TASK-BASED TOP-DOWN MODULATION NETWORK FOR MULTI-TASK-LEARNING APPLICATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

A general problem that received considerable recent attention is how to perform multiple tasks in the same network, maximizing both efficiency and prediction accuracy. A popular approach consists of a multi-branch architecture on top of a shared backbone, jointly trained on a weighted sum of losses. However, in many cases, the shared representation results in non-optimal performance, mainly due to an interference between conflicting gradients of uncorrelated tasks. Recent approaches address this problem by a channel-wise modulation of the feature-maps along the shared backbone, with task specific vectors, manually or dynamically tuned. Taking this approach a step further, we propose a novel architecture which modulate the recognition network channel-wise, as well as spatial-wise, with an efficient top-down image-dependent computation scheme. Our architecture uses no task-specific branches, nor task specific modules. Instead, it uses a top-down modulation network that is shared between all of the tasks. We show the effectiveness of our scheme by achieving on par or better results than alternative approaches on both correlated and uncorrelated sets of tasks. We also demonstrate our advantages in terms of model size, the addition of novel tasks and interpretability.
Code will be released.

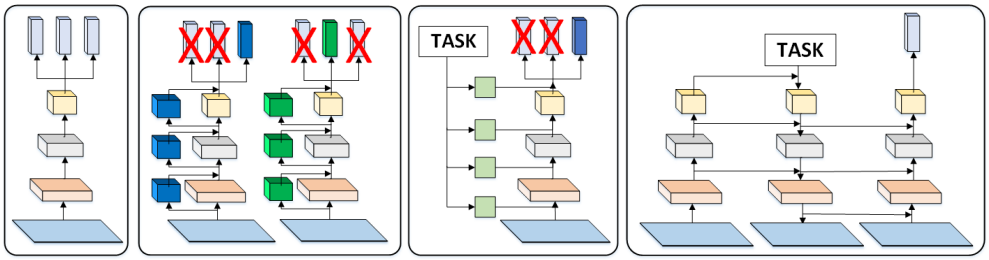
1 INTRODUCTION

The goal of multi-task learning is to improve the learning efficiency and increase the prediction accuracy of multiple tasks learned and performed together in a shared network.

Over the years, several types of architectures have been proposed to combine multiple tasks training and evaluation. Most current schemes assume task-specific branches, on top of a shared backbone (Figure 1a) and use a weighted sum of tasks losses, fixed or dynamically tuned, to train them (Chen et al., 2017; Kendall et al., 2018; Sener & Koltun, 2018). Having a shared representation is more efficient from the standpoint of memory and sample complexity and can also be beneficial in cases where the tasks are correlated to each other (Maninis et al., 2019). However, in many other cases, the shared representation can also result in worse performance due to the limited capacity of the shared backbone and interference between conflicting gradients of uncorrelated tasks (Zhao et al., 2018). The performance of the multi-branch architecture is highly dependent on the relative losses weights and the task correlations, and cannot be easily determined without a "trial and error" phase search (Kendall et al., 2018).

Another type of architecture (Maninis et al., 2019) that has been recently proposed uses task specific modules, integrated along a feed-forward backbone and producing task-specific vectors to modulate the feature-maps along it (Figure 1b). Here, both training and evaluation use a single tasking paradigm: executing one task at a time, rather than getting all the task responses in a single forward pass of the network. A possible disadvantage of using task-specific modules and of using a fixed number of branches, is that it may become difficult to add additional tasks at a later time during the system life-time. Modulation-based architectures have been also proposed by Strezoski et al. (2019) and Zhao et al. (2018) (Figure 1c). However, all of these works modulate the recognition network channel-wise, using the same modulation vector for all the spatial dimension of the feature-maps.

We propose a new type of architecture with no branching, which performs single task at a time but with no task-specific modules (Figure 1d). The core component of our approach is a top-down (TD)



(a) (b) (c) (d)

Figure 1: (a) Multi branched architecture, task specific branches on a top of a shared backbone, induces capacity and destructive interference problems, force careful tuning. Recently proposed architectures: (b) using tasks specific modules and (c) using channel-wise modulation modules. (d) **Our architecture:** a top-down image-aware full tensor modulation network with no task specific modules.

modulation network, which carries the task information in combination with the image information, obtained from a first bottom-up (BU1) network, and modulates a second bottom-up (BU2) network common for all the tasks. In our approach, the modulation is channel-wise as well as spatial-wise (a full tensor modulation), calculated sequentially along the TD stream. This allows us, for example, to modulate only specific spatial locations of the image depending on the current task, and get interpretability properties by visualizing the activations in the lowest feature-map of the TD stream. In contrast to previous works, our modulation mechanism is also "image-aware" in the sense that information from the image, extracted by the BU1 stream, is accumulated by the TD stream, and affects the modulation process.

