End-to-End Multi-Task Learning with Attention

Shikun Liu Edward Johns Andrew J. Davison

Department of Computing, Imperial College London
{shikun.liu17, e.johns, a.davison}@imperial.ac.uk

Abstract. In this paper, we propose a novel multi-task learning architecture, which incorporates recent advances in attention mechanisms. Our approach, the Multi-Task Attention Network (MTAN), consists of a single shared network containing a global feature pool, together with task-specific soft-attention modules, which are trainable in an end-to-end manner. These attention modules allow for learning of task-specific features from the global pool, whilst simultaneously allowing for features to be shared across different tasks. The architecture can be built upon any feed-forward neural network, is simple to implement, and is parameter efficient. Experiments on the CityScapes dataset show that our method outperforms several baselines in both single-task and multi-task learning, and is also more robust to the various weighting schemes in the multi-task loss function. We further explore the effectiveness of our method through experiments over a range of task complexities, and show how our method scales well with task complexity compared to baselines.

Keywords: multi-task learning, attention, convolutional neural network, adaptive task weighting

1 Introduction

Deep Convolutional Neural Networks (CNNs) have seen great success in a range of computer vision tasks, including image classification [1], semantic segmentation [2], depth estimation [3], and style transfer [4]. However, these networks are typically designed to achieve only one particular task. For more complete vision systems in real-world applications, a network which can perform multiple tasks simultaneously is far more desirable than building a set of independent networks, one for each task. This is more efficient not only in terms of memory and inference speed, but also in terms of data, since related tasks may share informative visual features. For example, the task of depth estimation with supervised learning may help with a semantic segmentation task, by learning features which are responsive to depth discontinuities, and hence which may provide cues for segmentation.

This type of learning is called Multi-Task Learning (MTL) [5,6,7], and in this paper we present a novel approach to MTL using attention masks, which further emphasises this ability to share complementary features. Compared to standard single-task learning, training multiple tasks whilst successfully learning a shared representation poses two key challenges:

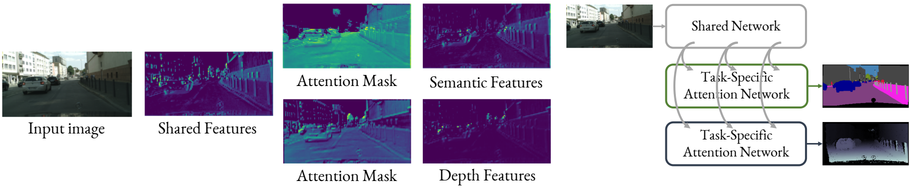
i) **Network Architecture (how to share)**: A multi-task learning architecture should express both *task-shared* features and *task-specific* features. As such, the network is encouraged to learn a generalisable representation (to avoid over-fitting), whilst also providing the ability to learn features tailored to each task (to avoid under-fitting).
ii) **Loss Function (how to balance tasks)**: A multi-task loss function, which weights the relative contributions of each task, should enable learning of all tasks with equal importance, without allowing learning to be dominated by the easier tasks. Manual tuning of loss weights is tedious and sub-optimal, and so automatically learning these weights, or designing a network which is robust to these weights, is highly desirable.

However, most prior multi-task learning approaches focus on only one of these two challenges, whilst maintaining a standard implementation of the other. In this paper, we introduce a unified approach which addresses both challenges cohesively, by designing a novel network which (i) enables both task-shared and task-specific features to be learned automatically, and consequently (ii) learns an inherent robustness to the choice of loss weighting scheme.

The proposed network, which we call the Multi-Task Attention Network (MTAN) (see Fig 1), is composed of a single shared network, which learns a global feature pool containing features across all tasks. Then for each task, rather than learning directly from the shared feature pool, a soft attention mask is applied at each convolution block in the shared network. In this way, each attention mask automatically determines the importance of the shared features for the respective task, allowing learning of both task-shared and task-specific features in a self-supervised, end-to-end manner. This flexibility enables much more expressive combinations of features to be learned for generalisation across tasks, whilst still allowing for discriminative features to be tailored for each individual task. Furthermore, automatically choosing which features to share and which to be task specific allows for a highly efficient architecture, with far fewer parameters than multi-task architectures which have explicit separation of tasks. [8,5].

We evaluated the performance with the tasks of semantic segmentation and depth estimation on the CityScapes dataset [9], with all implementations based upon the SegNet architecture [2]. Results show that our network outperforms several baselines for both single-task and multi-task learning, whilst also showing much greater robustness to the choice of weighting scheme in the loss function. As part of our evaluation of this robustness, we also propose a novel weighting scheme, Dynamic Weight Average (DWA), which adapts the task weighting over time by considering the rate of change of the loss for each task.

As further introspection into the benefits of multi-task learning, we also designed experiments to investigate how learning scales with task complexity, such as an increasing number of classes in semantic segmentation. Results showed that as task complexity increases, the relative performance of multi-task learning over single-task learning also increases due to the need to compress greater representation into a single network, and hence the increase in the benefits of sharing this information across multiple tasks.



