S-KANformer: Enhancing Transformers for Symbolic Calculations in High Energy Physics

Ritesh Bhalerao
VESIT
Mumbai, India
ritesh.work.personal@gmail.com

Eric Reinhardt
The University of Alabama
Tuscaloosa, AL
eareinhardt@crimson.ua.edu

Victor Baules
The University of Alabama
Tuscaloosa, AL
vbaules@crimson.ua.edu

Sergei Gleyzer
The University of Alabama
Tuscaloosa, AL
sgleyzer@ua.edu

Nobuchika Okada
The University of Alabama
Tuscaloosa, AL
okadan@ua.edu

Abstract

Squared amplitudes of particle collision interactions can be used to theoretically predict cross-sections in order to verify results of particle physics experiments. The calculation of amplitudes for a particle process is trivial, however, mapping those to a simplified squared amplitude expression can be very computationally expensive. Previous work has demonstrated the ability to map amplitudes to squared amplitudes substantially faster and with high accuracy using vanilla transformer models for simple processes. In this paper, we further explore the application of S-KANformer (transformer models infused with SineKAN layers). We present empirical evidence demonstrating that our model significantly outperforms the vanilla transformer in most tasks and shows greater robustness to varying factors such as batch size, dataset size, and sequence length. We also discuss some limitations and potential future directions of this work. Although more comprehensive studies need to be undertaken, this work shows promising directions for applications of S-KANformer, especially in domains involving symbolic calculations.

1 Introduction

In the realm of particle physics, particle interactions are commonly modelled theoretically using methods which aim to predict likelihoods of certain outcomes of those interactions. These likelihoods are generally modelled as cross-sections, for example, as likelihood of an interaction resulting in a certain type of particle passing through a surface area of a detector. Two key steps in determining these cross-sections are determining the amplitude of a wave function associated with the interaction and determining the squared amplitude from the amplitude.

Calculations of these squared amplitudes by hand are very error prone and inefficient. Also, the automated software available for such calculations (`Feyncalc`, `CompHEP` or `MARTY`) Shtabovenko (2024); Boos et al. (2004); Uhlrich et al. (2021) are time consuming.The proposed solution is to reframe the problem into a seq-to-seq learning task mainly using transformer-based models Vaswani (2017). The goal is to devise a system capable of generating accurate symbolic representation of the squared amplitudes conditional on the given input i.e. the symbolic amplitude equations or Feynman-diagrams. Inference time of these models will be orders of time lesser than existing calculation methods. In earlier works, this problem was tackled using vanilla transformers Alnuqaydan et al. (2023). In this paper, we extend the work further with the help of S-KANformer, our novel transformer model infused with Kolmogorov-Arnold Networks.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

2 Model description

2.1 SineKAN

Multi-layer perceptrons (MLPs) are the most crucial component of current neural networks. They are based on the principle of the Universal Approximation Theorem Hornik et al. (1989). In theory, a sufficiently large neural network can approximate any arbitrary function to a desired degree of accuracy. Recent works suggest a promising alternative to MLPs, namely Kolmogorov-Arnold Networks (KANs) Liu et al. (2024). Unlike traditional MLPs, which have fixed activation functions on the nodes (neurons), KANs place learnable activation functions on the edges of the computation graph. KANs are inspired by the Kolmogorov-Arnold Representation Theorem. The univariate functions used in the original KAN implementation were B-splines; however, they pose major computational challenges with increasing dimensionality. Some recent works suggest the use of alternative univariate functions other than B-splines Aghaei (2024a,b); SS (2024); Xu et al. (2024); Bozorgasl and Chen (2024); Ta (2024), some of which are more efficient. For S-KANformer, we explicitly make use of SineKAN Reinhardt and Gleyzer (2024), which is orders of magnitude faster than the original KAN implementation while maintaining similar performance.

Unlike the original B-spline KAN, SineKAN is based on sine functions. SineKAN architecture has shown promising results in satisfying universal approximation theorem and has proven to be numerically stable across multiple layers and different grid sizes. Mathematically, each layer can be expressed as:

$$y_i = \sum_j \sum_k (\sin(x_j \cdot \omega_k + \phi_{jk}) \cdot A_{ijk}) + b_i \tag{1}$$

Where y_i are the layer output features, x_j are the layer input features, ϕ_{jk} is a phase shift over the grid and input dimensions, ω_k is a grid frequency, w_{ijk} are the amplitude weights, and b_i is a bias term.

2.2 S-KANformer

The S-KANformer architecture replaces the Feed Forward Network (FFN) at the end of the last decoder block of the vanilla Transformer. Figure 1 describes the mentioned architecture, where the left block represents the encoder and the right block represents the final decoder block. Except for the final decoder block, all decoder blocks have the usual FFN after the cross multi-head attention, along with a residual connection, which is absent for KAN layers.

