Similarity-based equational inference in physics

Jordan Meadows^{1,*} and André Freitas^{1,2,†}

¹*Department of Computer Science, University of Manchester, Manchester M13 9PL, United Kingdom*

²*Idiap Research Institute, Rue Marconi 19, 1920 Martigny, Switzerland*

(Received 25 March 2021; accepted 21 June 2021; published 28 October 2021)

Automating the derivation of published results is a challenge, in part due to the informal use of mathematics by physicists compared to that of mathematicians. Following demand, we describe a method for converting informal handwritten derivations into datasets and present an example dataset crafted from a contemporary result in condensed matter. We define an equation reconstruction task completed by rederiving an unknown intermediate equation posed as a state, taken from three consecutive equational states within a derivation. Derivation automation is achieved via computer algebra system (CAS) by applying string-based CAS-reliant actions to states, which mimic mathematical operations and induce state transitions. We implement a symbolic similarity-based heuristic search to solve the equation reconstruction task as an early step towards multi-hop equational inference in physics.

DOI: [10.1103/PhysRevResearch.3.L042010](https://doi.org/10.1103/PhysRevResearch.3.L042010)

I. INTRODUCTION

Automating physical reasoning first involves the comprehension of physical concepts, language, and algebra, which form a cohesive informal mathematical explanation. Physics-inspired data-driven neural approaches are often used for the purpose of accurate calculation and simulation, but not for deriving equations [1–6]. Examples of equational [7–9] and conceptual [10] inference do not convey the complex arguments conducted within derivations, which combine assumed mathematics similar to premise selection, symbolic manipulation of equations, and reference to physical concepts. Automated theorem proving (ATP) is associated with mathematical rigor, but is not suitable for this type of informal equational and conceptual reasoning used by physicists [11,12]. Literature at this interface is limited and scarce [13], and there are few complete real-world derivations of published results that exist in a computer interpretable format.

Considering these limitations, this Letter proposes three contributions: (i) we present a dataset consisting of 368 equations compiled from the detailed derivation of a published equation in physics [14], represented as a sequence of states and actions, and we describe the dataset creation method; (ii) we define an *equation reconstruction task* by considering the smallest nontrivial derivation as three consecutive equational states with an unknown intermediate state that we aim to reconstruct; (iii) we propose the use of *symbolic similarity-based heuristics*, a knowledge base of assumed equations, and a set of granular mathematical operations

posed as string-related actions formulated in a computer algebra system (CAS) to solve the equation reconstruction task on the PhysAI-DS1 dataset [15]. We claim an approach accuracy of 56.2% considering both exact matching and the nonzero similarity cutoff regime, given the dataset, knowledge base, the set of actions, and the computer algebra system.

II. EQUATIONAL INFERENCE IN PHYSICS

Research at the border between artificial intelligence (AI) and physics has seen many recent successes. Data-driven approaches are popular, where many share the theme of feeding large datasets to physics-inspired architectures, learning physical model parameters often within the context of real-world problems [1–3,16–19]. In contrast to these simulation and calculation themed approaches, others aim to infer equations or conceptual insights from experimental or generated data from toy-model physics closer to symbolic regression [7–10,20,21], even recovering the latent network structure of dynamical systems from time series data [22]. While this class of approach is more similar to traditional theoretical physics, neural methods suffer from an explainability problem, and equational and conceptual reasoning are crucial components of physics derivations.

Automated theorem proving has seen little success in physics on the equational reasoning side, the literature is scarce [13], and efforts have been made to formalize physics towards ATP [12]. While Ref. [11] highlights five directions, we expand upon three in this work: (i) increasing the collection of physical theories we have in forms that can be used for training and validating symbolic reasoners in physics; (ii) developing representation methods and evaluation tasks which necessitate approaches that select suitable approximations, idealizations, and abstractions; and (iii) analysis of the nature of the informal argumentation used in physics.

Aligned with direction (i), we present 368 equations taken from physical theory [14] converted into a dataset as the first

*jordan.meadows@postgrad.manchester.ac.uk

†andre.freitas@manchester.ac.uk

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

contribution. Following direction (iii), this dataset is generated by a method which captures much of the derivation argumentation, by treating derivation progression as a finite state machine reliant upon computer algebra operations and representations. We address direction (ii) with an *equation reconstruction task* as a second contribution, which provides a medium for creating approaches which involve the selection of idealizations, approximations, and abstractions. As a third contribution, our similarity-based approach partially reconstructs the equational argument from PhysAI-DS1, which requires the use of symbolic approximations and the selection of supporting premises as a result of the physical theory that the comprised derivation represents.

