ATTENTION IS ALL YOU NEED FOR MIXTURE-OF-DEPTHS ROUTING

Advait Gadhikar[1, 2] ✉, **Souptik Majumdar**[1, 3] ✉, **Niclas Popp**[1, 4], **Piyapat Saranrittichai**[1], **Martin Rapp**[1], **Lukas Schott**[1],
[1]Bosch Center for Artificial Intelligence, Renningen, Germany
[2]CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
[3]University of Stuttgart, Stuttgart, Germany
[4]University of Tübingen, Tübingen, Germany
{advait.gadhikar@cispa.de, st184540@stud.uni-stuttgart.de}

ABSTRACT

Advancements in deep learning are driven by training models with increasingly larger numbers of parameters, which in turn heightens the computational demands. To address this issue, Mixture-of-Depths (MoD) models have been proposed to dynamically focus computations on the most relevant parts of the inputs, thereby enabling the deployment of large-parameter models with high efficiency during inference and training. However, conventional MoD models employ additional network layers specifically for the routing which are difficult to train, and add complexity to the model. In this paper, we introduce a novel attention-based routing mechanism *A-MoD* that leverages the existing attention map of the preceding layer for routing decisions within the current layer. Compared to standard routing, *A-MoD* allows for more efficient training as it introduces no additional trainable parameters and can be easily adapted from pre-trained transformer models. Furthermore, it can increase the performance of the MoD model. For instance, we observe up to 2% higher accuracy on ImageNet compared to standard routing and isoFLOP ViT baselines.

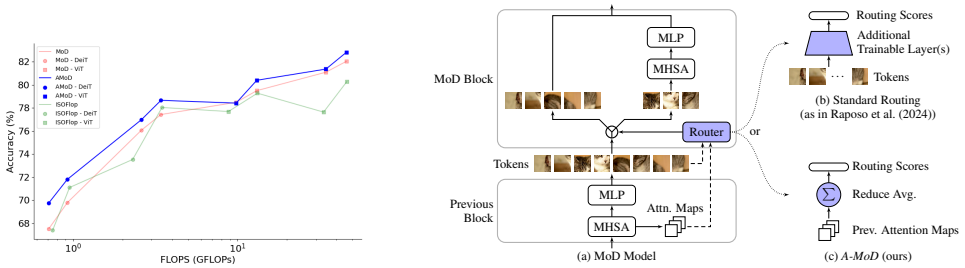


Figure 1: **Left** Accuracy vs FLOPs Pareto-curve for *A-MoD* compared with MoD and ISOFlop models on ImageNet-1k. **Right** MoD model (a) with standard routing (b) our *A-MoD* attention routing (c).

1 INTRODUCTION

Increasing the model size has enabled transformer-based deep learning models to achieve state-of-the-art performance across various domains, including computer vision (Dosovitskiy et al., 2021) and natural language processing (Hoffmann et al., 2022; Kaplan et al., 2020) – even unlocking emergent capabilities (Wei et al., 2022). However, the computational costs of these large models present significant challenges (Thompson et al., 2020). Therefore, reaching a Pareto-optimal model to maximize both efficiency and performance is crucial.

Jacobs et al. (1991) originally introduced conditional computation via mixture of experts, laying the foundations to increase model sizes while maintaining FLOPs, by dynamically activating only

a subset of the model parameters, termed experts, conditioned on the input. This principle allowed scaling towards outrageously large networks (Shazeer et al., 2016) and is leveraged at the forefront of current Large Language Models (LLMs) (Jiang et al., 2024).

Recently, Raposo et al. (2024) introduced Mixture-of-Depths (MoD) as a variant of mixture of experts. For MoD models, the computational costs are dynamically reduced by processing only a subset of tokens in a layer while the remaining tokens skip the layer (see Fig. 1). Compared to baselines with equivalent FLOPs, MoDs can perform favorably on language tasks. A crucial component of MoD is its router, which receives tokens as inputs and, given a user-defined capacity, determines which tokens should enter or skip a layer. The router usually consists of a linear layer that is jointly trained along with the model (Fig. 1).

The routing mechanisms heavily influence the model performance for multiple reasons. First, routing introduces noise into the training process, as the routing is a discrete decision and is often performed at multiple layers and per token. Second, routers depend on additional layers, and hence, need to be trained from scratch when adapting a vanilla pretrained model to an MoD model. Lastly, the router adds a small computational overhead to the sparsified model.

Hence, in this paper, we ask and address the question: *Can we improve the routing mechanism in MoD models based on information that is already available within the model, instead of using additional trainable parameters within the router?* We find the answer to our question in the attention mechanism of commonly used transformer architectures (Vaswani et al., 2017).

