], mAP (%) on Market-1501 [50] for **ReID**, Mean PCKh@0.5 (%) on MPII [1] for **Pose**, mIoU (%) on LIP [11] for **Parsing**. We also report overall multi-task performance Δ_m (%, see Section 4.1), which indicates the average relative improvement over the single-task baselines (ResNet-50-GN). * denotes using task specific batch normalization (TBN) [3] in the backbone. The best overall multi-task performance is in **bold**.

Methods	Backbone	ReID	Attribute	Pose	Parsing	Δ_m	#Param	#FLOPs
		mAP (↑)	MA (↑)	Mean (↑)	mIoU (↑)	(↑)	(M) ↓	(G) ↓
Single-task Networks	ResNet-50-GN	81.1	78.0	88.2	45.6	+0.0	123	41
(Upperbound)	ResNet-50-BN	83.0	78.3	88.4	45.4	–	123	41
Single-task Networks	ResNet-18-GN	74.9	76.9	87.0	42.4	–	63	24
(Baseline)	ResNet-18-BN	74.2	74.2	87.4	41.9	–	63	24
NDDR [10]	ResNet-18-GN	67.4	76.4	86.8	41.7	-7.3	68	26
Cross-stitch Network [27]		67.0	76.0	87.0	41.2	-7.7	63	24
RCM [15]	ResNet-50-GN	54.9	68.1	69.0	36.1	-21.9	141	80
SFG [2]		64.4	73.9	71.8	34.8	-17.0	52	20
GradNorm [4]		56.1	77.7	68.4	28.5	-23.1	52	18
MTAN [21]		42.7	77.4	86.0	41.9	-14.7	75	40
ASTMT [26]	ResNet-50-TBN*	50.6	78.9	87.0	43.6	-10.6	82	42
Multi-head Baseline	ResNet-50-BN	63.2	76.3	78.9	39.8	-11.9	52	18
	ResNet-50-TBN*	78.1	77.2	86.8	41.8	-3.7	52	41
	ResNet-50-GN	79.3	76.4	86.1	42.7	-3.3	52	18
GradSplit (Ours)	ResNet-50-GN	80.1	77.8	86.4	43.9	**-1.8**	52	18

Table 3. Comparison on **three** tasks: **ReID**, **Attribute**, and **Pose**.

Methods	Backbone	Attribute	ReID	Pose	Δ_m	#Param
		MA (↑)	mAP (↑)	Mean (↑)	(↑)	(M) ↓
Single-task	R50-GN	78.0	81.1	88.2	+0.0	85
	R18-GN	76.9	74.9	87.0	–	39
Cross-stitch [27]	R18-GN	76.3	72.7	86.8	-4.7	38
NDDR [10]		76.1	69.3	86.8	-6.2	42
GradNorm [4]	R50-GN	74.0	54.5	85.1	-13.8	38
MTAN [21]		77.4	50.0	85.5	-14.0	38
Multi-head	R50-GN	75.9	76.5	86.3	-3.5	38
GradSplit		77.6	80.2	86.3	**-1.3**	38

4.3. Multi-Task Rich Human Analysis

GradSplit consistently outperforms multi-head baseline under different multi-task settings As shown in Table 2-3, GradSplit gains further improvements from the multi-head baseline under different multi-task settings. For instance, under the four-task setting (Table 2), GradSplit consistently outperforms multi-head baseline: it is 0.8% in mAP, 1.3% in MA, 0.3% in Mean, and 1.2% in mIoU higher than baseline on ReID, Attribute, Pose, and Parsing, respectively.

We note that GradSplit achieves a better **accuracy-efficiency** trade-off compared to existing methods. In the four-task setting, the overall multi-task performance of GradSplit is comparable to that of the single-task network (upper bound), requiring only 50% of the parameters and FLOPs.

