

PAPER • OPEN ACCESS

SYMBA: symbolic computation of squared amplitudes in high energy physics with machine learning

To cite this article: Abdulhakim Alnuqaydan *et al* 2023 *Mach. Learn.: Sci. Technol.* **4** 015007

View the [article online](#) for updates and enhancements.

You may also like

- [Theoretical characterization of uncertainty in high-dimensional linear classification](#)
Lucas Clarté, Bruno Loureiro, Florent Krzakala et al.
- [Deep learning symmetries and their Lie groups, algebras, and subalgebras from first principles](#)
Roy T Forestano, Konstantin T Matchev, Katia Matcheva et al.
- [A meshfree moving least squares-Tchebychev shape function approach for free vibration analysis of laminated composite arbitrary quadrilateral plates with hole](#)
Songhun Kwak, Kwanghun Kim, Kwangil An et al.



PAPER

OPEN ACCESS

RECEIVED

21 September 2022

REVISED

8 December 2022

ACCEPTED FOR PUBLICATION

12 January 2023

PUBLISHED

27 January 2023

Original Content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



SYMBA: symbolic computation of squared amplitudes in high energy physics with machine learning

Abdulhakim Alnuqaydan^{1,2,*} , Sergei Gleyzer³ and Harrison Prosper⁴¹ Department of Physics and Astronomy, University of Kentucky, Lexington, KY, United States of America² Department of Physics, College of Science, Qassim University, Qassim, Saudi Arabia³ Department of Physics and Astronomy, The University of Alabama, Tuscaloosa, AL, United States of America⁴ Department of Physics, Florida State University, Tallahassee, FL, United States of America

* Author to whom any correspondence should be addressed.

E-mail: aal700@uky.edu**Keywords:** physics, high energy physics, machine learning

Abstract

The cross section is one of the most important physical quantities in high-energy physics and the most time consuming to compute. While machine learning has proven to be highly successful in numerical calculations in high-energy physics, analytical calculations using machine learning are still in their infancy. In this work, we use a sequence-to-sequence model, specifically, a transformer, to compute a key element of the cross section calculation, namely, the squared amplitude of an interaction. We show that a transformer model is able to predict correctly 97.6% and 99% of squared amplitudes of quantum chromodynamics and quantum electrodynamics processes, respectively, at a speed that is up to orders of magnitude faster than current symbolic computation frameworks. We discuss the performance of the current model, its limitations and possible future directions for this work.

1. Introduction

Most machine learning applications in high-energy physics focus on numerical data (see, for example, [1–4]), while only a few studies have investigated the application of machine learning to symbolic data [5]. Working with symbolic data is a challenging task both for humans and machines. Hand manipulation of large symbolic expressions is error prone, hence the routine use of domain-specific symbolic manipulation software tools [6, 7]. The question we address in this paper is whether it is feasible to encapsulate accurately the highly non-trivial symbolic manipulations encoded in these tools in a machine learning model? Our motivation is to amortize the time required to compute symbolic expressions by accepting an upfront cost in creating them, so that one can reap the benefits later of a symbolic calculation that is much faster than the software tools used to generate the symbolic data.

We use a *sequence-to-sequence* (*seq2seq*) model, specifically, a *transformer* [8], to compute symbolically the square of the particle interaction amplitude, a key element of a cross section calculation. To the best of our knowledge, this is a first application of such models to this task.

Cross sections are important quantities in high-energy physics because they link the real world of experimental physics with the abstract world of theoretical models. The calculation of a cross section can be an exceedingly complicated procedure, requiring many mathematical operations, including Lorentz index contractions, color factor calculation, matrix multiplications, Dirac algebra, traces, and integrals. Moreover, the complexity of these operations increases dramatically with the number of final state particles. It is a testament to the high mathematical sophistication and the depth of domain knowledge of the developers of these symbolic manipulation tools that such analytical calculations have been automated (see, for example, *FeynCalc* [6], *CompHEP* [9] or *MARTY* [7]). The squaring of amplitudes can result in very long expressions and correspondingly long computation times, an important challenge to overcome in practical applications. In this work, we address this challenge using symbolic machine learning and demonstrate a proof-of-concept for the symbolic calculation of the key element of cross section calculations.