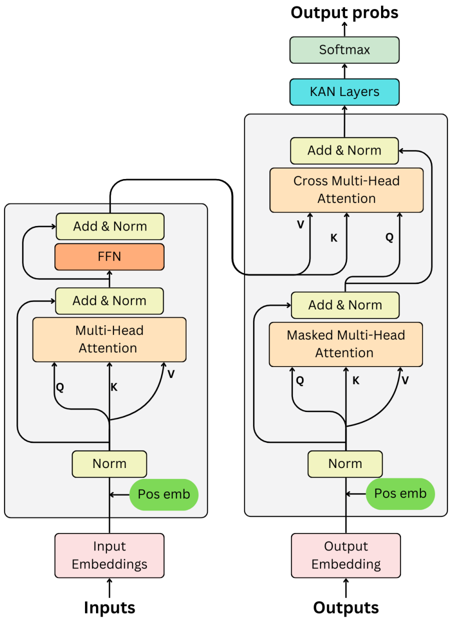

Figure 1: S-KANformer block diagram

2.3 Dataset

Data has been generated using the symbolic computation program MARTY Uhlrich et al. (2021) to obtain expressions for possible interactions in quantum electrodynamics (QED) mediated by photons, quantum chromodynamics (QCD) mediated by gluons, and electroweak theory (EW) mediated by $W^{+/-}$ and Z bosons. These processes we will generally describe as "A-to-B" where A is the number of incoming particles and B is the number of outgoing particles. We will focus in this work on tree-level processes in which there are no closed loops of propagators in the associated interaction Feynman diagram. The processes are restricted to 2-to-2 and 2-to-3 tree-level processes. For each process and numbers of incoming and outgoing particles, we generate an inclusive set of amplitudes and squared amplitudes. The sequences are expressed using the matrix element formalism, which employs Dirac/gamma matrices Dirac (1928) and spinor representations. Lorentz and spinor indices are used to indicate how the matrices and spinors should be contracted Minkowski (1908, 1910).

Generated sequences are tokenized using a custom tokenizer which separates the mathematically distinct components of the amplitude and squared amplitude expressions such as operations, indices, scalars, and matrices/tensors. After tokenization, we verify that there are no duplicate amplitude expressions arising from similar interaction diagrams present in the data which could constitute a data leakage. Full details of the tokenization can be found in our github repository A.1. For the current work, models were trained for only amplitude to squared amplitude calculation in QCD (2-to-2), QED (2-to-2 and 2-to-3) and EW (2-to-3) with `max_seq_lens`[1] of 544, 300, 1202 and 302 respectively.

3 Results & Discussions

After extensive experimentation with both Transformer and S-KANformer models across a variety of datasets, it was observed that S-KANformer generally outperformed or matched the performance of Transformers, but never performed worse. For all experiments, a **3-layer** encoder-decoder architecture was employed, with Feed Forward Network (FFN) dimensions set to **4096** for S-KANformer and **8192** for Transformers. In the S-KANformer models, only a single SineKAN layer was utilized, with a dimension of size **8192**. All models were trained to near convergence with a learning rate that decayed linearly with each epoch. No warm-up phase was used; instead, the norm of the gradients was clipped to a unit norm in most cases to improve convergence and enhance stability during the initial phases of training. Models are evaluated on basis of the sequence accuracy of the generated sequences with the original sequences, only exact match is taken into consideration. Testing is done on an unseen test set sized around 5 - 10% of the training data for that particular process. Detailed reports for the all the experiments can be found in our repository A.1. Following tables provide details about the training results according to the task.

Note: All models have an embedding size of **512**, **8** attention heads, and **~60M** parameters for Transformer, and **~65M** parameters for S-KANformer. The batch size is written in the form `batch_size_per_GPU X num_of_GPUs`.

Table 1: Comparison of Models Across Processes

Process Name	Training Samples	Model Name	Batch Size	Seq Acc(%)
QCD 2-2	41K	Transformer	64 x 2	89.40
QCD 2-2	41K	S-KANformer	64 x 2	91.13
QED 2-2	42K	Transformer	64 x 2	99.60
QED 2-2	42K	S-KANformer	64 x 2	99.60
EW 2-3	246K	Transformer	64 x 2	47.08
EW 2-3	246K	S-KANformer	64 x 2	66.57
QED 2-3	192K	Transformer	48 x 2	75.35
QED 2-3	192K	S-KANformer	48 x 2	95.50

[1] Maximum Sequence length is the maximum among the source and target sequence length for the particular dataset.