ATP methods have a strong reliance on logical formalization. In order to be comparable to ATP-based methods, derivations in physics would need to be translated into a logical form. There is a shared understanding that existing standard logical frameworks used in ATPs are limited and not aligned to the requirements of the nature of physics argumentation [11,12], and thus comparative baselines of this kind cannot be made apart from a very controlled fragment of the physics domain. Additionally, to our knowledge there are no efforts to automate computer algebra in physics, not at least at the discourse-level form expressed in physics papers. Here we present a baseline for the equation reconstruction task, with other approaches unable to be easily adapted to solve it.

Reinforcement learning has been applied in the context of regular [23] and equational [24] theorem proving, and nuclear physics [25]. Although such techniques may be useful for searching large state spaces, they are not currently applied in the context of computer algebra expressed at the surface form of physics papers. Instead, our contribution targets a granular understanding of the dialog between equational symbolic forms in physics and similarity metrics, and its utility as one component of the symbolic inference mechanism.

We develop and evaluate our early single-hop [26] approach in the context of a multi-hop extensible equational inference task, using a computer algebra sequence and subsequent dataset created using a method for converting contemporary physics derivations into CAS-interpretable data.

III. DATASET CONSTRUCTION METHODOLOGY

In theoretical condensed matter, Ref. [14] is concerned with manipulating topological features of polaritons in cavity-embedded honeycomb metasurfaces. The following equation is the first numbered result:

$$H_{\text{mat}} = \hbar\tilde{\omega}_0 \sum_{\mathbf{q}} (a_{\mathbf{q}}^\dagger a_{\mathbf{q}} + b_{\mathbf{q}}^\dagger b_{\mathbf{q}}) + \hbar\tilde{\Omega} \sum_{\mathbf{q}} (f_{\mathbf{q}} b_{\mathbf{q}}^\dagger a_{\mathbf{q}} + \text{H.c.}). \quad (1)$$

This is accompanied by a brief derivation highlighting the key equations. The challenge is to use this derivation as a guide to derive (1) with *CAS-level representation*, at the granularity level of physicists performing the derivation in real time, and to then store this derivation as a dataset.

FIG. 1. The first three consecutive equational states from the computer algebra derivation. Each state is shown associated with four main pieces of information stored in PhysAI-DS1. If *state 2* is considered unknown, the three states form a *derivation unit* and this diagram represents the equation reconstruction task, where the goal is to select the correct action and NSA to reconstruct *state 2* given states 1 and 3.

A. Manual derivation

We start by manually deriving (1) from a single first-quantized harmonic oscillator (as seen in Fig. 1). We avoid skipping any steps, which drastically lengthens the derivation, as in practice minute mathematical operations are taken for granted at the paper form, which may require its elicitation to enable CAS compliance. We frame the derivation as a finite state machine where the current equation is the current state, and we carefully choose an informal action (operation) and action argument, that will progress the derivation to the next state. The state, action, and argument sequences are recorded up to Eq. (1).

B. Computer algebra augmentation

This curated derivation serves as a guide for the approximate derivation to be recreated in the CAS. However, particular actions, e.g., “divide RHS by 2”, may not already explicitly exist in the CAS and must be constructed. There are limitations on how actions can be constructed, so the reconstructed derivation may not exactly mirror the handwritten version in general.

We use SYMPY [27], which allows the rendering and manipulation of equations in LATEX, such that equations can match those within papers. From the handwritten derivation which we represent as a *state sequence*, we define an initial 2-tuple (LHS, RHS) equational state using the left-hand side (LHS) and right-hand side (RHS) generated from the CAS, and build and store appropriate actions which induce state transitions to other 2-tuples, which closely match the handwritten state sequence where possible. Along the way, actions may require yet unused symbols and assumed equations to progress the derivation, both of which we collect in a *knowledge base* (KB). We employ the following conventions:

- (1) Equations are stored as (LHS, RHS) tuples.
- (2) Actions are functions which accept *two arguments*, including the current state and: no argument, a symbol, an equation. These secondary arguments are called non-state

arguments (NSAs). Respectively, these action categories are called *self-state*, *symbol-state*, and *equation-state* actions.

(3) Actions cause the *minimum possible state transition* while still progressing down the derivation sequence. In contrast, the action causing the maximum possible state transition would be called “immediately derive the goal state from the initial state”. A more granular action example is “divide RHS by 2”, which induces a comparatively minimal change in the state semantics. Actions are designed to operate at this resolution, but physicists work less granularly.

(4) No LHS is the same for any state in the sequence (by default). For example, if $(H, \hbar\omega_0)$ is the current state and the action “divide RHS by 2” is applied to it, then the next state in the sequence is $(H^{(1)}, \hbar\omega_0/2)$ with an index appended to the LHS because $\hbar\omega \neq \hbar\omega/2$. However, an action “remove index” exists for the specific case of algebraic manipulation with the LHS.

(5) If an action is not appropriate given its arguments, then the action returns the current state. (e.g., action = “divide current state RHS by KB equation RHS”, but the KB equation RHS is a vector).

