AdaMV-MoE: Adaptive Multi-Task Vision Mixture-of-Experts

Tianlong Chen[1*†], Xuxi Chen[1*], Xianzhi Du[2†], Abdullah Rashwan[3], Fan Yang[3]
Huizhong Chen[3], Zhangyang Wang[1], Yeqing Li[3]
[1]University of Texas at Austin, [2]Apple, [3]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].

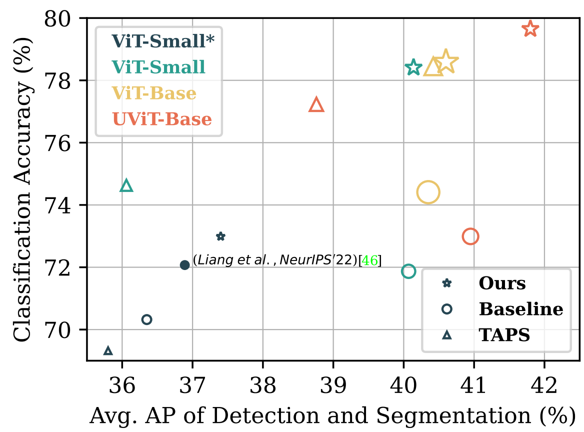

Figure 1. The multi-task vision recognition performance of various ViT architectures on ImageNet classification (y-axis), COCO object detection and instance segmentation (x-axis) benchmarks. Averaged results of detection's AP (%) and segmentation AP$^{\text{mask}}$ (%) are reported. Markers ☆ and ❍ denote ours and baseline approaches, respectively. A larger marker indicates more floating point operations (FLOPs) are used for inference. ViT-Small* is a reduced backbone variant with half transformer layers.

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

*Equal Contribution. †Work done while at Google.

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 32×32 for CIFAR [40] and 224×224 for ImageNet [24], while object detection is normally evaluated on the COCO [47] dataset with a higher resolution of 640×640 or 892×892. 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 subtask. 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 $\{6.66\% \sim 7.39\%$ accuracy, $0.87\% \sim 1.13\%$ AP, $0.84\% \sim 0.89\%$ $\text{AP}^{\text{mask}}\}$ 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, 23, 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)

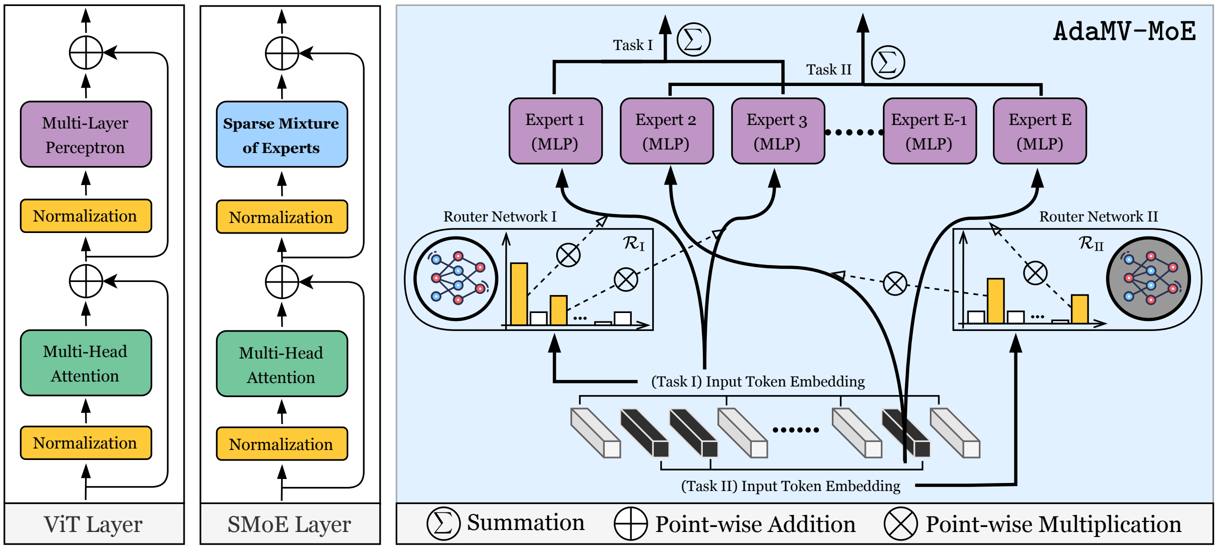

Figure 2. Overview of our framework. `AdaMV-MoE` contains both ViT (*left*) and SMoE (*middle*) layer, where the SMoE layer is built by replacing the original multi-layer perceptron (MLP) with a sparsely activated mixture of experts (MLPs). It enables multi-task learning by leveraging task-specific router networks (*right*). Each router network determines **how many** (*adaptive*) and **which** (*specialized*) experts are appropriate to activate the given task.

