
Tianlong Chen†‡, Xuxi Chen†, Xianzhi Du†, Abdullah Rashwan‡, Fan Yang‡, Huizhong Chen‡, Zhangyang Wang†, Yeqing Li‡
†University of Texas at Austin, ‡Apple, ¶Google
{tianlong.chen, xxchen, atlaswang}@utexas.edu, xianzhi@apple.com, {arashwan, fyangf, huizhongc, yeqing}@google.com

Abstract

Sparsely activated Mixture-of-Experts (MoE) is becoming a promising paradigm for multi-task learning (MTL). Instead of compressing multiple tasks’ knowledge into a single model, MoE separates the parameter space and only utilizes the relevant model pieces given task type and its input, which provides stabilized MTL training and ultra-efficient inference. However, current MoE approaches adopt a fixed network capacity (e.g., two experts in usual) for all tasks. It potentially results in the over-fitting of simple tasks or the under-fitting of challenging scenarios, especially when tasks are significantly distinctive in their complexity. In this paper, we propose an adaptive MoE framework for multi-task vision recognition, dubbed AdaMV-MoE. Based on the training dynamics, it automatically determines the number of activated experts for each task, avoiding the laborious manual tuning of optimal model size. To validate our proposal, we benchmark it on ImageNet classification and COCO object detection & instance segmentation which are notoriously difficult to learn in concert, due to their discrepancy. Extensive experiments across a variety of vision transformers demonstrate a superior performance of AdaMV-MoE, compared to MTL with a shared backbone and the recent state-of-the-art (SoTA) MTL MoE approach. Codes are available online: https://github.com/google-research/google-research/tree/master/moe_mtl.

1. Introduction

Multi-task vision recognition aims to simultaneously solve multiple objectives, which is commonly required in real-world applications. For instance, robotics [64] need to learn how to pick, place, cover, rearrange, and align objects simultaneously; autonomous vehicles [42] are expected to concurrently perform drivable area estimation, lane detection, pedestrian detection, and more. Classic multi-task learning (MTL) methods [52, 60, 28, 48, 67, 85, 51] learn a shared representation among different tasks and attach task-specific heads. Following the generic trend in visual recognition, recent MTL works leveraged Vision Transformers (ViTs) [25, 68, 49, 10] as the new unified backbone [7, 5].

However, such MTL models with a single backbone suffer from unstable training and inefficient inference. As pointed out by [56, 81, 17], the shared parameters might receive conflicted update directions from different objectives, and this negative competition usually leads to poor training convergence, biased representations, and inferior performance. Meantime, existing MTL regimes usually activate the whole network backbone, regardless of what tasks come. It causes a waste of computations in potential since various real-world MTL systems [64] perform one or a few tasks at each moment, which may only require the relevant model pieces. The sparsely activated Mixture-of-Experts (SMoE) serves as an encouraging remedy for tackling these two MTL bottlenecks. Specifically, a pioneering study [46] inserts SMoE layers into the MTL ViT by replacing its dense feedforward network with a series of sparsely
activated MoE experts (*e.g.*, multilayer perception (MLP)). Then, task-dependent routing policies are enforced to select a subset of task-relevant experts. Impressive results are demonstrated with this MTL MoE [46].

Despite these preliminary investigations, key challenges still persist in building an effective MTL system: *How to determine an appropriate network capacity for each task in MTL?* By treating it as a hyperparameter, performing the manual tuning for each task is laborious and infeasible due to the entanglement between tasks. Thus, a fixed model size across all tasks is a conventional setup of existing MTL approaches (*e.g.*, always using 4 experts in [46]). However, this rigid and sub-optimal design potentially sacrifices the learning of certain tasks, since excessive or insufficient network capacity leads to either over-fitting or under-fitting in simple or complex scenarios, respectively [72]. The disadvantages will be further amplified when optimizing multiple tasks with a substantial variation in task complexity. Take image classification and object detection tasks as examples. *First*, the common benchmarks for classification have a lower input resolution like $32 \times 32$ for CIFAR [40] and $224 \times 224$ for ImageNet [24], while object detection is normally evaluated on the COCO [47] dataset with a higher resolution of $640 \times 640$ or $892 \times 892$. *Second*, to obtain a satisfying performance, the routine network for detection [9] is usually larger than the ones for classification [69], such as ResNet-101 [34] versus ResNet-50. *Third*, as for the task objectives, object detection contains both object localization and recognition, and thus is more complicated than classification which can be essentially regarded as a sub-task. As discovered in [18, 33], their mismatched learning goals emphasize different feature proprieties (*i.e.*, location invariant [8] versus sensitive). Given such heterogeneity of task complexity, these two tasks are notoriously difficult to learn together with a shared feature extractor and unified model size. An adaptive mechanism is therefore demanded.

