ADAPTIVE SPARSITY MIXTURE-OF-EXPERTS (AS-MOE): EFFICIENT RESOURCE ALLOCATION IN MIXTURE-OF-EXPERTS MODELS

Hien The Liu, Anh Duc Nguyen, Huy Le Tu
University of Information Technology, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
{21522062 , 21520140, 21522173}@gm.uit.edu.vn

ABSTRACT

In this study, we propose an innovative enhancement to the Mixture-of-Experts (MoE) model, called Adaptive Sparsity Mixture-of-Experts (ASMoE). Traditional MoE Shazeer et al. (2017) models, while efficient in distributing computation across multiple experts, often suffer from imbalanced score distributions, leading to computational inefficiencies. ASMoE addresses this by dynamically adjusting the computational load of each expert based on their routing scores. By introducing a **ratio level** parameter, ASMoE reduces the number of activated hidden nodes for experts with lower scores, thus optimizing resource usage. Our experimental results demonstrate that ASMoE not only maintains high performance but also significantly improves computational efficiency. However, the model's runtime performance needs further optimization. This research opens new avenues for efficient resource allocation in large-scale neural networks.

1 INTRODUCTION

In recent years, large-scale models in general and Transformers specifically have shown great capabilities in many NLP tasks. However, nowadays, large-scale models can have up to billions of parameters. As a consequence, with the growth of the models' scale, the computational cost becomes higher, and it requires a large amount of hardware resources as well as memory requirements. The Mixture of Experts (MoE) Jacobs et al. (1991) Shazeer et al. (2017), which was first introduced in 1991, allows the model to scale up the number of parameters while maintaining an affordable computation. The traditional MoE method, while efficient in distributing computation across multiple experts, often suffers from imbalanced score distributions, leading to computational inefficiencies.

In this study, we propose an innovative enhancement to the Mixture-of-Experts (MoE) model, termed Adaptive Sparsity Mixture-of-Experts (ASMoE). ASMoE addresses this by dynamically adjusting the computational load of each expert based on their routing scores. By introducing a 'ratio level' parameter, ASMoE reduces the number of activated hidden nodes for the in-used experts, thus optimizing resource usage. To achieve the results in this research, we used the multilingual text-to-text transformer model (mT5) as the base model and integrated MoE and ASMoE techniques into mT5 on the multilingual VQA task.

To verify the effectiveness of ASMoE, we decided to use the UIT-EVJVQA datasetNguyen et al. (2023), a multilingual visual question answering dataset, which was first introduced in VLSP2022: UIT-EVJVQA Challenge: Multilingual Visual Question Answering. UIT-EVJVQA is the first multilingual Visual Question Answering dataset consists of three languages: English, Vietnamese, and Japanese. The question-answer pairs in the dataset were created by humans on a set of images taken in Vietnam, with the answer created from the input question and the corresponding image.

2 RELATED WORK

Mixture of Experts (MoE): MoE Jacobs et al. (1991) Shazeer et al. (2017) is a unique type of neural network in which the parameters are divided into several experts, or sub-modules, and conditional computation is carried out in an input-dependent manner. MoE allows models to be pretrained with much less computation, which means we can scale up the model or dataset size with the same budget as dense model. In the context of transformer, a MoE consists of two main components: sparse MoE layers and a router. Instead of using dense feed-forward network, MoE uses sparse layers, each layer has a specific number of experts, where each expert is a neural network, but they can be more complex networks or even a Moe itself, leading to hierarchical MoEs. The router plays a crucial role in MoE's efficiency and effectiveness, it acts as the decision-maker, determining which expert network (or a combination) is best suited to handle a specific piece of data. By directing the data to the relevant expert, the router ensures that the model leverages the most appropriate knowledge for the task. By utilizing MoE, recent study has achieved great results and MoE has gained increasing popularity in both NLP and CV tasks.