(a) Examples of attention features learned in MTAN

(b) Overview of MTAN

Fig. 1: (a) Examples of attention masks learned in layer 1 of the proposed network, for the tasks of semantic segmentation and depth estimation. (b) Overview of our proposal MTAN. The shared network takes in an image and learns task-shared features, whilst each attention network learns task-specific features, by applying attention masks to the shared network.

1.1 Contributions

- We propose a novel multi-task learning architecture, called the Multi-Task Attention Network (MTAN), which uses attention masks to enable learning of both task-shared and task-specific features in an end-to-end manner.
- We present experimental results which show that our architecture outperforms a number of single-task and multi-task architectures, and offers greater robustness to the choice of weighting scheme in the loss function.
- We demonstrate further results studying the relative performance of of multi-task learning over single-task learning, as the complexity of the tasks increases.

2 Related Work

The term Multi-Task Learning (MTL) has been broadly used in machine learning [10,11,7,12], with similarities to transfer learning [13,14] and continual learning [15]. Multi-task learning has also often implicitly been used without explicit reference. The most common example is that of a neural network pre-trained with a large-scale dataset, such as ImageNet [16], as a rich prior to a supervised fine-tuning [17] procedure. More explicitly, multi-task learning is often used with CNNs in computer vision to model two or even more related tasks jointly, such as image classification in multiple visual domains [18], or to couple dense prediction tasks such as the estimation of depth maps, surface normals and semantic segmentation [5,19], or for pose estimation and action recognition [20].

In this paper, we present a principled and unified approach to multi-task learning and thus address two important questions: how to design a good multi-task network architectures and how to balance feature sharing in multi-task learning?

Most multi-task learning network architectures are designed based on existing feed-forward deep neural networks. Some recent work includes [7] which proposed a multi-task network based on ResNet101 architecture [21], with a lasso-regularised combination of features from different layers, to encourage the network to separate features that are useful for different tasks. The Cross-stitch Network [5] used 'cross-stitch units' to combine two task-specific features, using a learnable linear combination of input activation maps. UberNet [22] proposed a common CNN trunk which based on VGG-Net [23], to perform as many as 7 tasks. The Progressive Networks method [8] applied a teacher-student relationship to accelerate training in multiple Atari games. However, some networks such as Cross-Stitch Networks and Progressive Networks are not parameter-efficient, since the network size increases linearly with the number of tasks. A desirable characteristic of multi-task learning is efficiency and this requires the network size to grow only gradually as extra tasks are added. Our proposed method aims to achieve this through the use of soft attention.

Regarding the balancing of feature sharing in multi-task learning, there is extensive experimental analysis in [5,6], with both papers arguing that

"*A different amount of sharing and weighting tends to work best for different tasks.*"

However, both of these approaches only applied a simple multi-task learning network, by taking a feedforward network and splitting it at the final layer, or exhaustively testing splitting different layers in a pair of tasks.

There has been some recent inspiring work focusing on task weighting in multi-task training. One approach is to use weight uncertainty [6], which modifies the loss functions of multi-task learning using homoscedastic task uncertainty based on a Gaussian approximation. Another method is that of GradNorm [24], which manipulates gradient norms over time to control the training dynamics.

3 Multi-Task Attention Network

We now introduce our novel multi-task learning architecture based on attention, the Multi-Task Attention Network (MTAN). The proposed architecture can readily be incorporated into any feed-forward network, and in the following section we demonstrate how to build MTAN upon an encoder-decoder network, SegNet [2]. This example configuration allows for image-to-image dense pixel-level prediction, such as semantic segmentation and depth prediction.

3.1 Architecture Design

MTAN consists of two components: a single shared network, and K task-specific attention networks. The shared network can be designed based on the particular task (in our case, SegNet, for image-to-image predictions), whilst each task-specific network consists of a set of attention modules, which link with the shared network. The attention modules apply a soft attention mask to the shared network, to determine the importance of each feature for the particular task. As such, the soft attention masks can be considered as feature selectors from the shared network, which are automatically learned in an end-to-end manner, whilst the shared network learns a compact global pool of features across all tasks.

A detailed visualisation of our network based on VGG-16 [23] is shown in Figure 2, which displays the encoder half of SegNet. The decoder half of SegNet is then symmetric to VGG-16. As shown, each attention module learns a soft attention mask, which itself is dependent on the features in the shared network at the corresponding layer. Therefore, the features in the shared network, and the soft attention masks, can be learned jointly to maximise the generalisation of the shared features across multiple tasks, whilst simultaneously maximising the task-specific performance due to the attention masks.

