Using Mixture of Experts to Fine-Tune Robotic Video Transformers

Muhammad Qasim Ali[1]
Winnie Trandinh[2] Zain Altaf[3] Alexander Wong[1]
Author[1], **Coauthor**[2], **UW Coauthor**[3], **UW VIP Lab Coauthor**[1]
[1]Vision and Image Processing Group, Systems Design Engineering, University of Waterloo
[2]Centre for Mechatronics and Hybrid Technologies, Computing and Software, McMaster University
[3]Biomedical Engineering, University of Waterloo
{m45ali, zaltaf, alexander.wong}@uwaterloo.ca
{trandint}@mcmaster.ca

Abstract

Video generation models have recently showcased impressive results, generating high quality visual features with realistic physics and motion. Such video generators are intriguing for robotics because after fine-tuning to the robotic embodiment, have the potential to serve as generalizable world models and real-world simulators. Among video generation approaches, masked video transformers provide a computationally efficient alternative to diffusion-based methods. Building on recent successes of Mixture of Experts (MoE) in transformer architectures, we propose a novel approach to improve pre-trained robotic video transformers using sparsely gated MoE. Our method replaces the feedforward layers of the transformer block with sparely gated MoE layers. We also introduce an innovative weight initialization scheme that improves training convergence while fine-tuning masked video transformers. We evaluate our method on the 1xgpt humanoid robotic dataset, demonstrating improvements in both cross-entropy loss (0.07 reduction) and LPIPS scores (0.007 reduction). Our findings suggest that MoE-based fine-tuning with strategic weight initialization can enhance the performance of robotic video transformers while maintaining computational efficiency through sparse expert activation.

1 Introduction

Large foundation models pre-trained on large and diverse datasets have led to incredible progress in generating text, images, and video [1]. Robotics is another domain where foundation models have showcased promising results ([2], [3]). For example, generative approaches to learning manipulation policies have shown remarkable behavior, such as the ability to execute complex trajectories. However, the performance of such robotic foundation models suffer from limited reliability and generalization, in part due to the lack of pre-existing internet-scale robotic datasets and the cost of collecting real-world robot trajectories.

Interactive real-world simulators, models that generate future image frames of the environment conditioned on past frames and actions, offer a solution to the data shortage. While robotic policies require successful trajectories to learn from ([4], [2], [3]), video simulators can leverage much broader and diverse datasets to generalize in many more domains. For example, video data on the internet, robotic data from different embodiments, and failed robot trajectories can still be useful to the video simulator to learn general-purpose visual features and intuitive physics of the world. Such a simulator or video generator could serve as a world model [5] for robotics. They can be used to train robotic policies on diverse scenarios, plan a sequence of actions without trying them in the real-world, and predict changes in the environment [6].

For video simulators to be effective in robotic planning and training robotic policies, they must be visually realistic, model the physics of the world accurately, and efficiently execute on robotic hardware. A popular approach to video generation is to train transformers with flow-matching [7] and denoising diffusion objectives ([8], [9]). However such models are computationally expensive to sample from during inference. An alternative approach to visual generation is using auto-

regressive and bi-directional transformers [10]. Such generative transformer based approaches are computationally efficient, taking orders of magnitude fewer sampling steps [11].

Sparsely gated mixture of experts (MoE) ([12]) has recently been used to improve performance of many language models ([13], [14]). Such architectures train a multitude of specialized experts, but activate only a sparse set of experts during any given inference. Sparsely gated MoE thus allow models to maintain efficiency while containing many parameters, by activating a smaller number of weights during inference.

In this paper, we explore the usage of mixture of experts in bi-directional transformers for robotic video generation. We investigate if adding MoE layers to a pre-trained robotic video generator improves performance. Furthermore, we propose a novel method to initialize MoE layers in such fine-tuning cases to enable faster training convergence. We evaluate our method on the tokenized frames of the 1xgpt humanoid robotic dataset [15]. Our findings showcase that the MoE layer with our weight initialization scheme yields an improvement of 0.07 on cross entropy loss and 0.007 on the Learned Perceptual Image Patch Similarity (LPIPs) metric.