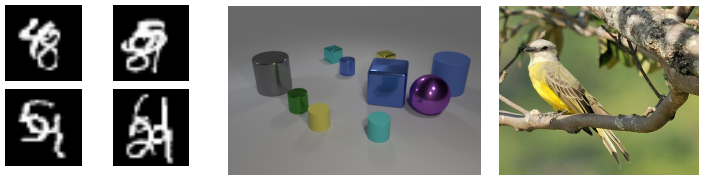
The main differences between our approach and previous approaches are the following: First, as mentioned, our approach does not use multiple branches or task-specific modules. We can scale the number of tasks with no additional layers. Second, our modulation scheme includes a spatial component, which allows attention to specific locations in the image, as illustrated in figure 2a for the Multi-MNIST tasks (Sabour et al., 2017). Third, the modulation in our scheme is also image dependent and can modulate regions of the image based on their content rather than location (relevant examples are demonstrated in figures 2b and 2c).

We empirically evaluated the proposed approach on three different datasets. First, we demonstrated on par accuracies with the single task baseline on an uncorrelated set of tasks with MultiMNIST while using less parameters. Second, we examined the case of correlated tasks and outperformed all baselines on the CLEVR (Johnson et al., 2017) dataset. Third, we scaled the number of tasks and demonstrated our inherent attention mechanism on the CUB200 (Welinder et al., 2010) dataset. The choice of datasets includes cases where the tasks are uncorrelated (Multi-MNIST) and cases where the tasks are relatively correlated (CLEVR and CUB200). The results demonstrate that our proposed scheme can successfully handle both cases and shows distinct advantages over the channel-wise modulation approach.

2 Related Work

Our work draw ideas from the following research lines:

Multiple Task Learning (MTL) Multi task learning has been used in machine learning well before the revival of deep networks (Caruana, 1997). The success of deep neural networks in single task performance (e.g. in classification, detection and segmentation) has renewed the interests of the computer vision community in the field (Kokkinos, 2017; He et al., 2017; Redmon & Farhadi, 2017). Although our primary application area is computer vision, multi task learning has also many application in other fields like natural language processing (Hashimoto et al., 2016; Collobert & Weston, 2008) and even across modalities (Bilen & Vedaldi, 2016). We further refer the interested reader to a review that summarizes recent work in the field (Ruder, 2017).



(a) M-MNIST examples (b) CLEVR example (c) CUB200 example

Figure 2: Images examples with their corresponding tasks. Our architecture benefits from its built-in image-aware, task dependent, localized attention mechanism. (a) M-MNIST examples, the tasks are to recognize the digits by their location. (b) CLEVR example, an example task is to determine whether there is a sphere to the right of a cylinder. (c) CUB200 example, an example task is to classify the bird's neck color.

Over the years, several types of architectures have been proposed in computer vision to combine the training and evaluation of multiple tasks. First works used several duplications (as many as the tasks) of the base network, with connections between them to pass useful information between the tasks (Misra et al., 2016; Rusu et al., 2016). These works do not share computations and cannot scale with the tasks. More recent architectures, which are in common practice these days, assume task-specific branches on a top of a shared backbone, and use a weighted sum of losses to train them. The joint learning of several tasks has proven to be beneficial in several cases (He et al., 2017) but can also decrease the results of some of the tasks due to a limited network capacity, uncorrelated gradients from the different tasks (sometimes called destructive inference) and different learning rates (Kirillov et al., 2019). A naive implementation of multi-task learning requires careful calibration of relative losses of the different tasks. To address these problem several methods have been proposed: "Grad norm" (Chen et al., 2017) dynamically tunes gradients magnitudes over time, to obtain similar learning rates of the different tasks. Kendall et al. (2018) uses a joint likelihood formulation to derive task weights based on the intrinsic uncertainty in each task. Sener & Koltun (2018) applies an adaptive weighting of the different tasks, to force a pareto optimal solution to the multi task problem.

Along an orthogonal line of research, other works suggested to add task-specific modules to be activated or deactivated during training and evaluation, depending on the task at hand. Liu et al. (2019b) suggests task specific attention networks in parallel to a shared recognition network. Maninis et al. (2019) suggests adding several types of low-weight task-specific modules (e.g. residual convolutional layers, squeeze and excitation (SE) blocks and batch normalization layers) along the recognition network. Note that the SE block essentially creates a modulation vector, to be channel-wise multiplied with a feature-map. Modulation vectors have been further used in Strezoski et al. (2019) for a recognition application and in Zhao et al. (2018) for a retrieval application and proved to decrease the destructive interference phenomena.

Our design, in contrast, does not use multi-branch architecture, nor task-specific modules. Our network is fully-shared between the different tasks. Compared to Zhao et al. (2018), we modulate the feature-maps in the recognition network channel-wise as well as spatial-wise, depending on both the task and the specific image at hand.

