# City Research Online

## City St George's, University of London

**Citation:** Arias-Tamargo, G., He, Y-H., Heyes, E., Hirst, E. & Rodriguez-Gomez, D. (2022). Brain webs for brane webs. Physics Letters B, 833, 137376. doi: 10.1016/j.physletb.2022.137376

This is the published version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

**Permanent repository link:** https://openaccess.city.ac.uk/id/eprint/29538/

**Link to published version:** https://doi.org/10.1016/j.physletb.2022.137376

**Copyright and Reuse:** Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk

Physics Letters B 833 (2022) 137376

Contents lists available at ScienceDirect

Physics Letters B

www.elsevier.com/locate/physletb

# Brain webs for brane webs

Guillermo Arias-Tamargo [a,b], Yang-Hui He [c,d,e,f], Elli Heyes [c,d,*], Edward Hirst [c,d], Diego Rodriguez-Gomez [a,b]

[a] Department of Physics, Universidad de Oviedo, C/ Federico García Lorca 18, 33007, Oviedo, Spain
[b] Instituto Universitario de Ciencias y Tecnologías Espaciales de Asturias (ICTEA) C/ de la Independencia 13, 33004 Oviedo, Spain
[c] Department of Mathematics, City, University of London, EC1V 0HB, UK
[d] London Institute for Mathematical Sciences, Royal Institution, London W1S 4BS, UK
[e] Merton College, University of Oxford, OX1 4JD, UK
[f] School of Physics, NanKai University, Tianjin, 300071, PR China

ARTICLE INFO

*Article history:*
Received 9 March 2022
Received in revised form 7 June 2022
Accepted 8 August 2022
Available online 17 August 2022
Editor: N. Lambert

ABSTRACT

We propose a new technique for classifying 5d Superconformal Field Theories arising from brane webs in Type IIB String Theory, using technology from Machine Learning to identify different webs giving rise to the same theory. We concentrate on webs with three external legs, for which the problem is analogous to that of classifying sets of 7-branes. Training a Siamese Neural Network to determine equivalence between any two brane webs shows an improved performance when webs are considered equivalent under a weaker set of conditions. This therefore suggests that the conjectured classification of 7-brane sets is not complete.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/). Funded by SCOAP[3].

## 1. Introduction & summary

Interacting UV complete Quantum Field Theories (QFTs) in $d > 4$ are notoriously hard to construct, yet very interesting for a number of reasons, ranging from those more theoretical –the study of the structure of QFT across dimensions, which has proven to be a vantage perspective on strong coupling physics and dualities– to those more applied –such as extra-dimensional models for our world, in many occasions string-inspired. Concentrating on the 5d case, one may try to construct 5d fixed point theories through an $\epsilon$-expansion of asymptotically free gauge theories in $4+\epsilon$ dimensions. Unfortunately, it is unclear whether this approach can be extended all the way to $d = 5$ by setting $\epsilon = 1$.

The situation is, however, very different for supersymmetric 5d QFTs. In [1], using String Theory, Seiberg argued that $SU(2)$ gauge theories with $N_f < 8$ flavor hypermultiplets can be regarded as a certain mass-deformation of strongly coupled non-lagrangian UV Super-Conformal Field Theories (SCFTs) with $E_{N_f+1}$ global symmetry, thus paving the road to the study of 5d QFTs by String Theory methods. Now, 5d SCFTs appear in different guises in String/M theory. One avatar is through compactifications on Calabi-Yau spaces [2]. Another is as low energy descriptions of systems of 5-branes in IIB String Theory [3,4], a scenario on which we shall focus here. However, it is not known whether these approaches allow the construction of all possible 5d SCFTs, nor whether a given SCFT admits a description in both languages.

Systems of 5-branes overlapping in 4+1 dimensions are completely described by their arrangement on a transverse 2d plane where each 5-brane looks like a segment. As these segments can meet and recombine, they form a "web" in the plane. Moreover, we will assume that the external 5-brane legs of the web end on a suitable 7-brane –which in the plane of the web looks like a point–, so that the web "hangs" from a 7-brane set. In this language a 5d SCFT is simply a 5-brane web such that all external legs meet at a point. This immediately suggests that it must be possible to construct and classify 5d SCFTs –or at least ones arising from branes– in a very simple way by simply listing all possible consistent brane webs. This classification program is of obvious interest, and would complement the work initiated in [5–10].

Nevertheless, one quickly realizes that, at least stated as above, the set of all consistent brane webs is hugely redundant, coming from two sources. First, the $SL(2,\mathbb{Z})$ duality of Type IIB String Theory translates into a global equivalence. Second, since the length of the external legs is not a parameter in the 5d QFT, one may imagine reducing this size (i.e., moving the external 7-brane along

* Corresponding author.
*E-mail addresses:* guillermo.arias.tam@gmail.com (G. Arias-Tamargo), hey@maths.ox.ac.uk (Y.-H. He), elli.heyes@city.ac.uk (E. Heyes), edward.hirst@city.ac.uk (E. Hirst), d.rodriguez.gomez@uniovi.es (D. Rodriguez-Gomez).

https://doi.org/10.1016/j.physletb.2022.137376
0370-2693/© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/). Funded by SCOAP[3].

the corresponding leg) and eventually crossing it to the other side. Since 7-branes come with a branch cut for the axiodilaton, rotating it so that it does not cross the web induces an $SL(2,\mathbb{Z})$ transformation of part of the web as well as the creation/annihilation of the appropriate number of 5-branes. This whole process –to which we will refer as a Hanany-Witten (HW) move– results in the equivalence of otherwise seemingly different webs [11].

Thus, coming back to our classification problem, the set of all possible 5d SCFTs arising from brane webs is the set of all consistent branewebs modulo the equivalence relation set by (a) $SL(2,\mathbb{Z})$ duality and (b) HW moves. In principle, implementing this program is straightforward. For instance, one could imagine charting the space of 5d SCFTs by first fixing the number of external legs, then drawing all possible webs and lastly keeping only one representative of each equivalence class under (a) and (b). However, in practice this is very hard, mostly due to the difficulty in implementing the equivalence relations. Thus, our classification problem seems insurmountable, as equivalent pairs need to be identified largely through trial and error. It is at this point where new Machine-Learning (ML) techniques, which came into String Theory in [12–16], may come to the rescue and provide an invaluable tool to accomplish this classification of webs.

Interestingly, since brane webs end on 7-branes, our problem is closely related to that of classifying sets of 7-branes (in fact, for the case of webs with three legs studied in this paper, both problems are equivalent). In turn, this problem, affected by the same ambiguities due to $SL(2,\mathbb{Z})$ and HW moves, was studied long ago in [17]. In that reference it is conjectured that sets of 7-branes are indeed fully characterized by a certain set of quantities (reviewed in Section 2). However, it turns out that this conjecture is not quite correct, as it is possible to construct sets of 7-branes with the same classifiers which are nevertheless not equivalent (we present explicit examples in Section 2). Our novel approach through ML may therefore shed further light on the classification of 7-branes as well.

ML, as a subfield of artificial intelligence, centres itself on the development of predominantly statistical tools to recognise and study patterns in large datasets. Neural Networks (NNs) are a primary tool within supervised ML, whose application on labelled data acts as a non-linear function fitting to map inputs to outputs, both represented as tensors over $\mathbb{Q}$ using decimals. In recent years the advancement of computational power has played perfectly into the hands of these many-parameter techniques, leading to a programme of application of these tools to datasets arising in theoretical physics [18–28] and the relevant mathematics [29–36]. Motivated by this, we initiate the program of applying ML techniques to the classification of 5-brane webs and 5d SCFTs, concentrating on the simplest case of webs with exactly three external legs. The goal will be to teach a Siamese Neural Network (SNN) [37], an architecture designed to determine the similarity of inputs, to recognize webs equivalent under both $SL(2,\mathbb{Z})$ and HW moves (the same architecture was proposed to condense the string landscape in [38]).