4 Conclusions

From the above results, it is evident that S-KANformer performs better than the Transformer in almost all cases by a significant margin. The design choice of replacing the FFN layer in only the final decoder block with SineKAN layers was primarily driven by the need to limit the increasing computational cost associated with swapping additional FFN layers with SineKAN layers. This approach draws inspiration from design paradigms like CNNs, where convolutional layers act as feature extractors before an MLP processes features for specific tasks, or NLP models, where only the final layers of pretrained transformers are adapted for downstream tasks. Additionally, it was revealed through experiments that additional swaps of FFN layers in decoder blocks as well as encoder blocks did not improve performance.

It is evident that S-KANformers are a promising avenue to consider, especially in symbolic tasks. The SineKAN layer used in these models, although faster than the original B-spline implementation, is still slower than traditional MLPs. This added computation has been mitigated to some extent by placing the layers only at the head of the transformer, but the concern persists. A S-KANformer with one SineKAN layer of size 8192 is, on average, 1.2 times slower than a transformer with almost the same number of parameters as the S-KANformer. Possible future work can include optimizing this architecture to reduce computations and time with increasing dimensionality. Additionally, more comprehensive studies are required to establish the efficacy of S-KANformers over Transformers.

5 Acknowledgement

This work was supported by the U.S. Department of Energy (DOE) under Award No. DE-SC0012447 (E.R., V.B., N.O., and S.G.). R.B. was a participant in the ML4SCI 2024 Google Summer of Code Program.

References

Alireza Afzal Aghaei. fkan: Fractional kolmogorov-arnold networks with trainable jacobi basis functions. *arXiv preprint arXiv:2406.07456*, 2024a.

Alireza Afzal Aghaei. rkan: Rational kolmogorov-arnold networks. *arXiv preprint arXiv:2406.14495*, 2024b.

Abdulhakim Alnuqaydan, Sergei Gleyzer, and Harrison Prosper. Symba: symbolic computation of squared amplitudes in high energy physics with machine learning. *Machine Learning: Science and Technology*, 4(1):015007, 2023.

Edward Boos, V Bunichev, M Dubinin, L Dudko, V Edneral, V Ilyin, A Kryukov, V Savrin, A Semenov, and A Sherstnev. Comphep 4.4—automatic computations from lagrangians to events. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 534(1-2):250–259, 2004.

Zavareh Bozorgasl and Hao Chen. Wav-kan: Wavelet kolmogorov-arnold networks. *arXiv preprint arXiv:2405.12832*, 2024.

P. A. M. Dirac. The quantum theory of the electron. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 117(778):610–624, 1928. doi: 10.1098/rspa.1928.0023.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.

Hermann Minkowski. Raum und zeit. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 18: 75–88, 1908. URL `https://archive.org/details/raumundzeit`. Lecture presented at the 80th Assembly of German Natural Scientists and Physicians, Cologne, September 21, 1908.

Hermann Minkowski. Die grundgleichungen für die elektromagnetischen vorgänge in bewegten körpern. *Mathematische Annalen*, 68:472–525, 1910. doi: 10.1007/BF01455871. URL `https://link.springer.com/article/10.1007/BF01455871`.

Eric AF Reinhardt and Sergei Gleyzer. Sinekan: Kolmogorov-arnold networks using sinusoidal activation functions. *arXiv preprint arXiv:2407.04149*, 2024.

Vladyslav Shtabovenko. New multiloop capabilities of feyncalc 10. *arXiv preprint arXiv:2407.01447*, 2024.

Sidharth SS. Chebyshev polynomial-based kolmogorov-arnold networks: An efficient architecture for nonlinear function approximation. *arXiv preprint arXiv:2405.07200*, 2024.

Hoang-Thang Ta. Bsrbf-kan: A combination of b-splines and radial basic functions in kolmogorov-arnold networks. *arXiv preprint arXiv:2406.11173*, 2024.

Grégoire Uhlrich, Farvah Mahmoudi, and Alexandre Arbey. –modern artificial theoretical physicist a c++ framework automating theoretical calculations beyond the standard model. *Computer Physics Communications*, 264:107928, 2021.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Jinfeng Xu, Zheyu Chen, Jinze Li, Shuo Yang, Wei Wang, Xiping Hu, and Edith C-H Ngai. Fourierkan-gcf: Fourier kolmogorov-arnold network–an effective and efficient feature transformation for graph collaborative filtering. *arXiv preprint arXiv:2406.01034*, 2024.

A Appendix / supplemental material

A.1 Repository

The code for S-KANformer, along with additional details regarding the experiments, is available at the following GitHub repository: github.com/Riteshbhalerao11/GSOC-24.