High School Student Institute of Mathematics and Informatics

Summer Research School Symposium 2021

Transformers predicting the future

Author(s):

Radostin Cholakov
High School of Mathematics
"Acad. Kiril Popov" - Plovdiv
radicho123@gmail.com

Scientific Advisor(s):

Todor Kolev
Comrade Cooperative
Sofia, Bulgaria
t.kolev@comrade.coop

Abstract

Recurrent Neural Networks were, until recently, one of the best ways to capture the timely dependencies in sequences. However, with the introduction of the Transformer, it has been proven that an architecture with only attention-mechanisms without any RNN can improve on the results in various sequence processing tasks (e.g. NLP). Multiple studies since then have shown that similar approaches can be applied for images, point clouds, video, audio or time series forecasting. Furthermore, solutions such as the Perceiver or the Informer have been introduced to expand on the applicability of the Transformer. Our main objective is testing and evaluating the effectiveness of applying Transformer-like models on time series data, tackling susceptibility to anomalies, context awareness and space complexity by fine-tuning the hyperparameters, preprocessing the data, applying dimensionality reduction or convolutional encodings and coming up with new sophisticated approaches. We are also looking at the problem of next-frame prediction and exploring ways to modify some existing solutions in order to achieve better performance and make them learn generalized knowledge.

1 Introduction

Since its introduction, the Transformer [1] has revolutionized how neural networks can process sequential data and is currently the go to solution for a wide variety of natural language processing tasks. Analogous models are also being applied for image and video, point clouds [2], sound and time series data.

The Transformer (Fig. 1) adopts an encoder-decoder structure where the core function of each encoder layer is to generate information about which parts of the inputs are relevant to each other. The decoder part does the opposite, taking all the encodings and using their incorporated contextual information to generate an output sequence.

The inputs and outputs (target sequences) are first embedded into an n-dimensional space and since there are no recurrent networks that can remember how sequences are fed into a model the positions are added to the embedded representation of each item.

Both the encoder and the decoder are composed of modules that can be stacked on top of each other multiple times, which is described by Nx in the figure. The modules consist mainly of Multi-Head Attention and Feed Forward layers.

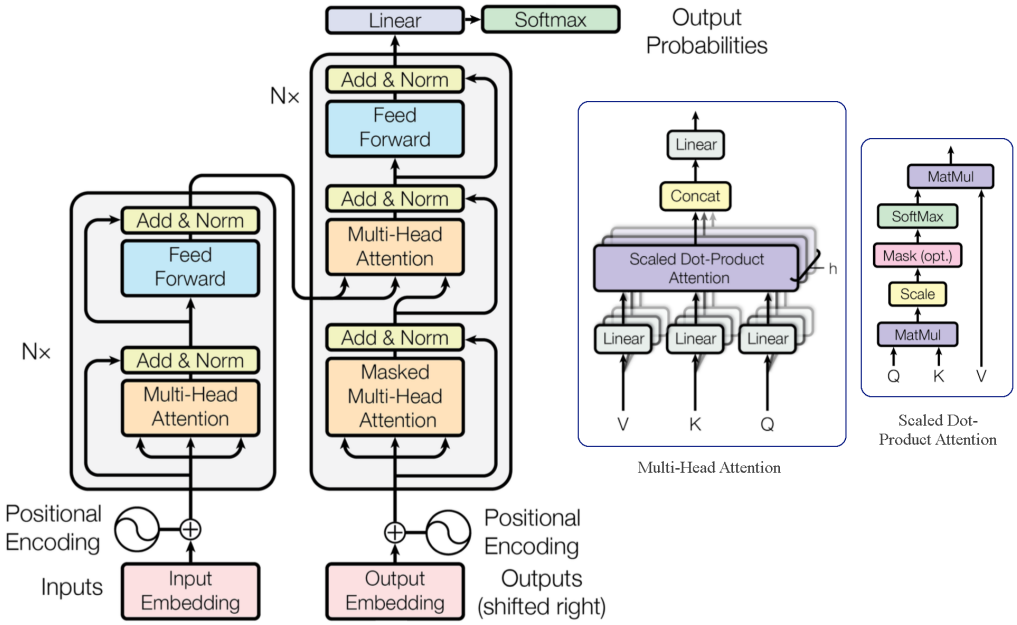

Figure 1: Transformer architecture.

