MATTE: Multi-task multi-scale attention
Gjorgji Strezoski, Nanne van Noord, Marcel Worringer
University of Amsterdam, Informatics Institute, Multix, Amsterdam, The Netherlands

A R T I C L E I N F O
Communicated by Nikos Paragios
Keywords:
Multi-task learning
Matting
Visual Decathlon
Ubernet

A B S T R A C T
In this work, we propose a general method for learning task and scale based attention representations in Multi-Task Learning (MTL) for vision. It relies on learning and maintaining cross-task and cross-scale representations of visual information, whose interaction contributes to a symmetrical improvement across the entire task pool. Apart from learning data representations, we additionally optimize for the most beneficial interaction between tasks and their representations at different scales. Our method adds an attention modulated feature as residual information to the processing of each scale stage within the model, including the final layer of task outputs. We empirically show the effectiveness of our method through experiments with current multi-modal and multi-scale architectures on diverse MTL datasets. We evaluate MATTE on high and low level vision MTL problems, against MTL and single task learning (STL) counterparts. For all experiments we report solid performance improvements in both qualitative and quantitative performance.

1. Introduction

Behavioral science research (Szumowska et al., 2018) has shown that performing multiple related tasks simultaneously, often increases productiveness and throughput without harming performance. With similar reasoning, multi-task learning (MTL) is the ability of a computational model to optimize for multiple learning tasks in a single training cycle. MTL has been formally defined by Caruana (1997) as the approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias (Thrun, 1996; Bakker and Heskes, 2003). As a training paradigm, MTL rests on the basis of sharing information between the tasks the model is training for, and using each individual control signal (loss) towards the benefit of the overall model performance. Sharing in MTL is performed using a parameter space, called the shared learning or shared parameter space which all of the tasks the model is training for use back-propagation (Caruana, 1997). This shared parameter space is what enables a single model to simultaneously represent and perform multiple tasks with different descriptive attributes.

MTL has numerous application domains. This ranges from simple recommender systems (Ning and Karypis, 2010; Zhang et al., 2019) which need to form a ranked list out of several meta-data modalities, to complex augmented reality (AR) systems (Lampropoulos et al., 2020) which need to perform efficient scene depth and surface normal estimations, instance segmentation, object recognition and alpha matting to generate a semantically correct scene. Representing, or even estimating, instance segmentation, object recognition and alpha matting to render a semantically correct scene. Representing, or even

https://doi.org/10.1016/j.cviu.2023.103622
Received 7 September 2022; Received in revised form 24 November 2022; Accepted 3 January 2023
Available online 16 January 2023
1077-3142/© 2023 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
tasks and possible scales, we maintain the information across network layers and task specific branches until the final output. This operation fits well within the promise of MTL and definition of attention, as it enhances useful information sharing between tasks. In our attention formulation, the attention mechanism decides which channel, at which scale, gets how much resources in our shared learning space, before being subjected to the training control signal. By introducing MATTE to MTL, we preserve both the local and non-local affinity information at each scale. MATTE helps us to alleviate the effects of destructive interference, and to generate a more informative embedding before the final output layers take over.

To evaluate MATTE we conduct experiments on NYUD-v2 (Misra et al., 2016), SUN-RGBD (Song et al., 2015), KITTI (Uhrig et al., 2017), Visual Decathlon (Rebuffi et al., 2017), Ubernet (Kokkinos, 2017) and the alpha matting dataset (Rhemann et al., 2009). In the experiments conducted on these datasets we cover both high- and low-level vision tasks in an MTL model compared to STL and MTL approaches, and evaluated the impact of the influence between tasks. The experimental results show that any local and non-local affinities can be learned and propagated to form meaningful final representations of all the tasks across all available scales. Through the ablation study we also show that MATTE has a measurable influence on destructive interference and identify the distinct effect of every part of our approach. Across all experimental setups and datasets we report continuous performance improvements using MATTE in existing multi-task and multi-scale approaches.

We report our contributions in this paper in four aspects:

- We propose MATTE, an universal MTL attention method for combined cross-scale and cross-task learning that can be added to any multi-scale network architecture.
- With MATTE we measure and combat destructive interference between tasks, a known and still existing problem in MTL methods.
- We conduct extensive experiments over multiple datasets and benchmarks, reporting competitive or superior performance and qualitative findings.

2. Related work

Our experiments span the topics of semantic and instance segmentation, depth estimation, surface normal estimation, object detection and classification as individual tasks in an MTL flow orchestration. From an architectural and learning perspective, we draw inspiration from ideas in attention, gating and learning space partitioning. This section is structured to cover the most relevant works to these topics.

2.1. Multi-task learning in vision

General multi-task learning methods rest on sharing complementary information between the task specific branches of a shared model in order to regularize their outputs, boost performance and improve inference efficiency. The same principles apply to MTL in computer vision, as a vast body of research shows that sharing information through building a shared representation between tasks, benefits both high and low level vision problems (Guo et al., 2018; Zhang et al., 2018; Strezoski et al., 2019; Liu et al., 2019; Vandenhende et al., 2020a; Martin et al., 2004; Wang et al., 2016).

In low level vision problems, the models are dealing with dense pixel estimation. In such cases building a shared representation layer between tasks as the branching point of the model does not implicitly take into account the effect of sharing information between different scales within the feature space for one task. Vandenhende et al. (2020b) show that different receptive field sizes (and therefore different scales) in building such representations alter how tasks interact with each other. For example, a dilated kernel applied to a depth estimation map provides much more information about the scene semantics then an eroded one. As such, building a shared representation between a semantic segmentation model and depth estimation model would make more sense, and yield a larger benefit with a dilated receptive field. Wang et al. (2016) confirm a similar finding in their study on the combination of semantic segmentation and estimating surface normals. Given the benefits of distinctly sharing through the task or scale dimensions, in dense pixel estimation tasks we incorporate both through an attention mechanism and concatenated residual component.