2 Problem Formulation

Let $t \in [0...T]$ represent time, $V \in \mathbf{R}^{H \times W}$ denote frames of a video, and a represent embeddings of actions taken by the robot. Robotic video simulators utilize past video frames $V_{1...t}$ to generate future video frame V_{t+1}. Such a next-frame video simulator can be called autoregressively to predict frame V_{t+2}, and so on. Interactive robot simulators generate future frames V_{t+1} conditioned on past video frames $V_{1...t}$ and past actions $a_{1...t}$. As we focus on architecture design, we use a non-interactive robot simulator (i.e. do not condition video generation on actions). In this paper, our robotic simulator is a pre-trained masked video transformer, and our goal is to improve its performance on the given dataset.

3 Background

Learning to generate video frames within the high dimensional $H \times W$ pixel space can be slow and computationally expensive. Variational Auto-encoders (VAEs) [16] can be used to compress the large image spaces into a much smaller $z \in \mathbf{R}^{H' \times W'}$ latent space. Vector Quantized Variational Auto-encoders (VQ-VAEs) [17] are also trained to reconstruct data samples from a variational objective; however, their latent space $z_k \in \mathbf{R}^D, k \in 1, ..., K$ is discretized into a finite vocabulary or codebook of size K using a quantization scheme. The smaller discretized latent spaces of VQ-VAEs allow for utilizing sequence-to-sequence transformers ([18]) rather than computationally expensive methods like denoising diffusion ([19], [20]) and Flow-Matching ([7], [21]). Furthermore, such approaches allow for drawing inspiration from language models, which also operate on discretized embeddings.

Various transformer-based approaches have been used to generate images and videos. DALL-E [22] used image transformers to generate images autoregressively. Mask-GIT [23] trained a bidirectional transformer to generate images from a quantized latent space using a masking training objective. It also proposed a novel non-autoregressive decoding method to synthesize images in finite and fewer computational steps than many prior diffusion [24] and autoregressive generation strategies. MAGVIT [25] extended Mask-GiT to video generation by using a spatio-temporal VQ-VAE and an improved masking scheme. MAGVIT2 [11] improves MAGVIT by designing an improved VQ-VAE tokenizer, yielding better image and video generation results than diffusion baselines and in much fewer sampling steps. Open-MAGVIT2 [26] contains an open-source implementation of the VQ-VAE tokenizer described in MAGVIT2.

Mixture of Experts [27] is an architecture design strategy based around training multiple experts that specialize in different behaviors and different subspaces of the input data. A gating mechanism is used to select the contribution of each expert in the output signal. MoE designs have found extensive applications in language models ([13], [14]). Many of these works utilize the sparsely gated Mixture of Experts variant [12], where contributions are weighted from a sparse set of experts rather than taking a weighted contribution of all experts.

4 Methodology

We detail the modifications we make to the architecture of the pre-trained transformer in 4.1. The weight initialization scheme is described in 4.2.

4.1 Architecture

We replace the feedforward block in each of the transformer blocks with sparsely gated MoE layers. The feedforward layers can account for as much as 90% of the parameters within multi-modal transformers [28]. By improving these feedforward layers, we can potentially have a substantial impact on the performance of the transformer.

Our mixture of experts layer MOE consists of N ex-

pert networks $\{f_1, ...f_N\}$ and a gating network g. Each expert f_i projects input x to some higher dimensional hidden embedding with a linear layer, applies a non-linear activation function, and finally projects the hidden embedding back to its original embedding size. The gating network is a linear network that predicts an N dimensional vector containing the weight/contribution of each expert in the final output. The gating network simply projects the input x to an N dimensional vector before applying a softmax activation. We use a sparsely gated MoE, where the weights associated with each expert (Eq.1) are sparsified with a topK function:

