TaskExpert: Dynamically Assembling Multi-Task Representations with Memorial Mixture-of-Experts

Hanrong Ye and Dan Xu$^{\boxtimes}$
Department of Computer Science and Engineering, HKUST
Clear Water Bay, Kowloon, Hong Kong
{hyeae, danxu}@cse.ust.hk

Abstract

Learning discriminative task-specific features simultaneously for multiple distinct tasks is a fundamental problem in multi-task learning. Recent state-of-the-art models consider directly decoding task-specific features from one shared task-generic feature (e.g., feature from a backbone layer), and utilize carefully designed decoders to produce multi-task features. However, as the input feature is fully shared and each task decoder also shares decoding parameters for different input samples, it leads to a static feature decoding process, producing less discriminative task-specific representations. To tackle this limitation, we propose TaskExpert, a novel multi-task mixture-of-experts model that enables learning multiple representative task-generic feature spaces and decoding task-specific features in a dynamic manner. Specifically, TaskExpert introduces a set of expert networks to decompose the backbone feature into several representative task-generic features. Then, the task-specific features are decoded by using dynamic task-specific gating networks operating on the decomposed task-generic features. Furthermore, to establish long-range modeling of the task-specific representations from different layers of TaskExpert, we design a multi-task feature memory that updates at each layer and acts as an additional feature expert for dynamic task-specific feature decoding. Extensive experiments demonstrate that our TaskExpert clearly outperforms previous best-performing methods on all 9 metrics of two competitive multi-task learning benchmarks for visual scene understanding (i.e., PASCAL-Context and NYUD-v2). Codes and models will be made publicly available.

1. Introduction

With the rapid development of deep learning models, many computer vision tasks can be effectively handled by deep networks, which strongly motivates researchers to de-
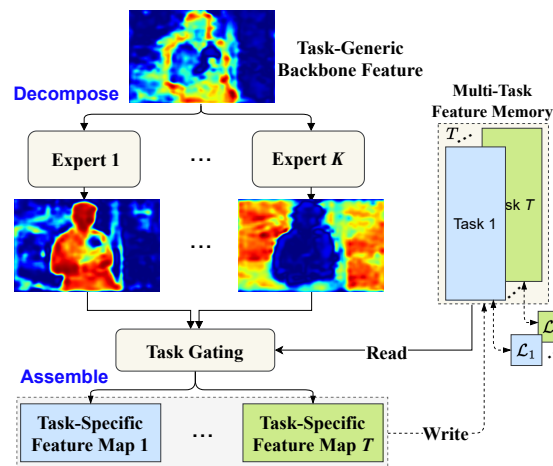

Figure 1: Motivative illustration of TaskExpert for dynamical decoding discriminative task-specific representations. A set of K expert networks decompose the task-generic backbone feature into K representative feature spaces. Then, the feature activations of an input from the multiple expert networks are dynamically assembled by the Task Gating network, to produce task-specific features for the T tasks. TaskExpert further performs long-range modeling of task-specific representations by designing a multi-task feature memory. It is updated at the current layer based on the assembled task-specific representations, and is used as an additional feature expert in the next layer.

velop unified multi-task networks to perform joint learning and inference on multiple distinct tasks [1, 2, 3, 4]. Multi-Task Learning (MTL) can avoid repetitive training of single-task models and is able to simultaneously generate predictions for different tasks with only one-time inference. On the other hand, as different computer vision tasks usually share a basic common understanding of input images, learning multiple tasks concurrently can help improve the representation power of each task and boost the overall performance of multi-task models [5, 6].

Learning discriminative task-specific features is a funda-

mental problem for MTL models. To achieve this objective, recent state-of-the-art works [7, 2, 8, 1] consider a decoder-focused paradigm, in which a shared backbone model with pretrained parameters is applied to learn generic image representations, and then separate task decoders are carefully designed to generate task-specific features. Although this paradigm is straightforward to achieve task-specific feature decoding, and produces promising multi-task prediction results, these models have two aspects of potential limitations: (i) The input task-generic feature from a backbone layer is fully shared for the different task decoders, which requires a careful design of the decoders (*e.g.* capacity and structure) for the different tasks to achieve task-discriminative decoding. (ii) The parameters of each task decoder are typically shared for different input samples, resulting in a static decoding of task-specific features, while a sample-dependant decoder is highly beneficial for learning more discriminative task features, as the input samples can be diverse, and sample-related context information is particularly important for multi-task learning and inference. All these aspects hamper the previous models from learning effective task-specific features and predictions.

To tackle the above-mentioned issues, this paper proposes a novel multi-task mixture-of-experts framework, coined as "TaskExpert", which dynamically performs the task-feature decoding for different inputs and tasks as depicted in Figure 1. Instead of using one shared backbone feature to decode different task features, the proposed TaskExpert introduces a set of task-generic expert networks that learn to ***decompose*** the backbone feature into a set of representative task-generic features. The feature decomposition enables the decoders to interact with finer-granularity task-generic feature spaces to produce more discriminative task features, as each expert can be responsible for modeling one representative feature space from the training data. Then, we dynamically ***assemble*** feature activations from different experts to produce task-specific representations, based on sample-dependent and task-specific gating scores predicted from a designed gating network. On the other hand, visual understanding tasks greatly benefit from representations of different network levels [9]. Thus, we further devise a "multi-task feature memory" to aggregate the dynamically decoded task-specific features produced at different network layers, and propose a "Memorial Mixture-of-Experts (**MMoE**)" module. For each task, MMoE utilizes its corresponding task feature from the multi-task feature memory as an additional feature expert, to improve task-specific feature decoding. In this way, MMoE enables long-range dynamic modeling of task-specific representation for each task throughout the entire backbone.

In summary, the main contribution is three-fold:

- We propose a multi-task mixture-of-experts model that allows effective decomposition of a task-generic fea-

ture from any backbone network layer and enables dynamically assembling discriminative multi-task representations by a task gating mechanism.