and embraces the conditional computation in an input-dependent way [37, 39, 12, 82]. The earliest variant of MoEs densely activates all experts for each input, and therefore it is computation-intensive [26]. Later on, [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 [27]. The effectiveness of SMoEs has been widely proved in various NLP [62, 44, 91, 88, 93, 38] and vision [58, 26, 2, 30, 74, 79, 1, 57] tasks. Particularly, the pioneering work [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. [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 [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 [3, 32, 51, 53, 15]. Yet it is hard to reach unless explicitly enforcing specialization and trimming down the redundancy among experts [13] like pre-defining a diverse expert assignment [22] or involving multiple routing policies [32]. iii) Representation collapse. Naïvely trained SMoE is prone to load imbalance, *e.g.*, only a few experts are frequently used while the others are scarcely activated. To alleviate this issue, [62] adds Gaussian noises to router networks; [44, 27] propose an auxiliary loss as the regularization; [45] formulates and solves a balanced linear assignment problem; [91] distributes the top-k relevant input for each expert; [59, 93] adopt deterministic hashing and stochastic routing; and [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, [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 [3], digital number images (MNIST) [32], and recommendation systems [51]. [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 [62] is proposed to scale up the model capacity while maintaining low per-inference costs. In this work, we consider SMoE for ViTs [25, 58], which inserts SMoE layers into every other transformer block. The SMoE layer contains a router network $\mathcal{R}$ and several experts $f_1, f_2, \cdots, f_{\mathrm{E}}$, where E is the number of experts. The expert module can be a few fully connected [62, 58] or convolutional layers [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 [29, 20] demonstrate their significance as memory networks to store substantial knowledge.

Another key component in SMoE layers, *i.e.*, $\mathcal{R}$, activates the top-k expert networks with the largest scores $\mathcal{R}(\boldsymbol{x})_i$ associated with input embedding $\boldsymbol{x}$, 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:

$$\boldsymbol{y}=\sum_{i=1}^{k}\mathcal{R}(\boldsymbol{x})_i\cdot f_i(\boldsymbol{x}),\mathcal{R}(\boldsymbol{x})=\texttt{TopK}(\texttt{softmax}(g(\boldsymbol{x})),k),$$

$$\texttt{TopK}(\boldsymbol{v},k)=\begin{cases}\boldsymbol{v} & \text{if } \boldsymbol{v} \text{ is the top } k\\ 0 & \text{otherwise}\end{cases}$$

where $f_i(\boldsymbol{x})$ stands for the feature representations produced from the expert f_i, which is weighted by $\mathcal{R}(\boldsymbol{x})_i$ to form the final output $\boldsymbol{y}$. The network g is the learnable part within a router $\mathcal{R}$ 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 $\mathcal{R}_j$ represents the router for the task j, and all expert networks $\{f_i\}|_{i=1}^{\mathrm{E}}$ are shared across tasks. The SMoE equipped with task-dependent router networks is defined as:

$$\boldsymbol{y}_j=\sum_{i=1}^{k_j}\mathcal{R}_j(\boldsymbol{x})_i\cdot f_i(\boldsymbol{x}),\mathcal{R}_j(\boldsymbol{x})=\texttt{TopK}(\texttt{softmax}(g_j(\boldsymbol{x})),k_j),$$

where k_j and $\boldsymbol{y}_j$ 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 $\mathcal{L}_{\mathrm{val}}^j$ on the validation set. If $\mathcal{L}_{\mathrm{val}}^j$ does not decay in the next Δn iterations, then we expand the activated model size by updating $k_j=k_j+1$.

Algorithm 1 Adaptive Expert Selection (AES).