The paper is organized as follows. In section 2 we note related work. In section 3 we provide the context for the work and define the relevant quantities involved in high-energy physics cross section calculations. We also introduce the seq2seq models, followed by the pertinent details of the datasets, transformer model and its training parameters in section 4. We present our results in section 5, followed by a discussion in section 6 and our conclusions in section 7

2. Related work

Seq2seq models are the basis of natural language processing (NLP) systems that are frequently used to translate from one language to another [10], summarize text [11] and even map images to textual summaries of them [12]. Transformer-based models [8] are now considered the state-of-the-art in NLP applications. The most well-known NLP models are: BERT (Bidirectional Encoder Representations from Transformers) [13] and GPT (Generative Pre-trained Transformer) [14]. These models have been used in medicine [15, 16] and DNA analysis [17].

Recently, seq2seq models have been used to perform symbolic calculations in calculus and to solve ordinary differential equations symbolically, achieving excellent results [18]. This model was able to predict correctly the symbolic solutions of integrals and found solutions to some problems that could not be completed using the the standard symbolic mathematics systems within a time limit orders of magnitude longer than the model execution time. Sequence-to-sequence transformer models have also been used to infer the recurrence relation of underlying sequences of numbers [19]. The authors used a transformer model to successfully find symbolic recurrence relations given the first terms of the sequence. Transformer models have also been used for symbolic regression [20].

In physics, symbolic machine learning, such as symbolic regression, have been applied to classical mechanics problems [21–23]. In [5], symbolic regression was used to extract optimal observables as an analytic function of phase space applied to Standard Model Effective Field Theory (SMEFT) Wilson coefficients using LHC simulated data. In our work, we take a further step and show that symbolic machine learning relying on transformers can be applied to complex, realistic, calculations in high-energy physics such as the calculation of squared amplitudes.

3. Background

We begin with an overview of the context of the work reported in this paper. More details can be found in [24], however, for the benefit of readers who are not high-energy physicists, we provide a concise description.

The Standard Model, one of the great intellectual achievements of the 20th century, describes all known elementary particles and their interactions through three of the four known fundamental forces, namely, the weak, electromagnetic, and strong forces (see, for example, [24]). The Standard Model is a Quantum Field Theory (QFT) specified in a mathematical expression called a Lagrangian. In QFT, the elementary particles are described in terms of quantum fields in space-time, where each type of particle is associated with a different field. The interactions among these particles are governed by fields, which are sometimes referred to as force carriers, whose details are precisely determined by the symmetries imposed on the Lagrangian.

There is a well-defined procedure to extract from the Lagrangian all the possible particle interactions of interest, each associated with a mathematical quantity called an amplitude. These amplitudes can be represented by Feynman diagrams, as shown in figure 1.

Calculating the cross section, for example of the process depicted in figure 1 and represented symbolically in figure 2, requires computing the squared amplitude and averaging and summing over the internal degrees of freedom of the particles.

As noted above, these calculations are typically performed with domain-specific computer algebra frameworks, such as FeynCalc [6], CompHEP [9] and MARTY [7]. Figure 2 shows one of the ‘shortest’ $2 \rightarrow 3$ quantum electrodynamic (QED) expressions after simplification, where in this process two incoming electrons $e(p_1)$ and $e(p_2)$ scatter into two electrons $e(p_3)$ and $e(p_4)$ and a photon $\gamma(p_5)$. Typical expressions can have hundreds of terms and the computational time can become a major challenge, especially if higher-order amplitudes are included to render predictions more precise.

The key insight in all the current uses of machine learning for symbolic applications is that many tasks can be viewed as a language translation problem. For example, a system that maps images to textual summaries of them can be viewed as translating from the language of images to a natural language. Likewise, algebraic manipulation such as the mapping of amplitudes to their squared form can be conceptualized as a language translation task. Since this task maps one sequence of symbols to another it is natural to consider seq2seq models.