We assume that the attention maps can be used to estimate the importance of a token, by averaging its interaction with other tokens. Based on that, we propose to aggregate the information in the attention maps and use it as an importance measure for token routing in MoD. We call our method attention routing for MoD: *A-MoD* (Fig. 1c). We find that *A-MoD* can outperform standard routing in MoD networks across a range of model sizes and tasks consistently (as shown in Fig. 1). Not only is our *A-MoD* parameter-free, but it can also be applied to adapt off-the-shelf pretrained transformer models to MoDs with almost no additional training. We further validate our method empirically and show that routing scores computed by *A-MoD* are better correlated with token importance estimates compared to routing scores from standard routers.

This paper presents a significant advancement in the application of MoD, which stems from natural language processing, to the visual domain. Our primary contributions are:

- We find that MoD is effective for visual tasks, providing empirical evidence that it can outperform vanilla models in terms of both FLOPs and performance.
- We introduce *A-MoD*, a parameter-free routing method for MoDs based on the attention maps to compute token importance and demonstrate that *A-MoD* outperforms a standard router on ImageNet.
- Compared to standard MoD, *A-MoD* consistently selects important tokens, and routing decisions correlate with leave-one-out token importance that is estimated by removing tokens.

2 Related Work

Attention Maps The attention mechanism (Bahdanau, 2014) enables models to learn long-range dependencies within sequences. Transformers across language (Vaswani et al., 2017) and vision (Dosovitskiy et al., 2021) leverage the attention mechanism in the language domain and have become a de facto standard model. For images, attention maps have been shown to focus on key areas of an image (Carion et al., 2020; Jetley et al., 2018) such as objects, which can be utilized for effective routing.

Mixture of Experts and Mixture-of-Depths Since their introduction over three decades ago (Jacobs et al., 1991; Jordan & Jacobs, 1993), Mixture of Experts (MoE) have been applied to various model types. Shazeer et al. (2016) introduced MoEs to scale transformer architectures (Ludziejewski et al., 2024). Subsequently, MoEs have achieved extensive empirical success across vision and language tasks (Puigcerver et al., 2024; Jain et al., 2024; Fedus et al., 2022; Riquelme et al., 2021). Raposo et al. (2024) recently introduced the Mixture-of-Depths (MoD) architecture (see Fig. 1),

where each transformer block processes only a subset of tokens, achieving a favorable compute-performance trade-off.

Routing Methods Routing mechanisms are required for most conditional computation blocks (Cai et al., 2024). In MoE models for transformers, the purpose of the router is to match tokens to experts such that performance is maximized. Various methods (Liu et al., 2024) have been proposed such as learned routers (Shazeer et al., 2016) with token choice or expert choice routing (Zhou et al., 2022), linear matching (Lewis et al., 2021), hashing inputs to match experts (Roller et al., 2021) and using reinforcement learning (Clark et al., 2022; Bengio et al., 2015; 2013). Explicitly learning the routers is the current state-of-the-art (Dikkala et al., 2023), however, this approach mainly proves effective with a larger number of routing parameters and is prone to training instabilities (Ramachandran & Le, 2019). Thus, training routers that consistently lead to strong performance remains an open problem.

Our work focuses on improving the MoD architecture. We propose a novel routing mechanism, based on attention maps, thereby eliminating the need for a standard router. The tokens are routed in a parameter-free manner without any extra computational overhead.

3 METHOD

In this section, we explain the Mixture-of-Depths (MoD) architecture and introduce our improved attention-based MoD routing algorithm, *A-MoD*. Given an input in terms of tokens $\mathbf{X}$, the output predictions are calculated by a model $f(\mathbf{X}; \Theta)$ consisting of L Transformer blocks parameterized by a set of learnable weights Θ. Each transformer block includes a Multi-Head Self-Attention block with H heads, followed by two fully-connected layers with GeLU activations.

MoD (Raposo et al., 2024) layers only process a subset of selected important tokens, while the remaining tokens skip the layer. Whether a token is skipped, is determined by token importance scores estimated by a routing algorithm. The standard mechanism is to use a dedicated router which learns the importance scores with an additional linear projection for each MoD layer. In contrast, our *A-MoD* computes the scores directly from the attention maps of previous layers without the need of additional parameters.