Input: Expert networks f_i ($i\in\{1,2,\cdots,\mathrm{E}\}$, routers $\mathcal{R}_j$ (j is the task index), the validation set $\mathcal{D}_{\mathrm{val}}^j$ for task j, the objective function $\mathcal{L}_{\mathrm{val}}^j$ on the validation set.
for a given task j **do**
  Initial the number of selected experts as $k_j \leftarrow 1$;
  Initial an indicator `Improved` as `True`;
  Initial the current best validation loss as $\mathcal{L}_{\mathrm{val(best)}}^j \leftarrow \infty$;
  while `True` **do**
    if $\mathcal{L}_{\mathrm{val(best)}}^j$ does not decrease for Δn iterations **then**
      if `not improved` **then**
        `break`;
      else
        $k_j \leftarrow k_j+1$; `improved` $\leftarrow$ `False`;
      end if
    end if
    Continue training the model;
    if $\mathcal{L}_{\mathrm{val}}^j < \mathcal{L}_{\mathrm{val(best)}}^j$ **then**
      $\mathcal{L}_{\mathrm{val(best)}}^j \leftarrow \mathcal{L}_{\mathrm{val}}^j$; `improved` $\leftarrow$ `True`;
    end if
  end while
  $k_j=k_j-1$ and fix k_j;
  Continue training to the target number of iterations.
end for
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 $\mathcal{L}_{\mathrm{val}}^j$ is larger than the previous best validation loss $\mathcal{L}_{\mathrm{val(best)}}^j$, 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 2. Detailed model sizes of (*Dense*) ViT variants.

Backbones	# Transformer Layers	# Attention Heads	Hidden Dimension	MLP Dimension
ViT-Small*	6	6	384	1536
ViT-Small	12	6	384	1536
ViT-Base	12	6	768	3072
UViT-Base	18	6	384	1536

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".

Backbone	Method		Classification	Object Detection			Instance Segmentation	# Parameters (M)	
			Accuracy(%)	AP(%)	AP_{50}(%)	AP_{75}(%)	AP^{mask}(%)	CLS	OD & IS
ViT-Small*	*Dense*	ViT for CLS	73.00	N.A.	N.A.	N.A.	N.A.	11.10	N.A.
	Dense	ViT for OD & IS	N.A.	39.75	61.71	42.77	36.10	N.A.	13.22
	Dense	MTL-ViT	68.30	36.35	58.79	38.86	34.01	13.67	
	Large Dense	MTL-ViT	70.32	37.74	60.27	40.58	34.97	20.41	
	Dense	TAPS	69.32	36.66	58.97	38.55	34.94	16.62	18.25
	Sparse	AdaMV-MoE **(Ours)**	**72.99**	**39.04**	**61.16**	**42.43**	**35.76**	16.33	19.00
ViT-Small	*Dense*	MTL-ViT	69.34	41.43	63.45	45.13	37.25	24.32	
	Large Dense	MTL-ViT	71.87	42.07	**64.48**	45.66	38.07	37.03	
	Dense	TAPS	74.63	37.38	60.15	39.89	34.74	27.86	30.22
	Sparse	AdaMV-MoE **(Ours)**	**78.41**	**42.16**	64.33	**45.73**	**38.12**	29.65	34.97
ViT-Base	*Dense*	MTL-ViT	74.18	42.63	64.31	46.53	38.30	91.10	
	Large Dense	MTL-ViT	74.41	42.47	64.19	46.12	38.23	123.87	
	Dense	TAPS	78.45	42.51	65.28	45.87	38.32	105.26	108.40
	Sparse	AdaMV-MoE **(Ours)**	**78.59**	**42.70**	**65.12**	**46.05**	**38.49**	112.37	123.00
UViT-Base	*Dense*	MTL-UViT	72.26	43.01	64.94	46.92	38.67	34.96	
	Large Dense	MTL-UViT	72.99	43.27	64.79	47.21	38.62	53.66	
	Dense	TAPS	77.23	40.58	63.41	43.72	36.94	39.68	41.45
	Sparse	AdaMV-MoE **(Ours)**	**79.65**	**44.14**	**65.54**	**48.17**	**39.51**	42.95	50.94

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 1.28M 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 $\{6 \times 10^{-3}, 1 \times 10^{-4}, 5 \times 10^{-4}\}$, 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 $\mathcal{D}_{\text{val}}$ 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 $16 \sim 64$ 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-