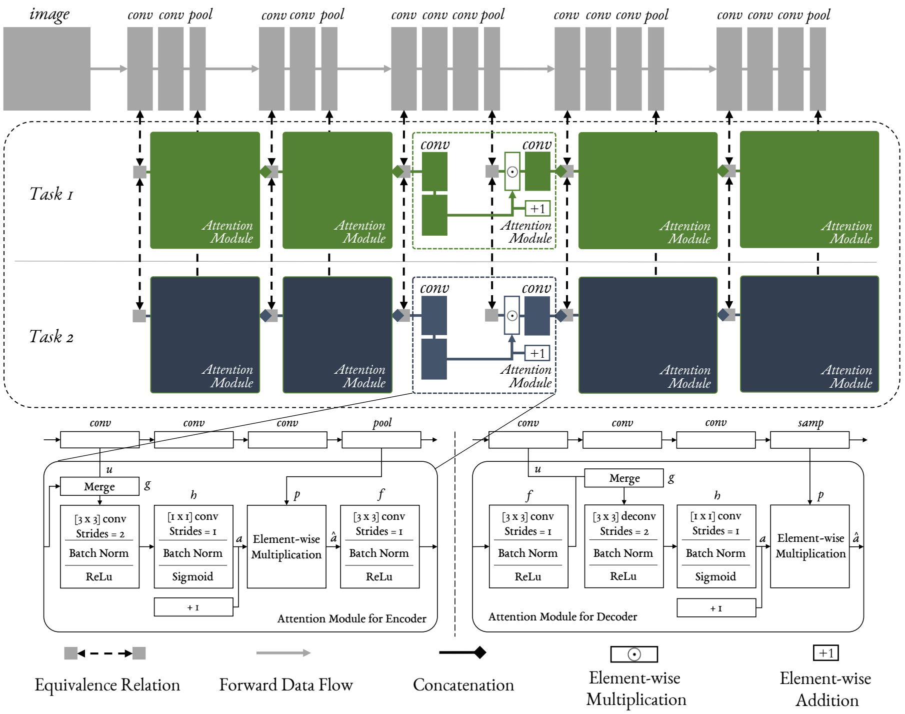


Fig. 2: Visualisation of MTAN based on VGG-16, showing the encoder half of SegNet (with the decoder half being symmetrical to the encoder). Task one (green) and task two (blue) have their own set of attention modules, which link with the shared network (grey). Here, the middle of the five attention modules has its structure exposed for visualisation, which is then further expanded in the bottom section of the figure, showing both the encoder and decoder versions of the module.

3.2 Task Specific Attention Module

The attention module is designed to allow the task-specific network to learn task-related features, by applying a soft attention mask to the features in the shared network, with one attention mask per task per feature channel. We denote the shared features in the j^{th} block of the shared network as $p^{(j)}$, and the learned attention mask in this layer for task i as $a_i^{(j)}$. The task-specific features $\hat{a}_i^{(j)}$ in this layer, are then computed by element-wise multiplication of the attention masks with the shared features:

$$\hat{a}_i^{(j)} = (a_i^{(j)} + 1) \odot p^{(j)} \ , \tag{1}$$

where $\odot$ denotes element-wise multiplication. The '+1' operation is the residual identity mapping motivated by [21,25], to help the network learn more robust attention maps by avoiding exploding or vanishing gradients, which may otherwise be caused by consecutive layer-by-layer multiplications.

As shown in Figure 2, the first attention module in the encoder takes as input only features in the shared network. But for subsequent attention modules in layer j, the input is formed by a concatenation of the shared features $u^{(j)}$, and the task-specific features from the previous layer $\hat{a}_i^{(j-1)}$:

$$a_i^{(j)} = h\left(g\left(\left[u^{(j)}; f\left(\hat{a}_i^{(j-1)}\right)\right]\right)\right), \quad j \geq 2 \ , \tag{2}$$

Here, f, g, h are convolutional (or deconvolutional) layers with batch normalisation, following a non-linear activation ReLu in f, g or Sigmoid in h. Both f and g are composed with a $[3 \times 3]$ kernel, while h uses a $[1 \times 1]$ kernel to match the channels between the concatenated features and the shared features. Furthermore, in function g, we apply a stride of size 2, to match the compressed / up-sampled resolution from the pooling / up-sampling operation. See Figure 2 for the equivalence of architecture between the encoder and decoder.

The attention mask $a_i^{(j)} \in [0, 1]$ is learned in a self-supervised fashion with back-propagation. If $a_i^{(j)} \to 0$, the attended feature maps are equivalent to global feature maps and the tasks share all the features. Therefore, we expect the performance to be no worse than that of a shared multi-task network, which splits into individual tasks only at the end of the network, and we show results demonstrating this in Section 4.

3.3 The Model Objective

In general multi-task learning with K tasks, a loss function with input $\mathbf{X}$ and task-specific labels $\mathbf{Y}_i, i = 1, 2, \cdots, K$, is defined as,

$$\mathcal{L}_{tot}(\mathbf{X}, \mathbf{Y}_{1:K}) = \sum_{i=1}^{K} \lambda_i \mathcal{L}_i(\mathbf{X}, \mathbf{Y}_i). \tag{3}$$

This is the linear combination of task-specific losses $\mathcal{L}_i$ with task weightings λ_i. In our experiments, we study the effect of different weighting schemes on the performance of MTAN, and on other multi-task learning approaches.