In this paper, we propose AdaMV-MoE, to address the aforementioned key barriers, by seamlessly customizing the current state-of-the-art (SOTA) MTL MoE [46]. To be specific, an adaptive expert selection mechanism is proposed to automatically determine the number of experts (or model capacity) in use for different vision tasks. We monitor the validation loss to adaptively determine activating more/less experts to prevent under-fitting/over-fitting. Our contributions are summarized below:

- We target the problem of multi-task vision recognition, and tackle the key challenge of choosing a suitable network capacity for distinctive tasks. According to training dynamics, our algorithm controls the task-specific model size in an adaptive and automatic manner.
- We introduce a customized MoE to resolve image classification, object detection, and instance segmentation simultaneously, which used to be a troublesome combination for MTL. Visualization of our learned task-specific routing decisions is provided and exposes specialization patterns, particularly for image contents.
- Extensive experiments are conducted to reveal the effectiveness of AdaMV-MoE in MTL, as shown in Figure 1. For example, our approaches surpass the vanilla MTL ViT with a shared feature extractor, by a significant performance margin of $(6.66\% \sim 7.93\%)$ accuracy, $0.87\% \sim 1.13\%$ AP, $0.84\% \sim 0.89\%$ AP$_{\text{mask}}$ for \{image classification, object detection, instance segmentation\} on ImageNet and COCO datasets with UViT-Base backbones [16].

2. Related Works

**Multi-Task Learning (MTL).** MTL resolves multiple objectives and produces corresponding predictions for input samples. It has been investigated for a long history, and numerous solutions are proposed ranged from classic learning algorithms [78, 36, 89, 4, 80, 43, 41] to modern deep neural networks. Deep learning methods generate shared feature representations to model the common information across tasks, which can be categorized into two groups, *i.e.*, encoder- and decoder-focused pipelines. The former [52, 60, 28, 48] allows the task interactions in the encoder and attaches task-specific heads on top of it as independent decoders. For example, [52] and [48] advocate the linear combination and attention mechanism to learn shared encoder representations among tasks, respectively. The latter [77, 87, 86, 70] first creates initial task-dependent features from decoders and then aggregates them to form the final per-task prediction. Such pipelines consume heavy computations since they need to at least execute all tasks once for the initial decoder features, which limits their practical usage in resource-constrained scenarios. In this paper, we mainly study encoder-focused architectures.

A conventional encoder architecture is a convolutional neural network (CNN) [48, 63, 84, 85]. As ViTs emerge, IPT [11] leveraged transformer-based models to solve multiple low-level vision tasks. [54] and [61] adopt similar architectures for the tasks of \{object detection, semantic segmentation\} and \{scene and action understanding, score prediction\} in the video, respectively. [7] further involves vision tasks from 3D domains. Our work considers jointly learning classification, object detection, and instance segmentation with ViT-based models. Note that it is highly non-trivial since classification and detection & segmentation emphasize location invariant [8] and sensitive features respectively, which potentially contradict each other. Besides, another theme in MTL investigates how to share and separate parameter spaces for learning task-agnostic and specific knowledge respectively [66, 71, 55, 6, 46].

**Mixture-of-Experts (MoE).** MoE duplicates some network components into a series of copies (named experts)
and embraces the conditional computation in an input-dependent way \cite{37,39,12,82}. The earliest variant of MoEs densely activates all experts for each input, and therefore it is computation-intensive \cite{26}. Later on, \cite{62,44,27} advocate a sparsely activated style for utilizing experts, called sparse MoE (SMoE). It greatly reduces the cost at both the training and inference stages, which grants impressive scalability and even allows enormous language models with trillions of parameters \cite{27}. The effectiveness of SMoEs has been widely proved in various NLP \cite{62,44,91,88,93,38} and vision \cite{58,26,2,30,74,79,1,57} tasks. Particularly, the pioneering work \cite{58} offers the first vision transformer-based SMoE for the image recognition task.

