Algebraically-Informed Deep Networks: A Deep Learning Approach to Represent Algebraic Structures

Mustafa Hajij[1] **Ghada Zamzmi**[2] **Matthew Dawson**[3]
Greg Muller[4] **Theodore Papamarkou**[5]

[1]University of San Francisco [2]HeartFlow [3] CIMAT
[4]University of Oklahoma [5]PolyShape

Abstract

Understanding how to uncover mathematical structure directly from data is a central challenge at the interface of deep learning and the mathematical sciences. In this work, we take a step toward bridging algebraic structures and neural networks by introducing algebraically-informed deep networks, a framework that represents any finitely-presented algebraic object using a collection of neural networks trained to satisfy its defining relations. Given a presentation $\langle S \mid R \rangle$, our method assigns a neural network to each generator and optimizes these networks so that they obey the algebraic relations in R. This provides a general computational mechanism for obtaining both linear and nonlinear representations of a wide class of algebraic structures, including groups, associative algebras, and Lie algebras. We demonstrate the applicability of the approach on algebraic and geometric structures central to low-dimensional topology. In particular, we show how the method recovers solutions to the Yang-Baxter equation, yields representations of braid groups and Temperley-Lieb algebras, and via the Reshetikhin-Turaev construction can be used to obtain link invariants. The results indicate that algebraic compatibility can be robustly enforced through gradient-based optimization, opening a path toward using deep learning as a tool for exploring algebraic structures and topological invariants.

1 Introduction

Over the past years, deep learning techniques have been used for solving partial differential equations Lagaris et al. [1998, 2000], Raissi [2018], Sirignano and Spiliopoulos [2018], obtaining physics-informed surrogate models Raissi et al. [2017], Rudy et al. [2017], computing the Fourier transform with deep networks Li et al. [2020], finding roots of polynomials Huang et al. [2004], and solving non-linear implicit system of equations Song et al. [2020]. In this work, we make one step towards building the bridge between algebraic/geometric structures and deep learning, and aim to answer the following question: How can deep learning be used to uncover the underlying solutions of an arbitrary system of algebraic equations?

To answer this question, we introduce *Algebraically-Informed Deep Networks* (**AIDN**), a deep learning method to represent any finitely-presented algebraic object with a set of neural networks. Our method uses a set of deep neural networks to represent a set of formal algebraic symbols that satisfy a system of algebraic relations. These deep neural networks are simultaneously trained to satisfy the relations between these symbols using an optimization paradigm such as stochastic gradient descent (SGD). The resulting neural networks are *algebraically-informed* in the sense that they satisfy the algebraic relations. We show that a wide variety of mathematical problems can be solved using this formulation.

1st Conference on Neural Information Processing Systems (NeurIPS 2025).

Next, we discuss a motivating example and present the applicability of our method on the well-known Yang-Baxter equation Jimbo [1989], which has been extensively studied in mathematics and physics.

1.1 Motivating Example: The Yang-Baxter Equation

To solve the set-theoretic Yang-Baxter equation, one seeks an invertible function $R : A \times A \rightarrow A \times A$, where A is some set, that satisfies the following equation:

$$(R \times id_A) \circ (id_A \times R) \circ (R \times id_A) = (id_A \times R) \circ (R \times id_A) \circ (id_A \times R). \quad (1)$$

Finding solutions of the above equation has a long history and proved to be a very difficult problem. For many decades, the Yang–Baxter equation[1] has been studied in quantum field theory and statistical mechanics as the master equation in integrable models Jimbo [1989]. Later, this equation was applied to many problems in low-dimensional topology Kassel and Turaev [2008]. For example, solutions of the Yang-Baxter equation were found to induce representations of the braid groups and have been used to define knots and 3-manifold invariants Turaev [2020]. Today, the Yang-Baxter equation is considered a cornerstone in several areas of physics and mathematics Vieira [2018] with applications to quantum mechanics Kauffman and Lomonaco Jr [2004], algebraic geometry Krichever [1981], and quantum groups Turaev [1988].

