PRETRAINED TRANSFORMERS AS UNIVERSAL COMPUTATION ENGINES

Kevin Lu
UC Berkeley
kzl@berkeley.edu

Aditya Grover
Facebook AI Research
adityagrover@fb.com

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

Igor Mordatch
Google Brain
imordatch@google.com

ABSTRACT

We investigate the capability of a transformer pretrained on natural language to generalize to other modalities with minimal finetuning – in particular, without finetuning of the self-attention and feedforward layers of the residual blocks. We consider such a model, which we call a Frozen Pretrained Transformer (FPT), and study finetuning it on a variety of sequence classification tasks spanning numerical computation, vision, and protein fold prediction. In contrast to prior works which investigate finetuning on the same modality as the pretraining dataset, we show that pretraining on natural language improves performance and compute efficiency on non-language downstream tasks. In particular, we find that such pretraining enables FPT to generalize in zero-shot to these modalities, matching the performance of a transformer fully trained on these tasks[1].

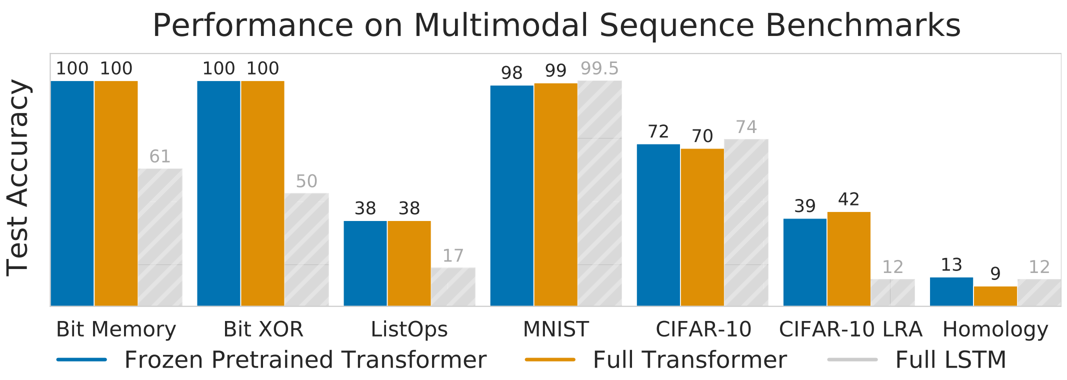

Figure 1: A *frozen* language-pretrained transformer (FPT) – without finetuning the self-attention and feedforward layers – can match the performance of a transformer fully trained on a downstream modality from scratch. We show results on diverse classification tasks (see Section A): numerical computation (Bit Memory/XOR, ListOps), image classification (MNIST, CIFAR-10), and protein fold prediction (Homology). We also show results for a fully trained LSTM to provide a baseline.

1 INTRODUCTION

The transformer architecture (Vaswani et al., 2017) has shown immense success in deep learning, serving as the backbone of massively large models for tasks in modeling of natural language (Brown et al., 2020), images (Dosovitskiy et al., 2020; Touvron et al., 2020), and proteins (Jumper et al., 2021), as well as multimodal datasets containing both images and text (Lu et al., 2019; Radford

[1]Code available at github.com/kzl/universal-computation.

et al., 2021). Inspired by these successes, we seek to explore the generalization capabilities of a transformer in transferring from one modality to another.

Classical approaches to sequence processing used recurrent neural network (RNN) based-approaches (Rumelhart et al., 1985; Hochreiter & Schmidhuber, 1997). In contrast, transformers utilize self-attention layers to extract features across tokens of a sequence, such as words (Vaswani et al., 2017) or image patches (Dosovitskiy et al., 2020). Furthermore, it has become common practice to train large models on unsupervised or weakly supervised objectives before finetuning or zero-shot generalization on a downstream task. However, the downstream tasks that have been studied are generally restricted to the same modality as the original training set: for example, train GPT (Radford et al., 2019) on a large language corpus, and finetune on a small task-specific dataset. Our goal in this work is to investigate finetuning on modalities distinct from the training modality.

We hypothesize that transformers, namely the self-attention layers, can be pretrained on a data-rich modality (e.g., natural language corpus) and identify feature representations that are useful for *arbitrary* data sequences, enabling effective downstream transfer of different modalities *without* expensive finetuning of the self-attention layers. In particular, we seek to investigate what pretrained language models (LMs) are capable of generalizing to other modalities with sequential structure, including numerical computation, image classification, and protein family prediction.