In this context, the attention-mechanism can be described as mapping a query and a set of key-value pairs to an output.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

Q is a matrix containing the query (vector representation of one item in the sequence), K are all the keys (vector representations of all the items in the sequence) of dimension d_k and V are the values of dimension d_v. This means that the weights are defined by how each item of the sequence (Q) is influenced by all the other items in the sequence (K). Additionally, the SoftMax[1] function is applied produce a distribution between 0 and 1. Those weights are then applied to all the words in the sequence that are introduced in V. They are the same vectors as Q for the first attention blocks in the encoder and decoder (self attention) but different for the module that has both encoder and decoder inputs (cross attention).

[1] https://en.wikipedia.org/wiki/Softmax_function

Instead of performing a single attention function the multi-head attention linearly projects the queries, keys and values h times in parallel.

After the multi-attention heads in both the encoder and decoder, there are pointwise feed-forward layers having identical parameters for each position, which can be described as a separate, identical linear transformation of each element from the given sequence.

In recent studies it has been shown that similar approaches could lead to significant performance boosts in tasks other than NLP. The Vision Transformer (ViT), Dosovitskiy et al. [3], attains excellent results in computer vision compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train. Solutions such as the VideoGPT, Yan et al. [4], showcase how to efficiently apply Transformers for video generation tasks. In the field of time series forecasting there are multiple proposals [5] on how Transformers can be modified to compensate for their susceptibility to anomalies while simultaneously leveraging the performance advantages.

With all this as a context we will examine if and how Transformers can be used for predicting future events, going from traditional approaches with time series data (e.g. weather or stock price forecasting) to more abstract tasks such as next-frame prediction in a video where the model should learn different movement patterns and additional dependencies.

2 Transformers for time series forecasting

Time series forecasting plays an important role in daily life to help people manage resources and make decisions. Although still widely used, taditional models, such as State Space Models [6] and Autoregressive[2] (AR) models, are designed to fit each time series independently and require practitioners' expertise in manually selecting trend, seasonality, etc. To tackle those challenges, recurrent neural networks [7] have been proposed as an alternative solution. Despite the emergence of various variants, including LSTM [8] and GRU [9], it is still hard to capture long-term dependencies in TS data. Unlike the RNN-based methods, Transformers allow the model to access any part of the history regardless of distance, making it potentially more suitable for grasping the recurring patterns with long-term dependencies.

2.1 Challanges and Solutions

As described in Li et al. [10], Transformers give impressive results for their performance advantages in forecasting tasks. However, their self-attention matches queries against keys insensitive to local context, which may make the model prone to anomalies and bring underlying optimization issues. Whether an observed point is an anomaly, change point or part of the patterns is dependent on its surrounding context. The similarities between queries and keys are computed based on their point-wise values without fully taking into account local context. In previous studies convolutional self-attention[3] has been proposed to ease the issue.

Another issue which may emerge is related to the space complexity of canonical Transformer which grows quadratically with the input length L, causing memory bottleneck.

Solutions such as the *Sparse Transformer*, Child et al. [11], with complexity of $O(n\sqrt{n})$ and the *LogSparse Transformer*, Li et al. [10], with complexity reduced to $O(n(\log n)^2)$ have been introduced. These approaches make long time series modeling feasible while retaining comparable to canonical Transformer results with much less memory usage.

2.2 Experiments and Results

During the research at SRS we compared two architectures - a standard RNN utilizing LSTM cells and a simple implementation of a Transformer (See Fig. 2) for forecasting how

[2] https://en.wikipedia.org/wiki/Autoregressive_model
[3] https://github.com/mlpotter/Transformer_Time_Series

the price of the S&P500 index [4] will change. They were trained on the same amount of data[5]: the daily closing value of the index from *3rd Jan 2000* to *31st Aug 2018*.

As shown in Fig. 2 the LSTM recurrent neural network barely learns to follow a trend whereas the Transformer architecture is able to capture more detailed dependencies and use them for future forecasting. *For example: In the short-term, the index price usually goes up after the quarterly reports of the big companies in a good year.*