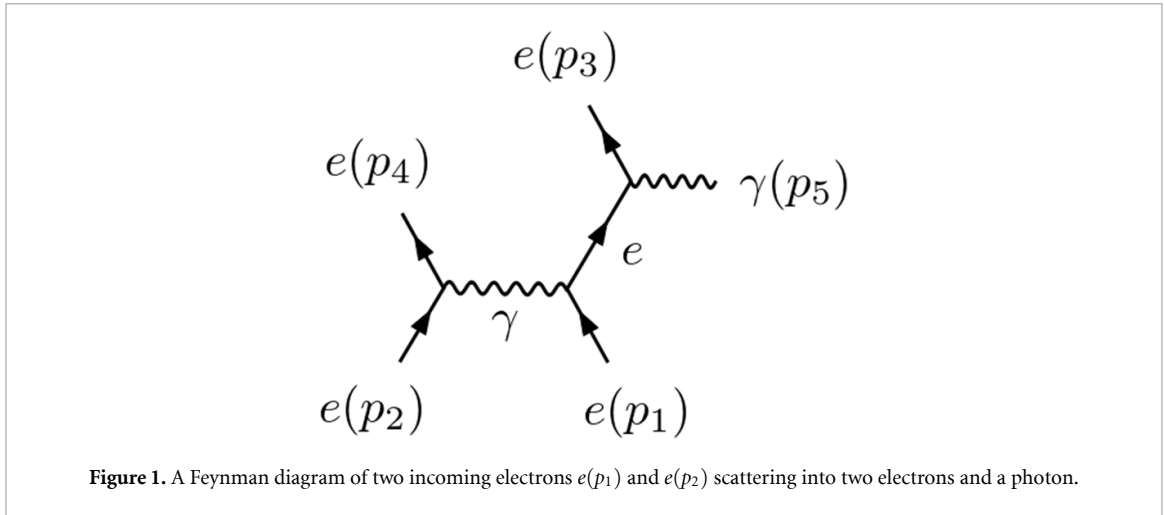


Figure 1. A Feynman diagram of two incoming electrons $e(p_1)$ and $e(p_2)$ scattering into two electrons and a photon.

- **The amplitude** ($e e \rightarrow e e \gamma$):

$$i\mathcal{M} = \frac{\frac{1}{2} i e^3 (p_{3\rho} \gamma_\rho^\epsilon \gamma_{\rho\eta} A_j^{\rho*}(p_5) e_{i\eta}^*(p_4) e_{i\epsilon}^*(p_3) e_{k\delta}(p_2) e_{i\delta}(p_1) - \frac{1}{2} p_{5\sigma} \gamma_{\rho\epsilon} \gamma_\rho^\epsilon \gamma_{\rho\eta} \gamma_\sigma^\epsilon A_j^{\rho*}(p_5) e_{i\eta}^*(p_4) e_{i\epsilon}^*(p_3) e_{k\delta}(p_2) e_{i\delta}(p_1))}{(m_e^2 - \vec{p}_2 \cdot \vec{p}_4) * \vec{p}_3 \cdot \vec{p}_5}}$$

- **The squared amplitude** ($e e \rightarrow e e \gamma$):

$$|\mathcal{M}|^2 = -\frac{e^6}{((\vec{p}_3 \cdot \vec{p}_5)^2 * (m_e^2 - \vec{p}_2 \cdot \vec{p}_4)^2)} (2m_e^6 + m_e^4 * (-\vec{p}_1 \cdot \vec{p}_3 - \vec{p}_1 \cdot \vec{p}_5 - \vec{p}_2 \cdot \vec{p}_4 + 2\vec{p}_3 \cdot \vec{p}_5) + m_e^2 * (\vec{p}_1 \cdot \vec{p}_2 * \vec{p}_3 \cdot \vec{p}_4 + \vec{p}_1 \cdot \vec{p}_2 * \vec{p}_4 \cdot \vec{p}_5 + \vec{p}_1 \cdot \vec{p}_4 * \vec{p}_2 \cdot \vec{p}_3 + \vec{p}_1 \cdot \vec{p}_4 * \vec{p}_2 \cdot \vec{p}_5 + \vec{p}_1 \cdot \vec{p}_5 * \vec{p}_3 \cdot \vec{p}_4 - \vec{p}_2 \cdot \vec{p}_4 * \vec{p}_3 \cdot \vec{p}_5) - \vec{p}_1 \cdot \vec{p}_2 * \vec{p}_3 \cdot \vec{p}_5 * \vec{p}_4 \cdot \vec{p}_5 - \vec{p}_1 \cdot \vec{p}_4 * \vec{p}_2 \cdot \vec{p}_5 * \vec{p}_3 \cdot \vec{p}_5)$$

Figure 2. Amplitude and squared amplitude of the $ee \rightarrow ee\gamma$ scattering process.