With further investigations, several downsides of SMoE are revealed, including: \(i\) Training instability. \cite{92} conducts a trade-off study of SMoE between its training stability and quality, where they show many classic tricks like gradient clipping stabilize training but sacrifice performance and the router \(z\)-loss \cite{92} seems to bring a win-win case. \(ii\) Poor specialization. The ideal outcome of SMoE is to divide and conquer certain tasks by tackling each piece problem with selected experts \cite{3,32,51,53,15}. Yet it is hard to reach unless explicitly enforcing specialization and trimming down the redundancy among experts \cite{13} like pre-defining a diverse expert assignment \cite{22} or involving multiple routing policies \cite{32}. \(iii\) Representation collapse. Na"{i}vely trained SMoE is prone to load imbalance, \textit{e.g.}, only a few experts are frequently used while the others are scarcely activated. To alleviate this issue, \cite{62} adds Gaussian noises to router networks; \cite{44,27} propose an auxiliary loss as the regularization; \cite{45} formulates and solves a balanced linear assignment problem; \cite{91} distributes the top-\(k\) relevant input for each expert; \cite{59,93} adopt deterministic hashing and stochastic routing; and \cite{14} promotes diversity during training, respectively. In this paper, we not only examine the aforementioned bottlenecks but also investigate new properties of routers such as policy convergence.

Several recent studies also explore the possibility of SMoE in the MTL scenarios. To be specific, \cite{51,3,32,31,90} use task-dependent router networks to select relevant parts of the model with a fixed size for each task. They show positive results in small-scale applications like classification for medical signals \cite{3}, digital number images (MNIST) \cite{32}, and recommendation systems \cite{51}. \cite{46} works on the efficient on-device MTL with a model-accelerator co-designed SMoE.

3. Methodology

3.1. Revisiting Sparse Mixture of Experts

SMoE \cite{62} is proposed to scale up the model capacity while maintaining low per-inference costs. In this work, we consider SMoE for ViTs \cite{25,58}, which inserts SMoE layers into every other transformer block. The SMoE layer contains a router network \(R\) and several experts \(f_1, f_2, \cdots, f_E\), where \(E\) is the number of experts. The expert module can be a few fully connected \cite{62,58} or convolutional layers \cite{73}, and we duplicate multi-layer perceptions (MLP) as expert networks shown in Figure 2. Note that MLPs in ViTs contain around 2/3 of total parameter counts, and \cite{29,20} demonstrate their significance as memory networks to store substantial knowledge.

Another key component in SMoE layers, \textit{i.e.}, \(R\), activates the top-\(k\) expert networks with the largest scores \(R(x)\), associated with input embedding \(x\), where \(i\) is the
Expert index. Normally, the number of selected experts $k$ is fixed and much smaller than the total number of experts $E$, which suggests the sparsely activated fashion of SMoE. The expert distribution can be formally depicted as below:

$$y = \sum_{i=1}^{k} y_i \cdot f_i(x), R(x) = \text{TopK}(\text{softmax}(g(x)), k),$$

where $f_i(x)$ stands for the feature representations produced from the expert $f_i$, which is weighted by $R(x)$, to form the final output $y$. The network $g$ is the learnable part within a router $R$ and it usually is one or a few layers MLP [62, 27]. TopK is a function that discards the small elements ranked after $k$. To reduce the negative effects of the imbalanced loading (or representation collapse [19]), we introduce regularization terms to balance the expert assignments, following the design and default hyperparameters in [58].

### 3.2. AdaMV-MoE: Adaptive Multi-Task Vision Recognition with Mixture-of-Experts

**Overview of AdaMV-MoE**  Our proposed framework, i.e., AdaMV-MoE, consists of task-dependent router networks and an adaptive expert selection (AES) mechanism. As described in Figure 2, input token embeddings are fed into corresponding router networks based on their task types. The task-dependent routers then choose the most relevant experts and aggregate their features for different tasks. The number of selected experts is dynamically decided according to the in-time training dynamics with AES.

**Task-dependent Routing Policies.** Let $R_j$ represents the router for the task $j$, and all expert networks $\{f_i\}_{i=1}^{E}$ are shared across tasks. The SMoE equipped with task-dependent router networks is defined as:

$$y_j = \sum_{i=1}^{k_j} R_j(x) \cdot f_i(x), R_j(x) = \text{TopK}(\text{softmax}(g_j(x)), k_j),$$

where $k_j$ and $y_j$ are the task-specific number of activated experts and output, respectively. As supported by Section 4, the discrepancy among different routing policies brings the entanglement of parameter spaces, resulting in mitigated gradient conflicts of MTL and enhanced performance.