Comparison to task-specific learning methods When updating the filter group assigned to task t, GradSplit drops gradients from the task losses that have negative relation with task t. As a result, it will result in filters that produce features specific to their corresponding tasks. This is significantly different from other methods based on task-specific modules [15, 2, 26]. ***(I)***: RCM [15] introduces Reparameterized Convolutions (RC) on an ImageNet pretrained backbone (fixed during optimization) to learn task-specific features. However, the pretrained feature is unsuitable for our problem where all samples are human images. This potentially prevents RCM from achieving the desired performance in our setting. For instance, it is 9.6 % lower than GradSplit in Attribute (as shown in Table 2). ***(II)***: SFG [2] stochastically re-purposes the convolution filters to be task-specific or shared. In our experiments, it did not improve accuracy from the baseline. ***(III)***: ASTMT [26] uses three task-specific modules (*i.e.*, Parallel RA [31], Squeeze-and-Excitation blocks [14], and task-specific BN (TBN) [3]) to learn task-specific features. ASTMT achieves the highest accuracy on both Attribute and Pose in four-task setting, however, it shows 50.6% mAP on ReID, which is 29.5% lower than GradSplit. Due to this imbalance, the overall multi-task performance Δ_m is 8.8% lower than GradSplit.

Task balancing methods We report the results of GradNorm [4] in Table 2 and Table 3. It aims to balance task losses by stimulating the task-specific gradients to be of sim-

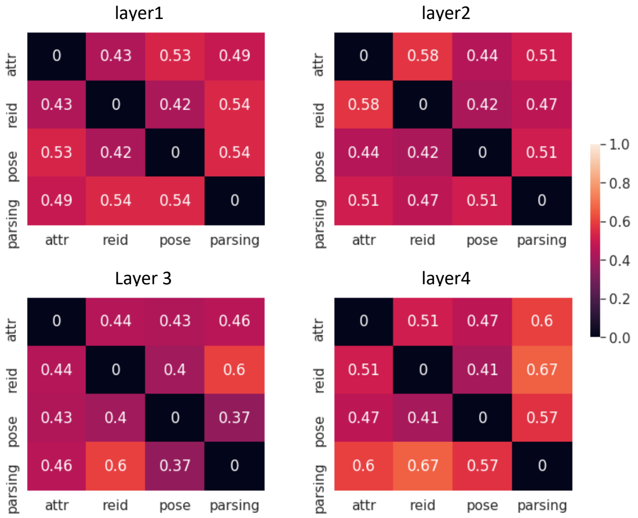

Figure 3. Example visualization of conflict frequency of task-specific gradients in different layers of multi-head baseline network (ResNet-50-GN). We quantify the relation of gradients amongst all tasks by calculating the cosine similarity scores. The conflict frequency in this figure represents how likely the gradients of two tasks have a negative correlation.

ilar magnitude. In our setting, the magnitude of pose loss is smaller than the magnitudes of other task losses. Under the multi-dataset learning setting, GradNorm failed to handle this imbalance effectively and achieved overall low accuracy. MGDA [32], which seeks to find Pareto optimal solutions, was also tested, but the overall accuracy was low. Please refer to the supplementary document for more results.

Comparison to backbone-focused architectures These methods consider feature sharing in the backbone. In Table 2 and Table 3, we compare GradSplit with four representative architectures, including MTAN [21], NDDR [10], and Cross-stitch [27]. Cross-Stitch and NDDR first train task-specific streams and then fine-tune the whole network, including the interaction module. Empirically, this way cannot bring useful information across tasks and failed to improve accuracy. The task-specific attention module proposed by MTAN only brings improvement over multi-head baseline on Attribute, while reporting low accuracy on the remaining tasks. Different from these methods, our GradSplit consistently brings improvements for all tasks over multi-head baseline.

4.4. Component Analysis and Discussion

Visualization of gradient conflict We show an example gradient conflict in the multi-head network via a conflict frequency map. We obtain the gradients for each task with a batch of 16 samples. We then quantify the relationship of task gradients by calculating the cosine similarity matrix. We repeat the process 20 times and report the average values. We define that there is a "gradient conflict" if the cosine similarity of gradients is negative given task gradients, following the practice in [46]. In Fig. 3, we show an example conflict frequency map at the first block in different layers of the multi-head baseline. We observe that values in the entries deviate from the chance of 0.5. Depending on the pairwise relation of tasks, conflicts occur more frequently if tasks are competing with each other. For example, we show ReID and Parsing conflict more often than average in layer 4.