From this sequence of states, actions, and arguments, we construct a dataset PhysAI-DS1, consisting of 368 consecutive entries. Features from PhysAI-DS1, from left to right, read *{State string length (LaTeX text), State string (LaTeX text), State string length (SymPy text), State string (SymPy text), State string length (SymPy tree), State string (SymPy tree), Action, Non-state argument (SymPy text), State type, Action type}*, which we have designed to capture the important aspects of a physics derivation. The minimum, average, and maximum equation string lengths for the SYMPY text considered in this Letter are, respectively, 52, 495, and 5476. The LATEX equivalent is 56, 582, and 6318. We consider three different representations for the states, while NSAs are expressed using the native representation of the CAS. Of the 368 states, 76 are categorized as type *integrative*, 217 are *consequent*, and 75 are *terminal*.

The states are categorized by their role in the derivation as follows. *Integrative states*: formed by the action “consider knowledge base equation”. *Consequent states*: follow integrative states in the sequence. *Terminal states*: directly precede integrative states. There are 35 unique actions split into self-state, symbol-state, and equation-state categories, described in convention (2) above. The final state 2-tuple in the dataset represented as an equation is given by

$$H^{(83)} = \hbar \sum_{\mathbf{q}} \tilde{\Omega}(\text{H.c.} + f_{\mathbf{q}} b_{\mathbf{q}}^{\dagger} a_{\mathbf{q}}) + \hbar \sum_{\mathbf{q}} \tilde{\omega}_0 (a_{\mathbf{q}}^{\dagger} a_{\mathbf{q}} + b_{\mathbf{q}}^{\dagger} b_{\mathbf{q}}), \quad (2)$$

which is mathematically equivalent but rearranged from (1), up to the LHS. SYMPY has a native method of ordering equations and this ordering problem persists throughout the derivation. The order of noncommutative terms is preserved however, and this issue is cosmetic. The names of the actions are, in some cases, misleading and may behave unintuitively. This reflects the limitations of our chosen CAS, as often the most obvious way to force a desired state transition during construction involves the invention of obscure string-based operational actions, as a result of the limitations of the com-

puter algebra system and the curated derivation. See Fig. 1 for key information associated with the equational states in PhysAI-DS1.

IV. EQUATION RECONSTRUCTION TASK

The *equation reconstruction task* aims to support the construction of AI-supported derivation systems operating at different levels of symbolic representation (from a string/natural language level to an algebraic object/operational level). The task considers three consecutive states in a derivation grouped into a *derivation unit* $\{s_{i-1}, s_i, s_{i+1}\}$, where two consecutive actions performed on s_{i-1} form the unit.

We aim to create a reconstruction $\hat{s}_i$ such that it is mathematically equivalent and sufficiently similar to an unknown s_i , given s_{i-1} and s_{i+1} .

In practice, this involves selecting the correct action and NSA to cause the transition $s_{i-1} \rightarrow s_i$ (with probability 1) given a suitable knowledge base and action set. This derivation unit translates along the state sequence and, combined with a suitable inference algorithm, outputs one reconstruction per translation.

States are non-Markov, which reflects the behavior of physicists who may use equations from the distant past to progress the derivation. Action and state sequences behave deterministically, and actions receive two arguments: the current state and the non-state argument (NSA). In addition to those in the KB, we allow for equation-type NSAs to be any equational state in the history up to s_i which violates the Markov condition.

A derivation unit treats the contained states as a “microderivation” with no explicit connection to the complete sequence. Although the state history is required by the actions for progression, it effectively becomes part of the KB which grows proportional to i . As follows, for N states in the full derivation, there are $N - 2$ fully independent reconstructions $\hat{s}_i$, which is equivalent to solving the equation reconstruction task for $N - 2$ independent derivations, each of size three.

We append dummy states onto the end and the beginning of the sequence, totaling $N = 370$ states. This results in 368 reconstructions comparable to the 368 states in the derivation. The first dummy state is given by $s_0 = (x, ?)$ and represents an entirely neutral equation akin to beginning with no prior knowledge. The second dummy state appended last is the 2-tuple form of Eq. (1) from Ref. [14] *passed through a computer algebra system*, such that the final state in PhysAI-DS1, and the dummy state, are given, respectively, as

$$s_{N-1} = \left(H^{(83)}, \hbar \sum_{\mathbf{q}} \tilde{\Omega}(\text{H.c.} + f_{\mathbf{q}} b_{\mathbf{q}}^{\dagger} a_{\mathbf{q}}) + \hbar \sum_{\mathbf{q}} \tilde{\omega}_0 (a_{\mathbf{q}}^{\dagger} a_{\mathbf{q}} + b_{\mathbf{q}}^{\dagger} b_{\mathbf{q}}) \right), \quad (3)$$

$$s_N = \left(H_{\text{mat}}, \hbar \sum_{\mathbf{q}} \tilde{\Omega}(\text{H.c.} + f_{\mathbf{q}} b_{\mathbf{q}}^{\dagger} a_{\mathbf{q}}) + \hbar \sum_{\mathbf{q}} \tilde{\omega}_0 (a_{\mathbf{q}}^{\dagger} a_{\mathbf{q}} + b_{\mathbf{q}}^{\dagger} b_{\mathbf{q}}) \right). \quad (4)$$

V. SYMBOLIC SIMILARITY-BASED SEARCH

A. Similarity measures

If we consider Eqs. (3) and (4) in the sequence, we can observe that they are equivalent but symbolically dissimilar, as the string representations of each LHS are not identical. We employ a similarity measure M to classify whether one state is equal *and similar* to another, with a hyperparameter ε .