**Adaptive Expert Selection (AES).** The optimal network size for various vision recognition tasks may alter significantly, due to the difference in task complexities. It is hard to conclude manually without laborious trial and error. We instead adopt an automatic algorithm AES to determine the $k_j$ in a data-driven way. As shown in Algorithm 1, it first computes the task-specific objective $L_{\text{val}}^j$ on the validation set. If $L_{\text{val}}^j$ does not decay in the next $\Delta n$ iterations, then we expand the activated model size by updating $k_j = k_j + 1$.

#### Algorithm 1 Adaptive Expert Selection (AES).

1. **Input:** Expert networks $f_i$ $(i \in \{1, 2, \ldots, E\})$, routers $R_j$ $(j$ is the task index), the validation set $D_{\text{val}}^j$ for task $j$, the objective function $L_{\text{val}}^j$, on the validation set.
2. **for** a given task $j$ **do**
   3. Initial the number of selected experts as $k_j \leftarrow 1$;
   4. Initial an indicator Improved as True;
   5. Initial the current best validation loss as $L_{\text{val}}^j(\text{best}) \leftarrow \infty$;
   6. **while** True **do**
      7. If $L_{\text{val}}^j(\text{best})$ does not decrease for $\Delta n$ iterations then
       8. **if** not improved **then**
          9. **end if**
       10. **else**
           11. $k_j \leftarrow k_j + 1$; improved $\leftarrow$ False;
           12. **end if**
       13. **end if**
       14. Continue training the model;
       15. If $L_{\text{val}}^j < L_{\text{val}}^j(\text{best})$ then
           16. $L_{\text{val}}^j(\text{best}) \leftarrow L_{\text{val}}^j$; improved $\leftarrow$ True;
           17. **end if**
      18. **end while**
      19. $k_j = k_j - 1$ and fix $k_j$;
   20. Continue training to the target number of iterations.
   21. **end for**
22. **Output:** AdaMV-MoE with task-dependent top-$k_j$ routers.

Existing literature [76] points out that a proper network expansion creates the possibility for escaping saddle points in the functional space and further decreases the objective values. Meanwhile, if $L_{\text{val}}^j$ is larger than the previous best validation loss $L_{\text{val}}^j(\text{best})$, we reduce the selected expert number by $k_j = k_j - 1$. Lastly, $k_j$ is fixed and the model is continually trained under reaching the target number of training iterations. The above procedures are repeated for all tasks.

### 4. Experiment

#### 4.1. Implementation Details

**Network Backbone.** Our experiments focus on ViT-based backbones, including ViT [25] and its advanced variant - UViT [16]. Varying the model size, we establish four ViTs of {ViT-Small*, ViT-Small, ViT-Base, UViT-Base}, of which the details are exhibited in Table 2.

<table>
<thead>
<tr>
<th>Backbones</th>
<th># Transformer Layers</th>
<th># Attention Heads</th>
<th>Hidden Dimension</th>
<th>MLP Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViT-Small</td>
<td>6</td>
<td>6</td>
<td>384</td>
<td>1536</td>
</tr>
<tr>
<td>ViT-Small*</td>
<td>12</td>
<td>6</td>
<td>384</td>
<td>1536</td>
</tr>
<tr>
<td>ViT-Base</td>
<td>12</td>
<td>6</td>
<td>768</td>
<td>3072</td>
</tr>
<tr>
<td>UViT-Base</td>
<td>18</td>
<td>6</td>
<td>384</td>
<td>1536</td>
</tr>
</tbody>
</table>

The backbone first takes input images from the classification and detection datasets and then extracts features that will further be processed by task-specific modules. A linear classification layer and detection & segmentation heads from Cascade Mask-RCNN [9] are chosen in our experiments. Following [58], ViT and SMoE layers are arranged alternatively. More details are in Section A1.
Table 1. Multi-task vision recognition performance of our proposed AdaMV-MoE. {Accuracy (\%)}, {AP (\%)}, {AP}_{50} (\%), {AP}_{75} (\%)}, and {AP^{mask} (\%)} are reported for classification (CLS) on ImageNet-1k, object detection (OD), and instance segmentation (IS) on COCO respectively. **# Parameters (M)** indicates the adaptively allocated network capacity. ViT-Small*/Small/Base [25] and UViT-Base [16] backbones are adopted, whose details are recorded in Table 2. ViT-Small* is a reduced variant of ViT-Small with half transformer layers. Comparisons are conducted with the baseline MTL-ViT and a recent state-of-the-art MTL approach TAPS [71]. The total number of experts E in our AdaMV-MoE is 8. {Dense and Large Dense, Sparse} means that the {entire, partial} network is used for each task at the training and inference stages, respectively. N.A. denotes “Not Applicable”.