4. Model description and training

In recent years, seq2seq transformer models have become one of the most powerful models for a variety of seq2seq applications [8]. Transformers use a mechanism called self-attention, whereby the model identifies and makes use of long-range dependencies among the symbols (or tokens) of the sequences. The architecture of the model consists of two parts: an *encoder* and a *decoder*. The encoder first maps the input sequence to a vector in a high-dimensional vector space in a process called *embedding*, and encodes the position of each token in the sequence, an operation referred to as *positional encoding*. Next, a measure of the degree to which a given token is related to other tokens, in a mechanism called multi-head attention, is calculated. During the training of the model, the decoder takes the output vector from the encoder, which encodes information about the input sequence to the encoder, together with the encoded target sequence one token at a time and outputs a sequence also one token at a time.

4.1. Datasets, model and training

We use the symbolic computation program MARTY [7] to generate expressions for possible interactions in QEDs, the theory of the electromagnetic force, and quantum chromodynamics (QCD), the theory of the strong forces. For our proof-of-concept, we restrict the scope 2-to-2 and 2-to-3 particle tree-level processes. All interactions involving off-shell and on-shell particles, anti-particles gauge bosons and triplet and anti-triplet color representations for QCD are included. Since it is possible for different amplitudes to yield the same squared expressions, we include such amplitudes in our dataset. Many of the QED and QCD interactions are non-trivial and lead to complicated expressions. All expressions are simplified with the Python symbolic mathematics module SymPy [25], factorized by particle masses, and organized into a standard format (first the overall factors, then the terms of the numerator, and third the denominator). For practical reasons, we exclude expressions longer than 264 tokens after simplification which excludes 5% and 26% of all QED and QCD expressions. This procedure yields 251 000 QED and 140 000 QCD expression pairs figure 3. The data are split into three sets: training, validation and test, 70%, 15% and 15%, respectively, and we choose a random sample of 500 expression pairs (from the test set) to evaluate the performance of the trained model.

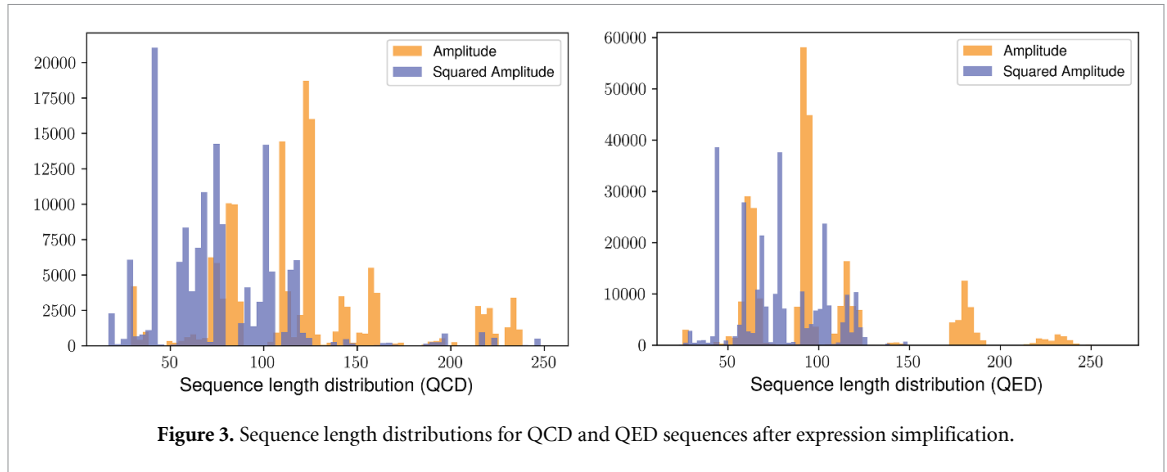


Figure 3. Sequence length distributions for QCD and QED sequences after expression simplification.

The `Tensorflow` [26] package is used to map symbols to integers, specifically with the `TextVectorization` class, which performs the tokenization, that is, the assignment of an integer to each symbol and the padding of sequences to make them of equal length. Each sequence is then converted to a vector built from these integers. The amplitudes are tokenized by operator (tensor) and its indices, while for squared amplitudes we tokenize them by each mass, product of momenta and numerical factor (for example, $4 * m_e^2 * p_1 \cdot p_2$ is three tokens) as there are a finite number of terms consistent with the physical dimension (in powers of mass) and conservation laws.