Top-Down Modulation Networks Neuroscience research provides evidence for a top-down context, feedback and lateral processing in the primate visual pathway (Gazzaley & Nobre, 2012; Gilbert & Sigman, 2007; Lamme et al., 1998; Hopfinger et al., 2000; Piëch et al., 2013; Zanto et al., 2010) where top-down signals modulate the neural activity of neurons in lower-order sensory or motor areas based on the current goals. This may involve enhancement of task-relevant representations or suppression for task-irrelevant representations. This mechanism underlies humans ability to focus attention on task-relevant stimuli and ignore irrelevant distractions (Hopfinger et al., 2000; Piëch et al., 2013; Zanto et al., 2010).

In this work, consistent with this general scheme, we suggest a model that uses top-down modulation in the scope of multi-task learning. Top down modulation networks with feedback, implemented as conv-nets, have been suggested by the computer vision community for some high level tasks (e.g. re-classification (Cao et al., 2015), keypoints detection (Carreira et al., 2016; Newell et al., 2016),

crowd counting (Sam & Babu, 2018), curriculum learning (Zamir et al., 2017) etc.) and here we apply them to multi-task learning applications.

3 APPROACH

We will first describe the vector modulation mechanism used in Zhao et al. (2018), then describe our architecture by details.

3.1 VECTOR MODULATION MECHANISM

A vector modulation of a given tensor X is defined as the multiplication between the elements of a vector w and the corresponding channel of the tensor X. Each element in the output tensor Y is defined by:

$$Y(y, x, ch) = X(y, x, ch) * w(ch) \tag{1}$$

where X is the tensor to be modulated and Y is the modulated tensor, both in the form $(HW \times C)$ where H, W, are the spatial dimensions of the tensors and C is their channel dimension. $w \in \mathbb{R}^C$, corresponding in size to the channel dimension of X and Y.

We further define a gated modulation module with a residual connection as:

$$Y(y, x, ch) = X(y, x, ch) + X(y, x, ch) * tanh(w(ch)) = X(y, x, ch) \otimes w(ch) \tag{2}$$

where the modulation vector w is gated with a *tanh* function before the multiplication and then added to the input tensor X through a residual connection. Unless stated otherwise we use the gated modulation as defined in Eq. 2 in all of our experiments. For simplicity we denote this operation by the symbol $\otimes$.

In the scope of Multi Task Learning, where several tasks exist, the modulation module is represented by a matrix $W \in \mathbb{R}^{C \times T}$ rather by a vector $w \in \mathbb{R}^C$, where T is the number of tasks.

Performing one task at a time, the current task, $t \in [1, T]$, is an input to the system and the modulated tensor Y depend on the specific t:

$$Y_t(y, x, ch) = X(y, x, ch) * W(ch, t) \tag{3}$$

Here, the rows of W are the modulation vectors for each task and W is end-to-end trainable. This modulation module has been used in Zhao et al. (2018), separately to every layer in the recognition network. The disadvantages of this module are that it ignores the spatial dimensions of the image and lack information from the image itself. Using the same strategy to explicitly optimize spatial-aware modulation tensors $\mathbf{W}(y, x, ch, t)$ was discussed in Zhao et al. (2018) but claimed it to be unfeasible due to their large dimensions. Our method address both of these issues in an efficient computational model.

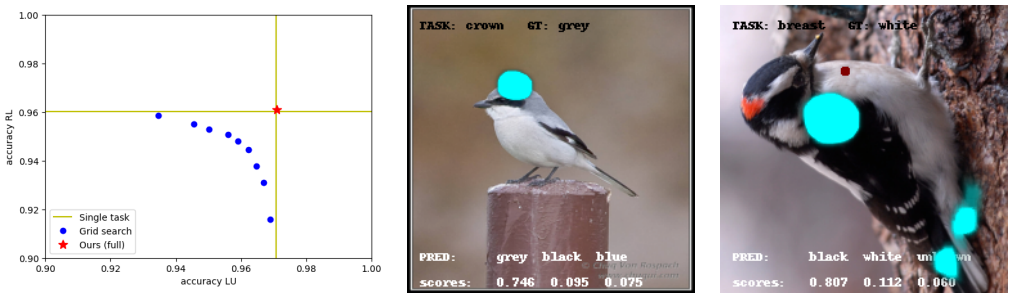
3.2 IMAGE-AWARE TASK-DEPENDENT TOP-DOWN TENSOR MODULATION

Our architecture uses image-aware, task dependent tensor modulation:

$$Y_t(y, x, ch) = X(y, x, ch) \otimes \mathbf{W}_{I,t}(y, x, ch)) \tag{4}$$

Where $\mathbf{W} \in \mathbb{R}^{HW \times C}$ is a modulation tensor that depends on the current task t and on the current image I and Y is the result of the element-wise multiplication between X and the gated $\mathbf{W}$ further passed through a residual connection.