We form a set of canonical algebraic actions referred by the PhysAI-DS1 dataset. Actions accept two arguments: the current state and a non-state argument (NSA). An NSA can be either None, a symbol, or an equation 2-tuple.

We form a knowledge base of requisite equations necessary for the derivation, and a set L consisting of every (non-numeric) symbol from each equation. All states prior to s_{i-1} are also included in the knowledge base. Additionally, if l_{i-1} is the set of symbols which constitutes s_{i-1} , and l_{i+1} is the set which constitutes s_{i+1} , then we append the symmetric difference $l_{i-1} \Delta l_{i+1}$ to L . As we know the form of s_{i-1} and s_{i+1} , we can use the constituent symbols to guide the derivation, which we achieve by including $l_{i-1} \Delta l_{i+1}$ in L , an element of which may be accepted as the argument of an action function as an NSA. The set R contains the state history up to s_i and the requisite equations of the knowledge base. We define a combined knowledge base $\mathcal{K} = L \cup R$.

Given the knowledge base $\mathcal{K}$, the set of actions $\mathcal{A}$, and the initial and goal states, respectively, $s_{i-1}, s_{i+1} \in \mathcal{S}$ for set of states $\mathcal{S}$, we aim to generate a state $\hat{s}_i \in \mathcal{S}$ such that the similarity measure $M(\hat{s}_i, s_i) \leq \varepsilon$ where $M : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$, and $\varepsilon \in \mathbb{R}_{\geq 0}$ is a hyperparameter determining whether $\hat{s}_i$ and s_i are considered both equal and similar. We choose M to be based upon one of five similarity measures:

(i) Levenshtein distance, $M(s_i, s_j) = M_L(s_i, s_j)$: Minimum number of single character insertions, substitutions, or deletions required to transform string s_i into s_j .

(ii) Damerau-Levenshtein distance, $M(s_i, s_j) = M_{DL}(s_i, s_j)$: Extension of Levenshtein distance to include single character transpositions.

(iii) Hamming distance, $M(s_i, s_j) = M_H(s_i, s_j)$: Minimum number of single character substitutions to transform s_i into s_j .

(iv) Jaro similarity, $M(s_i, s_j) = 1 - \text{sim}_J(s_i, s_j)$:

$$\text{sim}_J(s_i, s_j) = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3} \left(\frac{m}{|s_i|} + \frac{m}{|s_j|} + \frac{m-t}{m} \right) & \text{otherwise,} \end{cases}$$

where $|s_i|$ is the string length, m is the number of matching characters (considered matching if $\lfloor \max(|s_i|, |s_j|)/2 \rfloor - 1$ or less characters between them), and t is half of the number of transpositions.

(v) Jaro-Winkler similarity, $M(s_i, s_j) = 1 - \text{sim}_{JW}(s_i, s_j)$:

$$\text{sim}_{JW} = \text{sim}_J - lp(1 - \text{sim}_J),$$

where l is the length of a common prefix at the start of the string up to a maximum of four characters, and p is a constant scaling factor (defined by the Jellyfish library default for our case).

Taking M as the Levenshtein measure, the distance between s_N and s_{N-1} from the derivation state sequence is $M(s_{N-1}, s_N) = 5$, and therefore we require $\varepsilon \geq 5$ to determine

whether the two states are similar within the Levenshtein metric. Lower bounds on ε that capture the final reconstruction for each of the similarity measures can be obtained via

$$\varepsilon_{\text{low}} = M(s_N, s_{N-1}). \quad (5)$$

B. Heuristic search

Following the *equation reconstruction* task description, while we know the form of s_{i-1} and s_{i+1} with the s_i unknown, we can form reconstruction *candidates* $\hat{c}_i$ by applying an action $a \in \mathcal{A}$ to s_{i-1} such that $a : (s_{i-1}, k) \mapsto \hat{c}_i$, where $\hat{c}_i \in \mathcal{S}$ and $k \in \mathcal{K}$. We apply a second action to all possible $\hat{c}_i$ with the mapping $a' : (\hat{c}_i, k') \mapsto \hat{c}_{i+1}$, where, in general, $k \neq k'$ and $a \neq a'$, to obtain $\hat{c}_{i+1}$ for $(\hat{c}_i, \hat{c}_{i+1}) \in \mathcal{S}_{\text{final}}$.