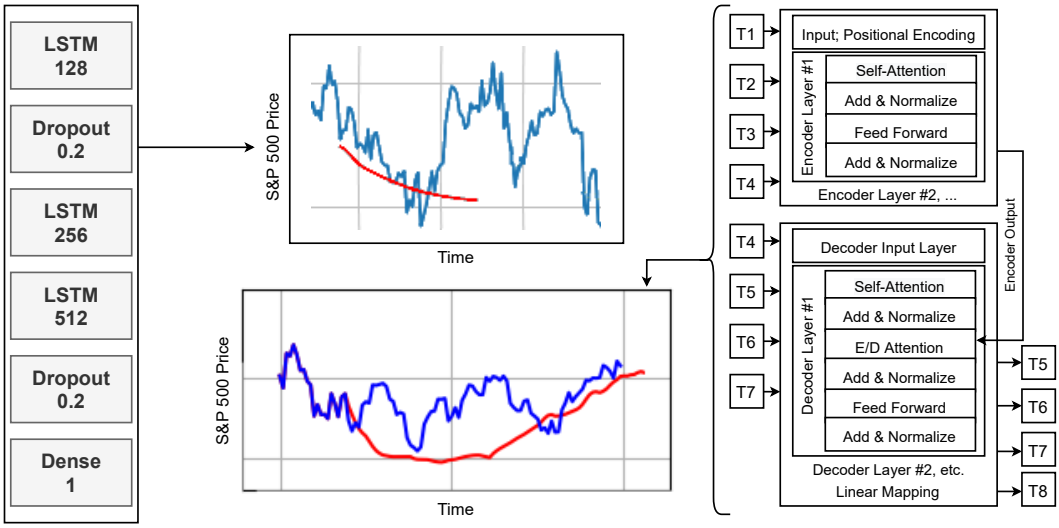

Figure 2: S&P500 price forecasting experiment. The graphics on the left and on the right describe the architectures. The top price chart shows how the RNN LSTM model forecasts pricing. The bottom chart shows the Transformer's predictions. Forecasts are colored in red and the actual prices - in blue.

3 Transformers for next-frame prediction

Another, more abstract way of thinking about future forecasting is next-frame prediction [12]. That is, predicting what happens next in the form of newly generated images, after a given amount of historical images. It refers to starting from continuous, unlabeled video frames and constructing a network that can accurately generate subsequent frames. Next-frame prediction is not only an experimental approach for video processing but a gateway to modelling machine learning architectures that can do more general assumptions and abstract reasoning.

3.1 Methods

The introduction of GPT and Image-GPT, Chen et al. [13] - a class of autoregressive Transformers that have shown tremendous success in modelling discrete data, inspired the creation of more and more Transformer-like solutions specialized for different tasks. During the SRS research we examined the VideoGPT, Yan et al. [4], a conceptually simple architecture for scaling likelihood based generative modeling to videos.

VideoGPT uses Vector Quantized Variational Autoencoder (VQ-VAE) [14] to learn downsampled latent representations of a given video. It employs 3D convolutions and axial self-attention [15] - generalization of self-attention that naturally aligns with the multiple dimensions of the tensors in both the encoding and the decoding settings. It allows for the vast majority of the context to be computed in parallel during decoding (Fig. 3).

[4] https://en.wikipedia.org/wiki/S%26P_500

[5] The data, some of the code, models and experiments described in this study are available on https://github.com/radi-cho/SRS21-public-data

A simple GPT-like architecture (Fig. 4) is then used to autoregressively model the discrete latents using position encodings.

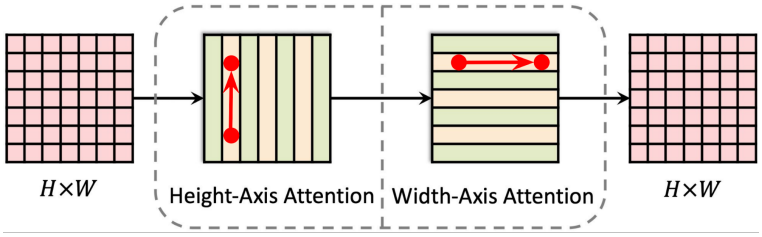

Figure 3: Axial Attention. The vertical layer provides 1-dimensional self-attention globally, propagating information within individual columns while the horizontal 1D layer allows for the capture of column-wise as well as row-wise information. That way the complexity of self-attention is reduced from quadratic (2D) to linear (1D).