In order to teach the computer the (in)equivalence of webs, we constructed a controlled dataset by implementing by hand $SL(2,\mathbb{Z})$ and HW moves on webs known to be inequivalent –as their classifiers following [17] are different. This problem turns out to be too complicated for the computer, which performs as well as random guessing. We obtained much better results by loosening the notion of equivalence. Indeed, by declaring that webs are equivalent if they share the same classifiers –which is a necessary but not sufficient condition for true equivalence– we constructed a dataset used to train and test the SSN on which it performed far better. We stress that the SSN did not explicitly see the classifiers used to construct the dataset. More explicitly, the SSN is only fed raw web data together with the information of equivalence/non-equivalence under the loosened version for training. From that, the web learns from the examples to identify equivalent/non-equivalent pairs in the independent test data under the loosened equivalence relation, and given that we know the ultimate existence of the classifiers discriminating webs –which we actually used to construct the dataset in the first place– it is natural to conclude that the SSN is somehow reconstructing the existence of the classifiers identified in [17]. This is an encouraging conclusion, also supported by topological data analysis, where a degree of clustering in the embeddings of the webs produced by the SSN is observed for the case of the loosened equivalence relation, as shown in Appendix A. In turn, this is to be compared to the failure of the SSN when faced with the problem of identifying equivalent/non-equivalent webs under the real equivalence relation. This lack of success can be turned around and used to infer an interesting lesson about the physics of sets of 7-branes, as it suggests that the full set of classifiers discriminating the true equivalence classes –if existent at all– are much more subtle. This work therefore serves as a prime example of how, in addition to helping us perform difficult computations, *ML also has the power to detect new patterns in mathematical data and show us that our current theories may be incomplete*.

This work initiates the application of ML to the study of branewebs and 5d SCFT's, and it leaves the door open for future development. In particular, it would be very interesting to improve the performance of ML on the full problem using the exact notion of equivalence –rather than the loosened version in terms of equal invariants. This may be achieved through consideration of more complicated architectures, such as graph neural networks or generative adversarial networks. One could also look for more suitable invariants of equivalence using unsupervised learning methods. Furthermore, since this work only considers classifying brane webs with three external legs, which is equivalent to classifying sets of 7-branes, as obvious next step would be to extend this to look at brane webs with more than three legs. Moreover, this work only scratches the surface of applications, as ML may provide new techniques to tackle very interesting problems. Besides the classification problem itself for webs with arbitrarily many external legs, it may be possible to apply ML to the characterization of the different phases of a given theory –such as [39]– or the construction of the corresponding magnetic quivers (see *e.g.* [40,41]).

## 2. Brane web data

Brane webs are configurations of linked $(p,q)$ 5-branes in type IIB String Theory [3,4]. These branes are objects with $p$ units of magnetic charge under the RR 2-form and $q$ units of magnetic charge under the NSNS 2-form of Type IIB String Theory. Supersymmetry requires that all the 5-branes share the $x^0,\ldots,x^4$ directions. Then, D5-branes (which correspond to (1,0) 5-branes) must be extended in the $x^5$ direction and pointlike in $x^6,\ldots,x^9$; NS5-branes (which correspond to (0,1) 5-branes) must be extended along $x^6$ and pointlike in $x^5,x^7,\ldots,x^9$; and general $(p,q)$ 5-branes must be extended along a line in the $(x^5,x^6)$ plane with slope $q/p$ and pointlike in the remaining directions. Moreover, different types of 5-branes can meet and recombine as long as charge conservation is preserved at the junction. Therefore, the resulting system forms a web in the $(x^5,x^6)$ plane. We will be working in the convention where all the charges are incoming at a junction; therefore the condition will be $\sum_{\text{junction}} p_i = \sum_{\text{junction}} q_i = 0$.

We assume that the brane webs end on 7-branes [42]. Indeed, 7-branes can be added without breaking supersymmetry (with some caveats to be discussed below) provided they are pointlike in the plane of the web and extended in the other directions. Since $(p,q)$ 5-branes can end on $[p,q]$ 7-branes, we can always assume our webs with $L$ external legs to end each on the appropriate $L$

external 7-branes.[1] The positions of the 7-branes naturally correspond to the mass-deformations of the 5d SCFT. As the location of a 7-brane along the external leg is not a parameter of the low energy theory, each 7-brane provides in principle one real parameter associated to its position. However, charge conservation and the symmetries of the plane of the web fix 3 such parameters. Thus, for a web with $L$ external legs –*i.e.*, $L$ external 7-branes– there are $L-3$ real mass deformations. A very important feature of 7-branes is that they act as magnetic sources for the string theory axiodilaton, which will be proportional to the complex logarithm $\log(x^5+ix^6)$. The branch cut is understood as a monodromy transformation: when crossing it fields undergo the appropriate action of type IIB $SL(2,\mathbb{Z})$ duality. Since the position of the cut is immaterial, one can always choose to orient the monodromy cut of all 7-branes so that it goes away from the brane web, such that none of the 5-branes cross it.

The presence of 7-branes places extra constraints on the web in order to preserve supersymmetry. Introducing the *self-intersection* $\mathcal{I}$ defined as (we only quote the version with exclusively external 7-branes)

$$\mathcal{I}=\left|\sum_{1\le i<j\le L}\det\begin{pmatrix}p_i & p_j\\ q_i & q_j\end{pmatrix}\right|-\sum_{i=1}^{L}[\gcd(p_i,q_i)]^2\,, \tag{2.1}$$

the condition for supersymmetry is [39,43]

$$\mathcal{I}\ge -2\,. \tag{2.2}$$

Since the dimension of the Coulomb branch –*i.e.* the rank– of the 5d SCFT is related to $\mathcal{I}$ as

$$d_{\rm CB}=\frac{\mathcal{I}+2}{2}\,, \tag{2.3}$$

the supersymmetry condition can be recast in a more physical way as demanding the dimension of the Coulomb branch to be an integer greater than or equal to zero.

Given these ingredients, our goal is to find a classification of 5-brane webs. To establish some order, we imagine fixing the number of external legs $L$ and constructing all webs with $L$ legs. In 5d SCFT language, this corresponds to constructing all possible 5d SCFTs which admit a fixed number of $L-3$ mass-deformations. Of course, there will be infinitely many such webs, since for a given $L$, there can be theories of arbitrary rank. However, even after fixing $d_{\rm CB}$, there will be infinitely many possible webs consistent with the specified data. The reason is that, as mentioned in the main text, the space of webs is hugely redundant[2]:

1. **S-duality of type IIB string theory:** We can act on the 5-branes and 7-branes of our setup with an $SL(2,\mathbb{Z})$ transformation. This amounts to changing all the $p$ and $q$ charges by

$$\begin{pmatrix}p_i\\ q_i\end{pmatrix}\mapsto\begin{pmatrix}p_i'\\ q_i'\end{pmatrix}=\begin{pmatrix}a & b\\ c & d\end{pmatrix}\begin{pmatrix}p_i\\ q_i\end{pmatrix}\,, \tag{2.4}$$

with $a,b,c,d\in\mathbb{Z}$ and $ad-cb=1$. Note that the charges of all the 7-branes are acted upon by the same $SL(2,\mathbb{Z})$ matrix at the same time (*i.e.* $\forall\, i=1\ldots,L$).