High level vision problems such as image classification, object detection, image captioning or attribute prediction also benefit from MTL methods (Misra et al., 2016; Zhang et al., 2016; Liu et al., 2015; Li et al., 2014; Zhang et al., 2014b; Strezoski et al., 2019; Zamir et al., 2018; Ranjan et al., 2019; Teichmann et al., 2018; Jou and Chang, 2016; Alami Mejjati et al., 2018). The main distinction between the MTL methods applied in high and low level vision problems is the clear separation of the high level vision learning space into a task specific section and a shared one. The benefits of a partitioned learning space have been carefully studied (Mallya et al., 2018; Mallya and Lazebnik, 2018; Zhao et al., 2018; Yamir et al., 2018; Zhang et al., 2019a; Strezoski et al., 2019), especially in cases where unrelated tasks seem to mutually harm their learning process (Zhao et al., 2018).

In this work, we preserve the partitioning of the learning space if already existing in architectures applied to high level vision problems, and encode it in the residual information forwarded to the processing of each scale.

2.2. Attention and learning space partitioning

As a manner of allocating the computational resources of the model towards the most informative signal (Olshausen et al., 1993; Itti and Koch, 2001, 2000; Larochelle and Hinton, 2010; Varvani et al., 2017), attention has been proven to help learning across many tasks in the multimedia, computer vision, and natural language processing domains. Regardless of the modality, attention mechanisms can appear
as spatial (Woo et al., 2018), cross-channel (Wang et al., 2017), cross-modality (Vandenende et al., 2020b), cross-task (Liu et al., 2019), or combined as (Woo et al., 2018) show with a combination of spatial and channel attention. In our work, we demonstrate the effectiveness of spatial attention method between tasks, scales and ground-truth modalities.

In MTL, attention can appear as an implicit component of the training architecture (Strezoski et al., 2019; Yang and Hospedales, 2017; Li et al., 2014), or as a standalone component (Liu et al., 2019; Mallya et al., 2018; Zhao et al., 2018) trained alongside the tasks at hand. Some regularization properties in MTL networks, such as the random influence of one task over the learning process of others, can be considered analogous to Shake-Shake Regularization (Gastaldi, 2017) which adds a random variable to both the forward and backward flows during training. Similarly, partitioning in the shared parameter space can also be considered an attention mechanism as dedicated units correspond to task specific control signals, which limits or enhances the influence of affected channels (Yamada et al., 2018). As regularization between tasks is one of the core properties of MTL, in this paper we calculate attention and affinity maps between both tasks and scales across multi-task embeddings. The final embedding from MATTE is a result of the attention modulated joint contribution of all tasks across all scales, controlled at each scale step.

3. Method

Parameter sharing is the foundation on which MTL methods are built. How to share and distribute resources among task interaction is a central research point in the field. In this section, we introduce attention based sharing between tasks and scales, as well their combined auxiliary embedding information. Fig. 1 shows a high level overview of our method, and how the different task and scale information is combined with the help of auxiliary outputs at each scale step. The process of training and forwarding the auxiliary output information during training is displayed step by step in algorithm 1.

Algorithm 1 Single Scale Training epoch for MATTE

1: procedure train(X, scale)
2: for \( X \) \( \in \) \( X_{\text{train}} \) do
3: \( \text{OUT}_{\text{scale}} \leftarrow \text{forward}(X) \) \triangleright Training loop
4: \( \text{RES}_{\text{scale}} \leftarrow \text{OUT}_{\text{scale}} \)
5: \( N\text{OUT}_{\text{scale}} \leftarrow \text{conv}_{1}(\text{OUT}_{\text{scale}}) \)
6: \( \text{EMB}_{\text{scale}} \leftarrow N\text{OUT}_{\text{scale}} \oplus \text{RES}_{\text{scale}} \)
7: if final scale then
8: \( \text{main}_{\text{out}}(\text{EMB}_{\text{scale}}) \)
9: else
10: \( \text{next}_{\text{scale}} \oplus \text{EMB}_{\text{scale}} \)
11: end

Algorithm 1 is repeated for each scale until the final output is reached. The final output gets the aggregated shared embeddings from each task at each prior scale when making the final predictions. Each scale has its own auxiliary output as in Fig. 2, from which a control signal is generated which trains our scale specific embeddings. In this way, we make sure that the shared task embedding is trained for each scale individually, and can measure the impact of the residual component propagated to the next scale.

In the sections below we use \( N \) as the set of tasks in each scale, \( S \) as the group of scales, and \( C, H, W \) as feature dimensions \((N \times S \times C \times H \times W)\). We denote \( \theta \) as the model parameters, \( F \) as the neural net, \( X \) as the input batch, \( L \) as the loss function and \( \delta \theta \) as the gradient. With this notation we can define \( f = f(X|\theta) \) as our model output. In a multi-task multi-scale environment the compound gradient is defined as \( \delta\theta = \sum_{i} \delta\theta_{i} + \sum_{j} \delta\theta_{j} \), where \( i \) is the task index, and \( j \) is the scale index. Defined as such, the parameter update rule is the standard compound gradient update defined as \( \delta\theta = \frac{\delta L}{\sum_{i} \delta\theta_{i}} \) where \( \delta\theta \) is the cumulative gradient update for all tasks.

3.1. Cross-task attention

Our method works with multiple scales of the input data. The first step is to learn a shared representation for each task within the first scale. As all tasks train simultaneously without additional sampling strategies applied, the resulting feature will be marginally biased towards a prevalent task (the one experiencing the smallest loss). After up-sampling this feature to be propagated to the next scale, much of the weaker task information will be suppressed (Srivastava et al., 2014).