The transformer model is implemented using `Tensorflow 2.8` [26] along with `Keras 2.8` [27] without structural modifications. The model has 6 layers and 8 attention-heads, with 512 embedding dimensions and 8192–16384 latent dimensions. We use *sparse categorical crossentropy* as the loss function, the Adam optimizer [28] with a learning rate of 10^{-4} and a batch size of 64. The training was performed for 50–100 epochs on two CASCADE-NVIDIA V100 GPUs which took about 12–24 h.

5. Results

The accuracy of the predicted symbolic expressions is assessed by taking a random sample of 500 amplitudes (from the test set) that have not been seen by the transformer model and predicting their squared amplitudes. We use three distinct metrics to assess the model accuracy.

- (a) **Sequence Accuracy:** the percentage of predicted symbolic expressions that *identically* match the correct expression.
- (b) **Token Score:** a measure of the number of tokens (symbols) predicted correctly in the correct location in the sequence.

$$Token\ Score = \frac{n_c - n_{ex}}{n_{act}} \quad \%,$$

where n_c is the number of tokens predicted correctly, n_{ex} is the number of extra tokens that the model predicts (if any), and n_{act} is the number of tokens in the correct sequence.

- (c) **Numerical Error:** the relative difference between the values of the predicted and the correct expressions when using random numbers between $\{0, 100\}$ for the momenta in the squared amplitude expression.

$$Numerical\ Error = \frac{x_{act} - x_{pred}}{x_{act}},$$

where x_{act} is the numerical value of the actual (correct) squared amplitude, x_{pred} is the predicted numerical squared amplitude. Repeating this process 100 times and varying the random numbers each time, we compute the root-mean-square-error (RMSE) whose distributions are shown in figure 4.

Table 1 summarizes the performance of our models. For QED, the model trained on QED expressions, correctly and identically predicts the squared amplitude for 493 out of 500 (98.6%) amplitudes. The remaining 7 squared amplitudes differ by one or more tokens. The overall token scores is 99.7%, i.e. the model correctly predicts at least 99.7% of the tokens on average. For QCD, the model trained on QCD expressions has a sequence accuracy of 97.4% and the token score is 98.9%. The average numerical error is 1.3×10^{-3} for QED and 8.8×10^{-3} for QCD (excluding two anomalous results with RMSE > 10 in QCD

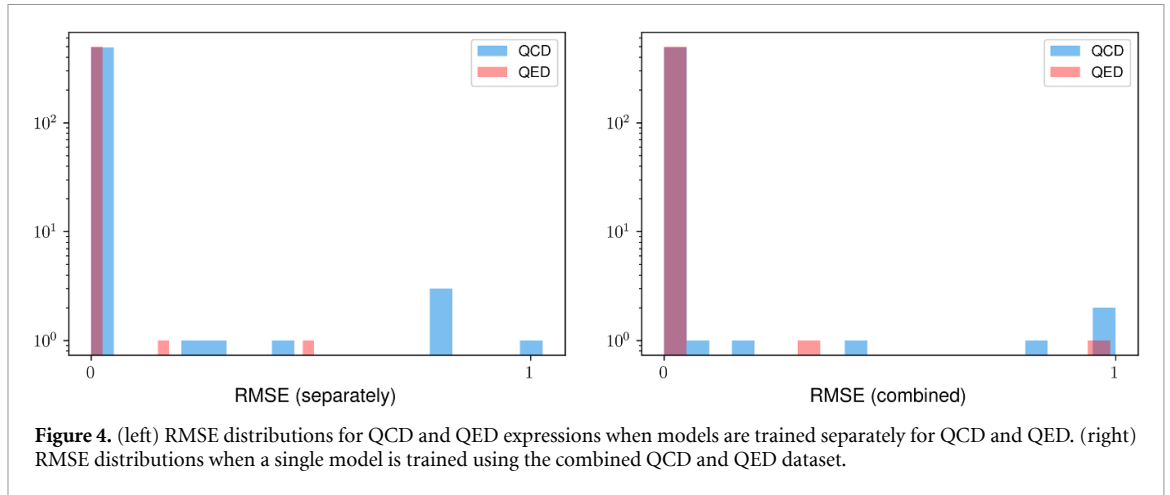


Table 1. Amplitude-squared amplitude Model results.