At this stage, multiple paths $\{s_{i-1}, \hat{c}_i, \hat{c}_{i+1}\}$ exist. By solving the following optimization problem, we can determine $\hat{s}_{i+1}$:

$$\hat{s}_{i+1} = \arg \min_{(\hat{c}_i, \hat{c}_{i+1}) \in \mathcal{S}_{\text{final}}} M(\hat{c}_{i+1}, s_{i+1}). \quad (6)$$

We employ the use of a heuristic $H(\hat{c}_i, s_{i+1}) : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$ to order the paths, as the search may end early under a specific condition. If we compare a state $\hat{c}_{i+1}$ such that $M(\hat{c}_{i+1}, s_{i+1}) = M(\hat{s}_{i+1}, s_{i+1}) = 0$, then the reconstruction candidate $\hat{c}_i$ from the path containing $\hat{c}_{i+1} = \hat{s}_{i+1}$ is selected as the reconstruction $\hat{s}_i$, which ends that iteration. Otherwise, all $\hat{c}_{i+1}$ are compared from each path, and the $\hat{c}_i$ of the path corresponding to the lowest $M(\hat{c}_{i+1}, s_{i+1})$ is taken as the $\hat{s}_i$. Based on the Levenshtein metric, H is defined as follows:

(a) $x(\hat{c}_i, s_{i+1}) = |l_s \setminus l_c|$, where l_s and l_c are the set of symbols that constitute s_{i+1} and $\hat{c}_i$, respectively. If x is large, it means there are many symbols in s_{i+1} that are not in $\hat{c}_i$, and the two are unlikely to be semantically linked.

(b) $y(\hat{c}_i, s_{i+1}) = M_L(c_{\min}, s_{\min})$, where $c_{\min}$ and $s_{\min}$ are the *subexpressions* in the trees of $\hat{c}_i$ and s_{i+1} , respectively, corresponding to the lowest M_L , given that the number of characters in the string representations of $c_{\min}$ and $s_{\min}$ is *less than or equal to 100*. This represents a similarity calculation for *small* subexpressions. If no suitable comparisons exist, then $y = M_L(\hat{c}_i, s_{i+1})$.

(c) $z(\hat{c}_i, s_{i+1}) = M_L(c_{\max}, s_{\max})$, where $c_{\max}$ and $s_{\max}$ are the *subexpressions* in the trees of $\hat{c}_i$ and s_{i+1} , respectively, corresponding to the lowest M_L , given that the number of characters in the string representations of $c_{\max}$ and $s_{\max}$ is *greater than 100*. This represents a similarity calculation for *large* subexpressions. If no suitable comparisons exist, then $z = M_L(\hat{c}_i, s_{i+1})$.

Given $\mathbf{r} = [n_1 x(\hat{c}_i, s_{i+1}), n_2 y(\hat{c}_i, s_{i+1}), n_3 z(\hat{c}_i, s_{i+1})]$ for $n_1, n_2, n_3 \in \mathbb{Z}_{\geq 0}$, the heuristic is given by

$$H(\hat{c}_i, s_{i+1}; n_1, n_2, n_3) = \|\mathbf{r}(\hat{c}_i, s_{i+1}; n_1, n_2, n_3)\|_2^2. \quad (7)$$

Each path $\{s_{i-1}, \hat{c}_i, \hat{c}_{i+1}\}$ corresponds to a heuristic value and the lowest-valued paths are searched first with the aim of finding a comparison $M(\hat{c}_{i+1}, s_{i+1}) = 0$ which terminates the search, else all comparisons are made.

The process is based on the notion that *if s_{i+1} can be obtained by applying two consecutive actions to s_{i-1} , then the state after one action may be s_i .*

TABLE I. The approach accuracy from the equation reconstruction experiments across the similarity measures. $\varepsilon_{\text{low}} = M(s_{N-1}, s_N)$ can be directly found by comparing the penultimate and final equations (states) in the derivation (state sequence). The hyperparameters used for each experiment were $(\varepsilon, n_1, n_2, n_3) = (\varepsilon_{\text{low}}, 10, 10, 10)$, where $\{n_i\}$ refer to the hyperparameters associated with Eq. (7), the heuristic.