Standard routing The standard approach to estimate routing scores in an MoD layer is using a linear layer that projects a token vector to a scalar representing its importance score. Formally, we consider the l-th transformer layer $f_l(\mathbf{X}^{l-1}; \theta_l)$ parameterized by a set of parameters θ_l with an input $\mathbf{X}^{l-1} = \left[\boldsymbol{x}_1^{l-1}; \boldsymbol{x}_2^{l-1}; \ldots; \boldsymbol{x}_N^{l-1}\right] \in \mathbb{R}^{N \times d}$ representing a token sequence of length N. Now, we can estimate token importance scores as $\boldsymbol{r}_i = (\mathbf{X}^{l-1}\mathbf{W}_r^l)_i$ where $\mathbf{W}_r^l \in \mathbb{R}^{d \times 1}$ is the parameter of the additional linear routing network. These tokens will be skipped or processed based on their scores as per the equation below:

$$\boldsymbol{x}_i^l = \begin{cases} r_i f_l\left(\mathbf{X}^{l-1}\right)_i + \boldsymbol{x}_i^{l-1} & \text{if} \quad \boldsymbol{r}_i \geq P_\beta(\mathrm{R}^l) \\ \boldsymbol{x}_i^{l-1} & \text{else} \end{cases} \tag{1}$$

Here, $P_\beta(\mathrm{R}^l)$ denotes the β-th percentile of all token importance scores R^l. β can be defined in terms of the capacity C as $\beta := 1 - \frac{C}{N}$, where $C \in (0, 1)$ is the capacity for the MoD layer. To learn the token importance scores during backpropagation, the output of the transformer layer is multiplied by the importance scores r_i, such that it can receive a non-zero gradient.

Attention routing In contrast to standard routing, we propose *A-MoD*, a method to compute routing scores based on attention without additional trainable parameters. *A-MoD* leverages the attention map of the previous layer to determine the routing scores for the current MoD layer, as shown in Figure 1c. The attention map $\mathrm{A}_h^{l-1} \in \mathbb{R}^{N \times N}$ of the h-th head from the previous layer can be computed as follows Vaswani et al. (2017) $\mathrm{A}_h^{l-1} = \text{softmax}\left((Q_h^{l-1})(K_h^{l-1})^T / \sqrt{d}\right)$. Here, $Q_h^{l-1} \in \mathbb{R}^{N \times d}$ and $K_h^{l-1} \in \mathbb{R}^{N \times d}$ are query and key matrices computed from the previous layer respectively, and d is the embedding dimension of query and key. Each element $\boldsymbol{a}_{h,ji}^{l-1}$ of A_h^{l-1} indicates how much information from the i-th token is considered when computing the j-th output. Aggregating $\boldsymbol{a}_{h,ji}^{l-1}$ across all rows yields a measure of the relevance of the i-th token with respect

Table 1: *A-MoD* **mostly outperforms MoD with standard routing and the isoFLOP baseline on ImageNet**, both for 50% and 12.5% capacity.

Model	Configuration	C$= 12.5\%$		C$= 50\%$	
		FLOPs (G)	**Accuracy (%)**	**FLOPs (G)**	**Accuracy (%)**
DeiT-Tiny	isoFLOP	0.75	67.4	0.95	71.1
	MoD	0.71	67.52	0.92	69.78
	A-MoD	0.71	**69.76**	0.92	**71.8**
DeiT-Small	isoFLOP	2.3	73.53	3.47	78.04
	MoD	2.6	76.07	3.42	77.43
	A-MoD	2.6	**76.98**	3.42	**78.66**
ViT-Base	isoFLOP	8.8	77.69	13.21	79.28
	MoD	9.8	**78.49**	13.1	79.5
	A-MoD	9.8	78.42	13.1	**80.4**
ViT-Large	isoFLOP	33.4	77.64	46.24	80.28
	MoD	34.5	81.1	45.92	82.04
	A-MoD	34.5	**81.37**	45.92	**82.82**

to all other tokens. In *A-MoD*, we propose to compute a token importance score by averaging the corresponding attention values across all rows and attention heads as $\boldsymbol{r}_i = \frac{1}{HN}\sum_{h=1}^{H}\sum_{j=1}^{N}\boldsymbol{a}_{h,ji}^{l-1}$. Based on the score computation above, the output from the l-th layer can then be calculated as:

$$\boldsymbol{x}_i^l = \begin{cases} f_l\left(\mathbf{X}^{l-1}\right)_i + \boldsymbol{x}_i^{l-1} & \text{if} \quad \boldsymbol{r}_i \geq P_\beta(\mathrm{R}^l) \\ \boldsymbol{x}_i^{l-1} & \text{else} \end{cases} \tag{2}$$

We note that, for *A-MoD*, we do not multiply the token scores $\boldsymbol{r}_i$ by the output, as the attention maps are already learnable in the previous layer. This preserves the original token output, promoting faster training when adapting from a vanilla pretrained checkpoint. In contrast, standard routing (see Eq. (1)) requires this term in order to learn the routing scores via backpropagation.