In Figure 1, we take a transformer model pretrained on natural language data, GPT-2 (Radford et al., 2019), and finetune only the input and output layers, as well as the positional embeddings and layernorm parameters. We call this model a Frozen Pretrained Transformer (FPT). On a range of tasks across a variety of modalities, FPT displays comparable performance to training the entire transformer or LSTM models, despite finetuning only $.1\%$ of the total number of parameters of the transformer model and none of the self-attention parameters. Additionally, we find FPT models also convergence faster during training. Our results suggest that the self-attention layers learned by a language model may have properties amenable to efficient universal computation. Through a series of experiments, we seek to investigate why language pretraining can transfer to other modalities by examining pretraining regimes, architecture choice, generalization abilities, model size, and importance of different sets of parameters in a transformer.

2 Methodology

We use a transformer model with linear input/output layers. Denote the embedding size/hidden dimension of the transformer as n_{dim}, the number of layers as n_{layers}, (note $n_{dim} = 768$ and $n_{layers} = 12$ for the base size models), the input dimension as d_{in}, the output dimension (number of classes) as d_{out}, and the maximum length of the sequence as l. We consider finetuning the following parameters of a pretrained GPT-2 model (Radford et al., 2019):

- **Output layer:** it is crucial to finetune the output layer since we are transferring to a completely new task – we use the simplest possible instantiation of an output network, being a single linear layer applied to the last output token output by the transformer. This highlights that almost all the computation is being performed by the frozen transformer. The output layer has $n_{dim} \times d_{out}$ parameters for the weight matrix. For example, for the base models on CIFAR-10, this comes out to $768 \cdot 10 = 7680$ parameters.

- **Input layer:** it is important to reinitialize a new input layer since we are reading in a new modality; in essence, we are learning how to query the transformer. This contrasts with prior unsupervised embedding evaluation techniques, such as linear probing – due to the change in modality, we instead should train the input layer as well, and evaluate if the frozen intermediate transformer model performs effective computation. The input layer has $d_{in} \times n_{dim}$ parameters for the weight matrix/embeddings, and an additional n_{dim} parameters if there is a bias term. For the base models on CIFAR-10, this comes out to $16 \cdot 768 = 13056$ parameters.

- **Layernorm parameters:** as is standard practice in other finetuning works (Rebuffi et al., 2017; Houlsby et al., 2019), we also finetune the affine layernorm parameters (scale and bias), which adapt to the statistics of the downstream task in a new domain. In GPT-2, layernorm is applied twice per block, so these are a total of $4 \times n_{dim} \times n_{layers}$ parameters. For the base models on CIFAR-10, these come out to $4 \cdot 768 \cdot 12 = 36684$ parameters.

- **Positional embeddings:** though we find these are surprisingly universal between modalities (see Section **??**), we generally see a small benefit to finetuning the positional embeddings which have a cheap parameter cost of $l \times n_{dim}$. For the base models on CIFAR-10, these come out to $64 \cdot 768 = 49512$ parameters.

Given the cheap linear scaling of these parameters, the parameter counts of large transformer models are dominated by the quadratic (in n_{dim} and l) self-attention and feedforward layers. For the base CIFAR-10 model with 124M parameters, these come out to approximately 0.086% of the network.

3 Empirical Evaluation

3.1 Can pretrained language models transfer to different modalities?

We investigate if the self-attention and feedforward layers – the main body – of a pretrained transformer can be applied to a classification problem in a different modality without finetuning. To do this, we apply our base method, where the input embedding layer, output readout layer, and layernorm parameters are finetuned.

Our results are shown in Figure 1. We find that across all seven tasks considered, FPT achieves comparable, if not marginally better performance than fully training a transformer. We believe these results support the idea that these models are learning representations and performing computation that is agnostic to the modality. We also note that both transformer variants significantly outperform LSTMs on some tasks, particularly ListOps and CIFAR-10 LRA, which have long sequence lengths of 512 and 1024, respectively.