To avoid the unfeasible computation burden of directly optimizing $\mathbf{W}$, we propose calculating $\mathbf{W}$ sequentially along a dedicated top-down (TD) stream in a coarse to fine manner by using standard



(a) (b) (c)

Figure 3: (a) Multi-MNIST accuracy scatter plot, top-right is better. (b), (c) A close look into the network decision making process; (b) good example: the crown area is well localized and the prediction follows the ground truth, (c) an error example: the breast isn't localized well enough.

convolutional and upsampling layers, to create feature-maps in the wanted spatial and channel dimensions. Practically, we optimize the parameters of the convolutional layers instead of directly optimizing the elements in $\mathbf{W}$. Calculating $\mathbf{W}$ in this manner also keeps the 2d-grid structure of the spatial information on the sequentially growing feature-maps along the TD stream. The inputs to the TD stream are the embedding of the current task t, and the feature-maps along the first bottom-up stream (BU1), added to the TD stream by lateral connections. The outputs of the TD stream are its feature-maps, which sequentially modulate the tensors along the recognition network (BU2) as describe in equation 4. The last of these feature-maps can optionally be used to attend regions of interest in the image, both in training mode (with an appropriate auxiliary loss) and in evaluation mode (to visualize the attention maps).

4 Databases and Experiments

4.1 Databases

We validate our approch on three different datasets:

MultiMNIST MultiMNIST (Sabour et al., 2017) is a multi-task learning version of the MNIST dataset in which multiple MNIST images are placed on the same image. We use 2, 3 and 4 classes experiments built as suggested by Sener & Koltun (2018). Several examples are demonstrated in Figure 2a. In the 2-classes experiment the tasks are: classifying the digit on the top-left (task-LU) and classifying the digit on the bottom-right (task-RL). We use 60K examples and directly apply LeNet (LeCun et al., 1998) as the underlying backbone in our experiments.

CLEVR CLEVR is a synthetic dataset, consists of 70K training images and 15K validation images, mainly used as a diagnostic dataset for VQA. The dataset includes images of 3D primitives, with multiple attributes (shape, size, color and material) and a set of corresponding (question-answer) tuples. We followed the work Liu et al. (2019a), which suggested to use CLEVR not as a VQA dataset, but rather as a referring expression dataset, and further adapt it to a multi-task learning methodology. The tasks in our setup consist of 4 questions ("Are there exactly two cylinders in the image?", "Is there a cube right to a sphere?", "Is there a red sphere?" and "Is the leftmost sphere in the image large?"), arbitrarily chosen, with various compositionary properties.

CUB200 is a fine grained recognition dataset that provides 11,788 bird images (equally divided for training and testing) over 200 bird species with 312 binary attribute annotations, most of them referring to the colors of specific birds' parts. In contrast to other work (Strezoski et al., 2019) that used all of the 312 attributes as a yes/no question, we re-organized the attributes as a multi task problem of 12 tasks (for 12 annotated bird's parts) each with 16 classes (the annotated colors + an unknown class). To demonstrate our interpretability capability, we further used the parts' location, annotated by a single point to each seen part, as an auxiliary target at the end of the TD stream.

Table 1: Performance on Multi-MNIST, uncorrelated tasks, higher is better. Our architecture achieves on-par accuracies to the single task baseline while using much less parameters. Enlarging the number of tasks costs no additional hardware.

	ALG	$\#P$	LU accuracy	RL accuracy	LL accuracy	RU accuracy
2 tasks	Single task	x2	97.07	96.02		
	Uniform scaling	x1.12	95.91	94.80		
	MOO	x1.12	95.66	95.19		
	ch-mod	x1.002	96.49	95.24		
	Ours	x1.29	**97.09**	**96.11**		
3 tasks	Single task	x3	**90.88**	**90.65**	83.19	
	Uniform scaling	x1.25	88.12	87.99	79.01	
	MOO	x1.25	87.43	87.15	79.69	
	ch-mod	x1.005	89.18	88.83	80.71	
	Ours	x1.31	**90.88**	90.50	**83.20**	
4 tasks	Single task	x4	**95.97**	92.86	93.36	94.74
	Uniform scaling	x1.37	92.59	88.81	88.38	91.84
	MOO	x1.37	93.26	90.50	89.91	92.57
	ch-mod	x1.007	93.54	90.72	90.19	92.03
	Ours	x1.32	95.79	**94.17**	**93.94**	**95.10**

4.2 EXPERIMENTS

architecture We use LeNet, VGG-11 and resnet-18 as our backbone BU architectures for the Multi-MNIST, CLEVR and CUB-200 experiments correspondingly. Each of the backbones has been divided to two parts; a first part that consists of the convolutional layers of the backbone and a second part with the fully connected layers (the classifier).