<table>
<thead>
<tr>
<th>Backbone</th>
<th>Method</th>
<th>Classification</th>
<th>Object Detection</th>
<th>Instance Segmentation</th>
<th># Parameters (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Accuracy (%)</td>
<td>AP (%)</td>
<td>AP_{50} (%)</td>
<td>AP_{75} (%)</td>
</tr>
<tr>
<td>Dense</td>
<td>ViT for CLS</td>
<td>73.00</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
</tr>
<tr>
<td>Dense</td>
<td>ViT for OD &amp; IS</td>
<td>N.A.</td>
<td>39.75</td>
<td>61.71</td>
<td>42.77</td>
</tr>
<tr>
<td>Dense</td>
<td>MTL-ViT</td>
<td>68.30</td>
<td>36.35</td>
<td>58.79</td>
<td>38.86</td>
</tr>
<tr>
<td>Large Dense</td>
<td>MTL-ViT</td>
<td>70.32</td>
<td>37.74</td>
<td>60.27</td>
<td>40.58</td>
</tr>
<tr>
<td>Dense</td>
<td>TAPS</td>
<td>69.32</td>
<td>36.66</td>
<td>58.97</td>
<td>38.55</td>
</tr>
<tr>
<td>Sparse</td>
<td>AdaMV-MoE (Ours)</td>
<td>72.99</td>
<td>39.04</td>
<td>61.16</td>
<td>42.43</td>
</tr>
<tr>
<td>Dense</td>
<td>MTL-ViT</td>
<td>69.34</td>
<td>41.43</td>
<td>63.45</td>
<td>45.13</td>
</tr>
<tr>
<td>Large Dense</td>
<td>MTL-ViT</td>
<td>71.87</td>
<td>42.07</td>
<td>64.48</td>
<td>45.66</td>
</tr>
<tr>
<td>Dense</td>
<td>TAPS</td>
<td>74.63</td>
<td>37.38</td>
<td>60.15</td>
<td>39.89</td>
</tr>
<tr>
<td>Sparse</td>
<td>AdaMV-MoE (Ours)</td>
<td>78.41</td>
<td>42.16</td>
<td>64.33</td>
<td>45.73</td>
</tr>
<tr>
<td>Dense</td>
<td>MTL-ViT</td>
<td>74.18</td>
<td>42.63</td>
<td>64.21</td>
<td>46.53</td>
</tr>
<tr>
<td>Large Dense</td>
<td>MTL-ViT</td>
<td>74.41</td>
<td>42.47</td>
<td>64.19</td>
<td>46.12</td>
</tr>
<tr>
<td>Dense</td>
<td>TAPS</td>
<td>78.45</td>
<td>42.51</td>
<td>65.28</td>
<td>45.87</td>
</tr>
<tr>
<td>Sparse</td>
<td>AdaMV-MoE (Ours)</td>
<td>78.59</td>
<td>42.70</td>
<td>65.12</td>
<td>46.05</td>
</tr>
<tr>
<td>Dense</td>
<td>MTL-UViT</td>
<td>72.26</td>
<td>43.01</td>
<td>64.94</td>
<td>46.92</td>
</tr>
<tr>
<td>Large Dense</td>
<td>MTL-UViT</td>
<td>72.99</td>
<td>43.37</td>
<td>64.79</td>
<td>47.21</td>
</tr>
<tr>
<td>Dense</td>
<td>TAPS</td>
<td>77.23</td>
<td>40.58</td>
<td>63.41</td>
<td>43.72</td>
</tr>
<tr>
<td>Sparse</td>
<td>AdaMV-MoE (Ours)</td>
<td>79.65</td>
<td>44.14</td>
<td>65.54</td>
<td>48.17</td>
</tr>
</tbody>
</table>

**Dataset and Task.** We examine our methods on ImageNet [24] and MS COCO 2017 [47] datasets, for classification and detection & segmentation tasks respectively. ImageNet contains 1.25M training images and 50K testing images of 1,000 classes, while MS COCO 2017 has 118K training images and 5K validation images. The input resolution is 224 × 224 for classification and 640 × 640 for object detection & instance segmentation.