Similarity measure	ε_{low}	Accuracy ($\eta = 0$)	Accuracy ($\eta = 1$)
Levenshtein	5	0.516	0.533
Damerau-Levenshtein	5	0.562	0.562
Hamming	224	0.508	0.799
Jaro	0.1643	0.514	0.690
Jaro-Winkler	0.0986	0.514	0.690

Extending to the multi-hop case, for a number of intermediate states $\mathcal{N}$, the number of derivation paths $\{s_{i-1}, c_i, \dots, c_{i+\mathcal{N}}\}$ grows as $(|\mathcal{A}||\mathcal{K}|)^{\mathcal{N}+1}$. Then, each Levenshtein comparison $M_L(\hat{c}_{i+\mathcal{N}-1}, s_{i+\mathcal{N}})$ in the heuristic, and final (e.g., Damerau-Levenshtein) comparison in the search $M_{DL}(\hat{c}_{i+\mathcal{N}}, s_{i+\mathcal{N}})$, has $O(mn)$ time complexity for equation string lengths m and n . However, this work targets stepwise, single-hop inference [26] as a unit of analysis, and end-to-end or multi-hop inference is currently outside our scope.

VI. EMPIRICAL EVALUATION

A reconstruction $\hat{s}_i$ is classified as a success in the evaluation if $M(\hat{s}_i, s_i) \leq \varepsilon$, where $\varepsilon = \eta \varepsilon_{\text{low}}$ for $\eta \in \mathbb{Z}_{\geq 0}$. The similarity value $\varepsilon_{\text{low}} = M(s_{N-1}, s_N)$ for a state sequence of length N is necessary in order to classify Eqs. (3) and (4) as equal and similar after a second action, and represents a unit of difference that is scalable with η . It can be directly found by comparing the penultimate and final equations (states) in the derivation (state sequence). The value $\eta = 0$ represents exact string matching, and $\eta = 1$ represents the minimum similarity required to reconstruct all equational states. At $\eta = 0$, false negatives may occur with no false positives and the accuracy is minimized, while for $\eta > 0$, the accuracy increases due to a decrease in false negatives and increase in false positives. We evaluate for accuracy at values of $\eta \in \{0, 1\}$. Table I describes the approach accuracy from the equation reconstruction experiments across the similarity measures.

From the Table I results, the approach based on the Damerau-Levenshtein metric outperforms the other string measures with respect to exact matches, and does not differ in accuracy upon introducing a unit of difference, ε_{low} . This suggests that Damerau-Levenshtein is less susceptible to the inclusion of false positives with increasing η . The Levenshtein-based measures both show low sensitivity to increasing η .

The Hamming measure approach results in the lowest exact matching accuracy, but the highest accuracy at unit difference, ε_{low} . This suggests that the measure is relatively more susceptible to false positives within our formalism, and less suitable for this task.

A. Categories of states and actions

Actions may be categorized depending upon their NSA as either *self-state*, *symbol-state*, or *equation-state*. States are

TABLE II. Number of instances associated with combinations of action-type and state-type pairs.

State type	Action type (count)		
	Self-state	Symbol-state	Equation-state
Integrative	0	0	76
Consequent	86	31	100
Terminal	20	18	37

categorized by their role in the state sequence as either *integrative*, *consequent*, or *terminal* (see Sec. III). There are two actions per state reconstruction (see Sec. IV), but only one action is directly responsible for the reconstruction. We consider this action, and the corresponding state, in the following analysis. Table II describes the total number of cases for each category of state versus each category of action. Table III describes the accuracy corresponding to each case.

At $\eta = 0$, consequent states formed by self-state actions and equation-state actions, and all integrative states, are most accurately reconstructed by the Damerau-Levenshtein approach. The standard Levenshtein approach reconstructs consequent states with symbol-state actions most accurately. Even in the exact matching case at $\eta = 0$, *integrative states are reconstructed with almost 90% accuracy, and all terminal states fail*.

At $\eta = 1$, the Levenshtein-based approaches share the lowest performance, where the Hamming approach displays nonzero accuracy in terminal states, and over 0.9 accuracy in the consequent self-state and symbol-state categories. This suggests that the Hamming approach quickly includes false positives with increasing η , as do the Jaro-based approaches particularly within the equation-state class, compared to the Levenshtein-based approaches.

B. Qualitative analysis and approach limitations

To provide additional insights into the limitations of our approach to solving the equation reconstruction task, we categorize known reconstruction failure types for the Damerau-Levenshtein variant as follows.

Terminal state problem. All terminal state reconstructions will fail—a claim supported by Table III. This arises due to the state after being *integrative* by definition, and integrative states are reconstructed by the specific action “consider knowledge base equation”. Integrative states represent the beginning of new derivation branches, and arriving at an integrative state after two consecutive actions is only dependent upon the second action being “consider knowledge base equation”. Therefore, any possible first action and subsequent reconstruction is suitable because of this condition on the second action. Terminal states represent 20.4% of the test set and around half of all reconstruction errors.

Multiple pathways. Reconstructions are ultimately determined by the final similarity check $M(\hat{c}_{i+1}, s_{i+1})$ such that $\hat{c}_{i+1}$ satisfies Eq. (6). Many $\hat{c}_i$ may be associated with a unique $\hat{c}_{i+1}$, and the lowest heuristic value $\hat{c}_i$ is then selected.