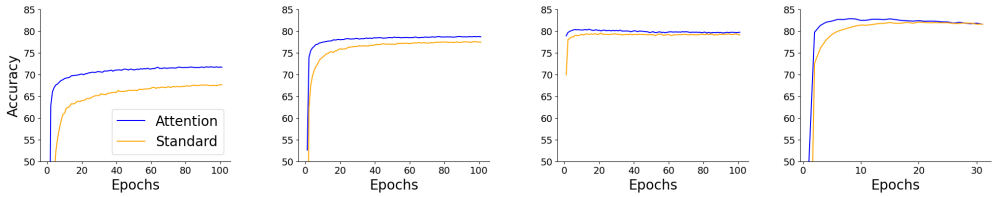
4 Experiments

We demonstrate the effectiveness of *A-MoD* across models on the ImageNet dataset(Russakovsky et al., 2015). Details of our training setup are provided in Appendix A.3.

***A-MoD* improves performance for finetuning** For finetining, we train each MoD model on ImageNet. Across all ViT models (ranging from 5M to 300M parameters), *A-MoD* mostly outperforms standard routing. Results for MoDs with 50% and 12.5% capacity are presented in Table 1. Through the training curves presented In Fig. 2 for 50% capacity and Fig. 5 for 12.5% capacity in the Appendix we highlight that *A-MoD* converges faster. Fig. 1 shows that *A-MoD* is Pareto-optimal for FLOPs vs accuracy, when compared with standard routing and isoFLOP baselines.

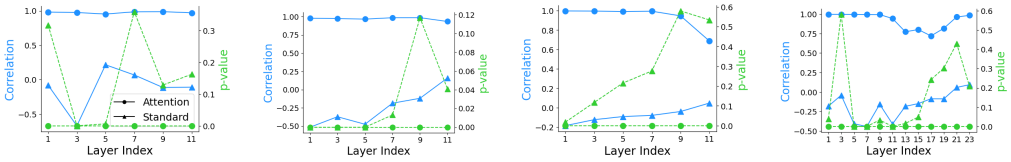
For the DeiT-Tiny model with 50% capacity (see Fig. 2(a)), *A-MoD* outperforms MoD by more than 2% and by 1% on the other larger models. Similarly, for 12.5% capacity, *A-MoD* outperforms standard routing on both DeiT-Tiny and Small and is on par for the larger variants. While *A-MoD* is marginally worse for the ViT-Base model for 12.5% capacity, it requires fewer epochs to converge as shown in the convergence plots in Fig. 5(c) (in the Appendix). Overall, Table 1 along with the training curves in Fig. 2 confirm that *A-MoD* can outperform MoDs with standard routing and the isoFLOP baselines.

Adapting from pretrained checkpoints As described in Eq. (2), *A-MoD* can compute routing scores solely based on the attention maps and it does not multiply the output of each MoD block with the routing score, thus largely conserving the token output. Both properties allow *A-MoD* finetuned from a pretrained checkpoint with attention routing to converge with minimal training. Fig. 2 illustrates that *A-MoD* enables much faster convergence, greatly reducing the required training time compared to standard routing.



(a) DeiT-T (b) DeiT-S (c) ViT-B (d) ViT-L

Figure 2: ***A-MoD* achieves better performance and faster convergence on ImageNet-1k.** Fine-tuning with *A-MoD*: Results comparing *A-MoD* with standard routing and isoFLOP baselines with 50% capacity on ImageNet.



(a) DeiT-T (b) DeiT-S (c) ViT-B (d) ViT-L

Figure 3: ***A-MoD* shows higher correlation between routing scores and leave-one-out token importance.** Correlation and p-values of the routing scores with layer-wise leave-one-out token importance on ImageNet.

Attention routing identifies important tokens To understand why *A-MoD* improves over standard routing, we investigate the routing scores and their correlation with leave-one-out (Hastie et al., 2009) token importance. Our goal is to estimate the relationship between the importance of a token and the routing score assigned to it by a standard or *A-MoD* router. Based on our empirical results, we conjecture that *A-MoD* weights are better correlated with token importance in comparison with standard routing, thus enabling *A-MoD* to always choose the most relevant tokens.

We first verify this claim by visualizing the routing in case of individual examples from ImageNet-1k as shown in Fig. 4. The figure highlights which patches of the image are chosen by the router in each MoD layer. In case of *A-MoD* (bottom), the router selects tokens that are part of the bird outline and face starting from the third MoD layer. In contrast, standard routing (top) selects more tokens that are part of the background, up to the last layer.

To quantify our qualitative observations, we compute the correlation of the routing scores with token importance estimates. For the importance of a token, we compute the change in loss of the model if that token is omitted in the vanilla transformer i.e. leave-one-out token importance. A large change in loss implies higher token importance and we would expect that token to have a higher routing score. The correlation of the routing scores for both standard routing and *A-MoD* with the token importance is shown in Fig. 3 along with the corresponding p-values.