As an example application, we show how the proposed **AIDN** can be utilized to solve the Yang-Baxter equation Etingof et al. [1998]. In particular, assuming $A \subseteq \mathbb{R}^n$ is a subset of a Euclidean space[2], **AIDN** realizes the desired solution R in equation 1 as a neural network $f_R(\theta)$, where $\theta \in \mathbb{R}^k$. Using SGD, we can find the parameters θ by optimizing a loss function, which essentially satisfies equation 1. More details are given in Section 4.1.

1.2 Related Work

Our work can be viewed as a part of the quest to discover knowledge and a step towards building learning systems that are capable of uncovering the underlying mathematical and physical laws from data. Examples of current deep learning efforts to solve problems in mathematics and physics include general methods to solve partial differential equations Sirignano and Spiliopoulos [2018], Lagaris et al. [1998, 2000], Raissi [2018] or more particular ones that are aimed at solving single equations such as the Schrödinger equation Mills et al. [2017]. Also, deep learning has been used to solve equations related to fluid mechanics Brunton et al. [2020], Wang et al. [2020], non-linear equations Mathia and Saeks [1995], Song et al. [2020] and transcendental equations Jeswal and Chakraverty [2018].

This work can also be viewed as a step towards advancing computational algebra Seress [1997], Lux and Pahlings [2010]. Although there is a large literature devoted to computing linear representations of finitely-presented algebraic objects Holt et al. [2005] and of finite groups in particular Steel [2012], Dabbaghian-Abdoly [2005] as well as few works about representation of algebras Fischbacher and de la Peña [1986], we are not aware of any algorithm that computes non-linear representations of algebraic structures. Further, existing works find the representations of algebraic structures in special cases Adams and Cloux [2008]; the majority of these algorithms utilize GAP Group et al. [2007], a system for computational discrete algebra. Our proposed **AIDN** can (1) compute both linear and non-linear representations of algebraic structures, (2) provide a general computational scheme that utilizes non-traditional tools, and (3) offer a different paradigm from the classical methods in this space.

1.3 Summary of Contribution

The main contributions of this work can be summarized as follows:

1. We propose **AIDN**, a deep learning algorithm that computes non-linear representations of algebraic structures. To the best of our knowledge, we are the first to propose a deep

[1] Technically, the term *Yang-Baxter equation* is utilized whenever the map R is linear. When the map R is an arbitrary map defined on a set, the term *set-theoretic Yang-Baxter* is used instead.

[2] We may consider real or complex Euclidean spaces, but for this example we will constrain our discussion on real-Euclidean spaces.

learning-based method for computing non-linear representations of any finitely presented algebraic structure.

2. We demonstrate the applicability of **AIDN** in low-dimensional topology. Specifically, we study the applicability of **AIDN** to braid groups and Templery-Lieb algebras, two algebraic constructions that are significant in low-dimensional topology.
3. We utilize **AIDN** for knot invariants discovery using deep learning methods. Specifically, using the Reshetikhin-Turaev construction we show that **AIDN** can be used to construct new link invariants.

The rest of the paper is organized as follows. Section 3 presents **AIDN**. Section 4 studies braid groups and Temperley–Lieb algebras.

2 Background

This section provides a brief introduction to neural networks and shows examples of the algebraic structures that can be defined on them. We only focus on real-neural networks with domains and co-domains in real Euclidean spaces for the sake of clarity and brevity. However, AIDN can be easily extended to complex-neural networks [**??**].

A *neural network* is a function $\mathrm{Net}:\mathbb{R}^{d_{\text{in}}}\to\mathbb{R}^{d_{\text{out}}}$ written as $\mathrm{Net}=f_L\circ\cdots\circ f_1$, where each layer $f_i:\mathbb{R}^{n_i}\to\mathbb{R}^{m_i}$ has the form $f_i(x)=\alpha_i(W_ix+b_i)$ with $W_i\in\mathbb{R}^{m_i\times n_i}$, $b_i\in\mathbb{R}^{m_i}$, and α_i applied coordinate-wise. Let $\mathcal{N}(\mathbb{R}^n)$ denote networks $\mathbb{R}^n\to\mathbb{R}^n$; it is closed under composition. For $\mathrm{Net}_1\in\mathcal{N}(\mathbb{R}^m)$ and $\mathrm{Net}_2\in\mathcal{N}(\mathbb{R}^n)$, define $(\mathrm{Net}_1\times\mathrm{Net}_2)(x,y)=(\mathrm{Net}_1(x),\mathrm{Net}_2(y))$. The space $\mathcal{N}(\mathbb{R}^n)$ becomes an associative $\mathbb{R}$-algebra via pointwise addition, scalar multiplication, and composition; its invertible elements form $\mathcal{G}(\mathbb{R}^n)$, and the commutator $[\mathrm{Net}_1,\mathrm{Net}_2]=\mathrm{Net}_1\circ\mathrm{Net}_2-\mathrm{Net}_2\circ\mathrm{Net}_1$ defines a Lie algebra structure. These operations allow many algebraic structures to be represented inside $\mathcal{N}(\mathbb{R}^n)$.