$$w = \text{softmax}(\text{topK}(g(x), k)) \tag{1}$$

$$\text{topK}(x, k)_i = \begin{cases} x_i, & \text{if } x_i \text{ in } k \text{ largest elements of } x \\ -\infty, & \text{otherwise.} \end{cases} \tag{2}$$

$$MOE(x) = \sum_{i=1}^{N} w_i f_i(x) \tag{3}$$

The topK function (Eq.2) retains the weights of the k largest entries and sets the other entries to $-\infty$. The softmax function ensures that the weights sum to one, and that the final contribution of entries with $-\infty$ is zero. The final output (Eq.3) of the MoE layer is simply a weighted sum between the activated experts f_i and the associated weights of the expert w_i. Such a sparse gating scheme saves computational resources, as the predictions of only a subset of the experts need to be computed. To encourage equal utilization of the experts, we incorporate router z-loss L_z and auxiliary load balancing loss L_B [29] into our training objective.

4.2 Weight Initialization

Naively replacing the feedforward layers with sparsely gated MoE layers leads to poor training convergence. After 10k training iteration, the training cross-entropy loss remained over 10, even though the training loss of the original fine-tuned model was 8.6.

Instead of initializing the MoE layers randomly, we instead initialize each of the N experts with the weights of the corresponding feedforward layer, which was previously fine-tuned on the dataset but replace by the MoE layer. The gating networks however are still initialized with random weights. Although each of the experts is initially identical, each expert eventually learns different weights due to the sparse aggregation scheme and the randomness caused by the gating network. In each mini-batch, the gating network picks a different set of experts, in part due to changes in the data and due to the randomness of expert selection by the gating network. Thus, the weights of each expert is optimized based on features from different data and combinations of previous experts. With this new initialization scheme, the training loss was able to match that of the fine-tuned model after 9k training iterations. Note that such an initialization scheme requires that each of the experts share the same architecture (hidden dimensions) with the original pre-trained transformer.

5 Experiments and Results

Dataset. We use the 1xgpt dataset [15] to train and evaluate our video simulator. The dataset contains over 100 hours of ego-centric video taken from the humanoid 1X robot. The humanoid robot executed various tasks such as folding a cloth, navigating a floor, and picking and placing various household objects.

Implementation Details. We build on top of the masked video transformer fine-tuned by the 1xgpt team [15]. An open-source implementation of the MAGVIT2 tokenizer is used to compress RGB images from 256×256×3 into latent embeddings of shape 16×16×256. We use the factorized version of the tokenizer with a total vocabulary of size 2^{18}. We train the transformer model in the quantized latent space of the tokenizer using a cross-entropy objective to match the predicted tokens with the masked ground truth tokens. We do not condition the video simulator on the action tokens.

Our transformer architecture uses 32 spatial-temporal attention layers as described in Genie. We replace feedforward layers in all 32 layers with our sparsely gated MoE layers. We use four experts with k set to 2. Each expert processes 256-dimensional embeddings with a multi-layer perceptron, using 1024 hidden units and a GELU activation function. We train for 100k training iterations using the Adam [30] optimizer, a learning rate of $1e^{-5}$, and a batch size of 12. We utilize the same masking scheme and image sampling scheme described in Mask-GiT, but sample images in two computational steps instead of twelve.

Metrics. After fine-tuning the model on our training dataset, we evaluate performance on a separate evaluation dataset. We use two metrics to evaluate performance: cross entropy loss and Learned Perceptual Image Patch Similarity (LPIPS) [31]. We evaluate cross-entropy loss between the predicted visual tokens and the ground truth visual tokens. LPIPS is a metric used to measure the perceptual similarity between two images. LPIPS is computed using the ℓ_2 distance between the deep feature representations (we use AlexNet representations) of the predicted and ground truth images.