We highlight a few important points for contextualizing these results. In particular, we find that it can be difficult to fully train a 12-layer transformer on some of these (relatively small) datasets, as training can either diverge/overfit or be unstable. For CIFAR-10, we report the full transformer results for a 3-layer model, and for CIFAR-10 LRA we report the number given for the 3-layer model from Tay et al. (2020). From an engineering perspective, this makes the full transformers harder to tune since we must choose model sizes that are stable and avoid overfitting – see Section 3.4 for more analysis. Furthermore, unlike some other works utilizing transformers for vision, we do not use convolutional input layers for a clean and fair analysis; only a linear layer is learned. Note that we also do not use 2D positional embeddings (or other domain-specific techniques), hence providing very weak inductive prior to the model.

3.2 How does language pretraining affect downstream performance?

Additionally, to further investigate the importance of language supervision, we also consider pretraining on the Bit Memory task, which simply allows the transformer to gain supervision working with arbitrary bit strings and performing memory/denoising. Our results are shown in Table 2. Pretraining on the bit memory task improves performance compared to the random models, but still lag behind training on natural language data. Additionally, we find that in terms of number of gradient steps/samples, pretraining on language leads to the fastest convergence, while pretraining on bit memory leads to somewhat slower convergence, and both pretraining methods converge significantly faster than the randomly initialized transformer models (more details in Section 3.5).

Table 1: Test accuracy of pretrained (FPT) vs randomly initialized (Random) models. The transformer is frozen. Random transformer initialization performs worse than pretraining with language.

Model	Bit Memory	XOR	ListOps	MNIST	C10	C10 LRA	Homology
FPT	100%	100%	38%	98%	68%	39%	13%
Random	76%	100%	37%	92%	62%	36%	9%

Additionally, to further investigate the importance of language supervision, we also consider pretraining on the Bit Memory task, which simply allows the transformer to gain supervision working with arbitrary bit strings and performing memory/denoising. Our results are shown in Table 2. Pretraining on the bit memory task improves performance compared to the random models, but still lag behind training on natural language data.

Table 2: Test accuracy of language-pretrained (FPT) vs bit memory-pretrained (Bit) models. Simply pretraining on this simple task performs well, but language provides additional supervision.

Model	**Bit Memory**	**XOR**	**ListOps**	**MNIST**	**C10**	**C10 LRA**	**Homology**
FPT	100%	100%	38%	98%	68%	39%	13%
Bit	100%	100%	37%	97%	63%	37%	8%

3.3 How important is the transformer self-attention architecture?

In Section 3.2 we found that the transformer architecture can already be fairly effective in this regime, even with only random parameters. Here, we consider using a random LSTM architecture instead of the transformer, allowing us to consider the raw effect of the architecture in this setting.

Table 3: Test accuracy of randomly initialized transformers vs randomly initialized LSTM models. Note that unlike in Figure 1, the LSTM here is frozen. Frozen LSTMs perform very poorly.

Model	**Bit Memory**	**XOR**	**ListOps**	**MNIST**	**C10**	**C10 LRA**	**Homology**
Trans.	76%	100%	37%	92%	62%	36%	9%
LSTM	50%	50%	17%	68%	34%	10%	6%

3.4 Does freezing the transformer prevent overfitting or underfitting?

Our general findings are that, in contrast to their fully trained counterparts, FPT models underfit, which lends them to further improvements by increasing model capacity. For example, in the CIFAR-10 LRA task, which is maximally difficult due to lack of inductive prior over the sequence (each pixel is fed in as an arbitrary token only ordered by a raster scan) and relatively small size of the CIFAR-10 dataset (50k images). In Table 4, we show the train/test gap between FPT versus training a 3-layer transformer from Tay et al. (2020), which we find to give stronger results than our experiments; in particular, they are much better than training a 12-layer transformer, which works poorly. Our results indicate that FPT is generally providing generalizable task representations without causing overfitting, whereas transformers can overfit arbitrarily poorly in low-data regimes.

Table 4: Train vs test accuracies on CIFAR-10 LRA task.

Model	**Number of Layers**	**Test Accuracy**	**Train Accuracy**
FPT (GPT-2)	12	39%	39%
Vanilla Transformer	3	42%	70%

3.5 Does language pretraining improve compute efficiency?

We consider the number of gradient steps to converge for FPT vs random transformer models. We generally find FPT models converge faster, which indicates that we can utilize the language pretraining to actually gain compute speedups for non-language tasks. Note that the bit memory pretraining method regime introduced in Section 3.2 generally falls between the two models, and notably is about 6× slower than FPT on Bit XOR, which is significantly better than random.