In our application to image-to-image pixel-level dense prediction, we perform the following two tasks, where $\hat{\mathbf{Y}}$ represents the network predicted result, and $\mathbf{Y}$ represents the ground-truth label:

Semantic Segmentation. For semantic segmentation, we apply a pixel-wise cross-entropy for each predicted class label from a depth-softmax classifier. We average the result for each valid pixel.

$$\mathcal{L}_1(\mathbf{X}, \mathbf{Y}_1) = -\frac{1}{pq} \sum_{p,q} \mathbf{Y}_1(p,q) \log \hat{\mathbf{Y}}_1(p,q). \tag{4}$$

Depth Estimation. For depth estimation, we apply an L_1 norm comparing the inverse predicted and ground-truth depth, as inverse depth can more easily represent points at infinite distances (such as the sky).

$$\mathcal{L}_2(\mathbf{X}, \mathbf{Y}_2) = \frac{1}{pq} \sum_{p,q} |\mathbf{Y}_2(p,q) - \hat{\mathbf{Y}}_2(p,q)|. \tag{5}$$

4 Experiments

In this section, we introduce the dataset used for validation in Section 4.1, and introduce several baselines for comparison to our MTAN method in Section 4.2. In Section 4.3, we introduce a novel adaptive weighting method, and in Section 4.4 we show the effectiveness of MTAN with various weighting methods compared with single and multi-task baseline methods. We show the visualisation of the learned attention masks in Section 4.5, and we explore how the performance of our method scales with task complexity in Section 4.6.

4.1 Datasets

Validation is carried out on the CityScapes dataset [9], which includes 2975 training and 500 validation high-resolution images, with publicly-available annotations. We use this dataset for two tasks: semantic segmentation and depth estimation. To speed up training, all training and validation images were resized to [128 × 256] resolution.

The dataset contains 19 classes for pixelwise semantic segmentation, together with pixelwise depth labels. The depth data for this dataset was calculated using the SGM algorithm [26], and is represented as inverse ground-truth depth. We pair the depth estimation task with three levels of semantic segmentation using 2, 7 or 19 classes (excluding the void group in 7 and 19 classes). Labels for the 19 classes are the original ground-truth labels, and the coarser 7 categories are defined as in the original CityScapes dataset. We further create a 2-class dataset with only background and foreground object classes. The details of these segmentation classes are presented in Table 1. Please note that both the 7 and 19-class CityScapes datasets have a void class which is not used in network training. We perform multi-task learning for 7-class CityScapes dataset in Section 4.4, with visualisation of these attention maps in Section 4.5, and we compare the 2/7/19-class CityScapes datasets in Section 4.6.

Table 1: Three levels of semantic classes for the CityScapes data used in our experiments.

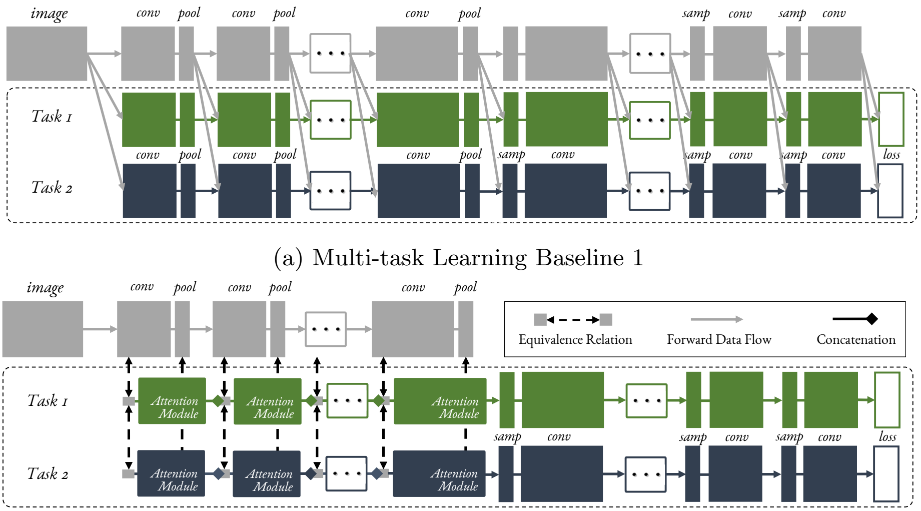
2-class	**7-class**	**19-class**
background	void	void
	flat	road, sidewalk
	construction	building, wall, fence
	object	pole, traffic light, traffic sign
	nature	vegetation, terrain
	sky	sky
foreground	human	person, rider
	vehicle	carm truck, bus, caravan, trailer, train, motorcycle

4.2 Baselines

Most image-to-image multi-task learning architectures are designed based on specific feedforward neural networks, and thus they are usually not directly comparable. Therefore, for a fair comparison across multi-task learning methods, we designed 4 networks (1 single-task + 3 multi-task) based on SegNet [2], which we consider as baselines:

- **STAN-SegNet:** Single-Task Attention Network, where we directly apply our proposed MTAN whilst only performing a single task.
- **SegNet, Split:** The vanilla feedforward SegNet, which splits at the last layer for the final prediction of two tasks.
- **MTLBL1-SegNet:** A shared network with two task-specific networks, where each task-specific network receives all features from the shared network, without any attention module. This is similar to the Cross-Stitch Network [5], but replaces the cross-stitches with an additional shared network for a closer comparison to our method. (Note that the original Cross-Stitch Network scales poorly with the number of tasks, whereas our method scales sub-linearly.)
- **MTLBL2-SegNet:** The encoder part of the network is the same as our proposed MTAN-SegNet, but we don't apply attention modules to the decoder. Each task-specific decoder network takes its only task-specific features and uses the standard SegNet decoder to return task predictions.

Both baselines MTLBL1-SegNet and MTLBL2-SegNet have more parameters than our proposed MTAN-SegNet, and were tested to validate that our proposed method's better performance is due to the attention modules, rather than simply due to the increase in network parameters. A detailed visualisation of these two baselines is presented in Figure 3.



(a) Multi-task Learning Baseline 1

(b) Multi-task Learning Baseline 2

Fig. 3: Two proposed Multi-task Learning Baseline based on SegNet.

4.3 Dynamic Weight Average

For most multi-task learning networks, training multiple tasks is difficult without finding the correct balance between those tasks. Recent approaches have attempted to address this issue [24,6]. To test whether our network is robust to various weighting methods, we propose a simple yet effective adaptive weighting method, named Dynamic Weight Average (DWA). Inspired by GradNorm [24], this learns to average task weighting over time by considering the rate of change of loss for each task. But whilst GradNorm requires access to the network's internal gradients, our DWA proposal only requires the numerical task loss, and therefore its implementation is far simpler.

We define the task weighting λ_k for task k as:

$$\lambda_k(t) := \frac{K \exp(w_k(t-1)/T)}{\sum_i \exp(w_i(t-1)/T)}, \quad w_k(t-1) = \frac{\mathcal{L}_k(t-1)}{\mathcal{L}_k(t-2)}, \tag{6}$$

Here, $w_k(\cdot)$ calculates the relative descending rate in the range $(0, +\infty)$, t is an iteration index, and T represents a temperature which controls the softness of task weighting, similar to [27]. A large T results in a more even distribution between different tasks. If T is large enough, we have $\lambda_i \approx 1$, and tasks are weighted equally. Finally, the softmax operator which is multiplied by K ensures that $\sum_i \lambda_i(t) = K$.

In our implementation, the loss value $\mathcal{L}_k(t)$ is calculated as the average loss in each epoch over several iterations. Doing so reduces the uncertainty from stochastic gradient descent and random training data selection. For $t = 1, 2$, we initialise $w_k(t) = 1$, but any non-balanced initialisation based on prior knowledge on training data could also be introduced.

4.4 Results on Multi-Task Learning

We now evaluate the performance of our proposed MTAN method in image-to-image multi-task learning, based on the SegNet architecture. Using the 7-class version of the CityScales dataset, we compare all the baseline methods introduced in Section 4.2, together with vanilla SegNet itself [2].

Table 2: 7-class semantic segmentation and depth estimation results on training and validation for the CityScapes dataset. We measure semantic segmentation as the mean intersection-over-union (mIoU) (the higher the better) and depth estimation as relative absolute error (RAE) (the lower the better). Column #P compares the number of network parameters, and the best performing combination of multi-task architecture and weighting is highlighted in bold.

Type	#P.	Architecture	Weighting	Semantic (mIoU)		Depth (RAE)	
				Train	Val	Train	Val
Single Task	1	Vanilla SegNet [2]	n.a.	0.7012	0.5097	0.3101	0.5027
	1.32	STAN - SegNet	n.a.	0.6950	0.5200	0.4535	0.5535
Multi Task	≈1	Vanilla SegNet, Split	Equal Weights	0.6850	0.5067	0.3542	0.5666
			Uncert. Weights [6]	**0.6965**	0.4957	0.4434	0.5726
			DWA, $T = 20$	0.6937	**0.5112**	**0.3540**	**0.4969**
	3.02	MTLBL1-SegNet	Equal Weights	0.7104	0.5164	0.4066	0.5166
			Uncert. Weights [6]	**0.7459**	0.5126	0.4480	0.5352
			DWA, $T = 20$	0.7126	**0.5202**	**0.3333**	**0.4491**
	2.06	MTLBL2-SegNet	Equal Weights	0.6672	0.5161	0.3817	0.5019
			Uncert. Weights [6]	**0.6833**	0.5130	0.4478	0.5440
			DWA, $T = 20$	0.6736	**0.5188**	**0.3714**	**0.4786**
	1.64	MTAN-SegNet	Equal Weights	0.6929	**0.5268**	0.3490	0.4246
			Uncert. Weights [6]	**0.6957**	0.5152	0.4318	0.4768
			DWA, $T = 20$	0.6863	0.5259	**0.3417**	**0.4184**