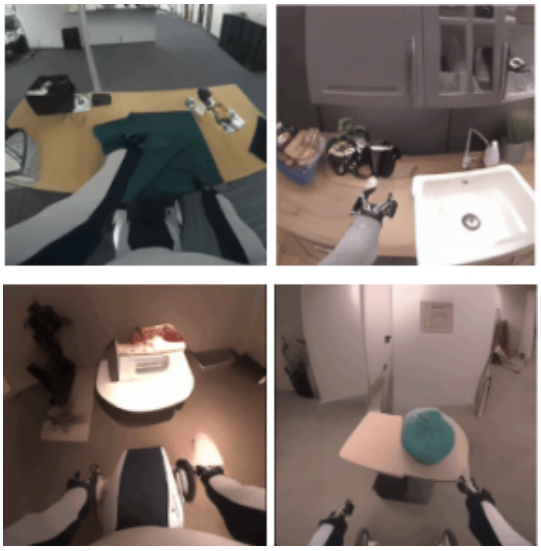

Figure 1: Sample frames generated after MoE fine-tuning model.

Rather than comparing raw pixel representations, LPIPS leverages deep features to assess similarity in ways that better correlate with qualitative and human perceptual quality.

Results. We show some qualitative results of generated frames in Figure 1. Table 1 shows that the MoE fine-tuning proposed in this paper improves both cross-entropy loss (0.07) and LPIPS score (0.007).

Metric	Pre-trained model	MoE fine-tuning
Cross entropy loss	9.2123	9.1412
LPIPS	0.2246	0.2174

Table 1: Comparison of cross-entropy loss and LPIPS with just the fine-tuned model and with the MoE fine-tuning suggested in this paper. Both metrics show improvement.

6 Conclusion

In this paper, we explored the application of Mixture of Experts to fine-tune masked video transformers. We concluded that replacing the feedforward layers of the transformer with Mixture of Expert layers improves the performance of the pre-trained video transformer. We also introduced a weight initialization scheme to help improve training convergence.

Acknowledgments

We thank 1xgpt for publicly releasing this humanoid dataset.

References

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang, "On the opportunities and risks of foundation models," 2022. [Online]. Available: https://arxiv.org/abs/2108.07258

[2] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," 2023. [Online]. Available: https://arxiv.org/abs/2307.15818

[3] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "Openvla: An open-source

vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[4] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," 2024. [Online]. Available: https://arxiv.org/abs/2303.04137

[5] D. Ha and J. Schmidhuber, "World models." *CoRR*, vol. abs/1803.10122, 2018. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1803.html#abs-1803-10122

[6] S. Yang, J. Walker, J. Parker-Holder, Y. Du, J. Bruce, A. Barreto, P. Abbeel, and D. Schuurmans, "Video as the new language for real-world decision making," 2024. [Online]. Available: https://arxiv.org/abs/2402.17139

[7] A. Polyak, A. Zohar, A. Brown, A. Tjandra, A. Sinha, A. Lee, A. Vyas, B. Shi, C.-Y. Ma, C.-Y. Chuang, D. Yan, D. Choudhary, D. Wang, G. Sethi, G. Pang, H. Ma, I. Misra, J. Hou, J. Wang, K. Jagadeesh, K. Li, L. Zhang, M. Singh, M. Williamson, M. Le, M. Yu, M. K. Singh, P. Zhang, P. Vajda, Q. Duval, R. Girdhar, R. Sumbaly, S. S. Rambhatla, S. Tsai, S. Azadi, S. Datta, S. Chen, S. Bell, S. Ramaswamy, S. Sheynin, S. Bhattacharya, S. Motwani, T. Xu, T. Li, T. Hou, W.-N. Hsu, X. Yin, X. Dai, Y. Taigman, Y. Luo, Y.-C. Liu, Y.-C. Wu, Y. Zhao, Y. Kirstain, Z. He, Z. He, A. Pumarola, A. Thabet, A. Sanakoyeu, A. Mallya, B. Guo, B. Araya, B. Kerr, C. Wood, C. Liu, C. Peng, D. Vengertsev, E. Schonfeld, E. Blanchard, F. Juefei-Xu, F. Nord, J. Liang, J. Hoffman, J. Kohler, K. Fire, K. Sivakumar, L. Chen, L. Yu, L. Gao, M. Georgopoulos, R. Moritz, S. K. Sampson, S. Li, S. Parmeggiani, S. Fine, T. Fowler, V. Petrovic, and Y. Du, "Movie gen: A cast of media foundation models," 2024. [Online]. Available: https://arxiv.org/abs/2410.13720