3 Method: Algebraically-Informed Deep Networks (AIDN)

The motivation example provided in Section 1.1 can be defined formally and generally as follows. Let $s_1,\dots,s_n$ be a collection of formal symbols (generators) that satisfy a system of equations $r_1,\dots,r_k$. We are interested in finding functions $f_{s_1},\dots,f_{s_n}$ (defined on some domain) that correspond to the formal generators and satisfy the same relations.

Let $S=\{s_i\}_{i=1}^n$ and $R=\{r_i\}_{i=1}^k$. Such a system $\langle S\mid R\rangle$ is called a *presentation*. Depending on the algebraic operations we allow, presentations encode different algebraic objects. For example, requiring associative multiplication with identity and inverses yields a group[3]. Finding functions $\{f_i\}$ that correspond to S and satisfy R is formally equivalent to finding a homomorphism from $\langle S\mid R\rangle$ to a target algebra where the $\{f_i\}$ live.

From this perspective, **AIDN** takes a finite presentation and produces neural networks $\{f_i(x;\theta_i)\}_{i=1}^n$, $\theta_i\in\mathbb{R}^{k_i}$, such that these nets correspond to the generators and satisfy the relations. We learn weights by minimizing

$$\mathcal{L}(f_1,\cdots,f_n):=\sum_{i=1}^{k}\|\mathcal{F}(r_i)\|_2^2, \tag{2}$$

where $\mathcal{F}(r_i)$ is the relation r_i written in terms of the networks and $\|\cdot\|_2$ is an L^2 norm, optimized with SGD Bottou [2012].

For the set-theoretic Yang–Baxter equation (Eq. 1), **AIDN** optimizes

$$\mathcal{L}(f_R):=\|(f_R\times id_A)\circ(id_A\times f_R)\circ(f_R\times id_A)-(id_A\times f_R)\circ(f_R\times id_A)\circ(id_A\times f_R)\|_2^2. \tag{3}$$

Invertibility for f_R can be enforced via invertible architectures (e.g., Jacobsen et al. [2018], Behrmann et al. [2019]) or by introducing $g_R(\alpha)$ and augmenting the loss:

$$\begin{aligned}\mathcal{L}(f_R,g_R):=&\|(f_R\times id_A)\circ(id_A\times f_R)\circ(f_R\times id_A)-(id_A\times f_R)\circ(f_R\times id_A)\circ(id_A\times R)\|_2^2\\&+\|f_R\circ g_R-id_{A\times A}\|_2^2+\|g_R\circ f_R-id_{A\times A}\|_2^2.\end{aligned}$$

[3] A group can have different presentations; determining which presentations give the trivial group is undecidable Lyndon and Schupp [1977], Rabin [1958].

We observed the second approach yields more stable solutions in practice. Algorithm 1 summarizes **AIDN**.

Algorithm 1: AIDN: Algebraically-Informed Deep Nets

Function *AIDN*($\langle S \mid R \rangle$, m), $S = \{s_i\}_{i=1}^n$, $R = \{r_i\}_{i=1}^k$
    foreach *Generator* $s_i \in S$ **do**
        Define network $f_i(x; \theta_i) \in \mathcal{N}(\mathbb{R}^m)$;
    $\mathcal{L}(f_1, \cdots, f_n) := \sum_{i=1}^k \|\mathcal{F}(r_i)\|_2^2$;
    Minimize $\mathcal{L}$ via stochastic gradient descent;
    return $\{f_i(\theta)\}_{i=1}^n$

Conceptually simple yet broadly applicable, **AIDN** relies on expressive networks and sufficient samples. Despite no guarantees of global optimality, we consistently found good solutions in practice; this aligns with observations on deep loss landscapes (e.g., Choromanska et al. [2015], Sagun et al. [2014], Shwartz-Ziv and Tishby [2017]). Architecture choices determine representation type (linear, affine, non-linear). We explore several in Sections 4.1 and 4.2.