**Baselines.** To support the effectiveness of our proposals, we consider three groups of comparison baselines: (1) Dense ViTs for single-task learning (STL), i.e., ViT for CLS and ViT for OD & IS. (2) Dense ViTs for multi-task learning, i.e., MTL-ViT. It shares the full feature extractor with task-specific heads attached. Large Dense implies a strengthened baseline that has a larger hidden dimension and more parameter counts as shown in Appendix A1. (3) TAPS [71], a recent state-of-the-art multi-task approach that advocates the task adaptive parameter sharing.

**Training and Evaluation Details.** The single-task learning baselines are trained with a batch size of 1, 024 and 256 for classification and object detection & instance segmentation, respectively. For MTL training, the batch sizes for the two tasks are 1, 024 and 128, respectively. During training, data augmentations are applied for both tasks. For classification, we use CutMix [83] and MixUp [65]. As for detection and segmentation, random scaling augmentations are employed to enhance the input samples.

Our ViTs are optimized with AdamW [50], a weight decay of {6 × 10⁻³, 1 × 10⁻⁴, 5 × 10⁻⁴}, initial learning rates (LR) of 3 × 10⁻³, {20, 2, 10}K iterations warm-up, and a cosine LR decay schedule for {CLS, OD & IS, MTL}. Multiple loss functions are involved in the model training, e.g., a cross-entropy loss for classification as well as the {class, box, mask} losses from Mask-RCNN for detection and segmentation. The default hyperparameters of [16] are inherited in our cases. As for AdaMV-MoE, we add two auxiliary loss terms of importance and loading regularizations [58] for router network learning. The coefficients of these two losses are set to 5 × 10⁻³ [58]. The value of Δn is set as 2000 when applying AES, and 1% of the training samples is randomly held out to construct a validation set D_{val} for AES. For TAPS and AdaMV-MoE, we first train the network to solve the classification task for 300K steps, and simultaneously train with two tasks (CLS and OD & IS) with additional 200K steps. For {Dense, Large Dense}, it is trained for 200K iterations. The ablation studies on the training steps are in Appendix A2. Each experiment uses 16 ~ 64 and 8 TPU-v3 for training and inference.

To evaluate the performance of trained ViTs, we report the test accuracy for classification, the validation {AP, AP_{50}, AP_{75}} for object detection tasks [16], and the validation AP^{mask} for instance segmentation [16]. Additionally, the number of activated parameters (in millions) is calcu-
Figure 3. The routing specialization of AdaMV-MoE at the fined-grained patch level. Upper shows the routing decisions of classification with ImageNet samples; Bottom presents the routing decisions of object detection and instance segmentation with COCO samples. Here we only visualize the top-2 selected experts whose indexes are indicated by the color of the patch’s boundary and content.

4.2. Superior Multi-Task Vision Recognition Performance of AdaMV–MoE

Comparisons with STL and MTL Approaches. We choose ViT-Small*/Small/Base and UViT-Base network architectures, considering their vanilla (Dense), widened (Large Dense), and SMoE (Sparse) variants. All methods are examined on the benchmark of ImageNet classification and COCO object detection & instance segmentation. The comparison results are collected in Table 1, where the following observations can be drawn:

1. Our AdaMV–MoE demonstrates great advantages with a clear performance margin compared to MTL baselines with a shared ViT feature extractor, i.e., (Dense, Large Dense) MTL-ViT.

In detail, AdaMV–MoE obtains \{(4.69%, 2.67%), (2.69%, 1.30%), (1.75%, 0.79%)\}, \{(9.06%, 6.54%), (0.73%, 0.09%), (0.87%, 0.05%)\}, \{(4.41%, 4.18%), (0.07%, 0.23%), (0.19%, 0.26%)\}, \{(7.39%, 6.66%), (1.13%, 0.87%), (0.84%, 0.89%)\} of \{Accuracy (%), AP (%), APMask (%)\} improvements for ViT-Small*/Small/Base and UViT-Base, respectively. It validates the effectiveness of our proposals.

2. AdaMV–MoE adaptively allocates adequate network capacity to resolve classification, detection, and segmentation tasks by activating different amounts of model parameters. For instance, our proposals spend fewer parameter counts for CLS while more parameter budgets for the challenging OD & IS tasks, e.g., 29.65M and 34.97M in the case of ViT-Small, which aligns with our intuition.