Vision Transformer (ViT) : Vision Transformer (ViT) Dosovitskiy et al. (2020) represents a significant departure from the convolutional neural network (CNN) paradigm that has long dominated the computer vision landscape. Whereas CNNs rely on local receptive fields and hierarchical feature extraction, ViT operates directly on image patches, treating them as input sequences for a transformer model. By leveraging the powerful self-attention mechanism inherent to transformers, ViT is able to capture long-range dependencies and global interactions between image regions, a capability that has proven advantageous compared to the localized processing of CNNs

Multilingual Pre-trained Text-to-Text Transformer (mT5): mT5 Xue et al. (2021) is a multilingual variant of T5, which was pretrained on a new Common Crawl-based dataset (mC4) covering 101 languages. Like T5, mT5 has a general-purpose text-to-text format, and its design is based on insights from a large-scale empirical study. It also used SentencePiece models s trained with the language sampling rates used during pre-training. Additionally, mT5 employed a character coverage of 0.99999 and enabled SentencePiece's "byte-fallback" function in order to support languages with extensive character sets, such as Chinese, guaranteeing that every string could be uniquely encoded. With all of those reasons, mT5 can handle various NPL tasks such as question answering, reading comprehension, translation,...

Encoder-Decoder Architectures for Image Captioning and VQA : Several recent works have explored the use of encoder-decoder architectures for various computer vision and language tasks, including image captioning and visual question answering. For image captioning, Anderson et al. (2018) proposed a model that combines bottom-up and top-down attention mechanisms to selectively attend to relevant visual features when generating captions. Their approach demonstrated state-of-the-art results on the MS-COCO image captioning benchmark. In the context of visual question answering, Teney et al. (2017) introduced a graph-based representation that captures the relationships between objects and their attributes in the image. This graph-based reasoning module was integrated into an encoder-decoder architecture to improve VQA performance. Furthermore, Xu & Saenko (2016) explored the use of a memory-augmented neural network for VQA, allowing the model to dynamically store and retrieve relevant information from the image and question to generate accurate answers. Moreover, Nicolson et al. (2022) also employed an encoder-decoder mechanism, specifically a VisionEncoder-Decoder architecture, to generate captions for medical images in the ImageCLEF 2021 challenge, demonstrating the effectiveness of this architecture in medical image captioning tasks

3 ASMoE's architecture Implementation

In this study, we introduce the Adaptive Sparse Mixture of Experts (ASMoE) model, which incorporates innovative strategies to enhance computational efficiency. Similar to the vanilla Mixture-of-Experts (MoE), we use a routing method to select the top-k experts best suited to process each type of token. Specifically, our method enhances computational efficiency by reducing the number of hidden units in each of the top-k experts based on their routing scores. Despite these modifications, we ensure that each token receives an appropriate amount of computation, with high routing scores receiving more computational resources and low routing scores receiving fewer. Additionally, we maintain a balancing loss to ensure an even distribution of workload among the experts, preventing bias toward any specific expert and thus avoiding the waste of a large number of parameters.

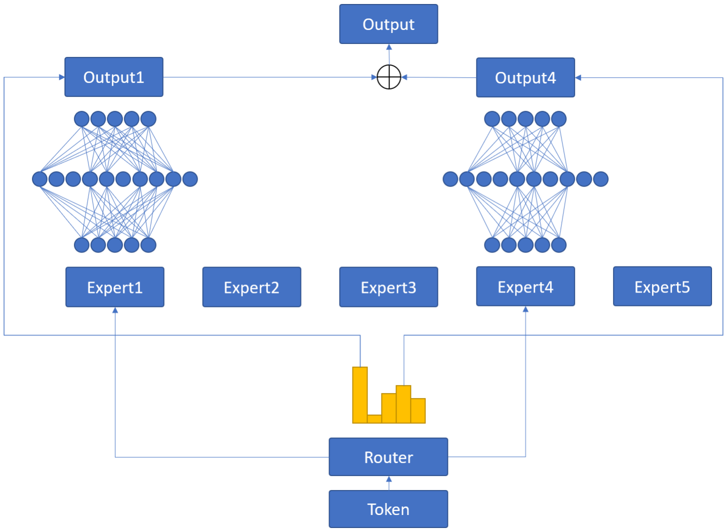


Figure 1: ASMoE's architecture

3.1 Hidden Unit Reduction via Random Indexing

To reduce the number of hidden units, we propose a technique that involves randomly indexing the weights of the linear layer. We then perform matrix multiplication between the tokens and the modified linear layer weights. By doing this, we can select a ratio of units based on the given routing scores. This approach prevents the model from requiring large computational resources for certain experts while ensuring that the final step, which aggregates the outputs from the top-k experts, remains computationally efficient. Unlike traditional Dropout, which zeros out certain outputs after full computation, our method avoids computation altogether for these units. Additionally, in the projection linear layer, we use the previous indices to select the corresponding weights for the tokens, and finally get the output with the desired form.