To remedy this, we attach a trainable attention block in the form of per task channel gating, which transforms our input from \( N \times S \times C \times H \times W \) to \( N \times S \times C_{\text{matte}} \times H \times W \), where \( C_{\text{matte}} \) represents the attention modulated features across all \( N \) tasks. The channel-wise transformation includes point-wise multiplication of the per task learned attention tensors for every channel. Different from other gating mechanisms (Strezoski et al., 2019; Woo et al., 2018; Mallya et al., 2018; Mallya and Lazebnik, 2018), our attention module preserves the task specific representation in the channel dimension similar to Liu et al. (2019). In fact, if attention is not applied, and the per task channels are only concatenated we empirically established that performance suffers despite the large resulting feature maps.

Following the attention block, a spatial pooling block (He et al., 2014) with a \( 1 \times 1 \) convolution output performs spatial pooling and dimensionality adjustment before propagating the embedding as a residual to the next scale or final output. Spatial pooling across the \( H \times W \) is suitable for cross-task operations as it does not interfere with the between task dynamics, nor does it affect the attention matrices.

It is important to note that in intermediate scale stages, the residual information propagates through the task specific heads before reaching the attention block. Every subsequent representation after the first...
scale, carries the information from the previous scale with it. This residual information is also concatenated to the final modulated feature at the final scale as illustrated in Fig. 1. The diagram of our cross-task attention module in an individual scale is displayed in Fig. 3.

3.2. Cross-scale attention

Similarly to applying attention between the tasks specific representations at any individual scale, across scale attention can improve both efficiency and performance. The benefit is twofold as by discarding channels with redundant information the memory footprint of the overall model is reduced, and the resulting feature is distilled by retaining features persistent across scales. In this capacity, our cross-scale attention block has a channel gating and modulating role just before the final output layers of the task specific branches. This module converts the input from \( N_t \times S \times C \times H \times W \) to \( N_t \times S_{\text{main}} \times C_{\text{matte}} \times H \times W \) dimensionality, where \( C_{\text{matte}} \) represents the attention modulated features across all \( S \) scales per task.

This attention block is trained based on the interference introduced between combinations of the different scale representations of the input. A simple way to measure interference between scale representations is to follow the gradients produced from the outputs on task combinations and make sure that they are not pulling in opposite directions. An example rule for applying attention to scales \( S_1 \) and \( S_2 \) of a single task which can positively or negatively influence each other:

\[
A_{S_1, S_2} = \begin{cases} 
1 - |\Delta \theta_{S_1} - \Delta \theta_{S_2}|, & \text{sign}(\Delta \theta_{S_1}, \Delta \theta_{S_2}) > 0 \\
0, & \text{sign}(\Delta \theta_{S_1}, \Delta \theta_{S_2}) \leq 0
\end{cases}
\]  

(1)

Eq. (1) does not symmetrically apply to task level attention as in most cases the gradient scale between tasks can differ significantly. As final output, our cross-scale attention block outputs a feature map containing all the channels that satisfy the positive condition of Eq. (1).

This final feature map is concatenated with the residual representations from the rest of the tasks and spatially pooled with 1 convolution, before reaching the output head. Finally the output head applies the final softmax that retains the most informative segments of our feature vector before producing output.

3.3. Combined embeddings and residuals

Each task and each scale in our model, outputs separate and combined embeddings which are followed through to the final output layer. As depicted in Fig. 1, these embeddings can be considered horizontal across all tasks in a single scale, and vertical across all scales for a single task. Horizontally, for every task we measure the influence of the per task gradients with respect to the attention coefficients in the attention maps. This enables us to capture which tasks interact or respond well to the same data input. Vertically, for every task we can measure which embedding produces the best score and how much of the other scales content can be beneficial for the total performance metric. As depicted in Fig. 1, both the attention and task specific features are generated in a single forward pass and propagated to the tail of the model where the MATTE attention tensors modulate the resulting feature vector. With fanning out auxiliary outputs at the branching point of our models, we are able to apply MATTE to distinct signals (task, scale, and combined) and let the final feature selection be done just before the final output.

Having a multi-scale backbone increases the quantity of additional, sometimes redundant information which can harm the overall feature quality if not properly scaled. In the process of adding the residual information from any previous scale, we achieve a distillation effect by measuring the impact of the residual and allowing the control signal of the next scale to influence the transferred values.

4. Experiments

In order to validate the contributions presented in this work, our experimental design takes into scope a number of classification, segmentation and dense prediction tasks. For this we consider a variety of datasets and benchmarks ranging from a multi-attribute based face dataset (CelebA), along a cross-dataset MTL benchmark (Visual Decathlon), to multi-output dense prediction (AlphaMatting, Adobe Image Matting, and VideoMatting.Com benchmark). To further explore the effects of the different attention components i.e. scale and task individually and combined, we perform an ablation study on the AlphaMatting dataset.

4.1. MTL in classification datasets

Multi-class classification datasets can be converted to a MTL by simply converting each existing class of the dataset into a binary task, assessing if a certain data sample belongs to a certain class. This experimental setup enables observing MTL in a homogeneous setting, where all data samples belong to the same dataset and have the same annotation form.

MNIST (Xiao et al., 2017) and CIFAR-10 (Krizhevsky et al., 2009) constitute the proof of concept part of our experimental design as they are well established benchmarks and provide simple indication of how different hyper-parameter setups tend to affect the method. For both dataset we define ten binary classification tasks and evaluate the accuracy, precision, and recall scores.

UCSD Birds (Wah et al., 2011) is a dataset that provides 11,788 bird images over 200 bird species with 312 binary attribute annotations. For state of the art comparison, we compare on ten target attributes obtained with spectral clustering using the FSIC as the similarity measure (Alami Mejjati et al., 2018). As we incrementally increase the selected number of attributes we define sets of 50, 100, 200, and 312 binary classification tasks for each of them. For this dataset the training and testing set have equal sizes and distributions. The attributes are sampled according to the ten attribute selection in Alami Mejjati et al. (2018) for the 10 task experiment and in order of the original annotation file for the rest of the experiments.