- We design a novel Memorial Mixture-of-Experts (MMoE) module by further introducing a multi-task feature memory, which can effectively interact (read and write) with the mixture of experts at different network layers to achieve long-range dynamic decoding of task-specific features by utilizing different levels of network representations.

- The proposed TaskExpert is extensively validated on two challenging multi-task visual scene understanding benchmarks (*i.e.* Pascal-Context and NYUD-v2). The results clearly show the effectiveness of TaskExpert for dynamic multi-task representation learning, and also establish new state-of-the-art performances on all 9 metrics on the benchmarks, regarding the comparison with previous best-performing methods using both transformer- and CNN-based backbones.

2. Related Works

Multi-Task Learning in Computer Vision In the field of multi-task learning (MTL) for computer vision problems [10, 11, 12, 13, 14, 15, 16, 17], some research works explore effective solutions through a perspective of model optimization and improve the optimization process of MTL via different multi-task loss designs [3, 6, 18, 19, 20] and gradient manipulations [21, 22, 23, 24]. These optimization strategies have also been demonstrated beneficial for alleviating the issue of task competition in multi-task learning [6]. From another perspective, several existing works also investigate designing better multi-task model architectures. These related works can be roughly divided into two groups: encoder-focused and decoder-focused methods. The encoder-focused methods [25, 26, 27, 28] target designing multi-task encoders, while the decoder-focused methods typically share the encoder (*i.e.* the backbone) for different tasks and design sophisticated decoders for task-specific feature learning and prediction [1, 2, 29, 8, 30, 31]. A recent state-of-the-art decoder-focused method (*i.e.* InvPT [7]) performs multi-task learning on powerful transformer architectures [32], and establishes a very strong baseline for multi-task visual scene understanding. It can effectively learn spatial and cross-task relationships globally. To adapt multi-task models to different tasks, researchers also propose to search for task-specific network structures for encoders [33, 34, 35] and decoders [2], or utilize task-specific learnable prompts to effectively capture task-specific information [36]. Despite promising performances of these works, they are still limited by static decoder network designs since their decoders cannot dynamically generate task-specific representations based on dif-

ferent inputs and tasks. In this paper, we propose TaskExpert, a novel and effective mixture-of-experts framework, which can achieve dynamic multi-task representation learning via feature decomposition and assembling through the entire backbone network.

Mixture-of-Experts Models With statistical motivation [37], Mixture-of-Experts (MoE) models are originally designed to control the dynamics of neural networks automatically [38, 39]. MoE learns a series of expert networks and a gating network. The outputs of expert networks are weighted by gating scores (or called "gate values") generated by a gating network before the weighting operation. In more recent works, some researchers use the gating scores as a criterion to sparsely select only one or a few experts. The sparse activation of experts enables a significant reduction of the computational cost when training large-scale models [40, 41, 42, 43]. Related to our work, [44] designs a multi-gate MoE to ensemble expert networks for different census analysis tasks with different gating networks, and [45, 46] propose sparse task-conditioned MoE networks for efficient multi-task learning, which generate predictions for *only one* task within a single forward network inference. In contrast to these works, our Memorial Mixture-of-Experts (MMoE) is a plug-and-play decoder module, which is designed for dynamically assembling discriminative task-specific features. Our TaskExpert can simultaneously generate multi-task predictions for all tasks in one single forward inference, leading to significantly higher multi-task training efficiency and enabling cross-task interaction learning. Moreover, MMoE models task-specific features throughout different network layers via the novel design of multi-task feature memory and help generate more discriminative task-specific features.

3. TaskExpert

3.1. Framework Overview

An overall illustration of our TaskExpert framework is shown in Figure 2. The input image is first projected to a token sequence and then fed into a series of classic visual transformer layers. Our framework can also be flexibly used with a CNN backbone. The proposed Memorial Mixture-of-Experts (MMoE) module is embedded at different network layers, and each module takes the task-generic backbone feature of that layer as input. It dynamically assembles and outputs discriminative task-specific representations at the corresponding layer. The MMoE further interacts (reads and writes operations) with the designed multi-task feature memory, to perform a long-range dynamic multi-task feature decoding, which benefits from different levels of network features. The reads utilize the feature memory as an additional feature expert in the MMoE module, and the writes update the memory at the layer. Finally, after be-
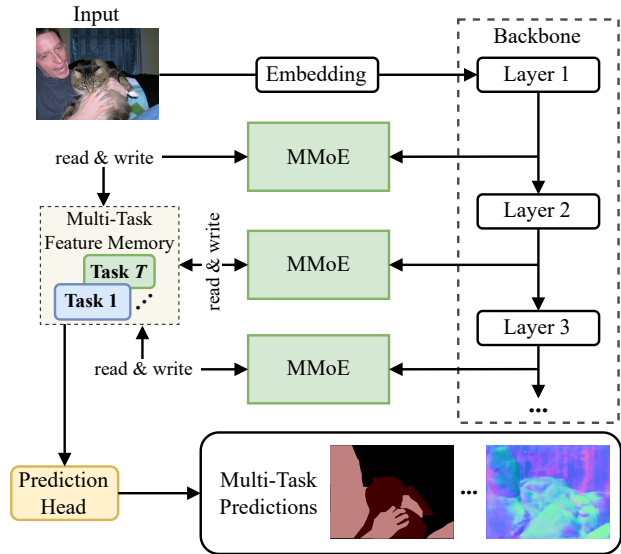

Figure 2: Overall architecture of the proposed TaskExpert model. Memorial Mixture-of-Experts (MMoE) modules are embedded at different backbone layers to dynamically decode task-specific feature for each task at the corresponding layer. The MMoE also interacts (reads and writes) with a designed multi-task feature memory to have a long-range dynamic multi-task feature decoding benefiting from different levels of network features. The reads utilize the memory as an additional feature expert in MMoE, and the writes update the memory at the layer. After being updated by the last network layer, the multi-task feature memory is used for the final prediction of each task.

ing updated by the outputs of all MMoE modules at different network layers, the task-specific features in multi-task feature memory are then used for generating the multi-task predictions with simple prediction heads. We elaborate on more details of our model in the following several sections.