Table 4. Analysis of GradSplit: 1) effect of applying Gradsplit on more layers; 2) effect of random dropout mask; 3) task-specific filters; and 4) GradSplit$_{\tau=\text{inf}}$: each group of filters is *only* updated by its assigned task loss. Layer i-j denotes applying GradSplit on four layers, from layer i to j. Hyper-parameter p is the drop rate of DropGrad [40]. ResNet-18-GN is used as the shared backbone.

Methods		Pose	Attribute	ReID		Parsing
		Mean	MA	Rank-1	mAP	mIoU
Multi-head Basel.		84.9	75.5	86.2	64.7	38.0
GradSplit	Layer 4	**85.4**	**77.1**	**89.2**	**71.4**	**39.1**
	Layer 3-4	85.0	77.1	88.0	68.0	38.3
	Layer 2-4	85.2	77.0	87.4	67.6	38.0
	Layer 1-4	84.6	77.0	87.6	66.9	36.6
DropGrad (p=0.50)		81.5	74.0	85.8	64.3	36.3
DropGrad (p=0.75)		81.5	73.9	85.3	63.7	36.8
GradSplit$_{\tau=\text{inf}}$		84.8	76.9	87.6	67.6	38.1

Which modules to apply GradSplit Table 4 shows the results of GradSplit when it is applied to a different combination of layers. The results are on the four-task setting, and both baseline and GradSplit use ResNet-18-GN as backbones. The Layer i-j denotes that GradSplit is applied on the i^{th} layer to the j^{th} layer. For example, Layer 2-4 means that GradSplit is applied to layer 2, 3, and 4. GradSplit can alleviate conflicting gradient issue but at the same time decreases the model capacity per task by reducing the number of dedicated filters. Meanwhile, some common features (*e.g.*, low-level patterns) can be shared across tasks and enhance representation ability when trained with joint loss. There is a trade-off between the above two factors. The results in Table 4 reflect this, showing the highest accuracy when GradSplit is applied to only Layer 4.

Comparison with regularization method GradSplit can be understood as a regularization method (discussed in Section 3.1). We compare GradSplit with DropGrad [40], which is a regularization method originally proposed for meta-learning. The key difference from GradSplit is that it drops gradients randomly from every filter so that the dominant task gradients of a filter can vary during training. Comparing DropGrad (p = 0.75) and Layer 4 in Table 4 implies that fixing the task-agnostic mask based on the inter-task relationship for gradient dropout is important. Note that when setting p = 0.75, the ratio of dropped gradients in DropGrad matches with GradSplit.

Table 5. Comparison when VGG-11 is employed as the backbone, for MPII, Market attributes, Market-1501 and LIP datasets.

Methods	Pose	Attribute	ReID		Parsing
	Mean	MA	Rank-1	mAP	mean Acc.
Multi-head Basel.	**83.6**	71.0	76.0	50.5	45.8
GradSplit	83.0	**72.6**	**78.4**	**53.3**	**46.0**

Table 6. Cross-dataset accuracy comparison on **Market-1501** for **Attribute**. In this table, the multi-task networks are trained under four-task setting. Note that, the Attribute annotation of Market-1501 is *not used* during training. We report the MA score (%) of common attributes appeared in Market-1501.

Methods	Backbone	Attribute (MA)
Single-task network	ResNet-50-BN	71.5
	ResNet-50-GN	73.0
MTAN [21]	ResNet-50-GN	75.5
GradNorm [4]	ResNet-50-GN	75.5
ASTMT [26]	ResNet-50-TBN	76.5
Multi-head baseline	ResNet-50-GN	74.6
	ResNet-50-TBN	73.8
GradSplit	ResNet-50-GN	**77.5**

Effect of inter-task relationship We evaluate an extreme variant of our method, *i.e.*, "GradSplit$_{\tau=\text{inf}}$", which uses the threshold $\tau = \text{inf}$ to obtain inter-task relation. This is equivalent to using the mask $\tilde{\mathbf{m}} = \mathbf{I}_T$ instead of $\mathbf{m}$, where $\mathbf{I}_T$ is an identity matrix of size T. In a nutshell, each group of filters is *only* updated by its assigned task loss. This practice potentially avoids the conflict issues and achieves some improvements over multi-head baseline in Table 4 (*e.g.*, 1.4% on Attribute and 2.9% mAP on ReID). However, without inter-task relationship, GradSplit$_{\tau=\text{inf}}$ cannot effectively manipulate gradient to learn the shared backbone, being inferior to GradSplit on all tasks.