Training. For each network architecture, we ran experiments with three types of weighting methods: equal weighting, weight uncertainty [6], and our proposed DWA (with hyper-parameter temperature $T = 20$, found empirically to be optimum). We trained all the models with stochastic gradient descent using a learning rate of 0.01 and momentum of 0.9, with a batch size of 8. During training, we divide the learning rate by 2 for every 50 epochs, for a total of 150 epochs (except for the weight uncertainty method which drops the learning rate after every 25 epochs). During training, we discovered that the weight uncertainty method [6] is not robust to learning rate. Therefore, to compare fairly to this method, we drop the learning rate significantly to avoid premature saturation of the gradients.

Results. Table 2 shows experimental results across all architectures, and across all loss function weighting schemes. For each architecture and weighting scheme combination, the best performance is highlighted in bold. By comparing validation results, our method outperforms all other methods, across all weighting schemes, and across both tasks. Moreover, our method has two key advantages. First, due to the efficiency of having a single shared feature pool with attention masks automatically learning which features to share, our method outperforms other methods without requiring extra parameters (column #P), and even with significantly fewer parameters in some cases. Second, our method is more robust to the choice of weighting scheme than other methods, and does not require cumbersome tweaking of loss weights. We can also see that our proposed DWA weighting method performs best across most of the baselines, whereas the uncertainty method [6] appears to overfit by displaying good training performance, but poorer validation performance.

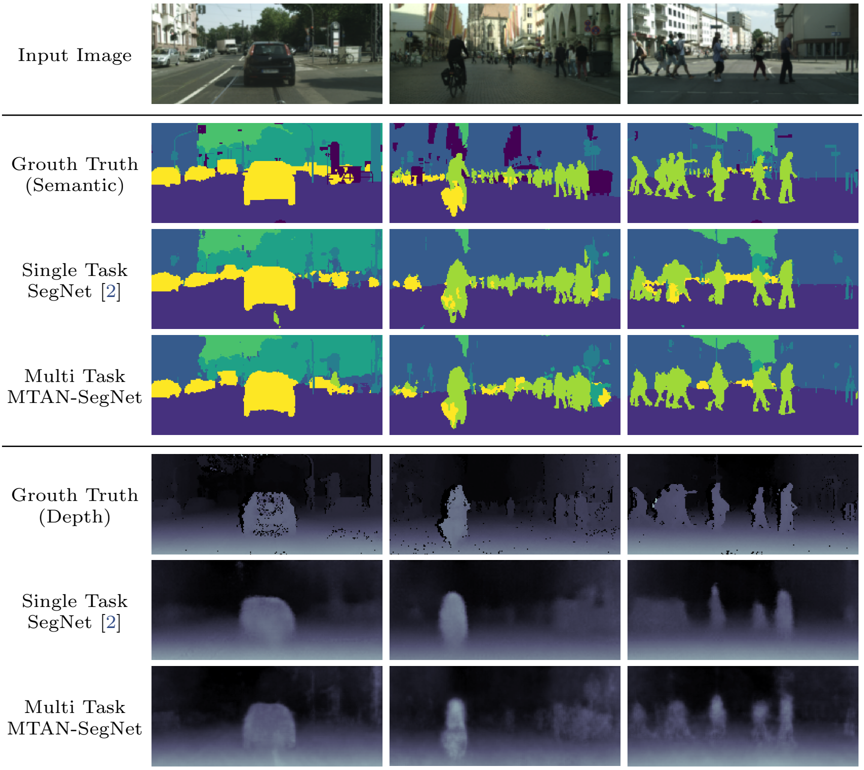
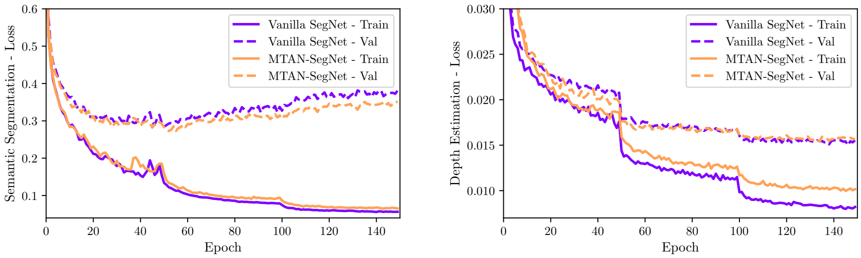


Fig. 4: CityScapes dataset validation results on 7-class semantic labelling and depth estimation all trained with equal weighting. The original images are cropped to avoid invalid points for better visualisation.