3.2. Multi-Task Backbone Structure

We consider a widely-used vision transformer (ViT) model [47] with L layers as the backbone of our multi-task MoE model. The input image is first divided into $H \times W$ patches and processed by the patch embedding and positional embedding modules. Then, we obtain a sequence of patch tokens $\mathbf{X}^0 = \{\mathbf{x}_1^0, \mathbf{x}_2^0, ..., \mathbf{x}_N^0\}$, where $N = H \times W$ and the number of channels is C. The patch tokens contain generic visual representations of the input image. We denote the l-th transformer layer as a mapping function f^l, and the updating of image patch tokens at different transformer layers can be formulated as $\mathbf{X}^l = f^l(\mathbf{X}^{l-1})$, where $\mathbf{X}^l \in \mathbb{R}^{N \times C}$ denotes the output patch tokens of the transformer at the l-th layer.
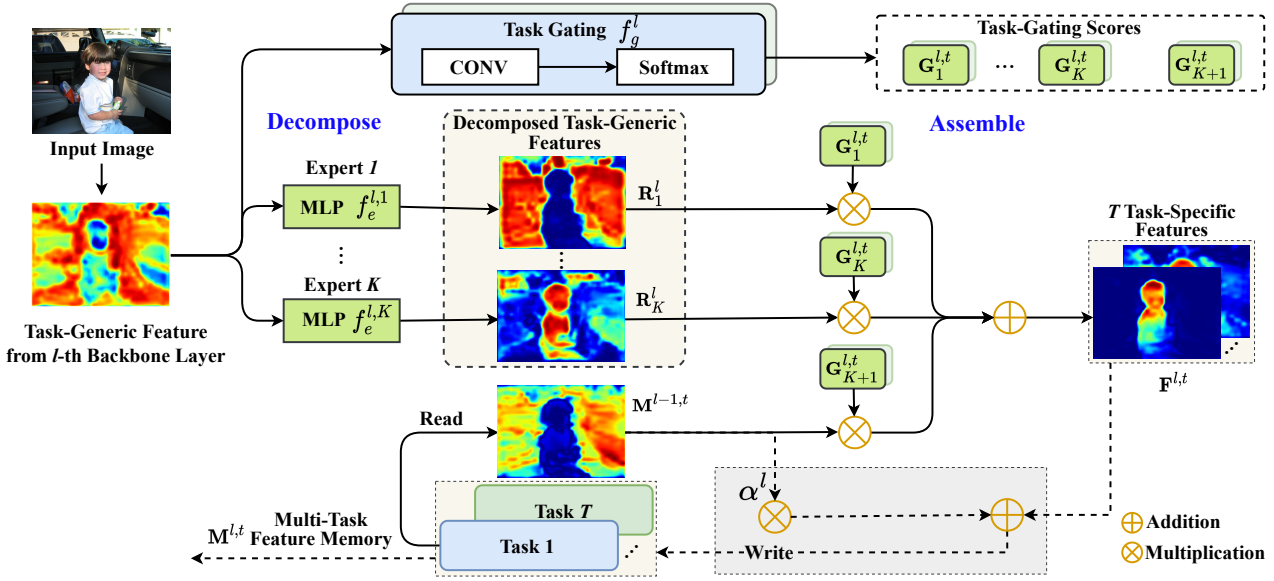
Figure 3: Illustration of the proposed Memorial Mixture-of-Experts (MMoE) module with the multi-task feature memory. MMoE accepts a task-generic feature from a backbone layer as input, and dynamically generates task-specific features for different tasks in a sample- and task-dependent manner. Specifically, MMoE decomposes the backbone feature to a set of K representative task-generic features $(\mathbf{R}_1^l, ..., \mathbf{R}_K^l)$ via the K expert networks. Then these features are used to assemble a set of T task-specific features based on dynamically generated task-gating scores from the task gating networks. To establish long-range modeling of task-specific representations crossing different network layers, the multi-task feature memory is designed to interact with the expert networks. The multi-task feature memory serves as an additional feature expert to decode the task-specific representations, and the decoded representations are also further used to update the multi-task feature memory.

3.3. Memorial Mixture-of-Experts (MMoE)

We present the details of several core parts of the proposed memorial mixture-of-experts (MMoE) module, including backbone generic-feature decomposition with a set of network experts, context-aware task-specific gating, the multi-task feature memory to have long-range modeling on task-specific features through interactions (reads and writes) with the mixture of experts, dynamic task-specific feature assembling, and multi-task prediction heads.

3.3.1 Backbone Feature Decomposition with Experts

As shown in Figure 3, to enable learning discriminative task-specific representations from the task-generic backbone feature, we first define a set of expert networks that decompose the backbone feature into several representative task-generic features in sub-feature spaces, representing distinct patterns of the backbone features, as can be observed in Figure 6. It allows more fine-grained decoding of task-specific features. Specifically, at the l-th backbone layer, let us assume that we use K expert networks $\{f_e^{l,i}(\cdot) : \mathbb{R}^{N \times C} \rightarrow \mathbb{R}^{N \times C}, i \in [1, K]\}$ to learn distinct task-generic feature projections, where superscript l denotes the layer index and i is the expert index. Each expert network employs the same Linear-BatchNorm-RELU-Linear structure but with different parameters. Through each expert network, we obtain a set of representative fea-

tures $\{\mathbf{R}_i^l \in \mathbb{R}^{N \times C}, i \in [1, K]\}$ by:

$$\mathbf{R}_i^l = f_e^{l,i}(\mathbf{X}^l), \tag{1}$$

where $\mathbf{X}^l$ is the l-th layer feature of the backbone.

3.3.2 Context-Aware Task-Specific Gating

After we have obtained a set of representative task-generic features decomposed from a backbone feature for all the tasks, we aim at decoding discriminative task-specific features for different tasks by automatically assembling these representative features. Thus, we propose to adaptively control the contribution of each expert for each patch token by designing a task-specific gating network.