CelebA (Liu et al.) consists of more than 200,000 face images with binary annotations on 40 face attributes related to age, expression, decoration, etc. The first 10 attributes from (Zhao et al., 2018) are selected for the 10 task experiment as more related to face appearance. We additionally report the results on 40 attributes to compare MATTE to Zhao et al. (2018) in a classification setting.

UT-Zappos50K (Yu and Grauman, 2017) is a large shoe dataset consisting of more than 50,000 catalog images collected from the web. This dataset contains four attributes of interest for our experiment, namely shoe type, suggested gender, height of the heel, and the shoe closure mechanism. As defined in Zhao et al. (2018), we define 4 classification tasks for a small scale test of MATTE over a real world dataset using the identical train, validation, and test splits from Veit et al. (2017), Zhao et al. (2018).

4.2. MTL in Matting datasets

Alpha matting as a computer vision task consists of three distinct processes (background/foreground segmentation, trimap estimation and image matting) resulting in a final output. The challenging aspect to tackle is the co-dependent nature of the task outputs leading to the final output — the alpha matte mask. This dependency between tasks can bring to light aspects of destructive interference in MTL learning and also show the importance of scale and receptive fields in MTL problems. For our MTL setup we define three tasks, semantic segmentation, trimap estimation and matting across three datasets.

The AlphaMatting Benchmark is a popular image matting benchmark, with eight objects for which trimaps and alphas are available

\[ \text{Alami Mejjati et al., 2018} \]
for training. In addition, there is a well defined evaluation protocol with a controlled set of testing images. Image matting as a task has a trimap image requirement, which is a pixel mapping of background, foreground and area of uncertainty pixels which we provide as an auxiliary input from the output of our trimap estimation task. The evaluation metrics for these tasks are explained in Section 4.4.

The **The VideoMatting project** (Erofeev et al., 2015) is the first public video matting benchmark which offers public training sequences and per frame ground-truth, as well as a public evaluation engine. For this dataset we process the videos in their native frame-rate and output a per frame result identically to the previously defined image matting tasks prior to evaluation.

### 4.3. Mixed MTL benchmarking

Heterogeneous MTL poses a different challenge as both the training and testing data often belong to different datasets and can accent the existing contrast between the tasks the model has to perform. The Visual Decathlon and UberNet benchmarks are well known benchmarks in heterogeneous MTL with a large variety of computer vision tasks.

**Visual Decathlon (VD)** (Rebuffi et al., 2017) is a benchmark that evaluates the ability of representations to capture simultaneously ten very different visual domains and measures their ability to perform well uniformly. While the images of this task are of a lower resolution (72 × 72 px.), they contain a wide variety of tasks such as pedestrian, digit, aircraft, and action classification, making it perfect for testing the generalization abilities of our method.

**UberNet** (Kokkinos, 2017) is the name of a MTL model designed to simultaneously address, boundary detection, normal estimation, saliency estimation, semantic segmentation, human part segmentation, semantic boundary detection, region proposal generation and object detection. For this purpose, UberNet trains on large MTL dataset with dense annotations. Unlike the Visual Decathlon, the nature of the annotations is quite diverse and it poses the challenge of dealing with missing annotations which demonstrate how attention can help bridge the missing annotation gap.

### 4.4. Evaluation metrics

In our experimental design we focus on heterogeneous datasets, where evaluation can differ significantly per task. Below we explain how we calculate the resulting scores of individual experiments and tasks that are not be evaluated with the standard metrics available for classification or regression problems.

The semantic segmentation and trimap estimation tasks are evaluated using mean intersection over union (mIoU). The VD challenge offers a per task and cumulative score with a maximum value of 10,000 (1,000 per task) based on the per-task accuracies using the official challenge metric.

We report results with SSDA and dsSSD (Rhemann et al., 2009) defined in Eqs. (2) and (3):

\[
SSDA = \frac{1}{px} \sum_{p} \left( \sum_{t} \frac{(d_{p,t} - \hat{d}_{p,t})^2}{\hat{d}_{p,t}} \right)
\]

\[
dSSD = \frac{1}{px} \sum_{p} \left( \sum_{t} \frac{d_{a_{p,t}} - da_{p,t}^{GT}}{da_{p,t}^{GT}} \right)
\]

These are metrics defined to capture the temporal dependencies across the processed frames. Their correlations with respect to human perception and individual frame matting metrics is explained in detail in Rhemann et al. (2009). The image matting tasks are evaluated using the SSD and MSE metrics.

We evaluated our attention approach in multiple classification settings in a classic MTL setup. The goal of our evaluation is to determine the effect of learning a hybrid form of attention across tasks and scales has on the final outcome in a MTL setting. In addition, we detect and correlate inherent connections between the tasks when the model is asked to combine channels across the feature maps extracted at different scales.

### 4.5. Experimental setup

For our experimental setup we use a MS-COCO pre-trained HRNet (Sun et al., 2019) and Res2Net (Gao et al., 2019) as our multi-scale backbone networks implemented in Pytorch 1.8 and Cuda 10.2. Our models were trained on 4 Nvidia 1080ti GPUs with LaPack version 3.9.0.\(^1\) The complexity of MATTE is approximately the same as the native complexity of the baseline models it is applied to. For example in a 50 layer network (ResNet-50) whose number of parameters is around 25M and the number of FLOPs for an image of 224 × 224 pixels is 4.2G, the multi-scale equivalent with MATTE has 500K additional parameters and the same FLOP count. For training a multi scale backbone in MTL mode, we adopted the low memory back-propagation method from Chen et al. (2016), Kokkinos (2017) and implemented an exception handling class for empty annotation values which persist the weight values when annotations are missing per class. Our exception handling enables to skip the nullable calculations which results in a 5% decrease in training time over conventional MTL training.