In our architecture, both BU streams consist of the first part of the backbone and share their weights. The TD stream, unless specified otherwise, is a replica of the BU stream, in terms of layers structure and number of channels, combined with upsampling layers. The classifier is only attached to the BU2 stream. Information is passed between the BU1, TD and BU2 streams using lateral connections implemented as 1x1 convolutions. A task embedding layer (a fully connected layer) is added on the top of the TD stream. See an illustration of the full scheme in figure 1d.

baselines We compare our method both to a "single task" approach, where each task is independently solved and to a "uniform scaling" approach, where a uniformly weighted sum of the individual losses is being minimized. In most of our experiments we have also compared our architecture to "ch-mod", a channel-wised vector modulation architecture (Zhao et al., 2018) and to a MOO (multi objective optimization approach) where the weights of loss items are dynamically tuned as suggested by Sener & Koltun (2018).

4.3 RESULTS

4.3.1 MULTIMNIST

We use the Multi-MNIST dataset to demonstrate our performance in case of uncorrelated tasks for 2, 3, and 4 tasks recognition problems with no additional hardware. All models trained using a standard LeNet architecture. We used a batch size of 512 images trained on 1 GPU with learning rate of $1e^{-3}$.

Figure 3a visualize the performance profile of the 2-classes experiment as a scatter plot of accuracies on task-LU and task-RL for the single task approach (vertical and horizontal lines correspondingly) and the multi-branched approach for several manually tuned loss weights (the blue dots). The scatter plot demonstrate a capacity problem, where better accuracies (above a certain limit) in one task

Table 2: Ablations on Multi-MNIST.

(a) number of channels

#ch	#P	LU ac	RL ac
dup	x1.29	97.09	96.11
10	x1.25	97.20	96.18
6	x1.10	96.88	96.05
2	x1.02	96.59	95.55
1	x1.01	96.30	95.59

(b) connectivity type

td	bu2	LU ac	RL ac
+	+	96.55	95.60
+	x	**96.99**	96.14
+	**+x**	**97.09**	**96.11**
x	+x	96.43	95.39
+x	+x	96.63	96.07

(c) laterals existence

	LU ac	RL ac
all	97.09	96.11
lat 3	96.62	95.77
lat 2	96.88	95.88
lat 1	96.65	95.69

(d) auxiliary losses

	LU ac	RL ac
bu2	97.09	96.11
bu2+bu1	96.75	95.71
bu2+td	96.82	95.56
bu2+td+bu1	96.84	95.53

cannot be achieved without being reflected as lower accuracies on another task. Our results are marked as a red star, showing equal accuracies to the single-task case.

Table 1 summarizes our results on the Multi-MNIST experiment while sequentially enlarging the number of tasks. Our method achieves on par results with the single-task baseline while using much less parameters (the third column shows the number of parameters as a multiplier of the number of parameter in a standard Lenet architecture). Other approaches, including the channel-wise modulation approach, achieve lower accuracy rates. Enlarging the number of tasks keeps this accuracy gap while decreases the ratio between the number of parameters in our scheme and the number of parameters in the single task or uniform scaling schemes.

4.3.2 Ablations on Multi-MNIST

We further conduct ablation studies on Multi-MNIST, to emphasis various aspects of our proposed architecture. Table 2 shows the ablation results, analyzed as follows:

Number of channels in the TD stream. Table 2a compares the results accuracies of our proposed architecture (first line, where the TD stream is a replica of the BU streams with 1, 10 and 20 channels in its feature-maps) with cheaper architectures with a different number of channels along the layers in the TD stream. Our experiments show a trend line (the accuracies decrease while the number of channels in the TD stream decrease) but also that optimizing the number of channels along the TD stream in terms of efficiency-accuracies tradeoff can be done. In this case increasing the number of channels in the lower feature-maps of the TD stream (10 instead of 1, second row in the table) results in better accuracies even while decreasing the number of channels of the higher feature-map in the TD stream (10 instead of 20).

Connectivity type. Our architecture uses two sets of lateral connections; the first set pass information between the BU1 stream to the TD stream and the second set pass information from the TD stream to the BU2 stream. Table 2b compares the results accuracies of our proposed architecture while using different connectivity types to the TD stream (first column) and to the BU2 stream (second column). Here $+$ is an addition connectivity, x is a multiplication connectivity and $+\otimes$ is a gated modulation with residual connection as described in Equation 2. The table shows higher accuracies while using addition connectivity on the TD stream and gated modulation connectivity on the BU2 stream. We further use these connectivity types in all of our experiments.