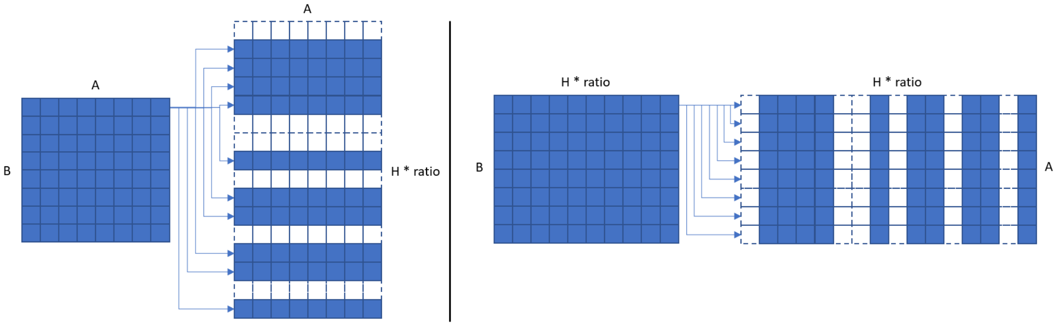


Figure 2: Indexing with hidden unit ratios

3.2 Token Grouping Using Threshold-Based Hidden Unit Ratios

However, each token will have different scores. Processing them individually is impractical, so to address this issue, we use a threshold to group tokens with similar routing scores together. These grouped tokens are then processed by an expert with a constant ratio of hidden units. The threshold is a hyperparameter defined as a series of intervals, such as $0 - \frac{1}{k}, \frac{1}{k} - \frac{2}{k}, \frac{2}{k} - \frac{3}{k}, \ldots, 1$, where k is the number of ranges. For example, if k is 4, the ranges would be $[0, 0.25], (0.25, 0.5], (0.5, 0.75], (0.75, 1]$. Correspondingly, the hidden unit ratios are defined as $\frac{1}{k+1}, \frac{2}{k+1}, \frac{3}{k+1}, \ldots, \frac{k}{k+1}$. By doing this, we can efficiently process tokens with similar routing scores without the need to handle each token individually or duplicate weights, both of which are inefficient in terms of memory and computation.

3.3 Ensuring Equal Computation Across Tokens

Finally, we ensure that each token in the input sequence receives an equal amount of computational effort. Although we utilize top-k experts for the final output, the reduction in hidden units for efficient computing ensures that the total computation required by the top-k experts is comparable to that of a single standard expert.

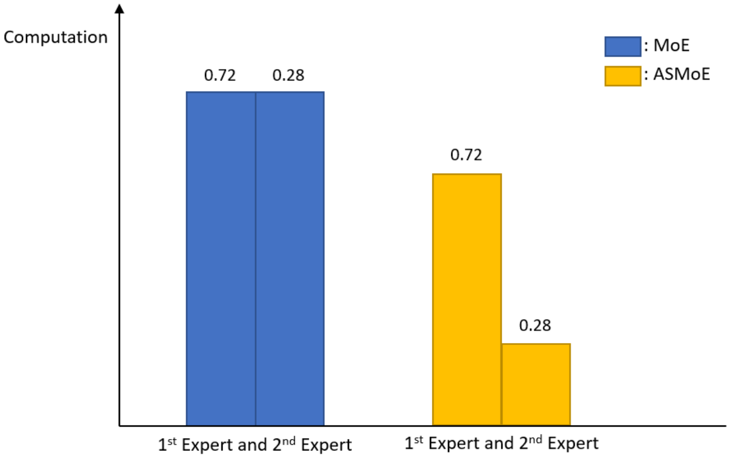


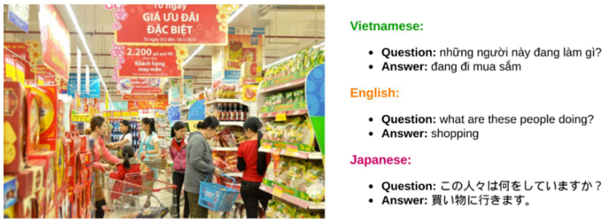
Figure 3: The computation cost between normal MoE and ASMoE