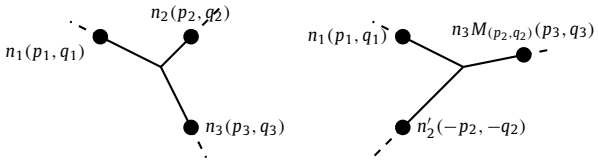


**Fig. 1.** Two brane webs equivalent by a HW move. The 5-branes are depicted as the lines in the $(x^5,x^6)$ plane; the dots represent the 7-branes; and the dashed lines the corresponding monodromy cut. In going from the web on the left to the one on the right, we have moved the 7-brane with label $i=2$ to the bottom of the junction. Sweeping with the monodromy clockwise changes the charges of the $[p_3,q_3]$ 7-brane by the action of $M_{(p_2,q_2)}$. After the transition, the new number of 5-branes $n_2'$ that hang from the $[p_2,q_2]$ 7-brane is determined by charge conservation. Note that we have changed the signs of the charges of the second 7-brane so that all the charges are ingoing.

2. **Hanany-Witten moves:** As mentioned, the position of the 7-brane along the external 5-brane has no effect on the low energy theory. This includes the possibility of moving the 7-brane to the other side of the 5-brane junction. If this is the case, two things happen: first, when the 7-brane crosses the junction, several 5-branes are created or annihilated; this is the well known Hanany-Witten transition. Second, in order to arrive back at the situation where the monodromy cut of the 7-brane goes away from the web without meeting any of the other objects, we need to sweep it 180 degrees. In doing this, the monodromy will act on one of the legs which we have not moved via an $SL(2,\mathbb{Z})$ transformation (see Fig. 1). The action of the monodromy of a $[p,q]$ 7-brane, if we sweep it clockwise, is given by

$$M_{(p,q)}=\begin{pmatrix}1-pq & p^2\\ -q^2 & 1+pq\end{pmatrix}\,, \tag{2.5}$$

and by $M_{(p,q)}^{-1}$ if we sweep it anti-clockwise. Note that in this case only one of the other external legs of the web is undergoing an $SL(2,\mathbb{Z})$ transformation.

Thus, in order to achieve our goal of classifying 5-brane webs we need to further mod out the possible webs with fixed $L$ by $SL(2,\mathbb{Z})$ and HW moves. However, while if given a web it is easy to construct many equivalent webs, the reverse problem of finding whether or not two given webs are equivalent is in general very difficult.

For the purposes of this work, the data specifying a brane web can be encoded in a *web matrix*, which lists the charges of the external $[p_i,q_i]$ seven-branes, as well as the number $n_i$ of five-branes hanging from each of them.

$$W=\begin{pmatrix}n_1p_1 & n_2p_2 & \cdots & n_Lp_L\\ n_1q_1 & n_2q_2 & \cdots & n_Lq_L\end{pmatrix}\,, \tag{2.6}$$

where $L$ is the number of external legs of the web, and we work with the convention that all the charges of the five-branes are ingoing and ordered anticlockwise around the junction.

At this point it is very interesting to note that the problem of classifying branewebs contains as a sub-problem that of classifying sets of 7-branes, which has a long history in the String Theory literature (and has ramifications to the mathematics of $SL(2,\mathbb{Z})$). Indeed, given that a necessary condition (though not sufficient, as more than one web can in principle be hung from the same 7-brane set) for two webs to be equivalent is that their respective sets of external 7-branes are equivalent; as a partial step, we could consider the external 7-branes alone, which are actually subject to the same $SL(2,\mathbb{Z})$ and HW ambiguities. This problem was studied in [17], where it was conjectured that inequivalent sets of $L$

[1] When labelling 5-,7-branes we assume $p,q$ to be coprime. Otherwise, $(p,q)=\gcd(p,q)\,(p',q')$, which represents $\gcd(p,q)$ 5-branes of type $(p',q')=(p/\gcd(p,q),q/\gcd(p,q))$ –and the same for 7-branes. The number of branes is often called *multiplicity*.

[2] Of course, there is a more mundane equivalence under relabelling of the legs. While it is not difficult to take care of this problem, we shall also use ML tools to account for it.

7-branes are characterized by the *total monodromy* of the web, defined as the product of the monodromies of all the 7-branes,

$$M_{tot} = M_{(p_1,q_1)} M_{(p_2,q_2)} M_{(p_3,q_3)} \,, \tag{2.7}$$

and the *asymptotic charge invariant* $\ell$, defined as

$$\ell = \gcd \left\{ \det \begin{pmatrix} p_i & p_j \\ q_i & q_j \end{pmatrix} , \forall i, j \right\} . \tag{2.8}$$

There are cases, however, where webs with the same classifiers give rise to different low energy theories. For instance, take the webs following 3-leg webs

$$W_1 = \begin{pmatrix} -4 & 3 & 1 \\ 18 & -18 & 0 \end{pmatrix} ,$$
$$W_2 = \begin{pmatrix} 3 & -4 & 1 \\ 18 & -18 & 0 \end{pmatrix} ,$$

corresponding to rank 2 theories. Even though these webs have the same classifiers – that is, the same number of external 7-branes, the same $\ell$ and the same total monodromy up to $SL(2,\mathbb{Z})$ –, there is no combination of $SL(2,\mathbb{Z})$/HW moves transforming one web into the other. In this particular case, $W_{1,2}$ are related by inversion of the y-axis in the plane of the web corresponding to parity in the 5d SCFT, therefore they are physically equivalent, yet by our definition the webs are not strongly equivalent. We choose to define strong equivalence without parity because this then also tells us about the classification of sets of 7-branes, in addition to 5d SCFTs. However, we believe that this phenomenon of brane webs being inequivalent despite having equal classifiers is generic. For example, consider the following 4-leg webs:

$$W_{(c)} = \begin{pmatrix} -3 & 0 & 2 & 1 \\ 1 & -1 & -1 & 1 \end{pmatrix} ,$$
$$W_{(e)} = \begin{pmatrix} -3 & 1 & 1 & 1 \\ 1 & -2 & 0 & 1 \end{pmatrix} ,$$

belonging to different theories (c) and e) in Figure 3 in [44]). Although it was already known that these webs correspond to different theories, the observation that they share the same classifiers is new and therefore stands as a valid counter example to the classification of equivalent brane webs via invariants. Thus, generically, the set of classifiers signalled in [17] form a necessary condition for two 7-branes to be equivalent (instead of a sufficient condition).

Since this paper serves as a proof of concept of the applicability of ML techniques to the classification of branewebs and 5d SCFTs, we restrict to the simplest case of $L = 3$. In this case charge conservation is enough to guarantee that, given a set of three 7-branes, only a unique web can be hung –up to the trivial possibility of considering multiple copies of it. Thus, for three-leg webs, their classification coincides as well with that of sets of three 7-branes.[3]

With this in mind, we define the following two notions of equivalent webs:

- **Strong equivalence:** Two webs are strongly equivalent if they can be transformed into each other by means of any combination of $SL(2,\mathbb{Z})$ and HW moves.
- **Weak equivalence:** Two webs are weakly equivalent if they have the same invariants (2.1), (2.7) (mod $SL(2,\mathbb{Z})$) and (2.8). Of course, if two webs are strongly equivalent they are also weakly equivalent.