4 AIDN for Finitely-presented Algebraic Structures

We demonstrate generality on structures significant to geometric topology and algebra: braid groups (Sec. 4.1) and Temperley–Lieb algebras (Sec. 4.2).

4.1 Braid Group

Background and diagrammatic definitions are provided in Hajij et al. [2020]. We focus here on the neural representation and results.

4.1.1 Representing the Braid Group Via Neural Networks

Let B_m be the braid group with generators and relations as defined classically. Using **AIDN**, one might expect to train $m-1$ networks, but the decomposition

$$\sigma_i = (\times^{i-1} id) \times \sigma \times (\times^{m-i+1} id), \tag{4}$$

reduces training to functions on two strands. This yields:

Lemma 1 *Let $m, n \geq 1$, $f, g \in \mathcal{N}(\mathbb{R}^n \times \mathbb{R}^n)$. Define $f_i, g_i \in \mathcal{N}((\mathbb{R}^n)^m)$ by $f_i := (\times^{i-1} id_{\mathbb{R}^n}) \times f \times (\times^{m-i+1} id_{\mathbb{R}^n})$ and similarly for g_i. If*

1. *$f_i f_{i+1} f_i = f_{i+1} f_i f_{i+1}$ for all $1 \leq i < m-1$,*
2. *$f_i g_i = id = g_i f_i$ for all $1 \leq i < m$,*
3. *$f_i f_j = f_j f_i$ for $|i-j| > 1$,*

then the map $F : B_m \to \mathcal{G}((\mathbb{R}^n)^m)$ given by $\sigma_i \mapsto f_i$, $\sigma_i^{-1} \mapsto g_i$ is a group homomorphism.

Thus we train two networks f, g that satisfy the functional braid relations (Fig. 1). We found jointly training inverses (f, g) works better than enforcing invertibility on f alone.

4.1.2 Performance of AIDN on Braid Group Representations

We train $f, g \in \mathcal{N}(\mathbb{R}^n \times \mathbb{R}^n)$ with architecture

$$\mathbb{R}^n \to \mathbb{R}^{2n+2} \to \mathbb{R}^{2n+2} \to \mathbb{R}^{100} \to \mathbb{R}^{50} \to \mathbb{R}^n, \tag{5}$$

under three regimes: **Linear** (identity activation, zero bias), **Affine** (identity activation, nonzero bias), and **Non-linear** (tanh, zero bias, identity last layer). Results are in Table 1; linear/affine after 2 epochs; non-linear after 600 epochs.

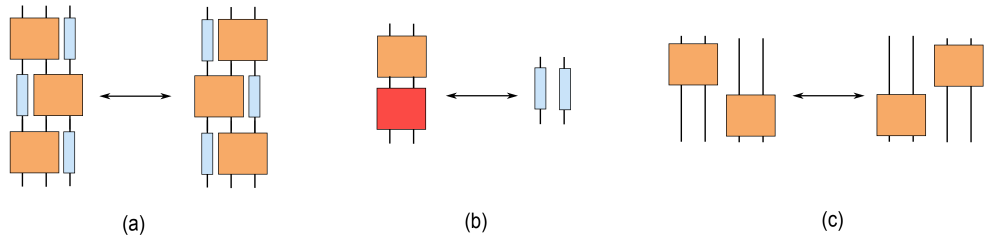

Figure 1: Functional braid relations for the neural maps: (a) Yang–Baxter-type, (b) inverse, (c) distant-commutation.