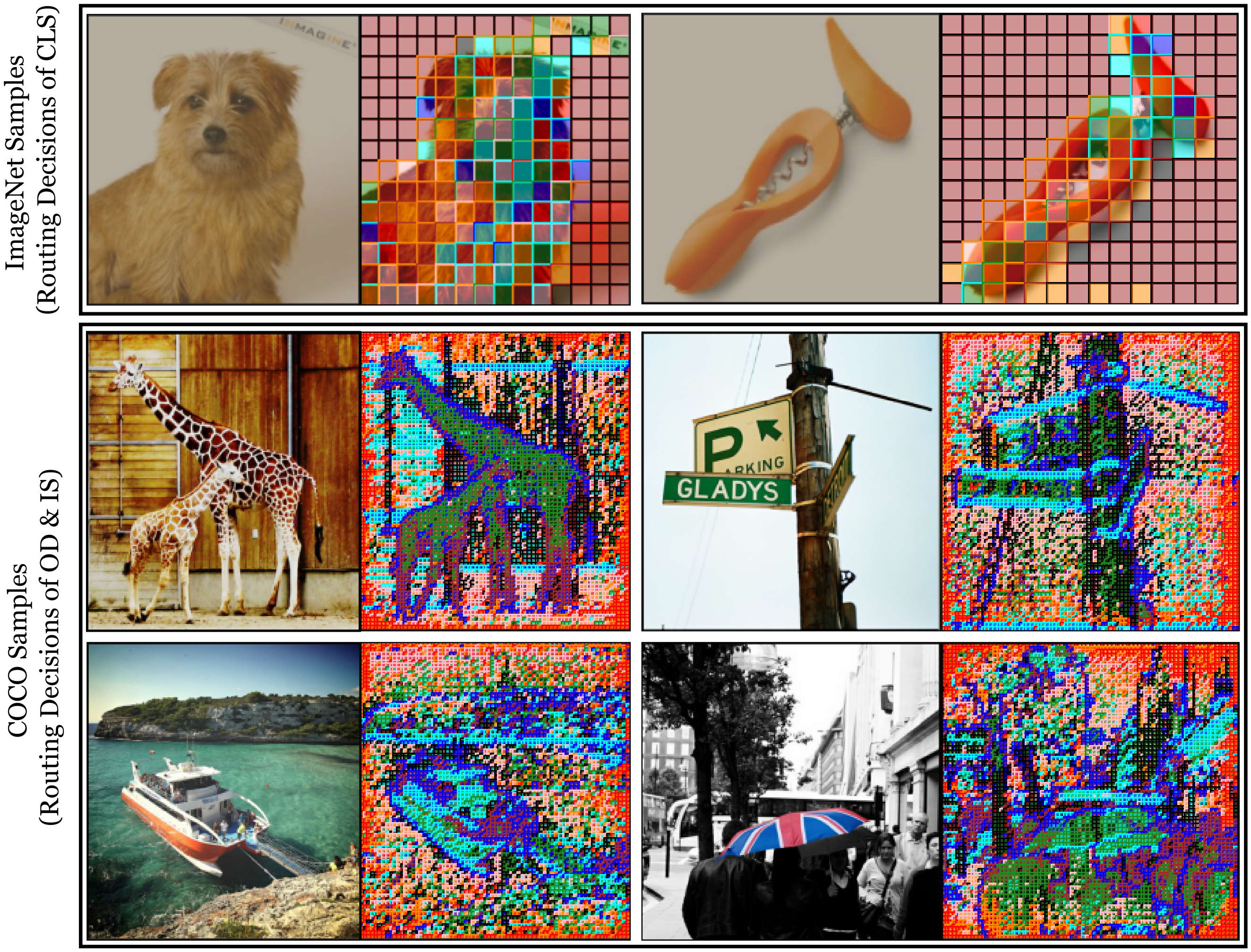

Figure 3. The routing specialization of `AdaMV-MoE` at the fined-grained patch level. *Upper* shows the routing decisions of classification with ImageNet samples; *Bottom* presents the routing decisions of object detection and instance segmentation with COCO samples. Here we only visualize **the top-2 selected experts whose indexes are indicated by the color of the patch's boundary and content.**

lated to imply the used model capacity for each task.

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: ❶ 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 (%), AP^{mask} (%)} improvements for ViT-Small*/Small/Base and UViT-Base, respectively. It validates the effectiveness of our proposals. ❷ `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. ❸ In additional, `AdaMV-MoE` consistently surpasses a recent SoTA MTL approach, *i.e.,* TAPS [71], by $\{0.14\% \sim 3.78\%$ Accuracy, $0.29\% \sim 4.78\%$ AP, $0.17\% \sim 3.38\%$ $\text{AP}^{\text{mask}}\}$ 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 $\mathcal{R}$ v.s. task-dependent $\mathcal{R}$; ii) # used experts, *i.e.*, activating fixed v.s. adaptive number of experts. "Ours w. task-dependent $\mathcal{R}$" and "Ours w. AES" present the same variant, which is also the one used to produce main results in Table 1.

Settings	Classification	Detection	Segmentation
	Accuracy(%)	AP(%)	AP^{mask}(%)
MTL-ViT	68.30	36.35	34.01
MTL-MoE [46]	72.07	38.53	35.24
Ours w. task-agnostic $\mathcal{R}$	72.56	37.54	34.71
Ours w. task-dependent $\mathcal{R}$	72.99	39.04	35.76
Ours w.o. AES	72.04	38.61	35.23
Ours w. AES	72.99	39.04	35.76

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 $\text{E} = 1$.