Table 5: Number of gradient steps until convergence for pretrained (FPT) vs randomly initialized (Random) models. Note that we use the same batch size and learning rate for both models.

Model	**Memory**	**XOR**	**ListOps**	**MNIST**	**C10**	**C10 LRA**	**Homology**
FPT	1×10^4	5×10^2	2×10^3	5×10^3	4×10^5	3×10^5	1×10^5
Random	4×10^4	2×10^4	6×10^3	2×10^4	4×10^5	6×10^5	1×10^5
Speedup	$4\times$	$40\times$	$3\times$	$4\times$	$1\times$	$2\times$	$1\times$

4 Conclusion

We proposed transferring a pretrained transformer-based language model for downstream tasks in non-language modalities. Through extensive empirical evaluation, we showed that these models could achieve performance competitive with transformers fully trained on the downstream task without having to finetune the self-attention and feedforward layers.

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

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.

Naomi K Fox, Steven E Brenner, and John-Marc Chandonia. Scope: Structural classification of proteins—extended, integrating scop and astral data and classification of new structures. *Nucleic acids research*, 42(D1):D304–D309, 2013.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Jie Hou, Badri Adhikari, and Jianlin Cheng. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 2018.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Kathryn Tunyasuvunakool, Olaf Ronneberger, Russ Bates, Augustin Žídek, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Anna Potapenko, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Martin Steinegger, Michalina Pacholska, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. High accuracy protein structure prediction using deep learning. 2021.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.

Thomas Miconi, Kenneth Stanley, and Jeff Clune. Differentiable plasticity: training plastic neural networks with backpropagation. In *International Conference on Machine Learning*, pp. 3559–3568. PMLR, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *Image*, 2:T2, 2021.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S Song. Evaluating protein transfer learning with tape. In *Advances in Neural Information Processing Systems*, 2019.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *arXiv preprint arXiv:1705.08045*, 2017.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*, 2020.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

A TASKS

We evaluate on a diverse set of classification tasks representative of different modalities. In particular, we are interested in if language models are inherently capable of *universal computation*, by which we mean the ability to learn representations for predictive learning across diverse modalities.

Bit memory. Similar to the task proposed by Miconi et al. (2018), we consider a bit memory task where the model is shown 5 bitstrings of length 1000. Afterwards, the model is shown a masked version of one of the bitstrings which has bits masked with probability 0.5, and the model is tasked with producing the original bitstring. The bitstrings are broken up into sequences of length 50, so that the model is fed 120 tokens of dimension 50.

Bit XOR. Similar to the bit memory task, the model is shown 2 bitstrings of length 5, where the model predicts the XOR of the two bitstrings. The bitstrings are shown 1 bit at a time, so the models are fed 10 tokens of dimension 1.

ListOps. Taken from Tay et al. (2020), the model is shown a sequence of list operations (e.g., `[ MAX 4 3 [ MIN 2 3 ] 1 0 ]`) and tasked with predicting the resulting output digit. This task evaluates the ability of a model to parse mathematical expressions and evaluate over a long context. The model is shown 1 token at a time, so the models are fed 512 tokens.

MNIST. We use the standard MNIST benchmark, where the model must classify a handwritten digit from a 32×32 black-and-white image. The tokens given to the model are 4×4 image patches, so the models are fed 64 tokens of dimension 16.

CIFAR-10. We use the standard CIFAR-10 benchmark (Krizhevsky et al., 2009), where each tokens given to the model are 4×4 image patches, so the models are fed 64 tokens of dimension 16.

CIFAR-10 LRA. This is a modified version of the above task taken from the Long Range Arena benchmark where the images are converted to grayscale and flattened with token length of 1 (Tay et al., 2020). As a result, the input sequence consists of 1024 tokens of dimension 1. This task is more challenging than vanilla CIFAR-10 classification above as the models must learn patterns over a significantly longer sequence length.

Remote homology detection. In this task, we are interested in predicting the fold for a protein, represented as an amino acid sequence. We use the datasets provided by TAPE (Rao et al., 2019; Fox et al., 2013; Hou et al., 2018), which generates a train/test split by holding out certain evolutionary groups. There are 20 common and 5 uncommon amino acids, and there are 1195 possible labels to predict. We only consider sequences of length less than 1024 for simplicity. The models are thus fed up to 1024 tokens of dimension 25.