For each token in the representative task-generic features, we compute a task-gating score to control its contribution to generating the task-specific features. The task-gating scores of each task are dynamically generated by a task-specific gating network, using the features $\mathbf{X}^l$ from the l-th network layer. We term the task-specific gating network with a short name "task gating", and denote it as $f_g^{l,t}(\cdot) : \mathbb{R}^{N \times C \rightarrow N \times K}$, where $t \in [1, T]$ (T is the total number of tasks) is a task index. The task-gating scores $\mathbf{G}^{l,t} \in \mathbb{R}^{N \times K}$ is computed as $\mathbf{G}^{l,t} = f_g^{l,t}(\mathbf{X}^l)$.

For the design of the gating networks, previous Mixture-of-Experts methods [42, 43, 41] use a simple linear layer followed by a Softmax function, which does not take into

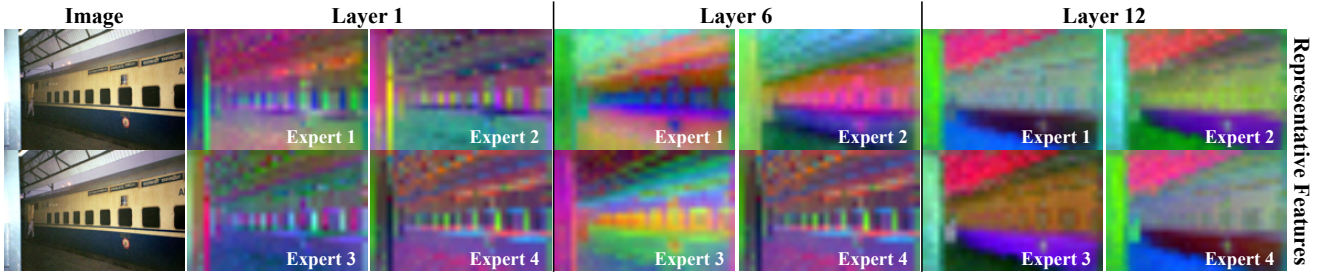| Image	Layer 1		Layer 6		Layer 12		
	Expert 1	Expert 2	Expert 1	Expert 2	Expert 1	Expert 2	Representative Features
	Expert 3	Expert 4	Expert 3	Expert 4	Expert 3	Expert 4	

Figure 4: PCA visualization of the representative features by task-generic experts across different layers. In the deeper layers, the features become increasingly responsive to high-level semantic areas. Conversely, features at more shallow layers show a heightened sensitivity to low-level patterns.

account the context of each image patch token, while using only the corresponding patch token when computing the gating score. As various challenging computer vision tasks (*e.g.* semantic segmentation and monocular depth estimation) require modeling and understanding each image patch with context information, we thus design a context-aware gating strategy. Specifically, context-aware gating adopts convolution kernels to extend the receptive field of the gating networks, and thus it can involve a local context of tokens in the computation of gating scores. We first reshape the input token sequence $\mathbf{X}$ (in $\mathbb{R}^{N \times C}$) as spatial maps (in $\mathbb{R}^{H \times W \times C}$), and then compute the gating scores of K experts with three layers, each having a 3×3 convolution with ReLU as intermediate activation function. We denote this computation procedure as "CONV", which will also be used later in Equation 3.

3.3.3   Multi-Task Feature Memory

To conduct long-range modeling of the multi-task representations decoded at different network layers that contain different levels of visual information, we design a multi-task feature memory to interact with the MMoE modules through read and write operations. The read operation uses the task feature memory as an additional feature expert for task-specific feature decoding in MMoE. The write operation updates the multi-task feature memory by using the decoded task-specific features from MMoE. In this way, the multi-task feature memory can aggregate the decoded task-specific features crossing different network layers. Specifically, the read operation is an identity mapping. The write operation can be performed as follows: given a decoded task-specific feature of the t-th task at the first layer, *i.e.* $\mathbf{F}^{1,t}$, the initialization of the multi-task feature memory $\mathbf{M}^{1,t}$ is obtained by $\mathbf{M}^{1,t} \leftarrow \mathbf{F}^{1,t}$. In the subsequent layers, suppose we decode task-specific features at every network layer, at the l-th layer ($l > 1$), we write the multi-task feature memory $\{\mathbf{M}^{l,t}, t \in [1, T]\}$ with the decoded task-specific feature $\mathbf{F}^{l,t}$ weighted by a task-specific learnable momentum value $\alpha^{l,t}$ as shown in Figure 3:

$$\mathbf{M}^{l,t} \leftarrow \mathbf{F}^{l,t} + \alpha^{l,t}\mathbf{M}^{l-1,t}, \ l > 1. \tag{2}$$

3.3.4   Dynamically Assembling Task-Specific Features

With the obtained representative task-generic features and task-specific gating scores as above-described, we can perform dynamic decoding of task-specific features, as the gating parameters are predicted in a sample-dependent manner. This is significantly different from existing decoder-focused multi-task structures, which share the decoding parameters for all the different input samples. Specifically, for the t-th task, we multiply every token of the representative task-generic features $\{\mathbf{R}_i^l, i \in [1, K]\}$ with their corresponding task-gating score $\{\mathbf{G}_i^{l,t}, i \in [1, K]\}$ and sum the tokens in the same spatial position to decode task-specific feature. Furthermore, to learn more discriminative features for each task, we utilize the corresponding task-specific feature in multi-task feature memory as an additional feature expert in MMoE, which contains long-range task-specific information learned throughout the entire network. It helps decode more effective and long-term task-specific features. As shown in Figure 3, for the MMoE at the l-th transformer layer ($l > 1$), we compute an additional gating score map $\mathbf{G}_{K+1}^{l,t}$ for the task-specific feature from the multi-task feature memory $\mathbf{M}^{l-1,t}$, which is performed by adding one more output channel into the task gating network $f_g^{l,t} : \mathbb{R}^{N \times C} \to \mathbb{R}^{N \times (K+1)}$. In summary, the task gating scores of the t-th task and the l-th layer, *i.e.*, $\mathbf{G}^{l,t} = \{\mathbf{G}_1^{l,t}, \mathbf{G}_2^{l,t}, ..., \mathbf{G}_{K+1}^{l,t}\}$, is computed as:

$$\mathbf{G}^{l,t} = \begin{cases} \text{Softmax}(\text{CONV}^{K+1}(\mathbf{X}^l)) & l > 1, \\ \text{Softmax}(\text{CONV}^{K}(\mathbf{X}^l)) & l = 1, \end{cases} \tag{3}$$

where the superscript K of CONV denotes the number of output convolution kernels. And accordingly, the decoded task-specific features at the l-th MMoE module is:

$$\mathbf{F}^{l,t} = \begin{cases} \sum_{i=1}^{K} \mathbf{G}_i^{l,t} \times \mathbf{R}_i^l + \mathbf{G}_{K+1}^{l,t} \times \mathbf{M}^{l-1,t} & l > 1, \\ \sum_{i=1}^{K} \mathbf{G}_i^{l,t} \times \mathbf{R}_i^l & l = 1. \end{cases} \tag{4}$$

Alternative sparse activation of experts As an alternative, we can also use "sparse connections" in MMoE, which means that we only activate K_{sel} selected experts to compute the task-specific features ($K_{sel} < K$) for each token [41]. In this case, for each token, we find the top K_{sel}
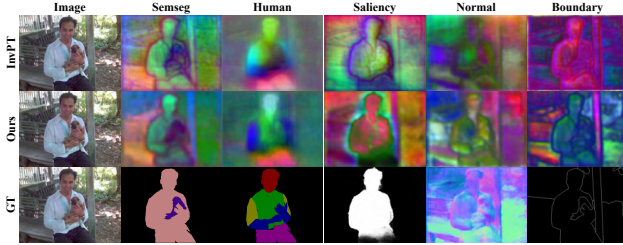
Figure 5: PCA visualization of the decoded final task features generated by InvPT [7] and our TaskExpert. Our model generates more discriminative features for all tasks.

experts to be activated based on the task-gating scores, and sum their outputs after being weighted by the task-gating scores. A comparison between the dense connection and sparse connection in MMoE is discussed in the experiments.

3.4. Generation of Multi-Task Predictions

At the final network layer L, we use task-specific prediction heads to directly generate prediction outputs for each task from the multi-task feature memory $\{\mathbf{M}^{L,t}, t \in [1, T]\}$. The prediction heads for different tasks have a similar structure, consisting of a 3×3 convolution layer with batch normalization and ReLU activation, followed by a linear layer for the final prediction.

4. Experiment

In this section, we validate the effectiveness of our proposal through several aspects: (i) ablation study, (ii) comparison with SOTA methods, and (iii) qualitative analysis.

4.1. Experimental Setup

Tasks and Benchmarks For evaluation, we adopt two popular benchmarks in multi-task learning as previous works [7, 1], *i.e.* PASCAL-Context [48, 49] and NYUD-v2 [50] datasets. PASCAL-Context contains 4,998 samples in train split and 5,105 samples in test split. It provides labels for multiple important visual understanding tasks, including semantic segmentation, human parsing, and object boundary detection. Previous work on multi-task learning generates pseudo labels for another two tasks [5]: surface normals estimation and saliency detection. Meanwhile, NYUD-v2 contains 1,449 samples in train split and 795 samples in test split. It collects labels for semantic segmentation and monocular depth estimation, from which the labels of surface normal estimation and object boundary detection can be generated. We utilize all the tasks provided in these datasets for evaluation.

Metrics We use the same evaluation metrics as previous works [7]. Specifically, semantic segmentation (Semseg) and human parsing (Parsing) tasks are evaluated by the mean Intersection over Union (mIoU). Monocular depth estimation task (Depth) is evaluated by Root Mean

Table 1: Ablation study on PASCAL-Context with a ViT-Base backbone. "MoE" denotes using Mixture-of-Experts module; "CG" denotes using context-aware gating; "Mem" denotes using our multi-task feature memory. ↓' indicates lower better and '↑' means higher better.

Model	Semseg mIoU↑	Parsing mIoU↑	Saliency maxF↑	Normal mErr↓	Boundary odsF↑	MTL Perf Δ_m↑
Baseline	76.40	65.70	84.63	13.83	69.50	-3.70
w/ MoE	77.62	66.37	84.65	13.77	70.40	-2.88
w/ MoE+CG	77.99	66.82	84.76	13.64	71.70	-2.07
w/ MoE+CG+Mem	**78.45**	**67.38**	**84.96**	**13.55**	**72.30**	**-1.46**

Square Error (RMSE). Surface normal estimation task (Normal) is evaluated by the mean error (mErr) of angles. Saliency detection task (Saliency) is evaluated by maximal F-measure (maxF). Object boundary detection task (Boundary) is evaluated by the optimal-dataset-scale F-measure (odsF). We calculate the mean relative difference across all tasks compared to the single-task baseline, which is noted as Multi-task Performance (MTL Perf Δ_m) [5].

Implementation Details We use 5 expert networks in MMoE for PASCAL-Context and 4 expert networks for NYUD-v2, which are the same as the corresponding number of tasks. We add MMoE at the end of each transformer layer if not otherwise stated. For model training, we follow the setting of InvPT [7] and use Adam optimizer to train the models for 40,000 iterations with a batch size of 4. Polynomial learning rate scheduler is used with a learning rate of 2×10^{-5} and a weight decay rate of 1×10^{-6}. We adopt the loss functions and weights for different tasks in [7]. The ViT backbones are pre-trained on ImageNet-22K [51].