Settings	Classification	Detection	Segmentation
	Accuracy(%)	AP(%)	AP^{mask}(%)
MTL-ViT	68.30	36.35	34.01
AdaMV-MoE w. $\text{E} = 4$	71.74	36.35	34.01
AdaMV-MoE w. $\text{E} = 8$	72.99	39.04	35.76
AdaMV-MoE w. $\text{E} = 16$	72.69	36.99	34.05
AdaMV-MoE w. $\text{E} = 32$	72.66	36.30	33.37

$\triangleright$ *Task-agnostic versus task-dependent routers* $\mathcal{R}$. Results in Table 3 tell that task-dependent routing policies benefit more than their task-agnostic counterpart, and enlarge the performance gains compared to the MTL-ViT baseline.

$\triangleright$ *With or without adaptive expert selection (AES).* Equipped with the AES, the activated model size is optimized for different tasks, which significantly boosts the MTL performance. To be specific, our w. AES outperforms its variant w.o. AES by $\{0.95\%$ Accuracy, 0.43% AP, 0.53% $\text{AP}^{\text{mask}}\}$ improvements, and a recently invented MTL-MoE [46] by $\{0.92\%$ Accuracy, 0.51% AP, 0.52% $\text{AP}^{\text{mask}}\}$ gains, which evidence the necessity of a customized network capacity for each task. Note that both ours w.o. AES and MTL-MoE [46] adopt a fixed and unified model size (or # selected experts) across all vision tasks, which potentially incurs inferior results.

$\triangleright$ *The number of experts.* Being one of the most important hyperparameters in an SMoE design, E roughly reflects the size of overall parameter spaces that allow explorations via dynamic routing. Table 4 reports AdaMV-MoE's results with $\{1, 4, 8, 16, 32\}$ experts, where AdaMV-MoE degrades into MTL-ViT if setting $\text{E} = 1$. We find that the performance of AdaMV-MoE saturates with more than 8 experts, and $\text{E} = 8$ seems a "sweet-point" in our multi-task vision recognition benchmark.

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.

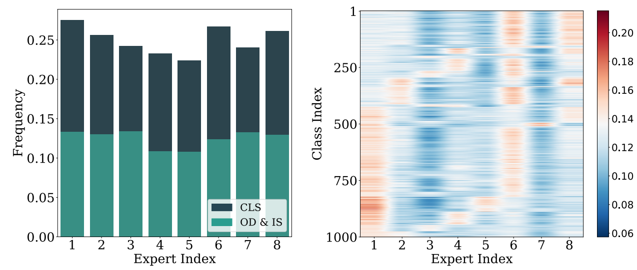

Figure 4. Analysis on the routing specialization at the task (*Left*) and class (*Right*) levels. The frequency of expert usage and the class-wise usage of classification are recorded in *Left* and *Right* figures, respectively. Visualizations are produced by AdaMV-MoE with ViT-Small. More qualitative results are in Appendix A2.

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.

$\triangleright$ *Task-level specialization.* From Figure 4 (*Left*), we find that ❶ there is an overall balanced loading across experts, suggesting a sufficient utilization of all model parameters; ❷ relatively, CLS prefers expert 1 & 6 and OD & IS use expert 1, 3, & 7 more, according to the frequency.

$\triangleright$ *Class-level specialization.* Based on Figure 4 (*Right*) which presents the class-wise expert usage of the last SMoE layer in AdaMV-MoE for classification, we observe that the expert 6 is preferred by most classes and other expert selections seem to correlate with class types, which coincide with the findings in [58]. Similar observations also exist for OD & IS, as shown in Appendix A2.

$\triangleright$ *Patch-level specialization.* In Figure 3, we visualize the expert assignments of AdaMV-MoE w. ViT-Small for each input patch. Specifically, the top-1 and top-2 of activated experts are implied by the color of the patch's boundary and content respectively, where different colors represent diverse experts. ❶ For classification, most patches from the background are assigned to two specific experts associated with the black patch boundary and red patch content, while varied experts are leveraged to deal with the main object in the foreground. ❷ As for the object detection 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.