Different network architectures In addition to ResNet-50 and ResNet-18, we use VGG-11 as backbone and report results in Table 5. We observe that GradSplit improves over multi-head baselines in three tasks and is comparable for the fourth (Pose), achieving overall good performance.

Cross-dataset analysis To further study the effect of multi-task learning and advantage of GradSplit, we conduct a cross-dataset analysis. Specifically, we report the results of Attribute on Market-1501 dataset. Note that, Market-1501 is used for ReID task and its Attribute annotations are not used during training. We have two observations in Table 6. **First**, multi-task learning can help cross-dataset testing. Compared with single-task network, all multi-task models achieve higher MA score on Market-1501. This implies that multi-task models learns more robust feature representations. **Second**, GradSplit gain the highest accuracy, which further validates its effectiveness.

Table 7. Comparison on **four** tasks when increasing the backbone capacity. Task-specific L4 branches out different task-head networks at the end of layer-3 of ResNet-50 and it uses layer4 structures for each task (R50-L4). R50-GN+ increase the backbone capacity of ResNet-50 by adding more convolutions to layer4.

Methods	Backbone	Attr	ReID	Pose	Parsing	Δ_m	#Param
		MA	mAP	Mean	mIoU	(↑)	(M)
Single-task	R50-GN	78.0	81.1	88.2	45.6	+0.0	123
Task-specific L4	R50-L4	76.8	78.2	86.4	43.5	-2.9	96
DropGrad (p=0.50)	R50-GN+	77.9	80.2	86.4	42.2	-2.7	72
Multi-head		77.1	80.4	87.8	46.9	+0.1	72
GradSplit		78.2	81.6	87.9	47.4	**+1.1**	72

Effect of backbone capacity. To study this, we add more convolutions to layer4 of ResNet-50-GN (denoted as R50-GN+). We also report results with task-specific layer4 (denoted as R50-L4) for each task. As shown in Table 7, using task-specific layer4 for each task cannot bring improvements. This is probably because there are still task conflicts in the shared backbone. Moreover, DropGrad does not work either. This further suggests that gradient removal based on inter-task relations is crucial. We also observe that the multi-head baseline matches the single-task performance (Δ_m=+0.1). GradSplit achieves a further +1.0% improvement and outperforms single-task accuracy on three tasks.

5. Conclusion

We present a framework to train a unified model for multiple human-related tasks. When tasks require task-specific features, they may compete each other with conflicting gradients during training, leading to lower overall accuracy. To alleviate this issue, we propose a novel training scheme called GradSplit, which enables each task to learn its assigned filters without the interference from other tasks. Moreover, Gradsplit enables each task-specific filter to *selectively* leverage all input features, producing more informative features for its assigned task. In the experiment, we extensively test GradSplit on four human-related tasks. We show that GradSplit consistently outperforms strong baselines and achieves a better accuracy-efficiency trade-off.

Limitations and future work To reduce the computational overhead for task relation estimation, it would be also interesting to study other strategies based on gradient similarity and loss changes. As GradSplit is a general approach which is applicable beyond human-related tasks, our future research includes the extension to other applications.

Acknowledgements This work was a part of Weijian Deng's internship at NEC Labs America. This work was also partially supported by the ARC Discovery Early Career Researcher Award (DE200101283) and the ARC Discovery Project (DP210102801).

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[2] Felix JS Bragman, Ryutaro Tanno, Sebastien Ourselin, Daniel C Alexander, and Jorge Cardoso. Stochastic filter groups for multi-task cnns: Learning specialist and generalist convolution kernels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1385–1394, 2019.

[3] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7354–7362, 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] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In *NeurIPS*, 2020.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 304–311. IEEE, 2009.

[8] Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, pages 27503–27516, 2021.

[9] Yuan Gao, Haoping Bai, Zequn Jie, Jiayi Ma, Kui Jia, and Wei Liu. Mtl-nas: Task-agnostic neural architecture search towards general-purpose multi-task learning. In *CVPR*, 2020.