4 Experiment

4.1 Dataset

In this research, we evaluate the performance of our model on UIT-EVJVQA dataset Nguyen et al. (2023). UIT-EVJVQA is a multilingual VQA dataset, used as a benchmark dataset for challenge of multilingual visual question answering at the 9th Workshop on Vietnamese Language and Speech Processing (VLSP 2022). This task attracted the participants from multiple teams from various universities and organizations

The dataset consists of 33,000+ pairs of question-answer over three languages: Vietnamese, English, and Japanese, on approximately 5,000 images taken from Vietnam, covering a wide range of visual concepts and reasoning tasks


Vietnamese:

- **Question:** những người này đang làm gì?
- **Answer:** đang đi mua sắm

English:

- **Question:** what are these people doing?
- **Answer:** shopping

Japanese:

- **Question:** この人々は何をしていますか？
- **Answer:** 買い物に行きます。

Figure 4: A sample from UIT-EVJVQA

For our experiments, we split the dataset into training, validation and test sets, with the ratios are respectively as figure 5. The training set is used to train the our model, the validation set is used for hyperparameters tuning, modification, and the test set is used for final evaluation.

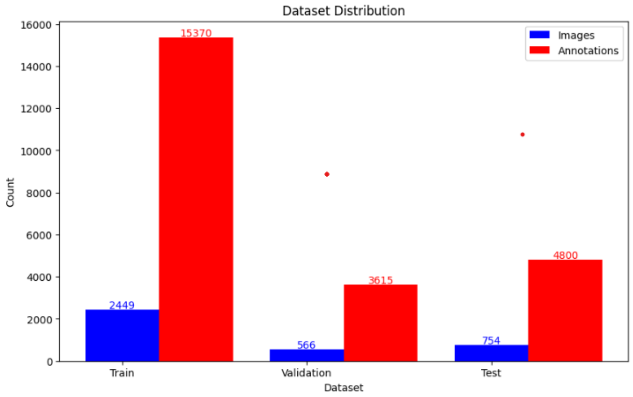


Figure 5: Distribution of Training, Validation and Test sets

4.2 Training Configuration

For comprehensive comparison and evaluation, we conducted experiments with three model variants to determine the performance and characteristics of our innovation on on the UIT-EVJVQA multilingual VQA dataset:

- ViT-mT5 Baseline Model
- Traditional MoE ViT-mT5 Model
- Adaptive Sparsity MoE (ASMoE) ViT-mT5 Model

In our investigation, we applied three different layer configurations - 6 layers, 8 layers, and 12 layers - to each of the three ViT-mT5 model variants: the baseline ViT-mT5, the MoE ViT-mT5, and the ASMoE ViT-mT5. The motivation behind this experimental design was to explore the impact of model depth on the complexity and resource requirements of the different architectures. By systematically increasing the number of layers, we aimed to explore how ASMoE performes and scales with depth compared to the baseline and traditional MoE ViT-mT5 model.

Table 1 shows that the baseline ViT-mT5 scales more efficiently than the MoE and ASMoE variants. As the layer count increases from 6 to 12, the baseline ViT-mT5's parameters rise from 249M to 305M and FLOPs from 83B to 119B. In contrast, both MoE and ASMoE ViT-mT5 models show larger increases in parameters, from 371M to 551M. However, ASMoE ViT-mT5 has a smaller increase in FLOPs (76B to 106B) compared to MoE ViT-mT5 (85B to 124B), indicating better computational efficiency. Despite this, ASMoE ViT-mT5 has higher training and inference times due to our single GPU setup, which cannot utilize parallel training, offsetting its FLOPs advantage.