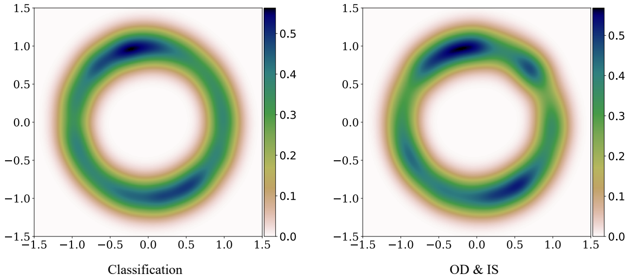

Figure 5. Analysis on the representation collapse of the hidden states from router networks. The diversity of these hidden states is calculated with Gaussian kernel density estimation and then is visualized as circle heatmaps. Darker areas have more concentrated features. A more uniformly distributed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

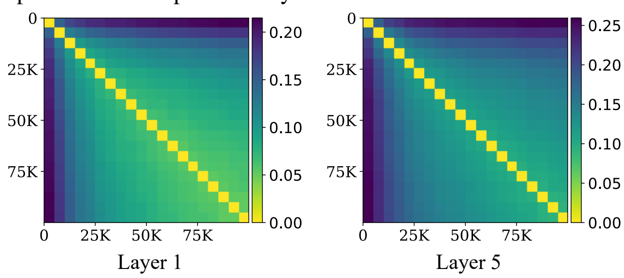

Figure 6. The convergence pattern of routing policies. Hamming distance results between routing decisions from different training iterations are produced by `AdaMV-MoE` with a ViT-Small* backbone. A shallow and a deep SMoE layer are examined.

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]. ❶ 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. ❷ 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*.

Settings of `AdaMV-MoE`	Classification	Detection	Segmentation
	Accuracy(%)	AP(%)	$\mathrm{AP^{mask}}$(%)
Early Layers	69.38	37.76	34.67
Middle Layers	72.67	38.49	35.04
Later Layers	73.19	38.00	34.92
Every Two Layers	72.99	39.04	35.76

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.

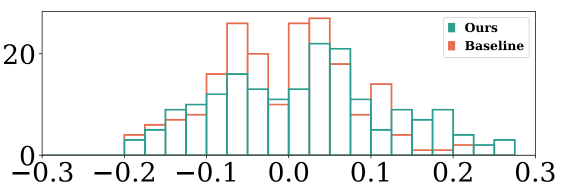

Figure 7. The distribution of `cosine` distance between training gradients computed from classification and detection objectives. Ours (`AdaMV-MoE`) and Baseline (MTL-ViT) results of ViT-Small's last ViT layer are collected.

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

[1] Alhabib Abbas and Yiannis Andreopoulos. Biased mixtures of experts: Enabling computer vision inference under data transfer limitations. *IEEE Transactions on Image Processing*, 29:7656–7667, 2020. 3

[2] Karim Ahmed, Mohammad Haris Baig, and Lorenzo Torresani. Network of experts for large-scale image categorization. In *European Conference on Computer Vision*, pages 516–532. Springer, 2016. 3

[3] Raquel Aoki, Frederick Tung, and Gabriel L Oliveira. Heterogeneous multi-task learning with expert diversity. *arXiv preprint arXiv:2106.10595*, 2021. 3

[4] BJ Bakker and TM Heskes. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 2003. 2

[5] Josh Beal, Hao-Yu Wu, Dong Huk Park, Andrew Zhai, and Dmitry Kislyuk. Billion-scale pretraining with vision transformers for multi-task visual representations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 564–573, 2022. 1

[6] Rodrigo Berriel, Stephane Lathuillere, Moin Nabi, Tassilo Klein, Thiago Oliveira-Santos, Nicu Sebe, and Elisa Ricci. Budget-aware adapters for multi-domain learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 382–391, 2019. 2

[7] Deblina Bhattacharjee, Tong Zhang, Sabine Süsstrunk, and Mathieu Salzmann. Mult: An end-to-end multitask learning transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12031–12041, 2022. 1, 2

[8] Valerio Biscione and Jeffrey Bowers. Learning translation invariance in cnns. *arXiv preprint arXiv:2011.11757*, 2020. 2

[9] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1483–1498, 2019. 2, 4

[10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1

[11] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12299–12310, 2021. 2

[12] Ke Chen, Lei Xu, and Huisheng Chi. Improved learning algorithms for mixture of experts in multiclass classification. *Neural networks*, 12(9):1229–1252, 1999. 3

[13] Tianyu Chen, Shaohan Huang, Yuan Xie, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. Task-specific expert pruning for sparse mixture-of-experts. *arXiv preprint arXiv:2206.00277*, 2022. 3

[14] Tianlong Chen, Zhenyu Zhang, Yu Cheng, Ahmed Awadallah, and Zhangyang Wang. The principle of diversity: Training stronger vision transformers calls for reducing all levels of redundancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12020–12030, 2022. 3