We observe that routing scores computed by *A-MoD* consistently have a very high correlation with token importance suggesting that attention routing assigns higher scores to important tokens. In contrast, standard routing sometimes even has a negative correlation with token importance, implying that it can assign higher scores to less important tokens. Moreover, all the p-values observed for *A-MoD* were lower than 10^{-8} whereas they were significant (in some layers even larger than 0.5) in case of standard routing, implying higher uncertainty in case of standard routing.

5 Conclusion

We propose *A-MoD*, a variation of Mixture-of-Depths (MoD) with attention routing instead of a standard router. To compute token importance for an MoD layer, *A-MoD* utilizes the attention maps from its previous layer, thereby achieving attention routing without additional parameters. In case of training from a pretrained checkpoint, leveraging trained attention information also leads to

increased training stability and faster convergence compared to vanilla MoD. Furthermore, we empirically demonstrate that *A-MoD* outperforms standard MoD across different model configurations and datasets while making better routing decisions.

REFERENCES

Dzmitry Bahdanau. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*, 2015.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*.

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts. *arXiv preprint arXiv:2407.06204*, 2024.

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*, pp. 213–229. Springer, 2020.

Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. Unified scaling laws for routed language models. In *International conference on machine learning*, pp. 4057–4086. PMLR, 2022.

Nishanth Dikkala, Nikhil Ghosh, Raghu Meka, Rina Panigrahy, Nikhil Vyas, and Xin Wang. On the benefits of learning to route in mixture-of-experts models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9376–9396, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

William Fedus, Jeff Dean, and Barret Zoph. A review of sparse expert models in deep learning. *arXiv preprint arXiv:2209.01667*, 2022.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

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.

Gagan Jain, Nidhi Hegde, Aditya Kusupati, Arsha Nagrani, Shyamal Buch, Prateek Jain, Anurag Arnab, and Sujoy Paul. Mixture of nested experts: Adaptive processing of visual tokens. *arXiv preprint arXiv:2407.19985*, 2024.

Saumya Jetley, Nicholas A Lord, Namhoon Lee, and Philip HS Torr. Learn to pay attention. In *International Conference on Learning Representations*, 2018.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. In *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, volume 2, pp. 1339–1344 vol.2, 1993. doi: 10.1109/IJCNN.1993.716791.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

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*, pp. 6265–6274. PMLR, 2021.

Tianlin Liu, Mathieu Blondel, Carlos Riquelme Ruiz, and Joan Puigcerver. Routers in vision mixture of experts: An empirical study. *Transactions on Machine Learning Research*, 2024.

I Loshchilov and F Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jan Ludziejewski, Jakub Krajewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, et al. Scaling laws for fine-grained mixture of experts. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024.

Joan Puigcerver, Carlos Riquelme Ruiz, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *International Conference on Learning Representations*, 2024.

Prajit Ramachandran and Quoc V. Le. Diversity and depth in per-example routing models. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=BkxWJnC9tX`.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

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*, 34:8583–8595, 2021.

Stephen Roller, Sainbayar Sukhbaatar, Jason Weston, et al. Hash layers for large sparse models. *Advances in Neural Information Processing Systems*, 34:17555–17566, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

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. In *International Conference on Learning Representations*, 2016.

Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, 2020.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/touvron21a.html`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10809–10818, 2022.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

A Appendix

A.1 Token importance

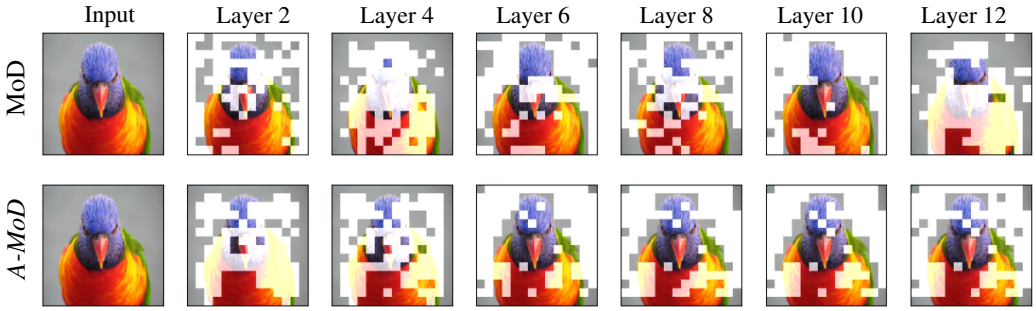


Figure 4: ***A-MoD* exhibits more meaningful routing compared to MoD.** Routing visualization: Example of DeiT-Small with 50% capacity on ImageNet. Each example shows tokens chosen by standard MoD (top) and *A-MoD* (bottom) for every MoD layer, white patches denote skipped. Each column represents a MoD layer as depth increases from left to right.