	$n = 2$			$n = 4$			$n = 6$		
braid group relation	Linear	Affine	Non-Linear	Linear	Affine	Non-Linear	Linear	Affine	Non-Linear
$f \circ g = id_{I^n}$	15×10^{-6}	12×10^{-6}	0.04	30×10^{-6}	24×10^{-6}	0.02	30×10^{-6}	31×10^{-6}	0.04
$g \circ f = id_{I^n}$	10×10^{-6}	11×10^{-6}	0.02	25×10^{-6}	18×10^{-6}	0.02	32×10^{-6}	30×10^{-6}	0.04
set-theoretic Yang Baxter	75×10^{-7}	70×10^{-7}	0.007	32×10^{-6}	29×10^{-6}	0.01	29×10^{-6}	27×10^{-6}	0.01

Table 1: L^2 error of braid relations after training f and g.

4.2 Temperley–Lieb Algebra

We test AIDN on the Temperley–Lieb algebra TL_m, a fundamental diagrammatic algebra in topology and mathematical physics. For a commutative ring $\mathcal{R}$ and integer $m \geq 2$, the algebra TL_m is generated by $U_1, \ldots, U_{m-1}$ and is defined by the relations $U_i U_{i+1} U_i = U_i$ for $1 \leq i \leq m-2$, $U_i U_{i-1} U_i = U_i$ for $2 \leq i \leq m-1$, $U_i^2 = \delta \, U_i$ for $1 \leq i \leq m-1$, and $U_i U_j = U_j U_i$ whenever $|i-j| > 1$, where $\delta \in \mathcal{R}$ is fixed. Although traditionally expressed via planar non-crossing diagrams, we work directly with this algebraic presentation.

4.2.1 Neural Representation Framework

To represent TL_m with neural networks, we adopt the reduction used for braids and train a single "two-strand" map. Let $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n$ be a neural network. For $1 \leq i \leq m-1$, define

$$f_i(\times^{i-1} id_{\mathbb{R}^n}) \times f \times (\times^{m-i+1} id_{\mathbb{R}^n}),$$

where $\times$ denotes horizontal concatenation and $id_{\mathbb{R}^n}$ is the identity on $\mathbb{R}^n$. Each f_i acts on $(\mathbb{R}^n)^m$ and plays the role of the generator U_i.

The Temperley–Lieb relations translate directly into functional equations for f: $f_i f_{i+1} f_i = f_i$ for $1 \leq i \leq m-2$, $f_i f_{i-1} f_i = f_i$ for $2 \leq i \leq m-1$, $f_i^2 = \delta f_i$ for $1 \leq i \leq m-1$, and $f_i f_j = f_j f_i$ whenever $|i-j| > 1$. These conditions ensure that the assignment $U_i \mapsto f_i$ extends uniquely to an algebra homomorphism $TL_m \to \mathcal{N}((\mathbb{R}^n)^m)$.

In effect, AIDN need only learn a single network f satisfying the above identities; the full representation of TL_m then follows automatically by inserting f into the appropriate tensor positions to obtain the maps $\{f_i\}_{i=1}^{m-1}$.

References

Jeffrey Adams and Fokko du Cloux. Algorithms for representation theory of real reductive groups. *arXiv preprint arXiv:0807.3093*, 2008.

Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582, 2019.

Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pages 192–204, 2015.

Vahid Dabbaghian-Abdoly. An algorithm for constructing representations of finite groups. *Journal of Symbolic Computation*, 39(6):671–688, 2005.

Pavel Etingof, Travis Schedler, and Alexandre Soloviev. Set-theoretical solutions to the quantum yang-baxter equation. *arXiv preprint math/9801047*, 1998.

U Fischbacher and JA de la Peña. Algorithms in representation theory of algebras. In *Representation Theory I Finite Dimensional Algebras*, pages 115–134. Springer, 1986.

GAP Group et al. Gap system for computational discrete algebra, 2007.

Mustafa Hajij, Ghada Zamzmi, Matthew Dawson, and Greg Muller. Algebraically-informed deep networks (aidn): a deep learning approach to represent algebraic structures. *arXiv preprint arXiv:2012.01141*, 2020.

Derek F Holt, Bettina Eick, and Eamonn A O'Brien. *Handbook of computational group theory*. CRC Press, 2005.

De-Shuang Huang, Horace HS Ip, and Zheru Chi. A neural root finder of polynomials based on root moments. *Neural Computation*, 16(8):1721–1762, 2004.

Jörn-Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon. i-revnet: Deep invertible networks. *arXiv preprint arXiv:1802.07088*, 2018.