To further investigate generalisation performance of our method compared to the single task SegNet, in Figure 5 we plot the learning curve with respect to the loss of both tasks. We can clearly see that our network is able to alleviate overfitting (reduces the gap between training and validation) compared to single task training, and produces a better generalisable feature representation. Figure 4 then shows qualitative results and comparison. We can see the advantage of multi-task learning particularly for depth estimation, where the edges of objects are clearly more pronounced compared with single-task training.



(a) Semantic Loss Curve

(b) Depth Loss Curve

Fig. 5: Learning curves of semantic and depth loss between vanilla SegNet [2] and our network MTAN-SegNet.

4.5 Attention Masks as Feature Selectors

To understand the role of the proposed attention modules, in Figure 6 we visualise the layer 1 attention masks learned with our network. We can see a clear difference in attention masks between the two tasks, with each mask working as a feature selector to mask out uninformative parts of the shared features, and focus on parts which are either task-specific, or task-shared. In particular, the attention masks have strong similarity to the shared features, and thus appear to act as a feature augmentation, whereas the attention maps for depth estimation appear to act as sparse feature extractors.

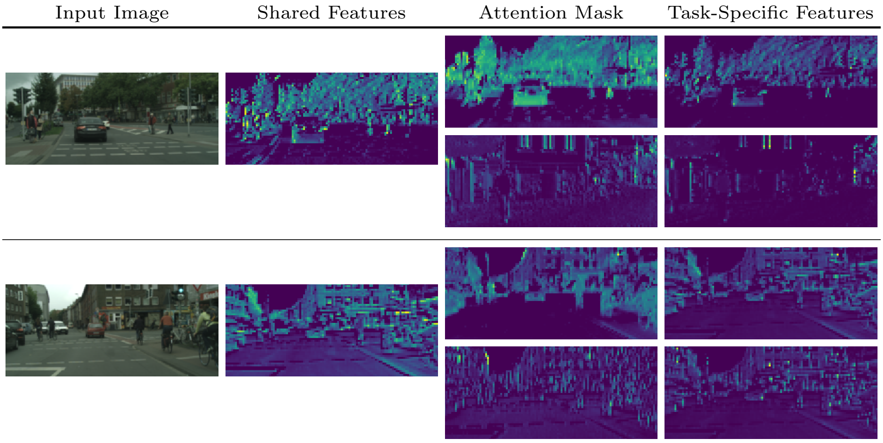


Fig. 6: Visualisation of the first layer of 7-class semantic segmentation and depth estimation attention features of our proposed network. Top row: semantic features; Bottom row: depth features. The colours for each image are rescaled to fit the data.

4.6 Effect of Task Complexity

For further introspection into the benefits of multi-task learning, we further evaluate our network on training data with different numbers of semantic classes, leaving the depth labels the same across all experiments. We trained the network with the same settings as in Section 4.4, but with all networks having equal loss weighting.

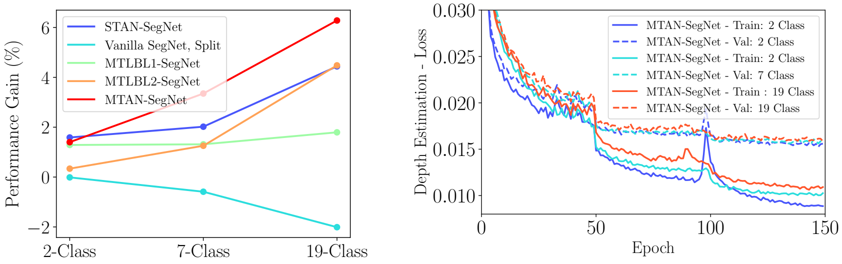
Results. Table 3 shows validation results for these experiments. Note that in single task depth estimation, changing semantic class will have no effect so we leave this part of the table blank. In Figure 7a, the performance gain of all multi-task methods (including our own), compared to the single-task SegNet [2], is shown.

There are three interesting findings in this experiment. First, we observe that the multi-task performance gain (over the single task network) increases as the number of semantic classes increases. In fact, for only a 2-class setup, the single-task framework performs best. However, for greater task complexity, the multi-task framework encourages the sharing of features, for a more efficient use of available network parameters, which then leads to better results. Second, the complimentary depth estimation task performs better when it trains with more semantic classes, owing to the greater provision of complementary information from which shared features can be learned for generalisation, further supporting the benefit of multi-task learning. Third, Figure 7b then shows that for multi-task learning, greater task complexity causes the gap between validation loss and training loss to actually decrease, and hence overfitting to decrease, due to the ability to share features across tasks using the supervised labels.

Table 3: 2/7/19-class semantic segmentation and depth estimation results trained with equal weighting on validation set of CityScapes dataset. We measure semantic segmentation as mean intersection-over-union (mIoU) (the higher the better) and depth estimation as relative absolute error (RAE) (the lower the better).