Lateral relative importance. Our architecture modulate the recognition network using 3 lateral connections. Table 2c shows the results accuracies of our proposed architecture while using only one lateral connection at a time. Using all the 3 lateral connections yields better accuracies than using any of them separately. On the other hand, using our architecture with only one lateral connection still yields better accuracies than using the Uniform scaling approach. The table also demonstrate that modulating lower feature-maps (third line) might yield better accuracies than modulate the embedding space only (second line).

Auxiliary losses. Our architecture, although mainly uses only one classification loss at the end of the BU2 stream, can be easily adapted to integrating two auxiliary losses, one at the end of the BU1 stream and the other on the image plane at the end of the TD stream. Table 2d shows no additional improvement while using these auxiliary losses on Multi-MNIST. Note that a TD auxiliary localization loss may be also used as a way to add interpretability capability to our scheme. We use this capability on the CUB200 experiment.

Table 3: Performance on CLEVR, higher is better. Our approach yields better accuracies also on correlated set of tasks with no additional hardware as tasks are added. Better accuracies are demonstrated both compared to the single task and uniform scaling approaches while using less parameters.

	ALG	$\#P$	que1 accuracy	que2 accuracy	que3 accuracy	que4 accuracy
2 tasks	Single task	x2	97.81	97.95		
	Uniform scaling	x1.5	98.09	**98.19**		
	ch-mod	x1.001	97.91	97.45		
	Ours	x1.56	**98.17**	**98.19**		
3 tasks	Single task	x3	96.92	97.81	**99.93**	
	Uniform scaling	x2	97.67	97.73	99.89	
	ch-mod	x1.001	97.01	97.55	99.90	
	Ours	x1.56	**98.25**	**97.92**	**99.93**	
4 tasks	Single task	x4	96.73	97.93	**99.94**	98.64
	Uniform scaling	x2.5	97.47	97.93	99.92	98.64
	Ours	x1.56	**98.43**	**98.16**	99.93	**98.66**

4.3.3 CLEVR

We used the CLEVR dataset to show our performance in case of correlated tasks (the questions on CLEVR are correlated) and to demonstrate our ability to enlarge the number of tasks with no extra hardware while keeping the targets accuracies.

Our results are summarized in Table 3. We trained all models using a VGG-11 architecture but decreased the number of channels in the output of the last convolutional layer from 512 to 128 to allow training with larger batch size. We used a batch size of 128 images trained on 2 GPUs with learning rate of $1e^{-4}$ using the Adam optimizer.

Table 3 shows that our results are better than both single task and uniform scaling approach while using much less parameters (the third columns shows the number of parameters of each architecture as a multiplier of the number of parameters in a single task VGG-11 backbone). Here, the channel-wise modulation approach uses the smallest number of parameters but also gets the worst results. The table also shows that enlarging the number of tasks (with no additional hardware) is not only feasible but also may improve the results of each task separately. We further note that we used a TD layers that are a replica of the VGG-11 BU layers. Further reducing the number of parameters by decreasing the channel dimensions in the TD stream can be easily done but is not our main scope in this work.

4.3.4 CUB 200

We used the CUB-200 dataset to further demonstrate our performance on correlated tasks in real-world images, scaling the number of tasks and using another type of backbone architecture (a Resnet backbone). In contrast to previous experiments, we did not aim at reducing the number of parameters (since we are using a Resnet backbone); rather we demonstrate better performance, and our built-in capability to visualize the attention maps at the end of the TD stream.

We trained all models using a Resnet-18 architecture. We used a batch size of 128 images trained on 2 GPUs with learning rate of $1e^{-4}$ using the Adam optimizer for 200 epochs. During training we add an auxiliary loss at the end of the TD stream. The target in this case is a one-hot 224x224 mask, where only a single pixel is labeled as foreground, blurred by a Gaussian kernel. During training, for each visible ground-truth annotated task/part, we minimize the cross-entropy loss over the 224x224 image at the end of the TD stream softmax output (which encourages a small detected area). Figures 3b and 3c demonstrate the advantages of this visualization, offering a close look into the reasons behind the decision making process of the network. Figure 3b is an example where the mask is well localized on the crown of the bird (the task) and the color is correctly predicted. Figure

Table 4: Performance on CUB200, higher is better. Our architecture is scalable with the number of tasks and outperforms other methods. All models trained for 200 epochs with lr 1e-4 using a resnet-18 backbone.

	wing	**uppertail**	**throat**	**nape**	**leg**	**eye**	**back**	**breast**	**forehead**	**belly**	**crown**	**bill**	**mean**
Single task	77.91	75.09	73.46	70.99	62.82	89.66	75.54	75.30	71.07	77.63	71.82	70.26	74.30
Uniform scaling	81.01	79.46	77.55	75.94	64.57	90.40	79.46	78.67	74.58	80.62	75.25	72.07	77.46
MOO	82.78	82.17	77.37	75.66	64.84	91.39	81.15	79.63	75.03	81.57	75.44	73.80	78.40
ch-mod	82.07	81.72	80.03	77.20	68.61	91.30	81.29	81.22	76.91	82.29	77.87	76.23	79.72
Ours	84.90	81.77	81.01	79.53	67.60	91.08	83.22	83.64	78.55	84.41	79.96	75.61	80.94

3c demonstrate an error case where the breast of the bird is not well localized by the mask and as a consequence the color is wrongly predicted.