Our goal is to teach a Neural Network to distinguish webs according to these two notions, by means of exposing it to big sets of webs where such equivalence is known. To begin with, note that webs with three external legs are described by 9 variables: $p_1, p_2, p_3, q_1, q_2, q_3, n_1, n_2, n_3$, that define the web matrix $W$ (2.6). These variables must satisfy the following three conditions: (1) $(p_i, q_i)$ must be coprime; (2) $\sum p_i = \sum q_i = 0$; and (3) the self-intersection (2.1) satisfies $\mathcal{I} \geq -2$. We then define the following datasets:

- **X**: we generate all possible $W$, for $p_i, q_i \in [-3, 3]$ and $n_i \in [1, 3]$, such that (1), (2) and (3) are satisfied. For each web, $W$, we compute $M, \ell$, and rank, and using this information we sort the webs into equivalence classes, of which there are 14, under the weak equivalence defined above. We take 48 webs[4] from each class to produce a total dataset $\mathbf{X} = \cup_{I=1}^{14} \mathbf{X}_I$ of 672 webs.[5] Members from different groups are necessarily inequivalent but members of the same group may also be inequivalent.
- **Y**: to create our second dataset we take one member from each group $\mathbf{X}_I$ –which are surely inequivalent– and perform a combination of HW and $SL(2,\mathbb{Z})$ moves and shuffle the order of the columns in the web matrices to generate sets, $\mathbf{Y}_I$, $I = 1, ..., 14$, where again $\#(\mathbf{Y}_I) = 48$, of equivalent web matrices. Combining these sets produces a total dataset $\mathbf{Y} = \cup_{I=1}^{14} \mathbf{Y}_I$ of 672 webs, where webs $W^i, W^j$ are equivalent (in the strong sense as defined above) if they belong to the same class $\mathbf{Y}_I$ and inequivalent otherwise.

## 3. Machine learning

Our goal is to see whether, given two webs $W^i, W^j$, a Siamese Neural Network (SNN) is able to determine whether or not they are equivalent. As the name suggests, SNNs consist of two or more identical sub-networks that share the same parameters, weights, and biases. These sub-networks output feature vectors (embeddings) of the inputs that are fed to a loss function, which calculates the similarity between the inputs. Here we train an SNN that takes as input the $2 \times 3$ web matrices $W^i$ and produces 10-dimensional embeddings, $\mathbf{x}^i = (x^i_1, ..., x^i_{10}) \in \mathbb{R}^{10}$ (a heuristically optimal choice), such that if two webs $W^i, W^j$ are equivalent the squared Euclidean distance between their embeddings, $d(\mathbf{x}^1, \mathbf{x}^2) = \sum_{\mu=1}^{10} (x^i_\mu - x^j_\mu)^2$, is less than some threshold. We train our network to minimise the triplet loss (B.2) using the Adam optimisation algorithm with triplet loss parameter $\alpha = 5$ and a learning rate of 0.0001. The triplet loss requires the input training data to be given in triplets $(A, P, N)$, where $A$ (anchor) is the reference web, $P$ (positive) an equivalent web, and $N$ (negative) a non-equivalent web. The distance between the anchor embedding and positive embedding is minimised while the distance between the anchor embedding and negative embedding is maximised. The network is trained on triplets generated from 80% of the data over 5 epochs using a batch size of 256 and 1000 steps per epoch. Batch size, epochs and steps per epoch were chosen from running a grid search algorithm on these hyperparameters. After training, we generate embeddings for the webs in the remaining 20% using the

[3] Restricting to webs with three legs has another technical advantage. Strictly speaking, the supersymmetry condition (2.2) is only valid for irreducible junctions, which are those which allow realisation of the full space of $L - 3$ *a priori* possible mass-deformations [39]. This represents an issue (whose nuances are currently not fully understood) which is not present for webs with $L = 3$.

[4] We take 48 webs from each class so that each set $\mathbf{X}_I$ is of equal size, where 48 is chosen as it is the size of the smallest equivalence class.

[5] Both the weak equivalence and strong equivalence datasets contain 672 webs which is small in ML terms, however, the SNN was trained on triplets taken from these datsets and so the size of the training data was actually much larger ($> 100,000$).

**Table 1**
Accuracy and MCC scores of SNN pairwise equivalence predictions on the test sets.

| | Accuracy | MCC |
|---|---|---|
| $\mathbf{X}_1 \cup \mathbf{X}_2$ | $1.0000 \pm 0.0000$ | $1.0000 \pm 0.0000$ |
| $\mathbf{Y}_1 \cup \mathbf{Y}_2$ | $0.5008 \pm 0.0139$ | $0.0025 \pm 0.0324$ |
| $\mathbf{X}$ | $0.7692 \pm 0.0997$ | $0.5523 \pm 0.2079$ |
| $\mathbf{Y}$ | $0.4864 \pm 0.0637$ | $-0.0268 \pm 0.1305$ |

**Table 2**
Rand index scores of $k$-means clustering of SNN web embeddings on the full datasets.

| | Rand Index |
|---|---|
| $\mathbf{X}_1 \cup \mathbf{X}_2$ | $1.0000 \pm 0.0000$ |
| $\mathbf{Y}_1 \cup \mathbf{Y}_2$ | $0.5031 \pm 0.0072$ |
| $\mathbf{X}$ | $0.8905 \pm 0.0053$ |
| $\mathbf{Y}$ | $0.3229 \pm 0.0456$ |

trained sub-network and compute the squared distance between the embeddings of all pairs in this set. If the distance is less than 1, the chosen threshold value, we determine the pair to be equivalent and inequivalent otherwise. The metrics used to measure the performance of the network are accuracy (B.4) and Matthew's Correlation Coefficient (MCC) (B.5) of the pairwise equivalence predictions. We use 5-fold cross validation, whereby the network is independently trained 5 times on different 80% partitions of the data, and tested on the remaining 20%, such that the union of the test sets gives the full dataset. Finally, on the full dataset of embeddings we applying $k$-means clustering to cluster the webs into equivalent groups, and comparing this to the true clustering we compute the Rand Index (RI) score. For more details on the ML methods used, see Appendix B.

The computations in this paper were carried out in `python` with the use of `Tensorflow` [45] and `Scikit-Learn` [46]. The SNN was built using code from [47]. Coding scripts and data can be found at this work's corresponding GitHub: https://github.com/elliheyes/MLBraneWebs.git.

### 3.1. Weak equivalence via invariants

We begin by considering the simplest case of just two weak equivalence groups, $\mathbf{X}_1$ and $\mathbf{X}_2$, and train our SNN to determine whether a pair of webs from the set $\mathbf{X}_1 \cup \mathbf{X}_2$ are weakly equivalent, meaning they belong to the same group. We note that the choice of $\mathbf{X}_1$ and $\mathbf{X}_2$ is arbitrary, in that we could have chosen any two groups, but we note that the results are similar in all cases and so it suffices to use this example. The accuracy and MCC scores of the pairwise equivalence predictions made by the SNN from the web matrices in the test set are given in Table 1. Amazingly we see that after training on 80% of $\mathbf{X}_1 \cup \mathbf{X}_2$ the network determines weak equivalence with 100% accuracy on the remaining 20%. The Rand Index (RI) score in Table 2 also shows that the $k$-means clusters agree completely with the true clusters. We visualise this result in Fig. 4, where the 10-dimensional output embeddings of the webs are reduced to 2-dimensions by t-SNE (t-distributed stochastic neighbour embedding [48], for an explanation see Appendix B). The webs group together into two distinct weakly equivalent clusters.