	Training Size	Sequence Acc.	Token Score	RMSE
QED (sequence)	251K	98.6%	99.7%	1.3×10^{-3}
QCD (sequence)	140K	97.4%	98.9%	8.8×10^{-3}
(QED + QCD) on QED	391K	99.0%	99.4%	2.5×10^{-3}
(QED + QCD) on QCD	391K	97.6%	98.8%	6.8×10^{-3}
QED (diagram)	258K	99.0%	99.7%	9.3×10^{-4}
QCD (diagram)	142K	73.4%	82.0%	0.3

only). The model trained on the combined QED and QCD dataset achieves a sequence accuracy of 99.0% for QED with a token score of 99.4% and an RMSE of 2.5×10^{-3} . The combined model gives a sequence accuracy of 97.6% and token score of 98.8% and RMSE of 6.8×10^{-3} for QCD.

6. Discussion

It has been demonstrated that even for a mathematical problem as complicated as squaring an amplitude, averaging and summing of over the internal degrees of freedom of particles, and manipulating the result into a meaningful form, it is possible to encapsulate that domain knowledge in a transformer model. Our model can ‘do the algebra’ with a sequence accuracy between 97.6% and 99.0%. The time of inference, on average, varies from one second to two seconds. The prediction time for QED is slightly longer (or similar) to the time taken by MARTY for the same calculation. For QCD, we find that our model is up to 2 orders of magnitudes faster than MARTY for some amplitudes, (especially those with anti-triplet color representations).

We observe, as it shown in tables 2 and 3, that the accuracy of the mapping is primarily affected by two factors: the amount of training data and the sequence length. The accuracy of the QED results is slightly higher than for QCD, which reflects the fact that the QED training sample is larger than the QCD sample and the QED sequences are shorter on average than those for QCD. This is especially true of the amplitude expressions, which are typically more complicated for QCD than for QED.

This indicates that our results are still data limited. Therefore, we should anticipate performance improvements if we are able to create and use larger datasets. There are several ways to achieve that. For example, additional tree-level processes can be included, such as 2-to-4 or 2-to-5, or by going beyond tree-level processes. Additionally, we can add predictions from theories of beyond the Standard Model physics as long as the mathematical structure of these theories is similar to that of the Standard Model.

The complexity of the transformer model increases quadratically with sequence length, which makes the training highly resource-intensive and affects the accuracy. This challenge can be addressed with variants of the basic transformer model that exhibit better scaling with sequence length [29, 30]. The length of the sequence can be further tuned by adjusting the tokenization process. Balancing the sequence length and number of tokens is an important hyperparameter optimization step in this class of models that we leave to future work.

Another issue that needs to be addressed is how to identify when a predicted squared amplitude is anomalous when there is no access to the correct answer. We have seen that a single token error involving a quantity with a large value, such as the top quark mass, impacts the numerical result severely. It would be

Table 2. Model performance on different sizes of QCD and QED dataset.

QCD	Train on:	$\frac{1}{10}$ data	$\frac{1}{5}$ data	$\frac{1}{3}$ data	$\frac{1}{2}$ data	full data (140 K)
	Sequence Accuracy:	52.0%	82.0%	91.0%	94.0%	97.4%
QED	Train on:	$\frac{1}{10}$ data	$\frac{1}{5}$ data	$\frac{1}{3}$ data	$\frac{1}{2}$ data	full data (251 K)
	Sequence Accuracy:	76.4%	89.6%	95.8%	98.0%	98.6%

Table 3. Model performance on different sequence lengths of QCD and QED dataset.

QCD	Maximum sequence length:	170 tokens	195 tokens	full length (256 tokens)
	Sample size:	123 K	126 K	140 K
	Sequence Accuracy:	99.6%	98.6%	97.4%
QED	Maximum sequence length:	170 tokens	185 tokens	full length (264 tokens)
	Sample size:	218 K	237 K	251 K
	Sequence Accuracy:	99.0%	98.8%	98.6%