4.2. Ablation Study

Model Variants Definition To examine the effectiveness of TaskExpert under a fair setting, we compare it to the multi-task baseline model and its variants with the same backbone. (1) The baseline model ("Baseline") uses ViT-Base as backbone, and designs task-specific decoders in each transformer layer to generate layer-wise task-specific features. The task-specific features from all layers are added together as multi-scale input of the corresponding prediction head for each task. These layer-wise decoders share the same structure as the expert networks used in MMoE. The prediction heads for all the tasks have a similar structure: Conv(3×3)-BN-ReLU-Linear, which are the same as those used in TaskExpert. (2) "TaskExpert w/ MoE" adds the expert and gating networks of MMoE without context-aware gating to Baseline. It uses gating networks to control the contribution of each expert to each task in a sample-dependent manner without context-aware gating or multi-task feature memory. The number of experts is the same as the number of tasks. (3) "TaskExpert w/ MoE+CG" further uses our context-aware gating strategy, which improves the gating networks by introducing the contextual information of each token. (4) "TaskExpert w/ MoE+CG+Mem" is the full version of our TaskExpert using multi-task feature
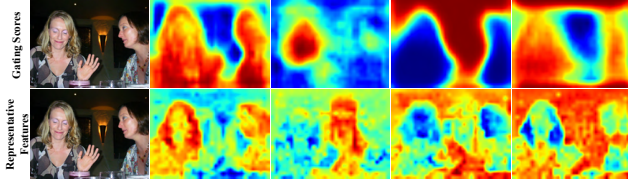
Figure 6: Visualization of the task-gating scores by task-specific gating networks and the decomposed representative features from expert networks in MMoE modules. The gating scores for different experts show distinct distributions, while the representative features are discriminative to different visual elements. These results suggest different experts are able to decompose diverse visual information from the backbone feature.
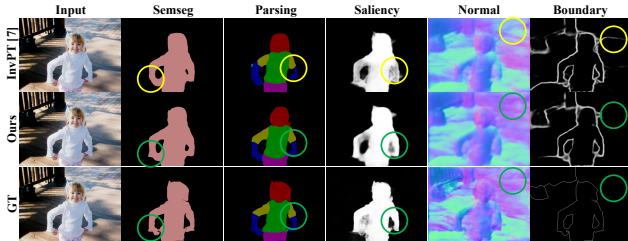

Figure 7: Qualitative comparison with SOTA method (*i.e.* InvPT [7]) on PASCAL-Context. Our TaskExpert generates better predictions with more accurate details as highlighted.

memory in the proposed MMoE module to establish modeling of long-term task-specific representation.

Key Results We analyze the performances of different model variants introduced above in Table 1. We find that using MoE brings a clear performance gain compared to the baseline model, which demonstrates the effectiveness of designing task-specific gating networks to aggregate the representative task-generic features for dynamically decoding task-specific features. For example, mIoU of semantic segmentation is improved from 76.40 to 77.62 (+1.22), and human parsing is improved from 65.70 to 66.37 (+0.67). Furthermore, the usage of context-aware gating strategy helps MoE achieves better performance by enabling the learning of contextual information for each token when computing the gating scores. For example, odsF of boundary detection is improved from 70.40 to 71.70 (+1.30) compared to only using MoE. Finally, with long-term modeling of task-specific representations via the proposed multi-task feature memory of Memorial Mixture-of-Experts, TaskExpert achieves significant performance improvement compared to the baseline model on different tasks, and the overall multi-task performance Δ_m is improved by **+2.24**.

Study of Spare and Dense Connections in MoE The original MoE models use the dense connection as the gating mechanism: all experts are activated and contribute to the output [38, 39]. Recent works propose to utilize sparse connection as the gating mechanism: for each token, only a

Table 2: Variants of TaskExpert on PASCAL-Context. The inference GPU memory is measured with batch size 1.

Variant	Semseg mIoU ↑	Parsing mIoU ↑	Saliency maxF ↑	Normal mErr ↓	Boundary odsF ↑	#Param	FLOPS	Inference GPU Mem
3 MMoE	78.04	66.54	84.85	13.62	71.20	174M	597G	8367M
6 MMoE	78.18	66.79	84.93	13.59	71.70	234M	659G	8851M
12 MMoE	78.45	67.38	84.96	13.55	72.30	347M	782G	12503M
ViT-S	75.04	62.68	84.68	14.22	68.80	**55M**	**204G**	**4187M**
ViT-B	**78.45**	**67.38**	**84.96**	**13.55**	**72.30**	347M	782G	12503M

Table 3: Comparison with multi-task MoE methods with ViT-S backbone on PASCAL-Context. The training GPU time is shown in the last column.

Variant	Semseg mIoU ↑	Parsing mIoU ↑	Saliency maxF ↑	Normal mErr ↓	Boundary odsF ↑	#Param	FLOPS	GPU Hours
M³ViT [45]	72.80	62.10	66.30	14.50	71.70	**42M**	420G	∼ 72
Mod-Squad [46]	74.10	**62.70**	66.90	**13.70**	**72.00**	50M	420G	∼ 72
Ours	**75.04**	62.68	**84.68**	14.22	68.80	55M	**204G**	**∼ 5**

Table 4: Comparison with the state-of-the-art methods with the same backbone on PASCAL-Context. Our TaskExpert clearly achieves superior performances on *all* tasks.

Model	FLOPs	#Param	Semseg mIoU ↑	Parsing mIoU ↑	Saliency maxF ↑	Normal mErr ↓	Boundary odsF ↑
PAD-Net [8]	124G	81M	53.60	59.60	65.80	15.30	72.50
MTI-Net [1]	161G	128M	61.70	60.18	84.78	14.23	70.80
ATRC [2]	216G	96M	67.67	62.93	82.29	14.24	72.42
PAD-Net w/ ViT-L [8]	773G	330M	78.01	67.12	79.21	14.37	72.60
MTI-Net w/ ViT-L [1]	774G	851M	78.31	67.40	84.75	14.67	73.00
ATRC w/ ViT-L [2]	871G	340M	77.11	66.84	81.20	14.23	72.10
InvPT [7]	669G	423M	79.03	67.61	84.81	14.15	73.00
TaskExpert (**ours**)	622G	420M	**80.64**	**69.42**	**84.87**	**13.56**	**73.30**

subset of experts contribute to the output [41], as explained in Section 3.3. In this experiment, we study the performances of using sparse connection (top 1 and top 2), and compare them with the default dense connection manner on PASCAL-Context. The results are demonstrated in Figure 8. It can be observed that activating more experts leads to better performance, and the dense connection achieves the best results. The reason is that it is able to freely utilize the full representative ability of all expert networks.