Motivated by this result we extend the investigation to consider the full dataset $\mathbf{X}$ of 14 classes. We observe from the results in Tables 1 and 2 that the performance scores have dropped to more modest values but the network is still performing significantly better than random guessing. The t-SNE plot in Fig. 6 also supports this result where we still see clusters for the weak equivalence classes, but the clusters are less distint. The mean squared

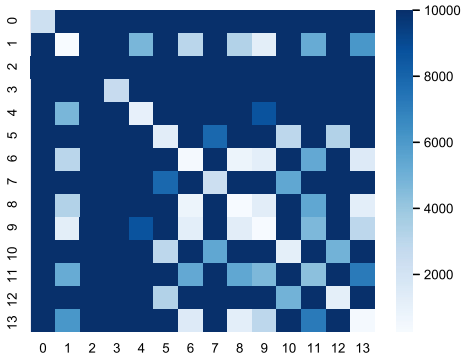


**Fig. 2.** Mean squared Euclidean distances between webs in $\mathbf{X} = \cup_{I=1}^{I=14} \mathbf{X}_I$.

Euclidean distances between the embeddings of each pair of webs from the 14 classes are displayed in Fig. 2. We see that the distances along the diagonal (i.e. for webs belonging to the same class) are close to 0, while distances between webs from different classes are often much larger than 0, which is the desired result. This plot also helps us to identify on which classes the network performs best.

### 3.2. Strong equivalence via generation

In the previous subsection we saw that an SNN is capable of identifying pairs of webs that satisfy the necessary condition for equivalence - that rank, $\ell$, and $M$, up to $SL(2,\mathbb{Z})$, are all equal. In this subsection we repeat the same procedure but consider strong equivalence, and hence the dataset $\mathbf{Y}$. We begin as before with the simplest case of just two strong equivalence classes $\mathbf{Y}_1$ and $\mathbf{Y}_2$ from $\mathbf{Y}$, and train the SNN to distinguish pairs of webs that belong to the same class $\mathbf{Y}_I$. We again note that the results with any other two classes, $\mathbf{Y}_i$ and $\mathbf{Y}_j$, are similar. As the scores in Table 1 show, the network fails to give any improvement on a random guess (which held for any choice of the 2 classes to compare). The RI score in Table 2 also supports this conclusion. It can be seen from the t-SNE plot in Fig. 5 that the embeddings of $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are mixed together and completely indistinguishable. Despite this poor result, we extend the investigation, as we did in Section 3.1, to the full dataset $\mathbf{Y}$. Mean distances between the embeddings of each pair of webs are displayed in Fig. 3. Comparing Figs. 2 and 3 we see that strongly equivalent webs are mapped relatively further away by the SNN than weakly equivalent webs, shown by darker elements along the diagonal, and non strongly equivalent webs are mapped relatively closer together than non weakly equivalent webs, shown by lighter elements off the diagonal. The scores in Tables 1 and 2 show that the SNN is no better at determining equivalence of web pairs from $\mathbf{Y}$ than $\mathbf{Y}_1 \cup \mathbf{Y}_2$ and is still no better than random guessing. This is supported by the t-SNE plot in Fig. 7 where we see no clusters for the classes $\mathbf{Y}_I$.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Elli Heyes reports a relationship with City University of London that includes: funding grants. Edward Hirst reports a relationship with Science and Technology Facilities Council that includes: funding grants. Yang-Hui He reports a relationship with Science and Technology Facilities Council that includes: funding grants. Guillermo Arias-Tamargo reports a relationship with Spanish Government that includes: funding grants. Diego Rodriguez-Gomez reports a relationship with Spanish Government that includes: fund-

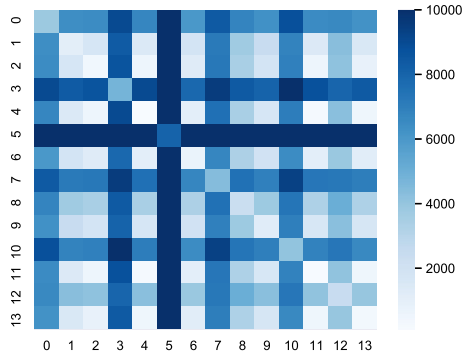


**Fig. 3.** Mean squared Euclidean distances between webs in $\mathbf{Y} = \cup_{I=1}^{I=14} \mathbf{Y}_I$.

ing grants. Diego Rodriguez-Gomez reports a relationship with Principado de Asturias that includes: funding grants. Guillermo Arias-Tamargo reports a relationship with Principado de Asturias that includes: funding grants. None.

## Acknowledgements

GAT is supported by the Spanish government scholarship MCIU-19-FPU18/02221. YHH would like to thank STFC for grant ST/J00037X/2. E. Heyes would like to thank City, University of London, for a PhD studentship and the States of Jersey for a postgraduate grant. E. Hirst would like to thank STFC for a PhD studentship. DRG and GAT are partly supported by Spanish national grant MINECO-16-FPA2015- 63667-P as well as the Principado de Asturias grant FC-GRUPIN-IDI/2018/000174. DRG would like to thank O. Bergman for conversations.

## Appendix A. Topological data analysis

***Raw web data*** To analyse the dataset of web information, both pre and post to embedding with the SNN's base model we turn to a tool from Topological Data Analysis (TDA). *Persistent homology* provides useful feature analysis of higher-dimensional datasets that cannot be feasibly plotted. The process plots the webs as points in their respective higher $d$-dimensional spaces, then creates a filtration of Vietoris-Rips complexes, where $\lambda$ points are connected in a $\lambda$-simplex if their $d$-dimensional balls drawn centred on each point intersect (up to $\lambda_{max} = d$). The filtration of complexes is then built during the process of the ball radii being continuously increased from 0 to $\infty$.

Throughout the filtration, $H_0$ features are all the points, born at radius 0, which then die as they become part of a connected component; *i.e.* as two components combine the larger feature takes priority and survives whilst the smaller feature (usually just a point) dies. The $H_1$ features, plotted on the same persistence diagrams, effectively represent 1d loop structures in the dataset. These are born as points connect to form a loop which is not a boundary of a union of 2d simplices in the complex. The feature then dies as the loop is filled in by the respective 2d simplices. Persistence diagrams plot the features as (birth, death) pairs, and were computed with the use of the *ripser* library [49] in `python`. For other uses of persistent homology in theoretical physics see [35,50,51].

Each web in the raw web datasets **X** and **Y** amounts to the contents of the web matrix (2.6) with $L = 3$. These 6 integers are plotted in $\mathbb{R}^6$ for the persistent homology analysis. Since datapoints are restricted to the integer lattice this translates to a grid-like distribution of features in the persistence diagrams, as only at specific radii can balls intersect and change the complex.

The analysis is performed for both datasets of 672 web datapoints, **X** in Fig. 8, and **Y** in Fig. 9. The HW moves in the generation process for **Y** lead to much larger web matrix entries, as reflected in the larger scales and finer grid structure in 9. For the **X** data the roughly uniform distribution of $H_0$ features indicates points are uniformly distributed in the space, as expected from the exhaustive generation procedure over the search space. Conversely the **Y** data has a slightly larger gap in the line (~125-145, note the much larger scale here) indicating there are clusters of points further from the main cluster, likely a result of some HW moves jumping datapoints away from the bulk.

Both datasets have $H_1$ features close to the diagonal, behaviour typical of noise; since there are no features far from the diagonal there is not a significant loop structure in the data which would otherwise indicate regions either omitted from the sampling or not physically plausible.

***SNN-embedded web data*** To further analyse the success of the SNN embedding procedure, beyond the t-SNE plots, we additionally use persistent homology to examine the embedded representations of the webs. This embedded web data is the result of the SNN's base model mapping the 6-parameter web matrix data into the $\mathbb{R}^{10}$ embedding space, where the aim of the SNN is to create an embedding model which separates inequivalent webs into their respective clusters.