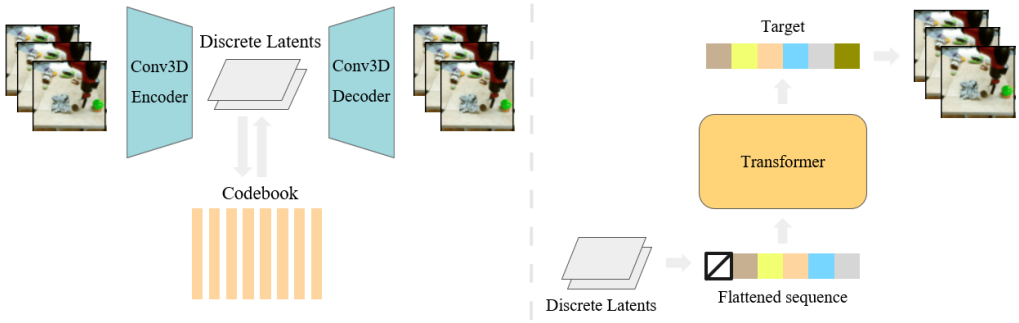

Figure 4: VideoGPT. The training pipeline is broken into two sequential stages. The first stage (Left) is similar to the original VQ-VAE training procedure. During the second stage (Right), VQ-VAE encodes video data to latent sequences as training data for the prior model.

Despite the simplicity and ease of training, the VideoGPT is able to generate samples competitive with state-of-the-art GAN [16] models. The experiments in the original paper mainly focus on creative video generation where the model samples a single frame and then tries to guess what the video is about. Although the VQ-VAE is trained fully unconditionally, conditional samples are still possible by training a conditional prior.

During SRS the model was modified and retrained to condition N frames and produce $N+M$ frames while decoding with a VQ-VAE trained with sequence length $N+M$. The conditioned frames are firstly fed into a 3D ResNet[6] [17], and then cross-attention is performed on the ResNet output during prior network training.

For the experiments we used a composition of two moving[7] MNIST, LeCun and Cortes [19], handwritten digits in a square box with dimensions of 64x64 pixels. They can bounce off the walls and go over each other.

Our main objective is to run the conditioned model and generate a few seconds (usually 4, 8, 16 or 32 frames) of video predicting the change of the position of the handwritten digits. Furthermore, if successful, the architecture can be repurposed and instead of feeding the

[6] https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035

[7] Pre-generated open-source moving MNIST datasets, such as http://www.cs.toronto.edu/~nitish/unsupervised_video/ introduced in Srivastava et al. [18] already exist. For our study moving videos were generated from the static MNIST images with a Python script in order to capture additional labelling information, e.g. which numbers are shown in the video, if they collide with one another, etc. For reference: https://gist.github.com/tencia/afb129122a64bde3bd0c.

result encodings to the VQ-VAE decoder, they can be passed to an additional classification neural network which performs labelling on the predicted future of the video. For example: to classify how likely it is the digits to collide in the next M frames.

3.2 Results

We have successfully trained a VQ-VAE with sequence lengths of 4, 8, 16 and 32 frames on the generated moving MNIST database. The final decoder reconstructions are more accurate than the pretrained models mentioned in the original paper (Fig 5).

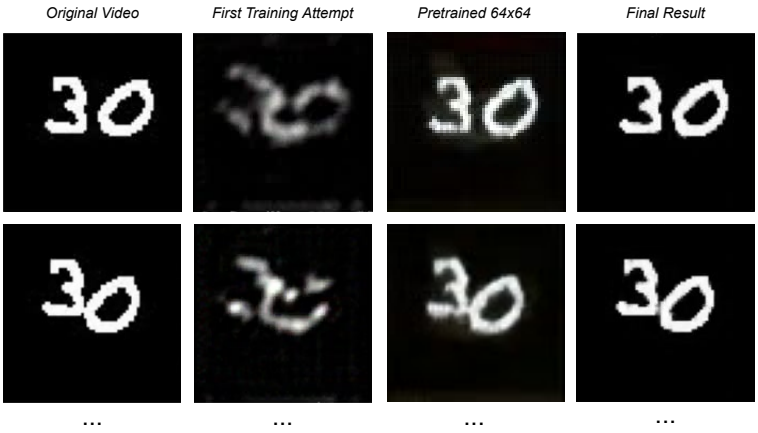

Figure 5: Representations of the original video from VQ-VAEs.

Moreover, the modified VideoGPT instance which predicts multiple frames in the future by conditioning historical data has been successful in the task of forecasting moving MNIST videos (Fig. 6). It has been tested for sequences of 4 (condition 2 frames to predict the next 2), 8 (condition 4 to predict 4) and 16 (condition 8 to predict 8).