Light-Weight Versions of TaskExpert To study the model performance and efficiency with a smaller parameter and memory budget, we conduct 2 sets of experiments in Table 3: **(i)** Using different numbers of MMoE modules on ViT-B backbone for PASCAL-Context. The MMoE modules are embedded evenly along the backbone's depth. **(ii)** Implementing TaskExpert on ViT-S, which is a smaller backbone. These smaller variants achieve reasonable performance with less parameter and memory consumption.

4.3. Comparison with SOTA Methods

We implement TaskExpert based on ViT-Large backbone with MMoE modules embedded at every two transformer layers, then we compare TaskExpert to recent SOTA methods in literature on both PASCAL-Context and NYUD-v2 in Table 4 and Table 5. We re-implement several CNN-based SOTA multi-task learning methods on ViT-L backbone for a more comprehensive comparison on PASCAL-Context. Compared with the previous SOTA
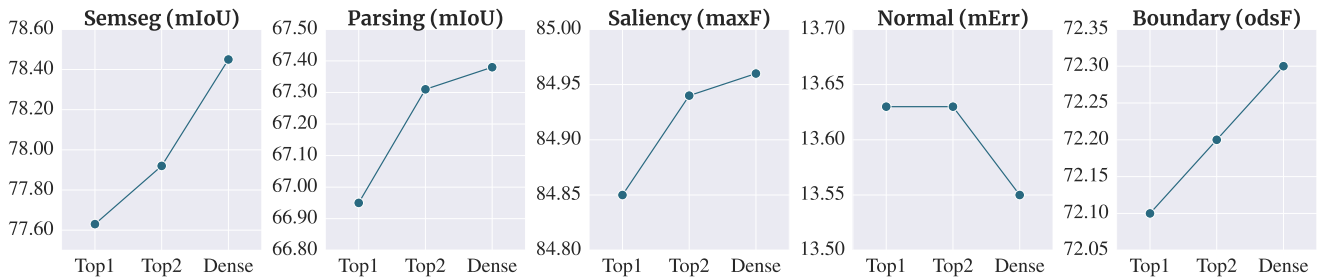
Figure 8: Comparison of TaskExpert with dense and sparse connection in MMoE. "Top 1" denotes only activating the expert with the highest gating score, "Top 2" means activating the experts with top 2 gating scores, and "Dense" activates all experts.

Table 5: Comparison with the state-of-the-art methods on NYUD-v2 dataset. Our TaskExpert achieves the best performances on *all* tasks.

Model	Backbone	Semseg mIoU ↑	Depth RMSE ↓	Normal mErr ↓	Boundary odsF ↑
Cross-Stitch [25]	HRNet18	36.34	0.6290	20.88	76.38
PAP [29]	HRNet18	36.72	0.6178	20.82	76.42
PSD [31]	HRNet18	36.69	0.6246	20.87	76.42
PAD-Net [8]	HRNet18	36.61	0.6270	20.85	76.38
MTI-Net [1]	HRNet48	45.97	0.5365	20.27	77.86
ATRC [2]	HRNet48	46.33	0.5363	20.18	77.94
InvPT [7]	ViT-L	53.56	0.5183	19.04	78.10
TaskExpert (**ours**)	ViT-L	**55.35**	**0.5157**	**18.54**	**78.40**

model (*i.e.* InvPT [7]), TaskExpert achieves clearly better performances on ***all 9 metrics*** of these two challenging benchmarks with the same backbone. Particularly, on PASCAL-Context, we improve semantic segmentation by **+1.61** and human parsing by **+1.81**. On NYUD-v2, semantic segmentation is improved by **+1.79**. It should be noted that, our gains are measured upon the very strong performance foundation established by InvPT (*e.g.* Semseg 79.03, Saliency 84.81 on PASCAL-Context). This superior performance clearly shows the effectiveness of the proposed TaskExpert. TaskExpert also maintains a ***higher efficiency*** with a reduction of 3 million in the number of parameters and 47 billion FLOPs compared to InvPT.

As underscored in the Related Work section, our TaskExpert is designed for a different setting from M^{3}ViT [45] and Mod-Squad [46]. Their models are constrained to generating a single task prediction per forward pass, while our model capably generates predictions for all tasks within a single forward pass. Therefore, the multi-task learning efficiency of our model is significantly higher than their methods. When utilizing the same backbone, it requires *only half the FLOPS*, as demonstrated in Table 3. Moreover, our model's training duration is markedly shorter since it learns all tasks concurrently in each iteration. While they necessitate around 18 hours of training with 4 A6000 GPUs as confirmed by their authors, our model completes training in under 5 hours using a single NVIDIA 3090 GPU.

4.4. Qualitative study

Qualitative Study of MMoE To understand better the learned gating scores and representative task-generic fea-

tures by the gating networks and expert networks in our MMoE modules, we present qualitative examples in Figure 6. As can be observed, TaskExpert effectively learns representative task-generic features with distinct activation distributions, focusing on different aspects of the image. Accordingly, the gating scores exhibit different response intensities to different areas for selectively assembling the experts' features for decoding task-specific features.

PCA Visualization of Representative Features of Task-Experts at Different Layers We conduct principal component analysis (PCA) on the representative features at layers 1, 6, and 12. As shown in Fig. 4, we visualize the first three components of the feature PCA as a color map. It reveals that the task experts at deeper layers capture more high-level semantic information of the input. On the other hand, in the shallower layers, the task-feature experts focus more on low-level visual patterns (*e.g.*, edges).