The $H_0$ analysis for the embedded **X** data in Fig. 10 shows a continuous stream of features indicating points merging together to form independent simplices, then the gap (~1.10-1.20) followed by a collection of features closer together indicates the separate clusters for each of the web classes combining together to finish the filtration. Since the features are relatively close they represent more symmetrically distributed clusters, nice behaviour since a priori no equivalence classes should be especially more related than others. This behaviour supports the success of the SNN embedding to separate the webs into clusters based on the equivalence used for training. The $H_0$ features for the embedded **Y** data, in Fig. 11, has similar behaviour, however the cluster separation is less uniform, indicated by the less consistent line where clusters aren't combining smoothly and then only a couple of features are separated from the main line ($> 8$), therefore some of the classes' clusters merge earlier, and are not as well separated as in the embedded **X** data, making this classification worse, and hence supporting the poorer learning results observed.

For both embedded datasets the $H_1$ features lie close to the diagonal, again indicating a lack of significant loop structures in the data clouds such that the clustering performs as expected. Relatively, the embedded **X** data has many more higher birth features, likely a consequence of the better separated clusters combining later in the filtration.

## Appendix B. Machine learning

### B.1. Siamese neural networks

Siamese Neural Networks (SNNs), first introduced in 1993 [37] to solve signature verification, are neural network architectures, made up of two or more identical sub-networks, that determine the similarity of inputs. The sub-networks $f_w$ map elements of a dataset $\mathcal{D}$ to $\mathbb{R}^d$, where $w$ denotes the weights and biases of the network. The goal is to train the network so that similar elements of $\mathcal{D}$ are mapped close together in $\mathbb{R}^d$, and dissimilar elements are mapped far apart. The $w$ are determined by extremising a loss function that is dependent on the squared Euclidean distance between the embeddings:

$$d_w(W_1, W_2) \equiv (f_w(W_1) - f_w(W_2))^2 \tag{B.1}$$

**Fig. 4.** t-SNE $\mathbf{X}_1 \cup \mathbf{X}_2$.

**Fig. 5.** t-SNE $\mathbf{Y}_1 \cup \mathbf{Y}_2$.

**Fig. 6.** t-SNE $\mathbf{X}$.

**Fig. 7.** t-SNE $\mathbf{Y}$.

t-SNE plots of the 10-dimensional web embeddings generated by the SNN of webs from the respectively labelled datasets, reduced to 2 dimensions.

where $f_w(W)$ denotes the embedding of input $W$. In this work we adopt the triplet loss function described below. This loss is then interpreted as our desired similarity score.

Since we are using the triplet loss our network is made up of three identical sub-networks that produce embeddings for the anchor, positive and negative input. Each of these sub-networks consists of a 2 dimensional convolution layer, with 8 filters, a kernel size of 2, and ReLU activation functions, followed by a dense layer, with 50 neurons and ReLU activation, and the output layer is a dense layer with 10 neurons. The output of these three sub-networks is fed to an external output layer that computes the triplet loss. A diagram of the network is represented in Fig. 12

### *B.2. Triplet loss*

The triplet loss takes as input a triple: anchor (A), positive (P), and negative (N), where the anchor and positive input are similar (*i.e.* equivalent webs) and the anchor and negative input are not. We want the distance between the anchor embedding and the positive embedding to be no larger than the distance between the anchor embedding and the negative embedding, that is:

$$||f_w(A) - f_w(P)||^2 \leq ||f_w(A) - f_w(N)||^2 \,.$$

Equivalently, we could write

$$||f_w(A) - f_w(P)||^2 - ||f_w(A) - f_w(N)||^2 \leq 0 \,.$$

This inequality is satisfied if the $f_w$ always outputs 0 (or another constant value), in other words if the embeddings are all equal. To prevent the network from doing this we modify the objective so that the difference needs to be strictly less than zero by introducing a hyperparameter $\alpha \in \mathbb{R}_{>0}$:

$$||f_w(A) - f_w(P)||^2 - ||f_w(A) - f_w(N)||^2 + \alpha \leq 0 \,.$$

The triplet loss function is then defined as:

$$\mathcal{L}(A, P, N) = \max(||f_w(A) - f_w(P)||^2 - ||f_w(A) - f_w(N)||^2 + \alpha, 0) \,. \tag{B.2}$$

This is the loss of a single triplet, the overall loss function of our network is computed as the mean of these individual losses over $\mu$ triplets:

$$J = \frac{\sum_{i=1}^{\mu} L(A^{(i)}, P^{(i)}, N^{(i)})}{\mu} \,. \tag{B.3}$$

### *B.3. Evaluation metrics*

The metrics used to evaluate the network performance, classifying equivalence of web pairs, were accuracy and Matthew's Correlation Coefficient (MCC):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \in [0, 1] \tag{B.4}$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}} \in [-1, 1] \tag{B.5}$$

where $TP, TN, FP, FN$ denote the number of true positive, true negative, false positive and false negative predictions respectively. Accuracy reflects the proportion of test web pairs whose equivalence is correctly determined. The MCC is similar to the Pear-