Repeating equations. If the s_{i+1} is an equation that has already been derived in the state history (up to the LHS

TABLE III. Equation reconstruction accuracy associated with combinations of action-type and state-type pairs. Italicized numbers represent the highest values from the seven possible combinations, across the five approaches.

Similarity measure	State type	Action-type accuracy ($\eta = 0$)			Action-type accuracy ($\eta = 1$)		
		Self-state	Symbol-state	Equation-state	Self-state	Symbol-state	Equation-state
Levenshtein	Integrative			0.776			0.829
	Consequent	0.605	0.742	0.560	0.628	0.742	0.560
	Terminal	0.000	0.000	0.000	0.000	0.000	0.000
Damerau-Levenshtein	Integrative			0.868			0.868
	Consequent	0.663	0.710	0.620	0.663	0.710	0.620
	Terminal	0.000	0.000	0.000	0.000	0.000	0.000
Hamming	Integrative			0.776			0.921
	Consequent	0.593	0.710	0.550	0.919	1.000	0.760
	Terminal	0.000	0.000	0.000	0.750	0.556	0.351
Jaro	Integrative			0.776			0.961
	Consequent	0.605	0.710	0.560	0.767	0.871	0.880
	Terminal	0.000	0.000	0.000	0.000	0.000	0.000
Jaro-Winkler	Integrative			0.776			0.961
	Consequent	0.605	0.710	0.560	0.767	0.871	0.880
	Terminal	0.000	0.000	0.000	0.000	0.000	0.000

index), then the first action will be “consider knowledge base equation” with the NSA as the repeated equation, and a second action chosen to not transition the state [equivalent to “do nothing”, convention (5)].

Action limitations. Actions are specifically made to induce state transitions between *known* states, i.e., from the curated derivation passed through the CAS. This means that upscaling the data or considering other datasets is time consuming, as the set of consecutive actions that forms a separate derivation may be entirely different. This would mean, for each new derivation, at least one pass of the full derivation sequence through the CAS must be made, and a new dataset must be constructed. Additionally, even within the same derivation, actions may not return mathematically correct states in all cases, as many are reliant on string manipulation and are not native to SYMPY directly. There is a considerable advantage in finding mathematically “robust” actions suitable for rigorous exploration of unknown states, but such work is outside our current scope.

Knowledge base selection. Premise selection is the task of selecting relevant mathematical statements aiming to maximize the probability of proving a given conjecture. The set of knowledge base equations has been handcrafted in the same vein as the action set, and they suffer similar limitations. It is a nontrivial task [28] to select suitable axiomatic equations or supporting statements capable of reconstructing derivations or proofs, and is outside our current scope.

VII. CONCLUSION

In response to the demand for informal mathematical datasets [11], we have created a method for producing datasets from physics derivations as a step towards multi-hop inference in physics. This method first involves a curated handwritten

derivation, then a recreation of the derivation in a suitably expressive computer algebra system, under an expressive representation scheme. We categorize states and actions within this methodology, and employ it to create a dataset *PhysAI-DS1* from a recent result in condensed matter [14]. We propose an equation reconstruction task which considers sets of three consecutive states within the derivation. The intermediate state is then considered unknown, and information from a knowledge base and the initial and final states may be used to mathematically infer the form of the unknown state by applying sequences of string-based CAS-reliant actions to the known state, and by utilizing similarity measures to approximate equality. As an early solution to the equation reconstruction task, we formulate primitive similarity-based heuristics with a search algorithm capable of reconstructing equations within *PhysAI-DS1* to 56.2% accuracy using the Damerau-Levenshtein metric, compared to baseline approaches using four distinct string similarity measures. We discuss limitations of the CAS-reliant similarity-based approach, and systematically characterize the performance of separate categories of action and state. Future work will focus on the physics explanation extraction process to account for different physical systems in a different *PhysAI* dataset. From this dataset, we aim to address multi-hop inference in physics via a neurosymbolic approach designed to account for both the linguistic and mathematical features of physics argumentation and explanation.

ACKNOWLEDGMENTS

This research was partially funded by EPSRC EnnCore (EP/T026995/1).

[1] G. Kissas, Y. Yang, E. Hwuang, W. R. Witschey, J. A. Detre, and P. Perdikaris, Machine learning in cardiovascu-

lar flows modeling: Predicting arterial blood pressure from non-invasive 4d flow MRI data using physics-informed neural