As in most MTL training pipelines (Guo et al., 2018; Zhang et al., 2018; Strezoski et al., 2019; Liu et al., 2019; Vandenhende et al., 2020a; Martin et al., 2004; Wang et al., 2016), our experiments seem to benefit from larger batch sizes. For each experiment, we use the maximum batch size to the memory capacity of our system. With respect to learning rates, the best performance and fastest convergence rates we experienced with smaller initial learning rates set to 5e-4, similar to Kokkinos (2017).

### 4.6. Ablation study

Our proposed attention mechanism depends on learning the combination of tasks and scales with the best mutual affinity. To establish a quantifiable metric of the influence of the chosen scale channels against the task channels, we conduct a qualitative and quantitative analysis.

---

\(^1\) https://github.com/leoxiaobin/deep-high-resolution-net.pytorch
study on the AlphaMatting dataset with three tasks with and without the individual attention modules. Our first experiment is founded on a bare multi-scale MTL backbone with a branch per output. In our second experiment we rely on the scale affinities as the main attention component, and in the third experiment rests on the task combinations. Finally we compare the three ablation approaches against our main experiment containing the same backbone and the combination of both scale and task attention factor combinations. Our ablation study continues in the Ubernet experimental setup where the subtraction of selected tasks from the MTL pipeline enables us to observe the effect of the between task influence with or without MATTE attention, further described in Section 5.4.

5. Results

As defined in Section 4.1, experiment one, we start our evaluation with a sanity check experiments on a small proof of concept dataset (MNIST). The intuitive tuples of tasks such as 1 and 7, 6 and 9, 2 and 5 or 3 and 8 quickly form after completing only 2 epochs. This behavior of the network confirms that the grouping of tasks makes visual sense and can be attributed to visible similarities in the input images (see Fig. 4).

5.1. Evaluation on Ubernet

Our Ubernet setup evaluation creates the opportunity of observing MATTE in a diverse annotation setting where labels can be missing in various tasks during training. The overall performance in Table 7 shows a consistent improvement along almost all tasks when attention is applied in both single and multi scale settings. In Table 7 we report an improvement in overall performance as well as in individual tasks singled out (Tables 2–4). It is through this experimental setup that we identify the influence MATTE has on destructive interference as well. As we report our performance, we are dealing with an increased feature

2 Note that a task cannot pair with itself.
entropy in experimental setups such as Ubernet. In such cases any model will clearly benefit from having more free parameters, additional nonlinearities and even larger batch sizes or strictly controlled learning rates (Chen et al., 2018). This means that training a larger model, or expanding the ones proposed in this work is likely to result in a high resolution output from MATTE on the Ubernet challenge. The sample input image is at the top left and the model outputs follow in both the first and second row.

Fig. 5. A high resolution output from MATTE on the Ubernet challenge. The sample input image is at the top left and the model outputs follow in both the first and second row.

Table 4

<table>
<thead>
<tr>
<th>Method</th>
<th># of tasks</th>
<th>ODS</th>
<th>OIS</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>MultiScale sPb</td>
<td>1</td>
<td>0.726</td>
<td>0.757</td>
<td>0.696</td>
</tr>
<tr>
<td>HED - Fusion</td>
<td>1</td>
<td>0.790</td>
<td>0.808</td>
<td>0.811</td>
</tr>
<tr>
<td>UberNet MTL</td>
<td>7</td>
<td>0.785</td>
<td>0.805</td>
<td>0.837</td>
</tr>
<tr>
<td>UberNet STL (sPb setup)</td>
<td>1</td>
<td>0.815</td>
<td>0.835</td>
<td>0.862</td>
</tr>
<tr>
<td>TAG (Single Scale)</td>
<td>7</td>
<td>0.830</td>
<td>0.838</td>
<td>0.874</td>
</tr>
<tr>
<td>Res2Net (Single Scale)</td>
<td>7</td>
<td>0.791</td>
<td>0.808</td>
<td>0.846</td>
</tr>
<tr>
<td>Res2Net (Multi Scale)</td>
<td>7</td>
<td>0.789</td>
<td>0.821</td>
<td>0.866</td>
</tr>
<tr>
<td>MATTE (Multi Scale) (CT)</td>
<td>7</td>
<td>0.789</td>
<td>0.821</td>
<td>0.866</td>
</tr>
<tr>
<td>MATTE (Multi Scale) (CS)</td>
<td>7</td>
<td>0.819</td>
<td>0.834</td>
<td>0.875</td>
</tr>
<tr>
<td>MATTE (Multi Scale)</td>
<td>7</td>
<td>0.836</td>
<td>0.840</td>
<td>0.886</td>
</tr>
</tbody>
</table>

Table 1 also contributes to our ablation study as it offers an observation of the applying task-wise attention on a multi-scale backbone, versus applying the MATTE hybrid form of attention. On average there is 72 ± 3% increase in the positive attention coefficients between task tuples when MATTE is applied against a plain multi-scale MTL approach. With only task-wise attention the average positive attention increase over a plain multi-scale MTL approach is 31 ± 9%. The inherent inductive bias that helps MTL build a useful shared representation is still present even without any form of attention applied during training, however the negative effects of destructive interference are present as much as the positive ones. To our second contribution, we report a significant decrease of negative influence between the tasks performed in a MTL setting using MATTE as a hybrid attention approach.

On the alphamatting.com benchmark (see Table 5), we compare MATTE to several STL approaches from the official competition site and multi-scale modified versions of the same approaches for a fair comparison. Compared to the rigid constraint of channel gating in Strezoski et al. (2019), our hybrid attention form is more flexible when it comes to influences from multiple scales in the final embedding and produces superior overall results. The latest submission on the official VD evaluation engine has the single scale (Zhao et al., 2018), which improves significantly with transitioning to a multi-scale backbone, but plateaus quickly after saturating the learnable modulation adjustment parameters. Table 6 shows the individual and overall scores, where MATTE scores the best overall challenge metric. For more qualitative findings please refer to the appendix.