Our quantitative results are summarized in Table 4. The results show better accuracies of our scheme compared to all baselines. We also show better accuracy compared with the channel-wise modulation scheme indicating of the advantages of our image-aware top-down modulation architecture.

5 SUMMARY

We proposed a novel architecture for multi-task learning using a top-down modulation network. Comparing with current approaches, our scheme does not use task-dependent branches or task-dependent modules, and the modulation process is executed spatial-wise as well as channel-wise, guided by the task and by the information from the image itself. We tested our network on three different datasets, achieving on par or better accuracies on both correlated and uncorrelated sets of tasks. We have also demonstrated some of the inherent advantages of our scheme: adding tasks with no extra hardware that result in a decrease in the total number of parameters while scaling the number of tasks, and allowing interpretability by pointing to relevant image locations.

More generally, multiple-task learning algorithms are likely to become increasingly relevant, since general vision systems need to deal with a broad range of tasks, and executing them efficiently in a single network is still an open problem. In future work we plan to adapt our described architecture to a wider range of applications (e.g. segmentation, images generation) and examine possible combinations of approaches such as combining partial-branching strategy with our TD approach. We also plan to study additional aspects of multi-task learning such as scaling the number of tasks and tackling the catastrophic forgetting problem.

REFERENCES

Hakan Bilen and Andrea Vedaldi. Integrated perception with recurrent multi-task neural networks. In *Advances in neural information processing systems*, pp. 235–243, 2016.

Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2956–2964, 2015.

Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4733–4742, 2016.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *arXiv preprint arXiv:1711.02257*, 2017.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167. ACM, 2008.

Adam Gazzaley and Anna C Nobre. Top-down modulation: bridging selective attention and working memory. *Trends in cognitive sciences*, 16(2):129–135, 2012.

Charles D Gilbert and Mariano Sigman. Brain states: top-down influences in sensory processing. *Neuron*, 54(5):677–696, 2007.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Joseph B Hopfinger, Michael H Buonocore, and George R Mangun. The neural mechanisms of top-down attentional control. *Nature neuroscience*, 3(3):284, 2000.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2901–2910, 2017.

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*, pp. 7482–7491, 2018.

Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9404–9413, 2019.

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*, pp. 6129–6138, 2017.

Victor AF Lamme, Hans Super, and Henk Spekreijse. Feedforward, horizontal, and feedback processing in the visual cortex. *Current opinion in neurobiology*, 8(4):529–535, 1998.

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Runtao Liu, Chenxi Liu, Yutong Bai, and Alan L Yuille. Clevr-ref+: Diagnosing visual reasoning with referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4185–4194, 2019a.

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*, pp. 1871–1880, 2019b.

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*, pp. 1851–1860, 2019.

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*, pp. 3994–4003, 2016.

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pp. 483–499. Springer, 2016.

Valentin Piëch, Wu Li, George N Reeke, and Charles D Gilbert. Network model of top-down influences on local gain and contextual interactions in visual cortex. *Proceedings of the National Academy of Sciences*, 110(43):E4108–E4117, 2013.

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

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pp. 3856–3866, 2017.

Deepak Babu Sam and R Venkatesh Babu. Top-down feedback for crowd counting convolutional neural network. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

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

Gjorgji Strezoski, Nanne van Noord, and Marcel Worring. Many task learning with task routing. *arXiv preprint arXiv:1903.12117*, 2019.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Amir R Zamir, Te-Lin Wu, Lin Sun, William B Shen, Bertram E Shi, Jitendra Malik, and Silvio Savarese. Feedback networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1308–1317, 2017.

Theodore P Zanto, Michael T Rubens, Jacob Bollinger, and Adam Gazzaley. Top-down modulation of visual feature processing: the role of the inferior frontal junction. *Neuroimage*, 53(2):736–745, 2010.

Xiangyun Zhao, Haoxiang Li, Xiaohui Shen, Xiaodan Liang, and Ying Wu. A modulation module for multi-task learning with applications in image retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 401–416, 2018.

A ADDITIONAL QUALITATIVE EXAMPLES ON CUB200

To demonstrate our interpretability capabilities we trained our proposed network with an auxiliary localized cross entropy loss at the last layer of the TD stream (details in section 4.3.4). Here we present several more examples of interest we did not include in the main text.