3. In additional, AdaMV–MoE consistently surpasses a recent SoTA MTL approach, i.e., TAPS [71], by \{0.14% ~ 3.78% Accuracy, 0.29% ~ 4.78% AP, 0.17% ~ 3.38% APMask\} on ImageNet and COCO datasets across four ViT backbones. Meantime, with ViT-Small*, it reaches competitive results compared to the single-task learning baselines, further showing the superiority of our algorithms.
Ablation Study of AdaMV-MoE. To investigate the contributions of each component in AdaMV-MoE, comprehensive experiments are conducted with ViT-Small* on multi-task vision recognition. As shown in Table 3 and Table 4, we conduct ablation on the router design, the need for adaptive network capacity during MTL, and the number of experts when employing AdaMV-MoE.

Table 3. Ablation studies on AdaMV-MoE of i) router selection, i.e., task-agnostic R v.s. task-dependent R; ii) # used experts, i.e., activating fixed v.s. adaptive number of experts. “Ours w. task-dependent R” and “Ours w. AES” present the same variant, which is also the one used to produce main results in Table 1.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Classification</th>
<th>Detection</th>
<th>Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy(%)</td>
<td>AP(%)</td>
<td>APmask(%)</td>
</tr>
<tr>
<td>MTL-ViT</td>
<td>68.30</td>
<td>36.35</td>
<td>34.01</td>
</tr>
<tr>
<td>MTL-MoE [46]</td>
<td>72.07</td>
<td>38.53</td>
<td>35.24</td>
</tr>
<tr>
<td>Ours w. task-agnostic R</td>
<td>72.56</td>
<td>37.54</td>
<td>34.71</td>
</tr>
<tr>
<td>Ours w. task-dependent R</td>
<td>72.99</td>
<td>39.04</td>
<td>35.76</td>
</tr>
<tr>
<td>Ours w.o. AES</td>
<td>72.04</td>
<td>38.61</td>
<td>35.23</td>
</tr>
<tr>
<td>Ours w. AES</td>
<td>72.99</td>
<td>39.04</td>
<td>35.76</td>
</tr>
</tbody>
</table>

Table 4. Ablation studies on # total experts (E) of our proposed AdaMV-MoE. MTL-ViT is the baseline that takes ViT as a shared backbone and multiple heads for different tasks. The backbone size of MTL-ViT is equal to the one of AdaMV-MoE with E = 1.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Classification</th>
<th>Detection</th>
<th>Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy(%)</td>
<td>AP(%)</td>
<td>APmask(%)</td>
</tr>
<tr>
<td>MTL-ViT</td>
<td>68.30</td>
<td>36.35</td>
<td>34.01</td>
</tr>
<tr>
<td>AdaMV-MoE w. E = 4</td>
<td>71.74</td>
<td>36.35</td>
<td>34.01</td>
</tr>
<tr>
<td>AdaMV-MoE w. E = 8</td>
<td>72.99</td>
<td>39.04</td>
<td>35.76</td>
</tr>
<tr>
<td>AdaMV-MoE w. E = 16</td>
<td>72.69</td>
<td>36.99</td>
<td>34.65</td>
</tr>
<tr>
<td>AdaMV-MoE w. E = 32</td>
<td>72.66</td>
<td>36.30</td>
<td>33.37</td>
</tr>
</tbody>
</table>

Q1: Is the expert selection specialized to different tasks, classes, and image contents? Yes. One key advantage of AdaMV-MoE is that it optimizes how many (i.e., adaptive network capacity) and which (i.e., dynamic routing) experts to activate for each task and input sample during MTL. We examine triple levels of routing specializations from coarse to fine-grained, including task, class, and patch levels.

Q2: Is the route selection normalized to different tasks, classes, and image contents? Yes. One key advantage of AdaMV-MoE is that it optimizes how many (i.e., adaptive network capacity) and which (i.e., dynamic routing) experts to activate for each task and input sample during MTL. We examine triple levels of routing specializations from coarse to fine-grained, including task, class, and patch levels.

4.3. In-Depth Dissection of AdaMV-MoE

Given the superiority of our AdaMV-MoE, we further offer an in-depth dissection by studying its i) specialization, ii) routing quality, iii) adequate positions to introduce SMoE layers, and iv) mitigation effects on gradient conflicts from multiple training objectives.
and instance segmentation, a clear patch-wise specialization is presented. For example, different image contents like the object boundary, the main body of objects, and the background are processed by distinctive and particular subsets of experts. In this way, the task is divided and conquered.