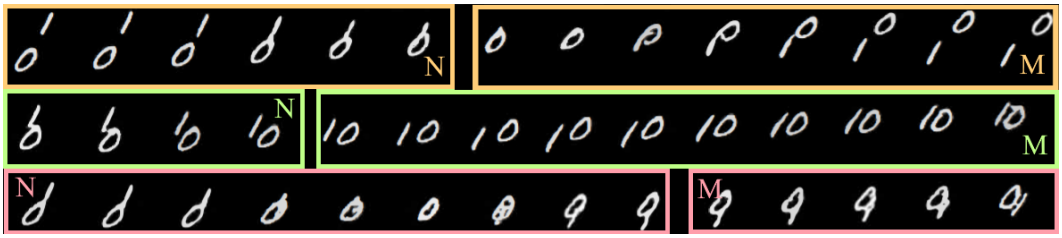

Figure 6: Next-frame prediction in a Moving MNIST video. Conditioning the first N frames and generating a video of length $N + M$ frames, where the M frames are newly generated.

Both the results from the VideoGPT and the time series experiments are a proof that with certain modifications the Transfromer can lead to high accuracy predictions and can replace traditional methods such as RNNs and CNNs in the field of future forecasting.

4 Future Work

We have shown how the generative VideoGPT [4] model can be tailored for future frame predictions as well as additional classification tasks. One of the directions for future development is to polish up those proposals and clear out some of the assumptions made in this paper to end up with a more stable and predictable architecture. In various studies (e.g. Wu et al. [20], Dehghani et al. [21], Liu et al. [22]), convolutions or recurrent structures have been used in order to manipulate, preprocess the encodings and enhance the Transformer's performance. We're interested in similar approaches in combination with the

encoder/decoder weights of a Transformer to be able to process concepts with higher levels of complexity.

If the two digits in the VideoGPT experiments will collide depends on wether they're going towards one another, but if the model is able to predict that they will bounce off a wall and then change directions to eventually collide is another level of abstraction and reasoning. Additionally, instead of videos the proposed pipeline can be used for encoded representations of other data types - time series, etc.

Another fruitful direction would be to explore the effectiveness of the Transformer as a part of Deep Reinforcement Learning[8] environments. We will be looking at already existing research (e.g. Zambaldi et al. [23], Chen et al. [24], Team et al. [25]) in order to come up with efficient solutions combining advantages from the worlds of supervised and reinforcement learning.

5 Conclusion

We have presented multiple ways of forecasting the future and how Transformer-like architectures can be adopted for such an use. We have looked at the possible solutions to problems emerging when Transformers are applied to time series data and the different levels of abstraction they can perform. RNNs and other standard solutions have been compared to newly introduced models. We have also modified the VideoGPT model to be used conditionally for next-frame prediction and proposed ways to upgrade it for future classification tasks and general reasoning. It can even be integrated as a part of Reinforcement Learning environments to enhance the behaviour of RL agents. We hope our work at SRS to be useful for future design of architectures in time series forecasting, video generation, decision making models, etc.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[2] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. *arXiv preprint arXiv:2103.03206*, 2021.

[3] 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.

[4] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

[5] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in Neural Information Processing Systems*, 32: 5243–5253, 2019.

[6] James Durbin and Siem Jan Koopman. Time series analysis by state space methods oxford university press, 2012.

[7] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

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

[8] https://en.wikipedia.org/wiki/Deep_reinforcement_learning

[9] Guizhu Shen, Qingping Tan, Haoyu Zhang, Ping Zeng, and Jianjun Xu. Deep learning with gated recurrent unit networks for financial sequence predictions. *Procedia computer science*, 131:895–903, 2018.

[10] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[11] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[12] Yufan Zhou, Haiwei Dong, and Abdulmotaleb El Saddik. Deep learning in next-frame prediction: a benchmark review. *IEEE Access*, 8:69273–69283, 2020.

[13] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12299–12310, 2021.

[14] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in neural information processing systems*, pages 14866–14876, 2019.

[15] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015.

[19] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

[20] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.

[21] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

[22] Zhouyong Liu, Shun Luo, Wubin Li, Jingben Lu, Yufan Wu, Chunguo Li, and Luxi Yang. Convtransformer: A convolutional transformer network for video frame synthesis. *arXiv preprint arXiv:2011.10185*, 2020.

[23] Vinicius Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, Murray Shanahan, Victoria Langston, Razvan Pascanu, Matthew Botvinick, Oriol Vinyals, and Peter Battaglia. Deep reinforcement learning with relational inductive biases. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HkxaFoC9KQ.

[24] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*, 2021.

[25] Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, et al. Open-ended learning leads to generally capable agents. *arXiv preprint arXiv:2107.12808*, 2021.

[26] Jorge Pérez, Pablo Barceló, and Javier Marinkovic. Attention is turing-complete. *Journal of Machine Learning Research*, 22(75):1–35, 2021.

[27] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017.

[28] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*, 2021.