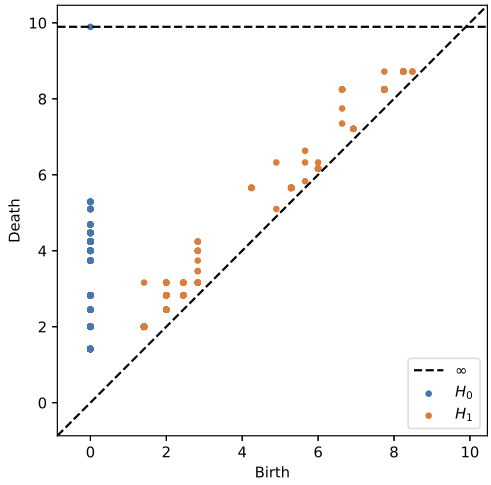


**Fig. 8.** $(n_i p_i, n_i q_i)$ **X** data.

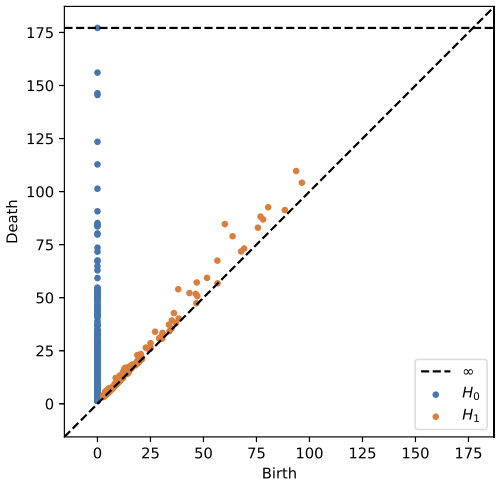


**Fig. 9.** $(n_i p_i, n_i q_i)$ **Y** data.

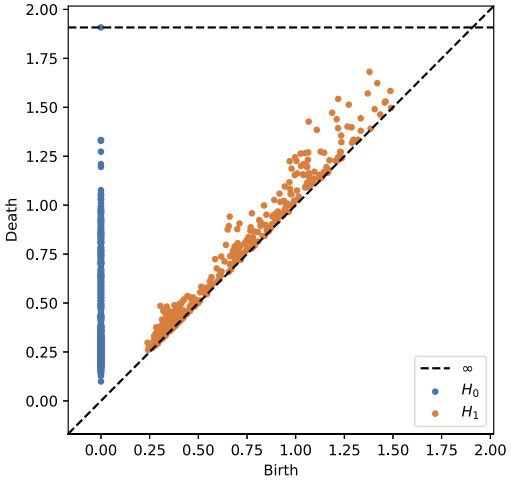


**Fig. 10.** SNN embedded **X** data.

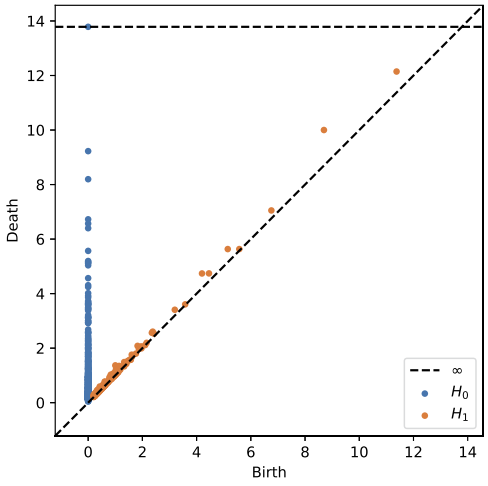


**Fig. 11.** SNN embedded **Y** data.

Persistence diagrams for $H_0$ and $H_1$ on the web $(n_i p_i, n_i q_i)$ data and SNN base model embedded data, for datasets **X** and **Y** respectively.

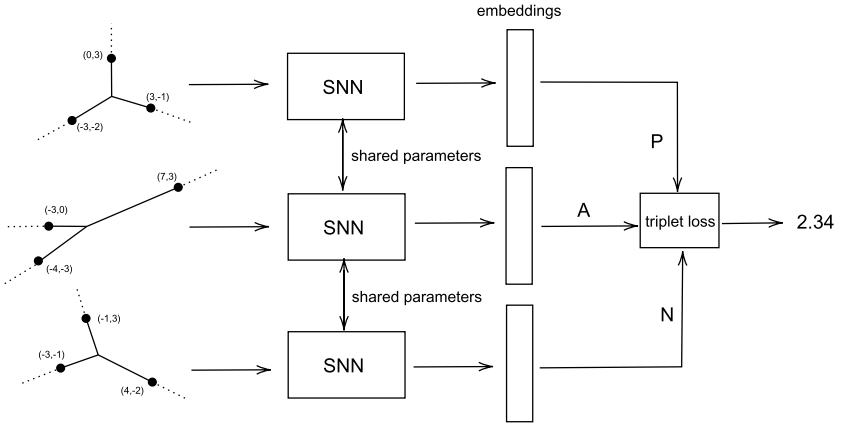


**Fig. 12.** Diagram of Siamese Neural Netowrk architecture using triplet loss.

son correlation coefficient, where a score of 1 indicates complete agreement between predictions and truth, 0 indicates that the predictions are no better than random guessing, and -1 indicates complete disagreement between the predicted and true equivalence.

### *B.4. K-means clustering*

*k*-means clustering sorts datapoints into *k* clusters, where each datapoint belongs to the cluster with the nearest centroid. The objective of the *k*-means algorithm is to minimise the sum of squared

distances between the centroid and the datapoints. Formally, given a set of datapoints $(\mathbf{x}_1, ..., \mathbf{x}_n)$, $k$-means clustering aims to partition the datapoints into $k$ sets $\mathbf{S} = (S_1, ..., S_k)$ so as to minimise

$$\sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} ||\mathbf{x} - \mu_i||^2 \tag{B.6}$$

where $\mu_i$ is the centroid of the set $S_i$. The first step is to choose the value for $k$, i.e. the number of clusters. Then the algorithm starts by randomly selecting $k$ points from the data to be centroids and assigning each datapoint to the cluster with the closest centroid. Then the new centroid of each cluster is computed and the datapoints are again assigned to the nearest cluster. This iterative two step process of computing the new centroid and assigning datapoints to clusters is repeated until the centroids do not change their position.

### B.5. Rand index

The *Rand Index* is a measure of similarity between two partitions. It considers all pairs of elements, counting pairs where both elements are assigned to the same subset, or different subsets, across the partitions; used here for comparison of the predicted and true partitions. Let $X = \{x_1, ..., x_n\}$ be a set of $n$ elements and $A = \{A_1, ..., A_r\}$, $B = \{B_1, ..., B_s\}$ be two partitions of $X$ into $r$ and $s$ subsets respectively. Then the Rand index, $R$, is given by

$$R = \frac{a+b}{a+b+c+d} \tag{B.7}$$

where

- $a$ is the number of pairs of elements in $X$ that are in the same subset in $A$ and the same subset in $B$.
- $b$ is the number of pairs of elements in $X$ that are in different subsets in $A$ and different subsets in $B$.
- $c$ is the number of pairs of elements in $X$ that are in the same subset in $A$ and different subsets in $B$.
- $d$ is the number of pairs of elements in $X$ that are in different subsets in $A$ and the same subset in $B$.

Therefore, $R$ exists in the range [0,1] where 1 corresponds to a perfect agreement between the partitions.

### B.6. t-distributed stochastic neighbour embedding (t-SNE)

t-distributed stochastic neighbour embedding (t-SNE) is a popular method, introduced in [48], to reduce high-dimensional data by embedding it in a low-dimensional space. It calculates similarities between datapoints in the high dimensional space and in the low dimensional space and tries to minimise the divergence between the two.

We can break t-SNE down into three steps:

1. Step 1: Given a set of $N$ points, $\mathbf{x}_1, ..., \mathbf{x}_N$ in high dimensional space, measure the similarities between points $\mathbf{x}_i$. The similarity of datapoint $\mathbf{x}_i$ to datapoint $\mathbf{x}_j$ is the conditional probability, $p_{j|i}$, that $\mathbf{x}_i$ would pick $\mathbf{x}_j$ as its neighbour. This probability is proportional to the probability density under a Gaussian centred at $x_i$.

$$p_{j|i} = \begin{cases} \frac{\exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||\mathbf{x}_i - \mathbf{x}_k||^2/2\sigma_i^2)}, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases} \tag{B.8}$$

where $\sigma_i$ are the standard deviations. If we take two points $\mathbf{x}_i, \mathbf{x}_j$, with $i \neq j$, then the values of $p_{i|j}$ and $p_{j|i}$ will be different. Therefore we define:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \tag{B.9}$$

2. Step 2: In the low dimensional space compute the similarity measures between datapoints, but instead of using a Gaussian distribution use a Student's t-distribution with one degree of freedom.

$$q_{ij} = \begin{cases} \frac{(1+||\mathbf{y}_i - \mathbf{y}_j||^2)^{-1}}{\sum_{k \neq l} (1+||\mathbf{y}_k - \mathbf{y}_l||^2)^{-1}}, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases} \tag{B.10}$$

3. Step 3: Compute the Kullback-Liebler divergence between the two probability distributions $p_{ij}$ and $q_{ij}$.

$$D_{KL}(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{B.11}$$

This divergence is minimised with respect to the datapoints $\mathbf{y}_i$ using gradient descent.

t-SNE has a hyper-parameter called perplexity, derived from $\sigma_i$, which is roughly a guess of the number of close neighbours each datapoint has and can have large effects on the resulting plot. Typical values of perplexity are in the range [5, 50]. In our use of t-SNE we use perplexity equal to 48 which is the true number of webs in each class.