Qualitative Comparison of Decoded Task Features In Fig. 5, we also conduct PCA on the final task features generated by InvPT [7] and our TaskExpert, and draw the first three components. The features for different tasks from TaskExpert show clear task-specific characteristics and are much more discriminative than those of InvPT.

Qualitative Comparison of Prediction Results To intuitively examine the prediction quality of TaskExpert compared to the previous best-performing method [7], we visualize the predictions of the two models on the test split of PASCAL-Context in Figure 7. Our TaskExpert generates visually better multi-task predictions.

5. Conclusion

In this paper, we present TaskExpert, a novel multi-task mixture-of-expert model establishing new SOTA performances on two challenging benchmarks. TaskExpert can effectively perform dynamic decoding of task-specific features, from a set of representative task-generic features learned by expert networks. TaskExpert further models the long-term task-specific representations at different layers via a designed multi-task feature memory. Extensive experiments clearly prove the effectiveness and efficiency of our TaskExpert model, and we hope that our TaskExpert can also be beneficial for the future study of dynamic representation learning problems.

Acknowledgement

References

[1] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *ECCV*, 2020.

[2] David Bruggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring relational context for multi-task dense prediction. In *ICCV*, 2021.

[3] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *CVPR*, 2020.

[4] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017.

[5] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *CVPR*, 2019.

[6] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018.

[7] Hanrong Ye and Dan Xu. Inverted pyramid multi-task transformer for dense scene understanding. In *ECCV*, 2022.

[8] 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 *CVPR*, 2018.

[9] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[10] Yu Zhang and Qiang Yang. A survey on multi-task learning. *TKDE*, 2021.

[11] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *TPAMI*, 2021.

[12] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018.

[13] Menelaos Kanakis, Thomas E Huang, David Brüggemann, Fisher Yu, and Luc Van Gool. Composite learning for robust and effective dense predictions. In *WACV*, 2023.

[14] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *ECCV*, 2022.

[15] Lukas Hoyer, Dengxin Dai, Yuhua Chen, Adrian Koring, Suman Saha, and Luc Van Gool. Three ways to improve semantic segmentation with self-supervised depth estimation. In *CVPR*, 2021.

[16] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *CVPR*, 2019.

[17] Xiangtai Li, Henghui Ding, Wenwei Zhang, Haobo Yuan, Jiangmiao Pang, Guangliang Cheng, Kai Chen, Ziwei Liu, and Chen Change Loy. Transformer-based visual segmentation: A survey. *arXiv preprint arXiv:2304.09854*, 2023.

[18] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *NeurIPS*, 2021.

[19] Siwei Yang, Hanrong Ye, and Dan Xu. Contrastive multi-task dense prediction. In *AAAI*, 2023.

[20] Wei-Hong Li, Xialei Liu, and Hakan Bilen. Learning multiple dense prediction tasks from partially annotated data. In *CVPR*, 2022.

[21] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In *NeurIPS*, 2020.

[22] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, 2018.

[23] Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *ICLR*, 2020.

[24] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *NeurIPS*, 2020.

[25] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016.

[26] 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 *CVPR*, 2019.

[27] Lijun Zhang, Xiao Liu, and Hui Guan. Automtl: A programming framework for automating efficient multi-task learning. In *NeurIPS*, 2021.

[28] Yuan Gao, Haoping Bai, Zequn Jie, Jiayi Ma, Kui Jia, and Wei Liu. Mtl-nas: Task-agnostic neural architecture search towards general-purpose multi-task learning. In *CVPR*, 2020.

[29] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *CVPR*, 2019.

[30] Xiaoya Zhang, Ling Zhou, Yong Li, Zhen Cui, Jin Xie, and Jian Yang. Transfer vision patterns for multi-task pixel learning. In *ACMMM*, 2021.

[31] Ling Zhou, Zhen Cui, Chunyan Xu, Zhenyu Zhang, Chao-qun Wang, Tong Zhang, and Jian Yang. Pattern-structure diffusion for multi-task learning. In *CVPR*, 2020.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszko-reit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[33] Simon Vandenhende, Stamatios Georgoulis, Bert De Bra-bandere, and Luc Van Gool. Branched multi-task networks: deciding what layers to share. In *ECCV*, 2020.

[34] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive feature shar-ing in multi-task networks with applications in person at-tribute classification. In *CVPR*, 2017.

[35] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learn-ing to branch for multi-task learning. In *ICML*, 2020.

[36] Hanrong Ye and Dan Xu. Taskprompter: Spatial-channel multi-task prompting for dense scene understanding. In *ICLR*, 2023.

[37] Michael I Jordan and Robert A Jacobs. Hierarchical mix-tures of experts and the em algorithm. *Neural Computation*, 6(2):181–214, 1994.

[38] Robert A Jacobs and Michael I Jordan. Learning piece-wise control strategies in a modular neural network architec-ture. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(2):337–345, 1993.

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

[40] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of ex-perts. *arXiv preprint arXiv:1312.4314*, 2013.

[41] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outra-geously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[42] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with sim-ple and efficient sparsity. *JMLR*, 23:1–39, 2022.

[43] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant mod-els with conditional computation and automatic sharding. In *ICLR*, 2021.

[44] 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 *KDD*, 2018.

[45] Hanxue Liang, Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tian-long Chen, Kai Zou, Yu Cheng, Cong Hao, and Zhangyang Wang. M3vit: Mixture-of-experts vision transformer for ef-ficient multi-task learning with model-accelerator co-design. In *NeurIPS*, 2022.

[46] Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik Learned-Miller, and Chuang Gan. Mod-squad: Designing mixture of experts as modular multi-task learners. *arXiv preprint arXiv:2212.08066*, 2022.

[47] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Syl-vain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[48] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: De-tecting and representing objects using holistic models and body parts. In *CVPR*, 2014.

[49] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[50] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[51] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.