A.2 Model Specs

We choose four different transformer-based architectures:

Table 2: Specifications of transformer-based models used in experiments

Model	**Parameters (M)**	**FLOPS (G)**
DeiT-Tiny	5.72	1.26
DeiT-Small	22.05	4.61
ViT-Base	86.57	17.58
ViT-Large	304.72	191.21

A.3 Training setup

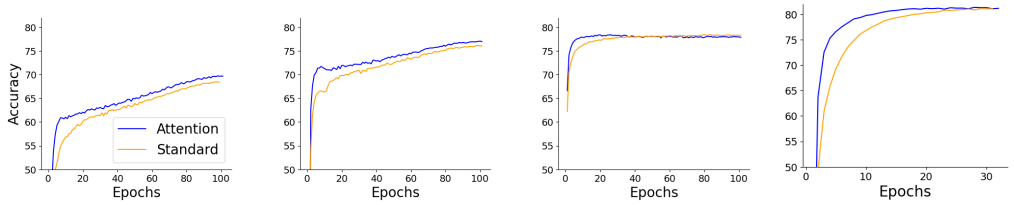
We evaluate *A-MoD* across four vision transformer architectures of varying sizes: DeiT-Tiny, DeiT-Small (Touvron et al., 2021), ViT-Base and ViT-Large (Dosovitskiy et al., 2021). Each MoD architecture is adapted from a vanilla pretrained checkpoint on ImageNet-1k (Russakovsky et al., 2015). Starting from this checkpoint, we train the MoD models with 50% and 12.5% capacity as described in Section 3, i.e., 50% and 12.5% tokens are processed in each MoD layer, respectively. Following Raposo et al. (2024), we alternate between MoD layers and dense layers in our MoD architecture i.e. every second layer is an MoD layer. Each model is trained with the AdamW optimizer (Loshchilov & Hutter, 2017) for 100 epochs using a batch size of 128 and a learning rate of $1e-5$ with a linear warmup followed by cosine annealing.

A.4 Additional results for finetuning on ImageNet-1k

Fig. 5 presents the convergence results for finetuning on ImageNet-1k with *A-MoD* at 12.5% capacity. Table 3 denotes the accuracy of *A-MoD* and standard routing without any training, after adapting the MoD weights from a vanilla pretrained checkpoint.

A.5 Effect of multiplying routing weights

We compare *A-MoD* with a modified version which multiplies the attention routing scores to the output of the MoD layer. Results in Fig. 6 show that multiplying the routing scores to the output can slow down convergence.

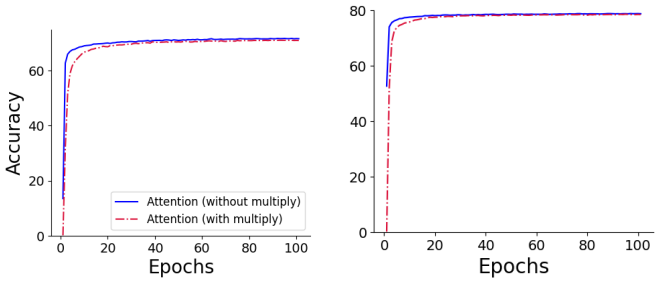


(a) DeiT-T (b) DeiT-S (c) ViT-B (d) ViT-L

Figure 5: ***A-MoD* achieves better performance and faster convergence on ImageNet-1k.** Finetuning with *A-MoD*: Results comparing *A-MoD* with standard routing and isoFLOP baselines for 12.5% capacity on ImageNet-1k.

Table 3: ***A-MoD* improves adaptation.** Accuracy of MoD on ImageNet-1k, adapted from a pretrained checkpoint, without any training.

Model	**Configuration**	**C = 50%**	**C = 12.5%**
DeiT-Tiny	MoD	4.45	0.42
	A-MoD	**52.6**	**0.97**
DeiT-Small	MoD	0.23	0.16
	A-MoD	**13.49**	**0.35**
ViT-Base	MoD	69.91	62.25
	A-MoD	**78.88**	**66.62**
ViT-Large	MoD	0.43	0.2
	A-MoD	**49.06**	**6.03**