SK Jeswal and Snehashish Chakraverty. Solving transcendental equation using artificial neural network. *Applied Soft Computing*, 73:562–571, 2018.

Michio Jimbo. Introduction to the yang-baxter equation. *International Journal of Modern Physics A*, 4(15):3759–3777, 1989.

Christian Kassel and Vladimir Turaev. *Braid groups*, volume 247. Springer Science & Business Media, 2008.

Louis H Kauffman and Samuel J Lomonaco Jr. Braiding operators are universal quantum gates. *New Journal of Physics*, 6(1):134, 2004.

Igor Moiseevich Krichever. Baxter's equations and algebraic geometry. *Functional Analysis and Its Applications*, 15(2):92–103, 1981.

Isaac E Lagaris, Aristidis Likas, and Dimitrios I Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5):987–1000, 1998.

Isaac E Lagaris, Aristidis C Likas, and Dimitris G Papageorgiou. Neural-network methods for boundary value problems with irregular boundaries. *IEEE Transactions on Neural Networks*, 11 (5):1041–1049, 2000.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Klaus Lux and Herbert Pahlings. *Representations of groups: a computational approach*, volume 124. Cambridge University Press, 2010.

R.C. Lyndon and P.E. Schupp. *Combinatorial Group Theory*. Number v. 89 in Classics in mathematics. Springer-Verlag, 1977. ISBN 9783540076421. URL `https://books.google.com.mx/books?id=2WUPAQAAMAAJ`.

Karl Mathia and Richard Saeks. Solving nonlinear equations using recurrent neural networks. In *World congress on neural networks, July*, pages 17–21, 1995.

Kyle Mills, Michael Spanner, and Isaac Tamblyn. Deep learning and the schrödinger equation. *Physical Review A*, 96(4):042113, 2017.

Michael O. Rabin. Recursive unsolvability of group theoretic problems. *Annals of Mathematics*, 67 (1):172–194, 1958.

Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, 19(1):932–955, 2018.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

Levent Sagun, V Ugur Guney, Gerard Ben Arous, and Yann LeCun. Explorations on high dimensional landscapes. *arXiv preprint arXiv:1412.6615*, 2014.

Akos Seress. An introduction to computational group theory. *Notices of the AMS*, 44(6):671–679, 1997.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

Yang Song, Chenlin Meng, Renjie Liao, and Stefano Ermon. Nonlinear equation solving: A faster alternative to feedforward computation. *arXiv preprint arXiv:2002.03629*, 2020.

Allan Kenneth Steel. Construction of ordinary irreducible representations of finite groups. 2012.

Vladimir G Turaev. The yang-baxter equation and invariants of links. *Inventiones mathematicae*, 92 (3):527–553, 1988.

Vladimir G Turaev. *Quantum invariants of knots and 3-manifolds*, volume 18. Walter de Gruyter GmbH & Co KG, 2020.

RS Vieira. Solving and classifying the solutions of the yang-baxter equation through a differential approach. two-state systems. *Journal of High Energy Physics*, 2018(10):110, 2018.

Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1457–1466, 2020.

TAG-DS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Paper is about algebraically informed Neural Networks and the paper presents the framework as well as a set of experiments.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We briefly discuss them at the end of the conclusion and provide an extended version in our appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Our main theorem and propositions are proven in the appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide these in our supplementary. We will also release our code and data publicly upon publication.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will also release our code and data publicly upon publication.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: For each experiment, we separately mention the hyperparameters, optimizer, data splits, and the curation of data.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Paper reports error bars over experiments and describes the number of runs these error bars are over.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute is not explicitly discussed; however, timings are not presented as part of the argument about the proposed approach and are not needed to replicate the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper focuses on a generalized framework for neural networks. The approach does not focus on any areas/applications of concern as highlighted within the Ethics Guidelines. Datasets in the paper do not contain any personally identifiable information/data about real people.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our work is largely theoretical and as such is not expected to make immediate social impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Paper presents a theoretical framework and does not release models or present results on data scraped from the internet, etc.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite datasets used in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will also release our code and data publicly upon publication. The repository has a license. We do not have assets of others within our data/code base.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No human-subject research was done.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [No]

    Justification: LLMs were only used for wording help.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.