5.3. Evaluation on the VD challenge

The VD challenge poses a different heterogeneous type of MTL problem, where each dataset is defined as a task. We implemented two scale based MTL approaches (Strezoski et al., 2019; Zhao et al., 2018) to fit a multi-scale backbone MTL flow with Res2Net for a fair comparison. Compared to the rigid constraint of channel gating in Strezoski et al. (2019), our hybrid attention form is more flexible when it comes to influences from multiple scales in the final embedding and produces superior overall results. The latest submission on the official VD evaluation engine has the single scale (Zhao et al., 2018), which improves significantly with transitioning to a multi-scale backbone, but plateaus quickly after saturating the learnable modulation adjustment parameters. Table 6 shows the individual and overall scores, where MATTE scores the best overall challenge metric. For more qualitative findings please refer to the appendix.

5.4. Destructive interference from surface normal estimation

When a pair of tasks react in an opposite manner to the same input, most often it is because the control signal is providing competing or even contradicting gradient directions over a shared layer of units. This phenomenon, known in Physics as destructive interference, occurs when two waves of equal frequency and opposite phases occupy the same physical space and cancel each other out. Similarly, in MTL the calculated gradients of two tasks with opposite directions over a shared layer of units. One of the contributions of our work lies in the ability to seamlessly combine these two types of attention in a multi-branched MTL model.

Table 1 also contributes to our ablation study as it offers an observation of the applying task-wise attention on a multi-scale backbone, versus applying the MATTE hybrid form of attention. On average there is 72 ± 3% increase in the positive attention coefficients between task tuples when MATTE is applied against a plain multi-scale MTL approach. With only task-wise attention the average positive attention increase over a plain multi-scale MTL approach is 31 ± 9%. The inherent inductive bias that helps MTL build a useful shared representation is still present even without any form of attention applied during training, however the negative effects of destructive interference are present as much as the positive ones. To our second contribution, we report a significant decrease of negative influence between the tasks performed in a MTL setting using MATTE as a hybrid attention approach.

On the alphamatting.com benchmark (see Table 5), we compare MATTE to several STL approaches from the official competition site and multi-scale modified versions of the same approaches for a fair comparison. Compared to the rigid constraint of channel gating in Strezoski et al. (2019), our hybrid attention form is more flexible when it comes to influences from multiple scales in the final embedding and produces superior overall results. The latest submission on the official VD evaluation engine has the single scale (Zhao et al., 2018), which improves significantly with transitioning to a multi-scale backbone, but plateaus quickly after saturating the learnable modulation adjustment parameters. Table 6 shows the individual and overall scores, where MATTE scores the best overall challenge metric. For more qualitative findings please refer to the appendix.

5.3. Evaluation on the VD challenge

The VD challenge poses a different heterogeneous type of MTL problem, where each dataset is defined as a task. We implemented two scale based MTL approaches (Strezoski et al., 2019; Zhao et al., 2018) to fit a multi-scale backbone MTL flow with Res2Net for a fair comparison. Compared to the rigid constraint of channel gating in Strezoski et al. (2019), our hybrid attention form is more flexible when it comes to influences from multiple scales in the final embedding and produces superior overall results. The latest submission on the official VD evaluation engine has the single scale (Zhao et al., 2018), which improves significantly with transitioning to a multi-scale backbone, but plateaus quickly after saturating the learnable modulation adjustment parameters. Table 6 shows the individual and overall scores, where MATTE scores the best overall challenge metric. For more qualitative findings please refer to the appendix.

5.4. Destructive interference from surface normal estimation

When a pair of tasks react in an opposite manner to the same input, most often it is because the control signal is providing competing or even contradicting gradient directions over a shared layer of units. This phenomenon, known in Physics as destructive interference, occurs when two waves of equal frequency and opposite phases occupy the same physical space and cancel each other out. Similarly, in MTL the calculated gradients of two tasks with opposite directions destroy each other’s influence over the trainable parameter space.

Destructive interference is clearly observable in our experimental setup when introducing the surface normal estimation problem to the pool of tasks the network performs. Table 8 shows an overall degradation in performance once the Surface Normal Estimation task is added in rows 3, 4, 5 and 7. It also notes a spike in performance in row 6 once learned in isolation (STL). Our reasoning relates to the performance
Table 5
Approach performance on the alphamatting.com benchmark in SAD and MSE across all classes, compared to STL learning approaches from the official benchmark results on a large input trimap. In our approach, the trimap is generated from the trimap generation branch with the interim background/foreground segmentation output as input.

<table>
<thead>
<tr>
<th>Approach</th>
<th>SAD</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Troll</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Doll</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Donkey</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elephant</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plant</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pineapple</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plastic bag</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATTE</td>
<td>9.5</td>
<td>0.4</td>
</tr>
<tr>
<td>MATTE (HRNet)</td>
<td>9.0</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Table 6
Performance on the Visual Decathlon Challenge with the model from (Rebuffi et al., 2017) [RB] with TRL sharing ratio of $\alpha = 0.6$ ($\omega = 0$ being the best performing baseline model of the challenge.).