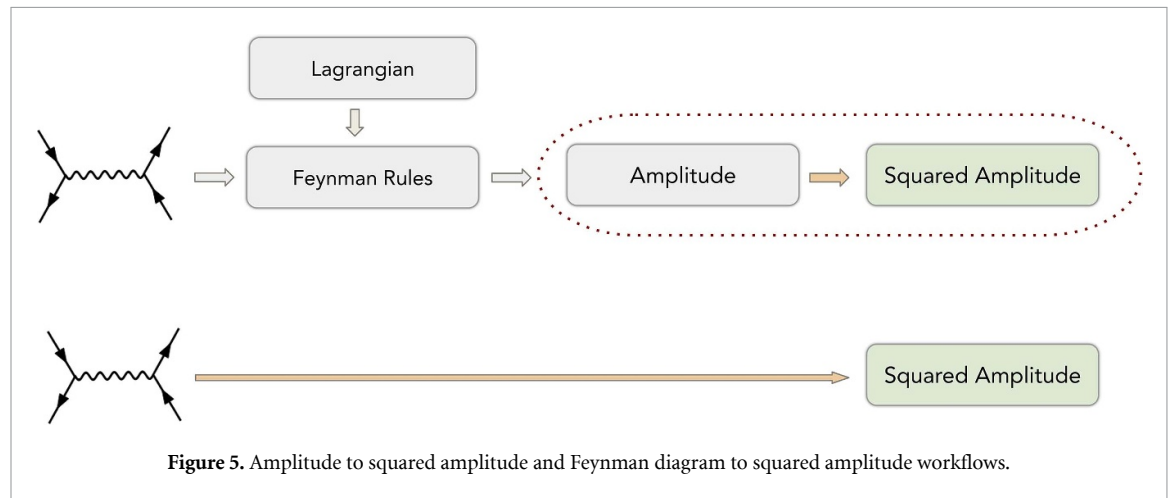


Figure 5. Amplitude to squared amplitude and Feynman diagram to squared amplitude workflows.

highly desirable to have a built-in solution rather than rely on human intervention by, for example, automating dimensional analysis (as each term should have the same dimension), or checking if the expression respects the conservation laws. While these practical solutions may be sufficient in some cases, a machine-learning solution whereby a confidence level can be assigned to the overall expression as well as on a per-token basis may be needed for this task. We leave this to future work.

As seq2seq models can be applied to all types of sequences, we can consider another interesting direction. Since Feynman diagrams themselves can be written down as sequences, we can map the Feynman diagram to the squared amplitude or to the cross section directly, as shown schematically in figure 5. A notable advantage of this is that the input Feynman diagram can be written by hand, if desired, without the need for a domain-specific tool to construct the amplitude. As a proof-of-concept of this idea, we test the transformer model on 2-to-3 QED and QCD processes following the same procedures as before. Remarkably, the QED Feynman diagram-based sequence model attains a sequence accuracy of 99.0% and token accuracy of 99.7% with average RMSE of 9.3×10^{-4} (see table 1). For QCD, the accuracy is much lower than for QED; the model for QCD attains a sequence accuracy of 73.4% and a token accuracy of 82.0% with average RMSE of 0.3 (excluding anomalous results with RMSE > 10). The current accuracy in QCD is lower compared to the models relying on amplitude sequence information, but this is expected as the number of input tokens is much smaller and may also indicate the need for a much larger training dataset, which we shall consider in a future work.

7. Conclusion

Our results demonstrate the ability of a symbolic deep learning model to learn the mapping between a particle interaction amplitude and its square to high accuracy despite the complexity of the mapping process. We observe that the accuracy strongly depends on the sequence length. The model has some limitations; however, the results obtained are sufficiently promising to motivate the search for further performance improvements that will be the focus of future work.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: <https://github.com/ML4SCI/SYMBAAHEP>.

Acknowledgments

We thank the University of Kentucky Center for Computational Sciences and Information Technology Services Research Computing for their support and use of the Lipscomb Compute Cluster and associated research computing resources. Special thanks to Grégoire Uhlich for technical help. This work was supported in part by the U.S. Department of Energy (DOE) under Award No. DE-SC0012447 (SG) and supported in part by the U.S. Department of Energy Award No. DE-SC0010102 (HP). This work was performed in part at Aspen Center for Physics, which is supported by National Science Foundation grant PHY-1607611.