[8] F. Zhu, H. Wu, S. Guo, Y. Liu, C. Cheang, and T. Kong, "Irasim: Learning interactive real-robot action simulators," 2024. [Online]. Available: https://arxiv.org/abs/2406.14540

[9] S. Yang, Y. Du, K. Ghasemipour, J. Tompson, L. Kaelbling, D. Schuurmans, and P. Abbeel, "Learning interactive real-world simulators," 2024. [Online]. Available: https://arxiv.org/abs/2310.06114

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[11] L. Yu, J. Lezama, N. B. Gundavarapu, L. Versari, K. Sohn, D. Minnen, Y. Cheng, V. Birodkar, A. Gupta, X. Gu, A. G. Hauptmann, B. Gong, M.-H. Yang, I. Essa, D. A. Ross, and L. Jiang, "Language model beats diffusion – tokenizer is key to visual generation," 2024. [Online]. Available: https://arxiv.org/abs/2310.05737

[12] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," 2017. [Online]. Available: https://arxiv.org/abs/1701.06538

[13] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. de las Casas, E. B. Hanna, F. Bressand, G. Lengyel, G. Bour, G. Lample, L. R. Lavaud, L. Saulnier, M.-A. Lachaux, P. Stock, S. Subramanian, S. Yang, S. Antoniak, T. L. Scao, T. Gervet, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mixtral of experts," 2024. [Online]. Available: https://arxiv.org/abs/2401.04088

[14] W. Cai, J. Jiang, F. Wang, J. Tang, S. Kim, and J. Huang, "A survey on mixture of experts," 2024. [Online]. Available: https://arxiv.org/abs/2407.06204

[15] 1X Technologies, "1X World Model Challenge," Jun. 2024.

[16] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022. [Online]. Available: https://arxiv.org/abs/1312.6114

[17] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," 2018. [Online]. Available: https://arxiv.org/abs/1711.00937

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[19] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," 2022. [Online]. Available: https://arxiv.org/abs/2204.03458

[20] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020. [Online]. Available: https://arxiv.org/abs/2006.11239

[21] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," 2023. [Online]. Available: https://arxiv.org/abs/2210.02747

[22] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," 2021. [Online]. Available: https://arxiv.org/abs/2102.12092

[23] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, "Maskgit: Masked generative image transformer," 2022. [Online]. Available: https://arxiv.org/abs/2202.04200

[24] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, D. Parikh, S. Gupta, and Y. Taigman, "Make-a-video: Text-to-video generation without text-video data," 2022. [Online]. Available: https://arxiv.org/abs/2209.14792

[25] L. Yu, Y. Cheng, K. Sohn, J. Lezama, H. Zhang, H. Chang, A. G. Hauptmann, M.-H. Yang, Y. Hao, I. Essa, and L. Jiang, "Magvit: Masked generative video transformer," 2023. [Online]. Available: https://arxiv.org/abs/2212.05199

[26] Z. Luo, F. Shi, Y. Ge, Y. Yang, L. Wang, and Y. Shan, "Open-magvit2: An open-source project toward democratizing auto-regressive visual generation," 2024. [Online]. Available: https://arxiv.org/abs/2409.04410

[27] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.

[28] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, "Palm: Scaling language modeling with pathways," 2022. [Online]. Available: https://arxiv.org/abs/2204.02311

[29] B. Zoph, I. Bello, S. Kumar, N. Du, Y. Huang, J. Dean, N. Shazeer, and W. Fedus, "St-moe: Designing stable and transferable sparse expert models," 2022. [Online]. Available: https://arxiv.org/abs/2202.08906

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. [Online]. Available: https://arxiv.org/abs/1412.6980

[31] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," 2018. [Online]. Available: https://arxiv.org/abs/1801.03924