(a) DeiT-T (b) DeiT-S

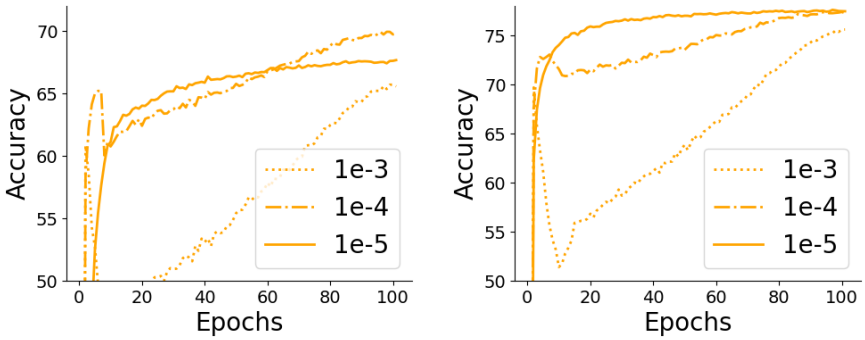
Figure 6: ***A-MoD* with routing scores multiplied to MoD output**. Multiplying the output of the MoD block with routing scores for *A-MoD* (red curve) compared to the proposed *A-MoD* without multiplication (blue curve).

A.6 Effect of learning rates

We identify the optimal learning rates for finetuning by conducting a sweep across a range of learning rates for finetuning on ImageNet-1k as shown in Fig. 7 and Fig. 8.

A.7 Comparison with token pruning methods

We compare our method *A-MoD* with other token-pruning and token-merging including Token Mergin (ToME) (Bolya et al.), A-ViT (Yin et al., 2022) and Dynamic-ViT (Rao et al., 2021) to validate the performance of *A-MoD*. We compare with the baseline results provided in Table 11 in Bolya et al. and Table 3 in Yin et al. (2022). However, we note that ToMe (Bolya et al.) trains their models with distillation while the other methods do not, which aids ToMe. Results are provided in Table 4.



(a) DeiT-T (b) DeiT-S

Figure 7: Sweep over learning rates on ImageNet-1k for standard routing.

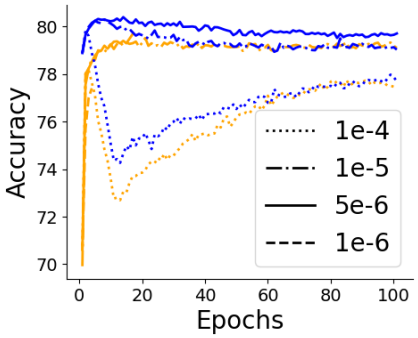


Figure 8: Sweep over learning rates on ImageNet-1k for standard routing and *A-MoD* on ViT-Base. Orange curves denote standard routing and blue curves denote attention routing.

Table 4: Comparison with other token-pruning and merging methods (* denotes training with distillation).

Model	Method	Top-1 Acc (%)	FLOPs (G)
DeiT-T	A-MoD	71.8	0.9
	A-ViT	71.0	0.8
	Dynamic ViT	70.9	0.9
	ToMe (with distillation)	71.69*	0.93
DeiT-S	A-MoD	78.66	3.42
	A-ViT	78.6	3.6
	Dynamic ViT	78.3	3.4
	ToMe (with distillation)	79.68*	3.43

A.8 Efficiency of *A-MoD*

To highlight the efficiency of A-MoD, we compare it with the baseline DeiT-S and report the top-1 accuracy on ImageNet. A-MoD is able to reduce the number of FLOPs by up to 18% without dropping performance, with standard training and no additional tricks. Results are provided in Table 5.

A.9 Training from scratch

We also provide results for training from scratch on ImageNet and observe that A-MoD outperforms standard routing as shown in Table 6.

Table 5: Comparison of A-MoD (70% capacity) with the vanilla DeiT-S baseline which has more FLOPs.

Model	**FLOPs (G)**	**Top-1 Accuracy (%)**
DeiT-S Baseline	4.6	79.6
A-MoD (C = 70%)	3.8	79.63

Table 6: Training from scratch comparison for *A-MoD* and MoD on ImageNet-1k.

Model	**Training Epochs**	**Method**	**Accuracy (%)**
DeiT-S	300	A-MoD	76.63
		MoD	75.90
ViT-Base	160	A-MoD	73.66
		MoD	72.47

A.10 Model Throughput

We provide a comparison of model throughput in Fig. 9. *A-MoD* has a higher throughput (img/s) in comparison to MoD and isoFLOP baselines. We also provide a breakdown of each method using the PyTorch profiler to highlight the CPU and GPU time used by each method for both the Attention layer and the MLP layer as shown in Fig. 10.