[15] Tianlong Chen, Zhenyu Zhang, AJAY KUMAR JAISWAL, Shiwei Liu, and Zhangyang Wang. Sparse moe as the new dropout: Scaling dense and self-slimmable transformers. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[16] Wuyang Chen, Xianzhi Du, Fan Yang, Lucas Beyer, Xiaohua Zhai, Tsung-Yi Lin, Huizhong Chen, Jing Li, Xiaodan Song, Zhangyang Wang, et al. A simple single-scale vision transformer for object localization and instance segmentation. *arXiv preprint arXiv:2112.09747*, 2021. 2, 4, 5, A13

[17] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR, 2018. 1

[18] Bowen Cheng, Yunchao Wei, Honghui Shi, Rogerio Feris, Jinjun Xiong, and Thomas Huang. Revisiting rcnn: On awakening the classification power of faster rcnn. In *Proceedings of the European conference on computer vision (ECCV)*, pages 453–468, 2018. 2

[19] Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, and Furu Wei. On the representation collapse of sparse mixture of experts. *arXiv preprint arXiv:2204.09179*, 2022. 4, 8

[20] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021. 3

[21] Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. Stablemoe: Stable routing strategy for mixture of experts. *arXiv preprint arXiv:2204.08396*, 2022. 8

[22] Yong Dai, Duyu Tang, Liangxin Liu, Minghuan Tan, Cong Zhou, Jingquan Wang, Zhangyin Feng, Fan Zhang, Xueyu Hu, and Shuming Shi. One model, multiple modalities: A sparsely activated approach for text, sound, image, video and code. *arXiv preprint arXiv:2205.06126*, 2022. 3

[23] Hal Daumé III. Bayesian multitask learning with latent hierarchies. *arXiv preprint arXiv:0907.0783*, 2009. 2

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 5

[25] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 3, 4, 5

[26] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013. 3

[27] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*, 2021. 3, 4

[28] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3205–3214, 2019. 1, 2

[29] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020. 3

[30] Sam Gross, Marc'Aurelio Ranzato, and Arthur Szlam. Hard mixtures of experts for large scale weakly supervised vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6865–6873, 2017. 3

[31] Shashank Gupta, Subhabrata Mukherjee, Krishan Subudhi, Eduardo Gonzalez, Damien Jose, Ahmed H Awadallah, and Jianfeng Gao. Sparsely activated mixture-of-experts are robust multi-task learners. *arXiv preprint arXiv:2204.07689*, 2022. 3

[32] Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed Chi. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[33] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927, 2019. 2

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[35] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016. A13

[36] Laurent Jacob, Jean-philippe Vert, and Francis Bach. Clustered multi-task learning: A convex formulation. *Advances in neural information processing systems*, 21, 2008. 2

[37] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991. 3

[38] Hao Jiang, Ke Zhan, Jianwei Qu, Yongkang Wu, Zhaoye Fei, Xinyu Zhang, Lei Chen, Zhicheng Dou, Xipeng Qiu, Zikai Guo, et al. Towards more effective and economic sparsely-activated model. *arXiv preprint arXiv:2110.07431*, 2021. 3

[39] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994. 3

[40] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[41] Abhishek Kumar and Hal Daume III. Learning task grouping and overlap in multi-task learning. *arXiv preprint arXiv:1206.6417*, 2012. 2

[42] Dong-Gyu Lee. Fast drivable areas estimation with multi-task learning for real-time autonomous driving assistant. *Applied Sciences*, 11(22):10713, 2021. 1

[43] Su-In Lee, Vassil Chatalbashev, David Vickrey, and Daphne Koller. Learning a meta-level prior for feature relevance from multiple related tasks. In *Proceedings of the 24th international conference on Machine learning*, pages 489–496, 2007. 2

[44] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020. 3

[45] Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. Base layers: Simplifying training of large, sparse models. In *International Conference on Machine Learning*, pages 6265–6274. PMLR, 2021. 3

[46] Hanxue Liang, Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tianlong Chen, Yu Cheng, Cong Hao, and Zhangyang Wang. M³vit: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design. In *36th Annual Conference on Neural Information Processing System*, 2022. 1, 2, 3, 7

[47] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 5

[48] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019. 1, 2

[49] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1

[50] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 5

[51] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1930–1939, 2018. 1, 3

[52] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3994–4003, 2016. 1, 2

[53] Sarthak Mittal, Yoshua Bengio, and Guillaume Lajoie. Is a modular architecture enough? *arXiv preprint arXiv:2206.02713*, 2022. 3