[10] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3205–3214, 2019.

[11] Ke Gong, Xiaodan Liang, Dongyu Zhang, Xiaohui Shen, and Liang Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 932–940, 2017.

[12] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *ICML*, 2020.

[13] 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.

[14] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[15] Menelaos Kanakis, David Bruggemann, Suman Saha, Stamatios Georgoulis, Anton Obukhov, and Luc Van Gool. Reparameterizing convolutions for incremental multi-task learning without task interference. In *Proceedings of the European conference on computer vision (ECCV)*, 2020.

[16] 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 conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[17] 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 Conference on Computer Vision and Pattern Recognition*, pages 6129–6138, 2017.

[18] Jogendra Nath Kundu, Nishank Lakkakula, and R Venkatesh Babu. Um-adapt: Unsupervised multi-task adaptation using adversarial cross-task distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1436–1445, 2019.

[19] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qingfu Zhang, and Sam Kwong. Pareto multi-task learning. *NeurIPS*, 2019.

[20] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. Improving person re-identification by attribute and identity learning. *Pattern Recognition*, 95:151–161, 2019.

[21] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019.

[22] Xihui Liu, Haiyu Zhao, Maoqing Tian, Lu Sheng, Jing Shao, Shuai Yi, Junjie Yan, and Xiaogang Wang. Hydraplus-net: Attentive deep features for pedestrian analysis. In *Proceedings of the IEEE international conference on computer vision*, pages 350–359, 2017.

[23] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[24] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10437–10446, 2020.

[25] Pingchuan Ma, Tao Du, and Wojciech Matusik. Efficient continuous pareto exploration in multi-task learning. In *International Conference on Machine Learning*, pages 6522–6531. PMLR, 2020.

[26] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1860, 2019.

[27] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[28] Lucas Pascal, Pietro Michiardi, Xavier Bost, Benoit Huet, and Maria A Zuluaga. Maximum roaming multi-task learning. In *AAAI*, 2021.

[29] Dripta S Raychaudhuri, Yumin Suh, Samuel Schulter, Xiang Yu, Masoud Faraki, Amit K Roy-Chowdhury, and Manmohan Chandraker. Controllable dynamic multi-task architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10955–10964, 2022.

[30] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516, 2017.

[31] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8119–8127, 2018.

[32] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pages 527–538, 2018.

[33] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, pages 9120–9132. PMLR, 2020.

[34] Chi Su, Jianing Li, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. Pose-driven deep convolutional model for person re-identification. In *Proceedings of the IEEE international conference on computer vision*, pages 3960–3969, 2017.

[35] Yumin Suh, Jingdong Wang, Siyu Tang, Tao Mei, and Kyoung Mu Lee. Part-aligned bilinear representations for person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–419, 2018.

[36] Ximeng Sun, Rameswar Panda, and Rogerio Feris. Adashare: Learning what to share for efficient deep multi-task learning. In *NeurIPS*, 2020.

[37] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3800–3808, 2017.

[38] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114. PMLR, 2019.

[39] Chiat-Pin Tay, Sharmili Roy, and Kim-Hui Yap. Aanet: Attribute attention network for person re-identifications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[40] Hung-Yu Tseng, Yi-Wen Chen, Yi-Hsuan Tsai, Sifei Liu, Yen-Yu Lin, and Ming-Hsuan Yang. Regularizing meta-learning via gradient dropout. *arXiv preprint arXiv:2004.05859*, 2020.

[41] Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Revisiting multi-task learning in the deep learning era. *arXiv preprint arXiv:2004.13379*, 2020.

[42] Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. *arXiv preprint arXiv:2010.05874*, 2020.

[43] Longhui Wei, Shiliang Zhang, Hantao Yao, Wen Gao, and Qi Tian. Glad: Global-local-alignment descriptor for pedestrian retrieval. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 420–428, 2017.

[44] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[45] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018.

[46] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *NeurIPS*, 2020.

[47] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3221, 2017.

[48] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[49] Liang Zheng, Yujia Huang, Huchuan Lu, and Yi Yang. Pose-invariant embedding for deep person re-identification. *IEEE Transactions on Image Processing*, 28(9):4500–4509, 2019.

[50] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015.