- networks, *Comput. Methods Appl. Mech. Eng.* **358**, 112623 (2020).
- [2] J. Liang and X. Zhu, Phillips-inspired machine learning for band gap and exciton binding energy prediction, *J. Phys. Chem. Lett.* **10**, 5640 (2019).
- [3] G. H. Teichert, A. Natarajan, A. Van der Ven, and K. Garikipati, Machine learning materials physics: Integrable deep neural networks enable scale bridging by learning free energy functions, *Comput. Methods Appl. Mech. Eng.* **353**, 201 (2019).
- [4] J. Collins, K. Howe, and B. Nachman, Anomaly Detection for Resonant New Physics With Machine Learning, *Phys. Rev. Lett.* **121**, 241803 (2018).
- [5] T. Bereau, R. A. DiStasio Jr, A. Tkatchenko, and O. A. Von Lilienfeld, Non-covalent interactions across organic and biological subsets of chemical space: Physics-based potentials parametrized from machine learning, *J. Chem. Phys.* **148**, 241706 (2018).
- [6] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, *Nat. Phys.* **14**, 447 (2018).
- [7] S.-M. Udrescu and M. Tegmark, Symbolic pregression: Discovering physical laws from raw distorted video, *Phys. Rev. E* **103**, 043307 (2021).
- [8] S.-M. Udrescu, A. Tan, J. Feng, O. Neto, T. Wu, and M. Tegmark, Ai Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity, [arXiv:2006.10782](https://arxiv.org/abs/2006.10782).
- [9] S. Kim, P. Y. Lu, S. Mukherjee, M. Gilbert, L. Jing, V. Čeperić, and M. Soljačić, Integration of neural network-based symbolic regression in deep learning for scientific discovery, *IEEE Trans. Neural Networks Learn. Syst.* **32**, 4166 (2020).
- [10] R. Iten, T. Metger, H. Wilming, L. del Rio, and R. Renner, Discovering Physical Concepts With Neural Networks, *Phys. Rev. Lett.* **124**, 010508 (2020).
- [11] E. Davis, Proof verification technology and elementary physics, in *Algorithms and Complexity in Mathematics, Epistemology, and Science* (Springer, New York, 2019), pp. 81–132.
- [12] C. Kaliszyk, J. Urban, U. Siddique, S. Khan-Afshar, C. Dunchev, and S. Tahar, Formalizing physics: Automation, presentation and foundation issues, in *International Conference on Intelligent Computer Mathematics* (Springer, New York, 2015), pp. 288–295.
- [13] N. S. Govindarajalulu, S. Bringsjord, and J. Taylor, Proof verification and proof discovery for relativity, *Synthese* **192**, 2077 (2015).
- [14] C.-R. Mann, T. J. Sturges, G. Weick, W. L. Barnes, and E. Mariani, Manipulating type-I and type-II Dirac polaritons in cavity-embedded honeycomb metasurfaces, *Nat. Commun.* **9**, 2194 (2018).
- [15] <https://github.com/jmeadows17/equational-inference> (unpublished).
- [16] J.-L. Wu, H. Xiao, and E. Paterson, Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework, *Phys. Rev. Fluids* **3**, 074602 (2018).
- [17] V. L. Deringer and G. Csányi, Machine learning based interatomic potential for amorphous carbon, *Phys. Rev. B* **95**, 094203 (2017).
- [18] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, Big data meets quantum chemistry approximations: The δ -machine learning approach, *J. Chem. Theory Comput.* **11**, 2087 (2015).
- [19] P. Baldi, P. Sadowski, and D. Whiteson, Searching for exotic particles in high-energy physics with deep learning, *Nat. Commun.* **5**, 4308 (2014).
- [20] T. Wu and M. Tegmark, Toward an artificial intelligence physicist for unsupervised learning, *Phys. Rev. E* **100**, 033311 (2019).
- [21] M. Raissi and G. E. Karniadakis, Hidden physics models: Machine learning of nonlinear partial differential equations, *J. Comput. Phys.* **357**, 125 (2018).
- [22] Z. Zhang, Y. Zhao, J. Liu, S. Wang, R. Tao, R. Xin, and J. Zhang, A general deep learning framework for network reconstruction and dynamics learning, *Appl. Network Sci.* **4**, 110 (2019).
- [23] C. Kaliszyk, J. Urban, H. Michalewski, and M. Olšák, Reinforcement learning of theorem proving, [arXiv:1805.07563](https://arxiv.org/abs/1805.07563).
- [24] J. Piepenbrock, T. Heskes, M. Janota, and J. Urban, Learning equational theorem proving, [arXiv:2102.05547](https://arxiv.org/abs/2102.05547).
- [25] M. Luo and L. Liu, Automatic derivation of formulas using reinforcement learning, [arXiv:1808.04946](https://arxiv.org/abs/1808.04946).
- [26] S. Min, E. Wallace, S. Singh, M. Gardner, H. Hajishirzi, and L. Zettlemoyer, Compositional questions do not necessitate multi-hop reasoning, [arXiv:1906.02900](https://arxiv.org/abs/1906.02900).
- [27] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh *et al.*, SYMPY: Symbolic computing in PYTHON, *Peer J. Comput. Sci.* **3**, e103 (2017).
- [28] D. Ferreira and A. Freitas, Premise selection in natural language mathematical texts, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, 2020), pp. 7365–7374.