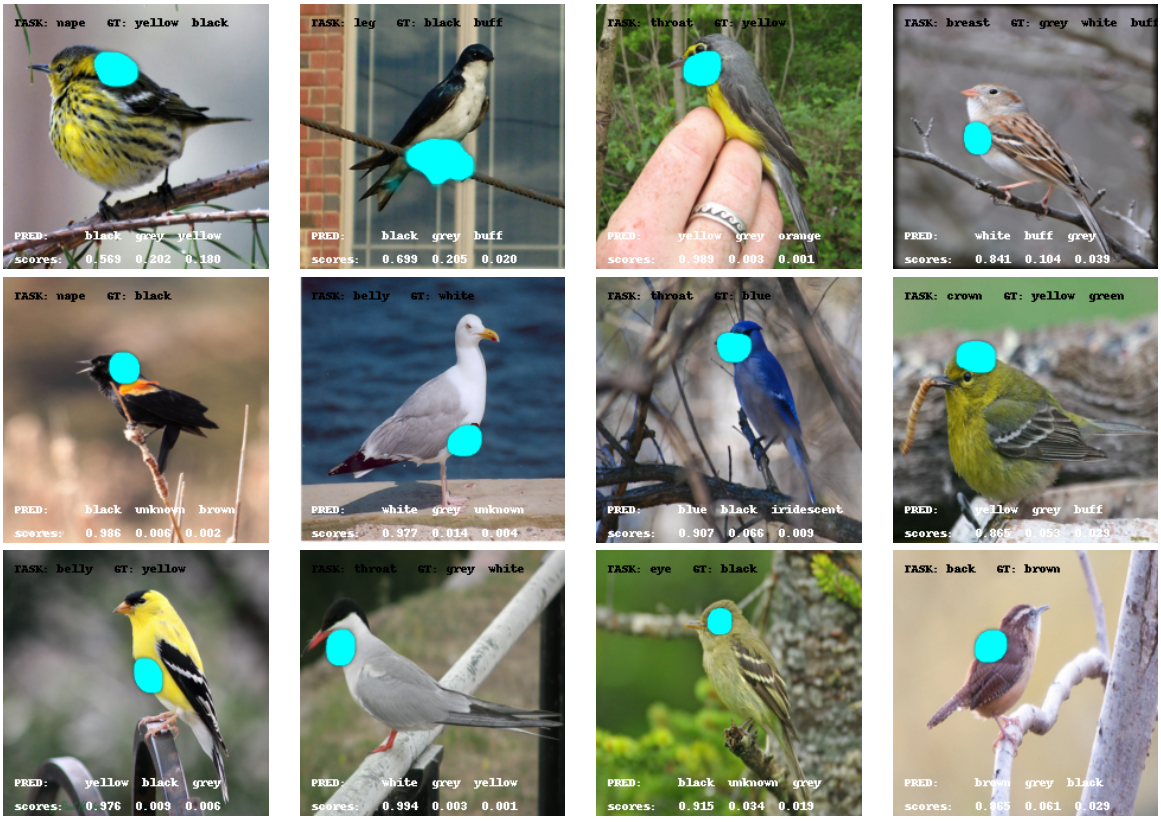


Figure 4: More qualitative examples to demonstrate our ability to identify the relevant regions that most affected the network prediction. In all of these images the target part (the task, shown in the upper part of each image), is precisely localized and the prediction (shown in the lower part of each image) follows the ground truth. Best viewed in color while zoomed in.

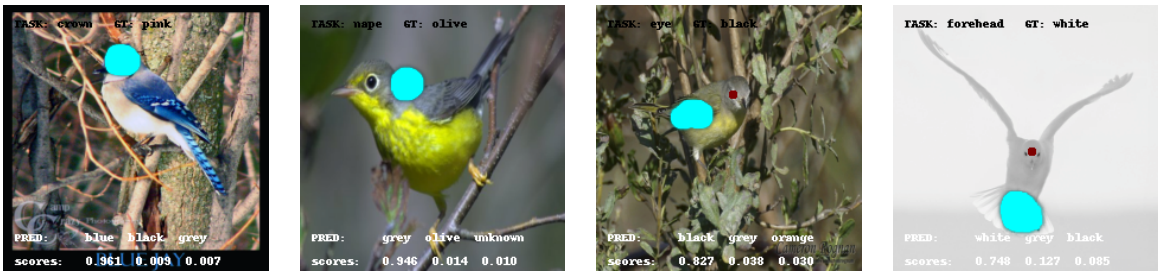


Figure 5: Error cases. Left images demonstrate good examples, counted as failure cases due to annotations errors. Our network successfully localize the asked part and correctly predict its color. Right images demonstrate bad localization examples. Ground truth classes were still predicted, with a very high score, maybe due to the correlated nature of the tasks. Best viewed in color while zoomed in.