Model	Layers	Params (M)	Training Time (h)	Inference Time (s)	FLOPs (B)
ViT-mT5	6	249	5.40	0.87	83
	8	267	5.42	1.08	95
	12	305	5.45	1.22	119
MoE ViT-mT5	6	371	5.3	1.34	85
	8	431	5.27	1.50	98
	12	551	5.7	2.17	124
ASMoE ViT-mT5	6	371	6.58	3.04	76
	8	431	6.90	3.26	86
	12	551	7.95	4.26	106

Table 1: Model Complexity

For the model training, we used a cross-entropy loss function with the ignore_index parameter set to the padding token ID from the tokenizer. By setting the ignore_index parameter, we can avoid the contribution and effect of padding tokens, which appeared frequently during the training process and might negatively affect the model's learning and optimization procedure. For optimization, we applied the AdamW algorithm with betas of (0.85, 0.95) and a weight decay of 0.0001. Furthermore, we employed an ExponentialLR scheduler with a gamma value of 0.85 to gradually reduce the learning rate during training. All of these hyperparameters were consistently implemented across the three model variants, with a batch size of 16

4.3 Results

We compare our baseline model, which is ViT-mT5, with the MoE ViT-mT5 and ASMoE ViT-mT5 models. All models use the same encoder - ViT, and the difference bewteen three model is the decoder. Although they all use the same decoder which is mT5's one, the MoE ViT-mT5 and ASMoE ViT-mT5 models replace the dense layers with the MoE and ASMoE modules respectively.

The results of our 3 models on the are shown in table 2 2. Under the base training configuration, the 6 layers ViT-MoE achieved the highest results of all metrics, with Bert Score at 0.73, Bleu at 0.07. And the ASMoE variant's performance is as good as ViT-mT5 with slightly differences. In the 8-layer configuration the ASMoE ViT-mT5 achieved better result than its counterpart in all metrics .Under larger setting with 8 layers and 12 layers, the ViT-mT5 model has completely outperformed both MoE and ASMoE models in all aspects.

With the same training configuration, the ViT-mT5 is likely to perform better than both ViT-MoE and ViT-ASMoE models as the number of layers goes up. We can see that the more complex the MoE models' architecture is, the more epoch-to-train the they need to achieve better result. Since they have n-experts (8 experts in this study) and only use top-k experts in one forward pass per layer, the model will need a greater training time to cover all the experts compares to normal FFN models. As the number of layers goes up the ASMoE variant will perform better than the MoE one as shown in the 8 and 12 layers section in 2. But still, we need further more training and test to verify this in a bigger scenario.

Layers	Model	Bert Score	Bleu1	Bleu2	Bleu3	Bleu4	Rouge	Meteor
6	ViT-mT5	0.715	0.141	0.072	0.036	0.016	0.206	0.141
	MoE ViT-mT5	**0.730**	**0.148**	**0.076**	**0.038**	**0.019**	**0.207**	0.147
	ASMoE ViT-mT5	0.702	0.143	0.072	0.035	0.016	0.199	**0.151**
8	ViT-mT5	**0.727**	**0.144**	**0.076**	**0.039**	**0.019**	**0.210**	**0.147**
	MoE ViT-mT5	0.706	0.124	0.059	0.028	0.011	0.189	0.121
	ASMoE ViT-mT5	0.719	0.142	0.071	0.034	0.016	0.205	0.143
12	ViT-mT5	**0.716**	**0.134**	**0.067**	**0.031**	**0.015**	**0.200**	**0.135**
	MoE ViT-mT5	0.696	0.123	0.054	0.024	0.011	0.191	0.122
	ASMoE ViT-mT5	0.667	0.128	0.058	0.027	0.011	0.168	0.128

Table 2: Experiment Results

In addition to the model performance metrics, we also analyze the validation loss curves to gain further insights into the model behavior, by using Figure 6 to show the loss plots for the three ViT-mT5 model variants across the different layer configurations, as well as observe the comprehensive and optimizing ability of 3 model variants.

Investigations into the architectural stability of the Vit-mT5 model reveal a noteworthy characteristic - the model exhibits a remarkable degree of consistency in performance, regardless of the number of layers employed in the architecture. This trend suggests that the core design of the Vit-mT5 model possesses an inherent scalability, enabling it to maintain its capabilities with high fidelity even as the depth of the network is increased. Such stability in the face of architectural complexity is a desirable trait, as it allows for the seamless deployment of the Vit-mT5 model across a wide range of applications and problem domains without the need for extensive architectural tuning or modifications.