<table>
<thead>
<tr>
<th>Run</th>
<th>VD Score</th>
<th>Aircraft</th>
<th>Cifar-100</th>
<th>Daumler</th>
<th>DTD</th>
<th>GTSRB</th>
<th>Imagenet-12</th>
<th>Omniglot</th>
<th>SVHN</th>
<th>UCF-101</th>
<th>VGG-Flowers</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResAdapt (Rebuffi et al., 2017)</td>
<td>2851.3</td>
<td>299.8</td>
<td>195.9</td>
<td>155.4</td>
<td>261.5</td>
<td>472.6</td>
<td>224.1</td>
<td>337.1</td>
<td>282.8</td>
<td>231.6</td>
<td>390.2</td>
</tr>
<tr>
<td>mtMTL (Zhao et al., 2018)</td>
<td>3207.0</td>
<td>259.1</td>
<td>224.9</td>
<td>625.8</td>
<td>249.4</td>
<td>726.8</td>
<td>304.1</td>
<td>311.4</td>
<td>197.0</td>
<td>232.1</td>
<td>377.8</td>
</tr>
<tr>
<td>Task Routing (Strezoski et al., 2019)</td>
<td>2919.3</td>
<td>305.2</td>
<td>204.1</td>
<td>165.9</td>
<td>273.3</td>
<td>469.2</td>
<td>228.4</td>
<td>345.1</td>
<td>272.8</td>
<td>252.1</td>
<td>403.2</td>
</tr>
<tr>
<td>Res2Net (Gao et al., 2019)</td>
<td>3364.7</td>
<td>389.3</td>
<td>301.8</td>
<td>412.0</td>
<td>219.6</td>
<td>401.2</td>
<td>308.9</td>
<td>327.2</td>
<td>302.1</td>
<td>265.5</td>
<td>433.9</td>
</tr>
<tr>
<td>HRNet (Sun et al., 2019)</td>
<td>3166.5</td>
<td>394.9</td>
<td>318.2</td>
<td>235.5</td>
<td>350.9</td>
<td>499.9</td>
<td>263.1</td>
<td>396.5</td>
<td>299.3</td>
<td>382.9</td>
<td>435.3</td>
</tr>
<tr>
<td>TAG (Foly et al., 2021)</td>
<td>3576.5</td>
<td>394.9</td>
<td>318.2</td>
<td>235.5</td>
<td>350.9</td>
<td>499.9</td>
<td>263.1</td>
<td>396.5</td>
<td>299.3</td>
<td>382.9</td>
<td>435.3</td>
</tr>
<tr>
<td>MATTE [RB]</td>
<td>3757.6</td>
<td>422.3</td>
<td>243.1</td>
<td>533.7</td>
<td>223.7</td>
<td>710.3</td>
<td>303.9</td>
<td>322.1</td>
<td>282.8</td>
<td>231.6</td>
<td>439.6</td>
</tr>
<tr>
<td>MATTE (Res2Net)</td>
<td>3396.8</td>
<td>399.2</td>
<td>305.6</td>
<td>416.1</td>
<td>212.7</td>
<td>395.2</td>
<td>311.7</td>
<td>330.1</td>
<td>299.3</td>
<td>302.1</td>
<td>421.1</td>
</tr>
<tr>
<td>MATTE (HRNet)</td>
<td>3584.4</td>
<td>405.0</td>
<td>312.1</td>
<td>502.0</td>
<td>200.4</td>
<td>699.6</td>
<td>281.1</td>
<td>299.6</td>
<td>283.5</td>
<td>234.2</td>
<td>439.6</td>
</tr>
</tbody>
</table>

Table 7
This table reports the overall performance on the UberNet experimental setup with an identical multi scale backbone (Res2Net).

<table>
<thead>
<tr>
<th>Method</th>
<th>Detection</th>
<th>Boundaries</th>
<th>Salience</th>
<th>Parts</th>
<th>Surface Normals</th>
<th>S. Boundaries</th>
<th>S. Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP</td>
<td>ODS</td>
<td>OBS</td>
<td>AP</td>
<td>MF</td>
<td>mIoU</td>
<td>MF</td>
</tr>
<tr>
<td>MTI Net (Multi Scale)</td>
<td>79.7</td>
<td>0.800</td>
<td>0.802</td>
<td>0.841</td>
<td>0.820</td>
<td>38.1</td>
<td>24.3</td>
</tr>
<tr>
<td>UberNet MTI (Single Scale)</td>
<td>76.2</td>
<td>0.779</td>
<td>0.805</td>
<td>0.836</td>
<td>0.820</td>
<td>36.7</td>
<td>24.2</td>
</tr>
<tr>
<td>UberNet MTI (Multi Scale)</td>
<td>75.3</td>
<td>0.772</td>
<td>0.802</td>
<td>0.836</td>
<td>0.814</td>
<td>34.2</td>
<td>27.7</td>
</tr>
<tr>
<td>UberNet MTI (Optimal)</td>
<td>77.8</td>
<td>0.785</td>
<td>0.805</td>
<td>0.837</td>
<td>0.822</td>
<td><strong>48.8</strong></td>
<td>24.3</td>
</tr>
<tr>
<td>Res2Net (Single Scale)</td>
<td>77.3</td>
<td>0.770</td>
<td>0.801</td>
<td>0.833</td>
<td>0.825</td>
<td>38.0</td>
<td>24.8</td>
</tr>
<tr>
<td>TAG (Single Scale)</td>
<td>80.3</td>
<td>0.594</td>
<td>0.810</td>
<td>0.818</td>
<td>0.830</td>
<td>43.1</td>
<td>26.1</td>
</tr>
<tr>
<td>MATTE (Single Scale)</td>
<td>81.1</td>
<td>0.831</td>
<td>0.822</td>
<td>0.85</td>
<td>0.83</td>
<td>45.3</td>
<td>27.6</td>
</tr>
<tr>
<td>Res2Net (Multi Scale)</td>
<td>79.1</td>
<td>0.781</td>
<td>0.811</td>
<td>0.847</td>
<td>0.829</td>
<td>42.5</td>
<td>25.3</td>
</tr>
<tr>
<td>MATTE (Multi Scale)</td>
<td>84.2</td>
<td>0.836</td>
<td>0.840</td>
<td>0.886</td>
<td>0.830</td>
<td>48.1</td>
<td>28.9</td>
</tr>
</tbody>
</table>

Table 8
Observing the effect of adding Surface Normal Estimation as an additional task to Object Detection and Semantic Segmentation with Res2Net. Due to the difference in the metrics, we normalized the Mean Angle Distance with to the scale of 0 to 100 with equation $\frac{\text{Mean Angle Dist}}{180} \times 100$.