Type	Method	Semantic (mIoU)			Depth (RAE)		
		2-class	7-class	19-class	2-class	7-class	19-class
Single Task	Vanilla SegNet [2]	0.8004	0.5097	0.2675	-	-	-
	STAN - SegNet	**0.8127**	0.5200	0.2794	-	-	-
Multi Task	Vanilla SegNet, Split	0.7999	0.5067	0.2621	**0.5289**	0.5666	0.4748
	MTLBL1-SegNet	0.8103	0.5164	0.2723	0.5419	0.5166	0.4893
	MTLBL2-SegNet	0.8027	0.5161	0.2795	0.6204	0.5019	0.4700
	MTAN-SegNet	0.8112	**0.5268**	**0.2843**	0.6067	**0.4246**	**0.4513**

As a deeper look into the performance change in MTAN-SegNet compared to single task SegNet, we further provide performance change in Figure 8 of our method compared to the single-task SegNet. Here, the categories are sorted in increasing order of the mean percentage image coverage. We can see our network outperform the single task baseline in 16 out of 19 classes. In particular, there are the greatest performance boosts in small to mid sizes of classes, whilst classes



(a) Performance Gain

(b) Depth Loss Curve

Fig. 7: (a) Performance gain in percentage from all single-task and multi-task baseline methods compared with results produced SegNet. (b) Depth loss from training and validation set in our network.

with large image coverage tend to already perform well with single-task learning. This further highlights the power of multi-task learning when laballed data is scarce, but suggests that when sufficient data is available, single-task learning can still be effective.

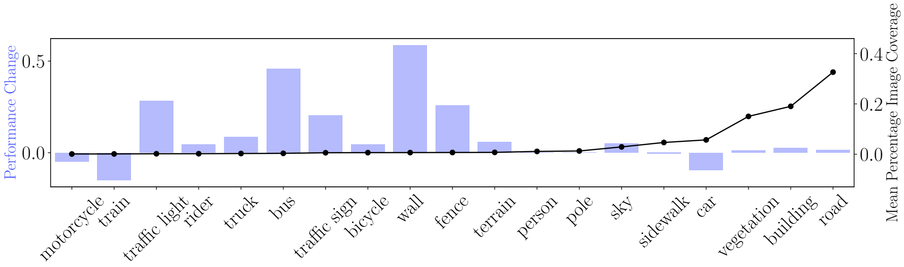


Fig. 8: Performance comparison between MTAN-SegNet and SegNet trained in each 19 dense class. We report performance gain in bar chart (purple) sorted in mean percentage image coverage (black) for each class.

5 Conclusions

In this work, we have presented a new method for multi-task learning, the Multi-Task Attention Network (MTAN). This consists of a global feature pool, together with task-specific attention modules for each task, which allows for automatic learning of both task-shared and task-specific features. Experiments on the CityScapes dataset for the tasks of semantic segmentation and depth estimation, show our method to outperform competing methods across both tasks, whilst also showing robustness to the particular weighting scheme used in the loss function. Due to our method's ability to share weights through attention masks, our method is parameter efficient, and the architecture is simple to implement and train. We have also shown that as task complexity increases, the performance of multi-task learning increases at a greater rate than that of single-task learning, and our proposed method demonstrates this behaviour even more strongly than other multi-task learning methods.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
2. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561 (2015)
3. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Advances in neural information processing systems. (2012) 1097–1105
4. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision, Springer (2016) 694–711
5. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 3994–4003
6. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. arXiv preprint arXiv:1705.07115 (2017)
7. Doersch, C., Zisserman, A.: Multi-task self-supervised visual learning. In: The IEEE International Conference on Computer Vision (ICCV). (Oct 2017)
8. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
9. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 3213–3223
10. Caruana, R.: Multitask learning. In: Learning to learn. Springer (1998) 95–133
11. Evgeniou, T., Pontil, M.: Regularized multi–task learning. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2004) 109–117
12. Kumar, A., Daume III, H.: Learning task grouping and overlap in multi-task learning. arXiv preprint arXiv:1206.6417 (2012)
13. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10) (2010) 1345–1359
14. Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S.: Transfer feature learning with joint distribution adaptation. In: Proceedings of the IEEE international conference on computer vision. (2013) 2200–2207
15. Thrun, S., Pratt, L.: Learning to learn. Springer Science & Business Media (2012)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
17. Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P., Bengio, S.: Why does unsupervised pre-training help deep learning? Journal of Machine Learning Research **11**(Feb) (2010) 625–660
18. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Learning multiple visual domains with residual adapters. arXiv preprint arXiv:1705.08045 (2017)
19. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2650–2658

20. Gkioxari, G., Hariharan, B., Girshick, R., Malik, J.: R-cnns for pose estimation and action detection. arXiv preprint arXiv:1406.5212 (2014)
21. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification. arXiv preprint arXiv:1704.06904 (2017)
22. Kokkinos, I.: Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
24. Chen, Z., Badrinarayanan, V., Lee, C.Y., Rabinovich, A.: Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. arXiv preprint arXiv:1711.02257 (2017)
25. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European Conference on Computer Vision, Springer (2016) 630–645
26. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Transactions on pattern analysis and machine intelligence **30**(2) (2008) 328–341
27. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)