[54] Eslam Mohamed and Ahmad El Sallab. Spatio-temporal multi-task learning transformer for joint moving object detection and segmentation. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1470–1475. IEEE, 2021. 2

[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
[56] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*, 2015. 1
[57] Svetlana Pavlitskaya, Christian Hubschneider, Michael Weber, Ruby Moritz, Fabian Huger, Peter Schlicht, and Marius Zollner. Using mixture of expert models to gain insights into semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 342–343, 2020. 3
[58] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34, 2021. 3, 4, 5, 7, A13
[59] Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason E Weston. Hash layers for large sparse models. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 3
[60] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4822–4829, 2019. 1, 2
[61] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Video multitask transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2
[62] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 3, 4
[63] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, pages 9120–9132. PMLR, 2020. 2
[64] Charles Sun, Jedrzej Orbik, Coline Manon Devin, Brian H. Yang, Abhishek Gupta, Glen Berseth, and Sergey Levine. Fully autonomous real-world reinforcement learning with applications to mobile manipulation. In *CoRL*, volume 164 of *Proceedings of Machine Learning Research*, pages 308–319. PMLR, 2021. 1
[65] Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip S Yu, and Lifang He. Mixup-transformer: dynamic data augmentation for nlp tasks. *arXiv preprint arXiv:2010.02394*, 2020. 5
[66] Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: Learning what to share for efficient deep multi-task learning. *Advances in Neural Information Processing Systems*, 33:8728–8740, 2020. 2
[67] Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-task bayesian optimization. *Advances in neural information processing systems*, 26, 2013. 1
[68] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, volume 139, pages 10347–10357, July 2021. 1
[69] Trieu H Trinh, Minh-Thang Luong, and Quoc V Le. Selfie: Self-supervised pretraining for image embedding. *arXiv preprint arXiv:1906.02940*, 2019. 2
[70] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *European Conference on Computer Vision*, pages 527–543. Springer, 2020. 2
[71] Matthew Wallingford, Hao Li, Alessandro Achille, Avinash Ravichandran, Charless Fowlkes, Rahul Bhotika, and Stefano Soatto. Task adaptive parameter sharing for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7561–7570, 2022. 2, 5, 6
[72] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12695–12705, 2020. 2
[73] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X Yu. Long-tailed recognition by routing diverse distribution-aware experts. *arXiv preprint arXiv:2010.01809*, 2020. 3
[74] Xin Wang, Fisher Yu, Lisa Dunlap, Yi-An Ma, Ruth Wang, Azalia Mirhoseini, Trevor Darrell, and Joseph E Gonzalez. Deep mixture of experts via shallow embedding. In *Uncertainty in artificial intelligence*, pages 552–562. PMLR, 2020. 3
[75] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021. A13
[76] Lemeng Wu, Dilin Wang, and Qiang Liu. Splitting steepest descent for growing neural architectures. *Advances in neural information processing systems*, 32, 2019. 4
[77] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018. 2
[78] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8(1), 2007. 2
[79] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. *Advances in Neural Information Processing Systems*, 32, 2019. 3
[80] Kai Yu, Volker Tresp, and Anton Schwaighofer. Learning gaussian processes from multiple tasks. In *Proceedings of the 22nd international conference on Machine learning*, pages 1012–1019, 2005. 2
[81] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020. 1

[82] Seniha Esen Yuksel, Joseph N. Wilson, and Paul D. Gader. Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1177–1193, 2012. 3

[83] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 5

[84] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11197–11206, 2020. 2

[85] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018. 1, 2

[86] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for semantic segmentation and depth estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 235–251, 2018. 2

[87] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4106–4115, 2019. 2

[88] Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Moefication: Conditional computation of transformer models for efficient inference. *arXiv preprint arXiv:2110.01786*, 2021. 3

[89] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Clustered multi-task learning via alternating structure optimization. *Advances in neural information processing systems*, 24, 2011. 2

[90] Qianyu Zhou, Ke-Yue Zhang, Taiping Yao, Ran Yi, Shouhong Ding, and Lizhuang Ma. Adaptive mixture of experts learning for generalizable face anti-spoofing. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6009–6018, 2022. 3

[91] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. Mixture-of-experts with expert choice routing. *arXiv preprint arXiv:2202.09368*, 2022. 3

[92] Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. Designing effective sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022. 3

[93] Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Jianfeng Gao, and Tuo Zhao. Taming sparsely activated transformer with stochastic experts. In *International Conference on Learning Representations*, 2022. 3