<table>
<thead>
<tr>
<th>Task Combination</th>
<th>Obj.</th>
<th>Sem.</th>
<th>Surf. Normals</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obj. Detection</td>
<td>83.2</td>
<td>n/a</td>
<td>n/a</td>
<td>83.2</td>
</tr>
<tr>
<td>Obj. Det. and Sem. Seg.</td>
<td>83.9</td>
<td>84.3</td>
<td>n/a</td>
<td>84.1</td>
</tr>
<tr>
<td>Obj. Det. Sem. and Surf. Norm.</td>
<td>82.0</td>
<td>81.7</td>
<td>86.9</td>
<td>83.5</td>
</tr>
<tr>
<td>Obj. Det. and Surf. Norm.</td>
<td>81.0</td>
<td>n/a</td>
<td>88.2</td>
<td>84.6</td>
</tr>
<tr>
<td>Sem. Seg. and Surf. Norm.</td>
<td>n/a</td>
<td>80.2</td>
<td>87.4</td>
<td><strong>83.8</strong></td>
</tr>
<tr>
<td>Surf. Norm.</td>
<td>n/a</td>
<td>81.5</td>
<td>89.5</td>
<td>84.1</td>
</tr>
<tr>
<td>PRIO + Obj. Det. Sem. and Surf. Norm.</td>
<td>81.0</td>
<td>80.3</td>
<td>86.8</td>
<td>83.2</td>
</tr>
<tr>
<td>MATTE + Obj. Det. and Surf. Norm.</td>
<td><strong>83.1</strong></td>
<td>n/a</td>
<td>90.0</td>
<td><strong>86.6</strong></td>
</tr>
<tr>
<td>MATTE + Sem. Seg. and Surf. Norm.</td>
<td>n/a</td>
<td>82.6</td>
<td>89.9</td>
<td>86.3</td>
</tr>
<tr>
<td>MATTE + Obj. Det. Sem. and Surf. Norm.</td>
<td><strong>83.1</strong></td>
<td>84.1</td>
<td>89.2</td>
<td><strong>85.5</strong></td>
</tr>
</tbody>
</table>

issues reported in Kokkinos (2017) as well. The surface normal estimation problem of the UberNet setup poses a harder challenge in an MTL setup as the label definition of the input is of a more continuous and geometric nature. The result of that ground-truth difference is that the backbone networks with their ImageNet and Coco pre-trained weights do not pose a relevant feature bed without additional nonlinearities and heavy fine-tuning for surface normals.

Using MATTE clearly allows for competitive performance in cases when destructive interference occurs, because the attention modules filter out the features relevant to the largest overall and per
task improvement. Table 8 shows that the degradation in performance of the surface normals task done in STL mode, compared to the native task performance done in an MTL setting is 2.3% in row 10 compared to row 6, but significantly smaller — 4.7% (between row 6 and 7) than an MTL setup without MATTE and an explicit task prioritization approach (Guo et al., 2018).

6. Conclusion

In this work we show the power of a combined scale and task-wise attention in multi-scale MTL models. For our primary contribution, we propose a MTL attention mechanism, MATTE, that leverages the combined representational power of multi-scale backbones and MTL models for learning superior representations and boosting performance. We achieve such behavior by increasing the influence of channels across tasks and scales beneficial to performance, using a trainable non-linearity over task and scale affinity tensors. By applying attention in such manner MATTE combats destructive interference across tasks and scales in multi-scale MTL. Our qualitative and quantitative findings suggest a clear benefit of using MATTE as an attention mechanism in MTL in both high level tasks such as image classification, and dense output tasks such as segmentation, trimap generation and matting in both videos and images.

Our dense output experiments with segmentation and matting tasks give a more intuitive outlook on the effect of MATTE as an attention mechanism, and also treat trimap estimation as a learnable task rather than a manual transformation. Having this type of dynamic trimap estimation, leads us closer to a true end to end matting approach, where the size of the uncertainty area in a trimap is dictated by the matting model. While having multiple trimap candidates and iteratively applying the trimap to the refinement of the final matting result works very well, it still generates redundant data and model iterations to produce the final result. With MATTE the conventional ensemble of trimap approach to matting can be replaced by a trimap generated to produce best matting results with the underlying model, reducing the redundant trimap generation, reducing the number of I/O operations if interim results are stored and producing competitive or better performance.

Our experiments with the Ubernet setup helped us observe how our method would cope with missing annotations in a MTL dataset, but also measure the effect an attention mechanism such as MATTE has on destructive interference. Instead of freezing training for a branch with missing annotations, it can be inherently handled by the attention coefficients masking the resulting gradient for the specific branch. Attention is also a promising way of addressing destructive interference by making the model aware of how tasks interact with each other. In MTL setups with multi scale feature extraction, instead of amplifying the negative effects, using MATTE on the multi scale features can help in learning a superior representation.

The true strength of MTL over STL approaches is best visible when there is a structured sharing approach, in a tightly related set of tasks over a descriptive dataset. We conclude that even in tasks as related as segmentation, trimap estimation and matting negative interference can affect the final outcome if the shared representation is not properly modulated. With MATTE, we present an approach that can enables multi-scale MTL approaches to achieve the promise MTL models are built for.

CRediT authorship contribution statement

**Gjorgji Strezoški:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft. **Naanne van Noord:** Validation, Writing – review & editing, Supervision. **Marcel Worring:** Validation, Resources, Writing – review & editing, Supervision, Project administration, Funding application.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References