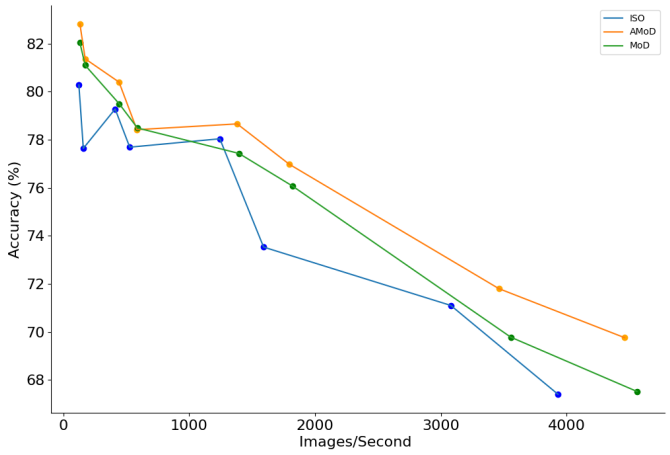


Figure 9: Accuracy vs Throughput for MoD vs ISOFlop Models with Batch Size 100 on Nvidia A100 GPU.

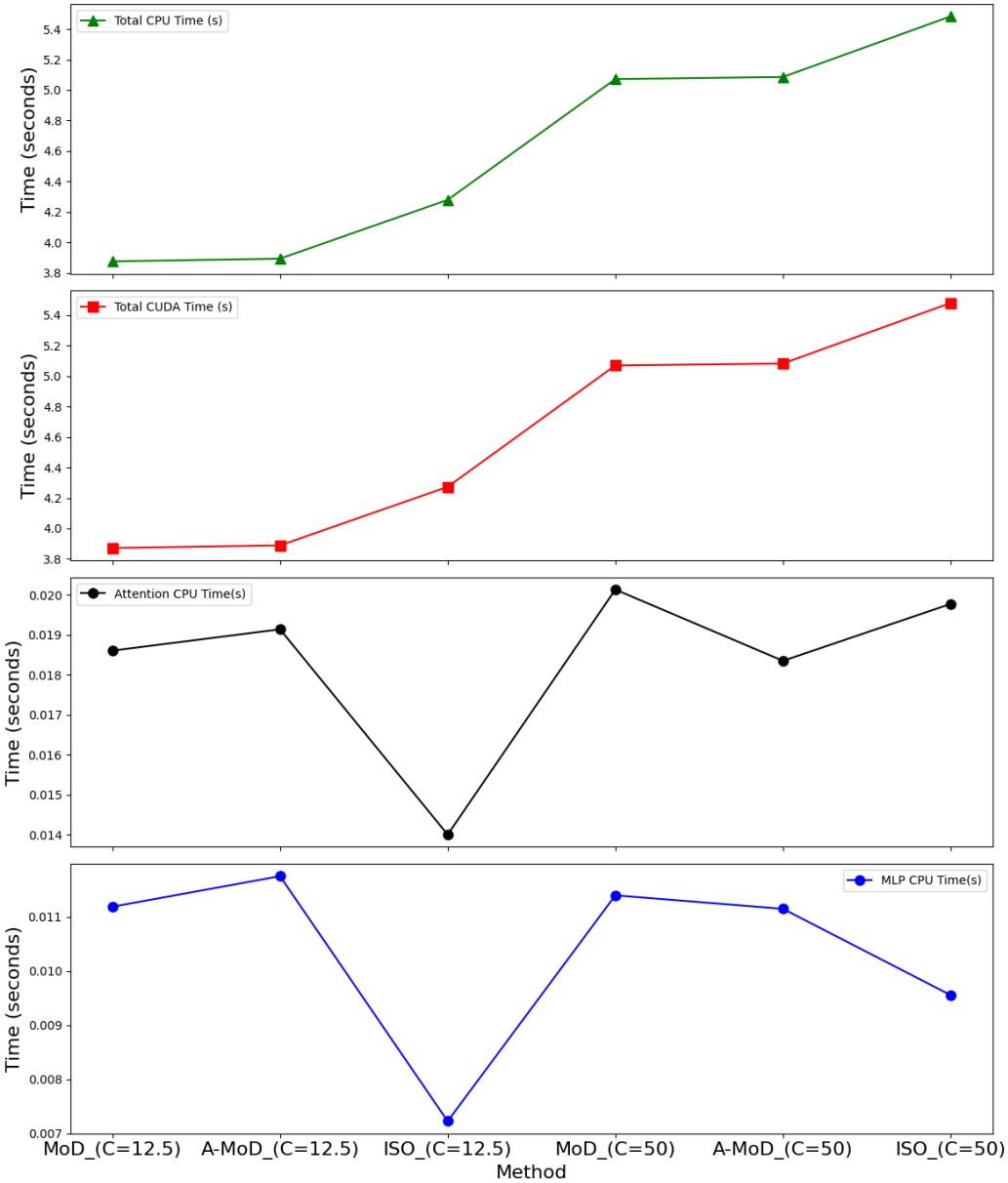


Figure 10: Profiling *A-MoD*, MoD and isoFLOP **ViT-Base** methods on Nvidia A100 GPU. The x-axis shows different models from left to right: MoD, *A-MoD* and isoFLOP for both C=12.5% and C=50%.