ORCID iD

Abdulkhakim Alnuqaydan  <https://orcid.org/0000-0003-2460-6181>

References

- [1] Albertsson K *et al* 2018 Machine learning in high energy physics community white paper (arXiv:1807.02876)
- [2] Bourilkov D 2019 Machine and deep learning applications in particle physics *Int. J. Mod. Phys. A* **34** 1930019
- [3] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborová L 2019 Machine learning and the physical sciences *Rev. Mod. Phys.* **91** 045002
- [4] Feickert M and Nachman B 2021 A living review of machine learning for particle physics (arXiv:2102.02770)
- [5] Butter A, Plehn T, Soybelman N and Brehmer J 2021 Back to the formula—LHC edn (arXiv:2109.10414)
- [6] Shtabovenko V, Mertig R and Orellana F 2020 FeynCalc 9.3: new features and improvements *Comput. Phys. Commun.* **256** 107478
- [7] Uhlich G, Mahmoudi F and Arbey A 2021 MARTY—modern artificial theoretical physicist a c++ framework automating theoretical calculations beyond the standard model *Comput. Phys. Commun.* **264** 107928
- [8] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł and Polosukhin I 2017 Attention is all you need (arXiv:1706.03762)
- [9] Boos E, Bunichev V, Dubinin M, Dudko L, Edneral V, Ilyin V, Kryukov A, Savrin V, Semenov A and Sherstnev A 2004 CompHEP 4.4: automatic computations from Lagrangians to events *Nucl. Instrum. Methods Phys. Res. A* **534** 250–9
- [10] Ott M, Edunov S, Grangier D and Auli M 2018 Scaling neural machine translation (arXiv:1806.00187)
- [11] Guan W, Smetannikov I and Tianxing M 2020 Survey on automatic text summarization and transformer models applicability 2020 *Int. Conf. on Control, Robotics and Intelligent System (CCRIS 2020)* (New York: Association for Computing Machinery) pp 176–84
- [12] Liu W, Chen S, Guo L, Zhu X and Liu J 2021 CPTN: full transformer network for image captioning *CoRR* (arXiv:2101.10804)
- [13] Devlin J, Chang M-W, Lee K and Toutanova K 2018 Bert: pre-training of deep bidirectional transformers for language understanding (arXiv:1810.04805)
- [14] Karpathy A and Fei-Fei Li 2014 Deep visual-semantic alignments for generating image descriptions (arXiv:1412.2306)
- [15] Yikuan Li, Rao S, Solares J R A, Hassaine A, Canoy D, Zhu Y, Rahimi K and Salimi-Khorshidi G 2019 Behrt: transformer for electronic health records (arXiv:1907.09538)
- [16] Hatamizadeh A, Tang Y, Nath V, Yang D, Myronenko A, Landman B, Roth H and Daguang X 2021 Unetr: transformers for 3D medical image segmentation (arXiv:2103.10504)
- [17] Yanrong J, Zhou Z, Liu H and Davuluri R V 2021 Dnabert: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome *J. Bioinform.* **37** 2112–20
- [18] Lample G and Charton Fçois 2019 Deep learning for symbolic mathematics (arXiv:1912.01412)
- [19] d’Ascoli Sphane, Kamienny P-A, Lample G and Charton Fçois 2022 Deep symbolic regression for recurrent sequences (arXiv:2201.04600)
- [20] Valipour M, You B, Panju M and Ghodsi A 2021 Symbolicpt: a generative transformer model for symbolic regression (arXiv:2106.14131)
- [21] Cranmer M D, Xu R, Battaglia P and Ho S 2019 Learning symbolic physics with graph networks (arXiv:1909.05862)
- [22] Lemos B, Jeffrey N, Cranmer M, Shirley H and Battaglia P 2022 Rediscovering orbital mechanics with machine learning (arXiv:2202.02306)
- [23] Udrescu S-M and Tegmark M 2019 AI Feynman: a physics-inspired method for symbolic regression (arXiv:1905.11481)
- [24] Goldberg D 2017 *The Standard Model in a Nutshell* (Princeton, NJ: Princeton University Press)
- [25] Meurer A *et al* 2017 Sympy: symbolic computing in python *PeerJ Comput. Sci.* **3** e103
- [26] Abadi Min *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems (available at: www.tensorflow.org/)
- [27] Chollet F *et al* 2015 Keras (available at: <https://github.com/fchollet/keras>)
- [28] Kingma D P and Jimmy B 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)
- [29] Zaheer M, Guruganesh G, Dubey A, Ainslie J, Alberti C, Ontanon S, Pham P, Ravula A, Wang Q, Yang Li and Ahmed A 2020 Big bird: transformers for longer sequences (arXiv:2007.14062)
- [30] Beltagy I, Peters M E and Cohan A 2020 Longformer: the long-document transformer (arXiv:2004.05150)