When examining the comparative performance of the MoE Vit-mT5 and ASMoE Vit-mT5 variants, an intriguing pattern emerges. In the 6-layer configuration, the MoE Vit-mT5 model demonstrates a discernible advantage over its ASMoE counterpart. However, as the model complexity is further increased, with 8 and 12 layer architectures, the ASMoE Vit-mT5 models begin to outshine the MoE variants. This observation suggests that as the model architecture becomes more complex, the ASMoE approach is better equipped to harness the additional capacity and deliver superior performance. The superior performance of the ASMoE Vit-mT5 in deeper models implies that the specialized expert mechanism employed in the ASMoE architecture is particularly well-suited for handling the increased complexity, allowing it to extract and leverage more nuanced features and patterns from the data in a more effective manner.

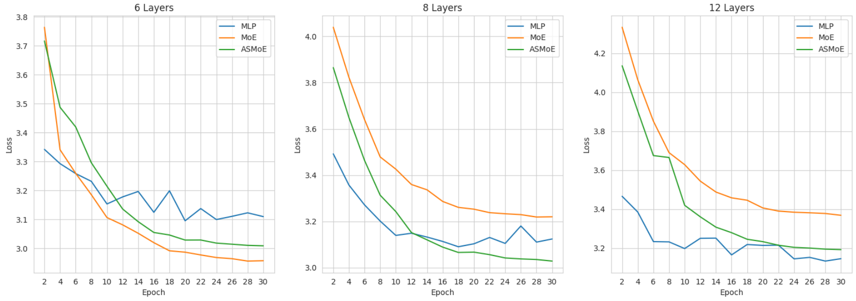


Figure 6: Validation Loss Curves for 3 variants followed by 3 kinds of layer configuration

5 Conclusion

In conclusion, this work presents a novel approach to Mixture-of-Experts (MoE) architectures, which we have termed Adaptive Specialized MoE (ASMoE). Through our experimental evaluations, we have demonstrated that the proposed ASMoE technique holds significant promise as an effective model design, particularly for more complex neural network architectures.

While the base Vit-mT5 model exhibits a remarkable stability in performance, regardless of the depth of the network, our findings indicate that this stability may also limit its ability to fully capitalize on increased model complexity. In contrast, the MoE and ASMoE variants show greater malleability, with the ASMoE approach emerging as the superior option as the models become more intricate.

The superior performance of ASMoE in comparison to traditional MoE when dealing with deeper, more complex neural networks suggests that the specialized expert mechanism employed in the ASMoE framework is particularly well-suited for extracting and leveraging nuanced features and patterns from data in large-scale, high-capacity models. This insight positions the ASMoE approach as a compelling choice for applications involving large language models and other complex deep learning architectures, where the ability to adaptively allocate computational resources and specialize experts can lead to significant performance gains.

Overall, the introduction of the ASMoE framework, and its empirical validation against the base Vit-mT5 and MoE variants, represents an important step forward in the ongoing pursuit of efficient and high-performing neural network models for a wide range of real-world applications.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077–6086, 2018.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1):79–87, 03 1991. ISSN 0899-7667. doi: 10.1162/neco.1991.3.1.79. URL https://doi.org/10.1162/neco.1991.3.1.79.

Ngan Luu-Thuy Nguyen, Nghia Hieu Nguyen, Duong TD Vo, Khanh Quoc Tran, and Kiet Van Nguyen. Vlsp2022-evjvqa challenge: Multilingual visual question answering. *arXiv preprint arXiv:2302.11752*, 2023.

Aaron Nicolson, Jason Dowling, and Bevan Koopman. Imageclef 2021 best of labs: The curious case of caption generation for medical images. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 190–203. Springer, 2022.

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.

Antoine Simoulin and Benoit Crabbé. How many layers and why? an analysis of the model depth in transformers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pp. 221–228, 2021.

Damien Teney, Lingqiao Liu, and Anton van Den Hengel. Graph-structured representations for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2017.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, 2018.

Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pp. 451–466. Springer, 2016.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer, 2021. URL https://arxiv.org/abs/2010.11934.