Figure 5. Analysis on the representation collapse of the hidden states from router networks. The diversity of these hidden states is calculated with Gaussian kernel density estimation and then is visualized as circle heatmaps. Darker areas have more concentrated features. A more uniformly distributed circle heatmap means a more balanced expert usage and a lower risk of representation collapse. Results are produced by AdaMV-MoE with ViT-Small.

**Q2: What is the quality of learned routing policies?**

*High quality in terms of less routing collapse and good policy convergence.* We study AdaMV-MoE’s routing policies from the perspectives of collapse [19] and convergence [21].

To study its routing collapse, we plot the diversity of hidden features from router networks as shown in Figure 5. The heatmaps from both CLS and OD & IS demonstrate uniformly distributed hidden states, suggesting a balanced expert assignment and less representation collapse [19] which are consistent with observations concluded from Figure 4.

For the policy convergence, we choose a shallow and a deep SMoE layer of our ViT-Small∗, and present the Hamming distance between routing decisions from different training iterations in Figure 6. We notice the routing converges well after the first 25K iterations and the shallow SMoE layer has a higher convergence speed, which enjoys less routing fluctuation and better sample efficiency [21]. Such high-quality routing policies potentially explain the superiority of AdaMV-MoE.

**Q3: Where should we insert the SMoE layers?**

*Later layers.* For a vision transformer, there are various options to replace the original ViT layer with an SMoE layer. We compare different design choices such as adopting SMoE in the Early, Middle, Later, and Every Two layers, where each AdaMV-MoE variant has half ViT and half SMoE layers. Results in Table 5 reveal that only enforcing SMoE to early layers incurs inferior MTL performance. A possible reason is that early layers are usually responsible for learning common features like basic shapes or colors, which should be shared across classes during vision recognition tasks.

Table 5. Ablation studies on positions of introduced SMoE layers. Results are produced by AdaMV-MoE with ViT-Small∗.

<table>
<thead>
<tr>
<th>Settings of AdaMV-MoE</th>
<th>Classification</th>
<th>Detection</th>
<th>Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy(%)</td>
<td>AP(%)</td>
<td>AP(mask)(%)</td>
<td></td>
</tr>
<tr>
<td>Early Layers</td>
<td>69.38</td>
<td>37.76</td>
<td>34.67</td>
</tr>
<tr>
<td>Middle Layers</td>
<td>72.67</td>
<td>38.49</td>
<td>35.04</td>
</tr>
<tr>
<td>Later Layers</td>
<td>73.19</td>
<td>38.00</td>
<td>34.92</td>
</tr>
<tr>
<td>Every Two Layers</td>
<td>72.99</td>
<td>39.04</td>
<td>35.76</td>
</tr>
</tbody>
</table>

**Q4: Does AdaMV-MoE alleviate the issue of gradient conflicts from diverse tasks?**

*Yes.* First, AdvMV-MoE naturally disentangles parameter spaces for different tasks thanks to its sparse and conditional computing manner. Second, as shown in Figure 7, for the common parameters for all tasks, the gradient conflicts are generally reduced by our proposals, *e.g.*, less negative and more positive cosine distance between training gradients from CLS and OD.

Figure 7. The distribution of cosine distance between training gradients computed from classification and detection objectives. Ours (AdaMV-MoE) and Baseline (MTL-ViT) results of ViT-Small’s last ViT layer are collected.

5. Conclusion

In this paper, we present an adaptive multi-task vision recognition framework, aiming at the automatic design of used network capacity for distinctive tasks. Our proposals seamlessly customize the current SoTA MTL Mixture-of-Experts, and optimize the task-specific model size by adaptively activating or deactivating experts. Extensive investigations across various ViT architectures consistently demonstrate the performance improvements from our approach, on the challenging benchmark of ImageNet classification and COCO object detection & instance segmentation. Future work includes the extension of multi-modal multi-task learning like “Pathway” systems.

Acknowledgement

Z. Wang is supported by a Google Research Scholar award and a Google TensorFlow Model Garden Award.
References


[55] Pedro Morgado and Nuno Vasconcelos. Nettailor: Tuning the architecture, not just the weights. In Proceedings of
the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3044–3054, 2019. 2


[75] Lemeng Wu, Dilin Wang, and Qiang Liu. Splitting steepest descent for growing neural architectures. Advances in neural information processing systems, 32, 2019. 4