## References


[1] N. Seiberg, Five-dimensional SUSY field theories, nontrivial fixed points and string dynamics, Phys. Lett. B 388 (1996) 753–760, arXiv:hep-th/9608111.
[2] D.R. Morrison, N. Seiberg, Extremal transitions and five-dimensional supersymmetric field theories, Nucl. Phys. B 483 (1997) 229–247, arXiv:hep-th/9609070.
[3] O. Aharony, A. Hanany, Branes, superpotentials and superconformal fixed points, Nucl. Phys. B 504 (1997) 239–271, arXiv:hep-th/9704170.
[4] O. Aharony, A. Hanany, B. Kol, Webs of (p,q) five-branes, five-dimensional field theories and grid diagrams, J. High Energy Phys. 01 (1998) 002, arXiv:hep-th/9710116.
[5] P. Jefferson, H.-C. Kim, C. Vafa, G. Zafrir, Towards classification of 5d SCFTs: single gauge node, arXiv:1705.05836 [hep-th].
[6] L. Bhardwaj, G. Zafrir, Classification of 5d $\mathcal{N}$ = 1 gauge theories, J. High Energy Phys. 12 (2020) 099, arXiv:2003.04333 [hep-th].
[7] F. Apruzzi, C. Lawrie, L. Lin, S. Schäfer-Nameki, Y.-N. Wang, Fibers add flavor, part I: classification of 5d SCFTs, flavor symmetries and BPS states, J. High Energy Phys. 11 (2019) 068, arXiv:1907.05404 [hep-th].
[8] F. Apruzzi, C. Lawrie, L. Lin, S. Schäfer-Nameki, Y.-N. Wang, Fibers add flavor, part II: 5d SCFTs, gauge theories, and dualities, J. High Energy Phys. 03 (2020) 052, arXiv:1909.09128 [hep-th].
[9] L. Bhardwaj, Flavor symmetry of 5d SCFTs. Part I. General setup, J. High Energy Phys. 09 (2021) 186, arXiv:2010.13230 [hep-th].
[10] L. Bhardwaj, Flavor symmetry of 5*d* SCFTs. Part II. Applications, J. High Energy Phys. 04 (2021) 221, arXiv:2010.13235 [hep-th].
[11] A. Hanany, E. Witten, Type IIB superstrings, BPS monopoles, and three-dimensional gauge dynamics, Nucl. Phys. B 492 (1997) 152–190, arXiv:hep-th/9611230.
[12] Y.-H. He, Deep-learning the landscape, arXiv:1706.02714 [hep-th].
[13] Y.-H. He, Machine-learning the string landscape, Phys. Lett. B 774 (2017) 564–568.
[14] D. Krefl, R.-K. Seong, Machine learning of Calabi-Yau volumes, Phys. Rev. D 96 (6) (2017) 066014, arXiv:1706.03346 [hep-th].
[15] F. Ruehle, Evolving neural networks with genetic algorithms to study the String Landscape, J. High Energy Phys. 08 (2017) 038, arXiv:1706.07024 [hep-th].
[16] J. Carifio, J. Halverson, D. Krioukov, B.D. Nelson, Machine learning in the String Landscape, J. High Energy Phys. 09 (2017) 157, arXiv:1707.00655 [hep-th].
[17] O. DeWolfe, T. Hauer, A. Iqbal, B. Zwiebach, Uncovering the symmetries on [p,q] seven-branes: beyond the Kodaira classification, Adv. Theor. Math. Phys. 3 (1999) 1785–1833, arXiv:hep-th/9812028.
[18] A. Ashmore, Y.-H. He, B.A. Ovrut, Machine learning Calabi–Yau metrics, Fortschr. Phys. 68 (9) (2020) 2000068, arXiv:1910.08605 [hep-th].
[19] M.R. Douglas, S. Lakshminarasimhan, Y. Qi, Numerical Calabi-Yau metrics from holomorphic networks, arXiv:2012.04797 [hep-th].

[20] V. Jejjala, D.K. Mayorga Pena, C. Mishra, Neural network approximations for Calabi-Yau metrics, arXiv:2012.15821 [hep-th].
[21] L.B. Anderson, M. Gerdes, J. Gray, S. Krippendorf, N. Raghuram, F. Ruehle, Moduli-dependent Calabi-Yau and SU(3)-structure metrics from Machine Learning, J. High Energy Phys. 05 (2021) 013, arXiv:2012.04656 [hep-th].
[22] J. Halverson, B. Nelson, F. Ruehle, Branes with brains: exploring string vacua with deep reinforcement learning, J. High Energy Phys. 06 (2019) 003, arXiv:1903.11616 [hep-th].
[23] C.R. Brodie, A. Constantin, R. Deen, A. Lukas, Machine learning line bundle cohomology, Fortschr. Phys. 68 (1) (2020) 1900087, arXiv:1906.08730 [hep-th].
[24] S. Krippendorf, D. Lust, M. Syvaeri, Integrability ex machina, Fortschr. Phys. 69 (7) (2021) 2100057, arXiv:2103.07475 [nlin.SI].
[25] J. Bao, S. Franco, Y.-H. He, E. Hirst, G. Musiker, Y. Xiao, Quiver mutations, Seiberg duality and machine learning, Phys. Rev. D 102 (8) (2020) 086013, arXiv:2006.10783 [hep-th].
[26] J. Bao, Y.-H. He, E. Hirst, J. Hofscheier, A. Kasprzyk, S. Majumder, Hilbert series, machine learning, and applications to physics, Phys. Lett. B 827 (2022) 136966, arXiv:2103.13436 [hep-th].
[27] H.-Y. Chen, Y.-H. He, S. Lal, M.Z. Zaz, Machine learning etudes in conformal field theories, arXiv:2006.16114 [hep-th].
[28] M. Larfors, A. Lukas, F. Ruehle, R. Schneider, Learning size and shape of Calabi-Yau spaces, arXiv:2111.01436 [hep-th].
[29] S. Gukov, J. Halverson, F. Ruehle, P. Sułkowski, Learning to unknot, Mach. Learn. Sci. Technol. 2 (2) (2021) 025035, arXiv:2010.16263 [math.GT].
[30] Y.-H. He, Machine-learning mathematical structures, arXiv:2101.06317 [cs.LG].
[31] Y.-H. He, E. Hirst, T. Peterken, Machine-learning dessins d'enfants: explorations via modular and Seiberg–Witten curves, J. Phys. A 54 (7) (2021) 075401, arXiv:2004.05218 [hep-th].
[32] J. Bao, Y.-H. He, E. Hirst, Neurons on amoebae, arXiv:2106.03695 [math.AG].
[33] J. Bao, Y.-H. He, E. Hirst, J. Hofscheier, A. Kasprzyk, S. Majumder, Polytopes and machine learning, arXiv:2109.09602 [math.CO].
[34] H.-Y. Chen, Y.-H. He, S. Lal, S. Majumder, Machine learning Lie structures & applications to physics, Phys. Lett. B 817 (2021) 136297, arXiv:2011.00871 [hep-th].
[35] D.S. Berman, Y.-H. He, E. Hirst, Machine learning Calabi-Yau hypersurfaces, Phys. Rev. D 105 (6) (2022) 066002, arXiv:2112.06350 [hep-th].
[36] P. Berglund, B. Campbell, V. Jejjala, Machine learning kreuzer-skarke Calabi-Yau threefolds, arXiv:2112.09117 [hep-th].
[37] J. Bromley, J.W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, R. Shah, Signature verification using a siamese time delay neural network, Int. J. Pattern Recognit. Artif. Intell. 7 (1993).
[38] Y.-H. He, S. Lal, M.Z. Zaz, The world in a grain of sand: condensing the string vacuum degeneracy, arXiv:2111.04761 [hep-th].
[39] O. Bergman, D. Rodríguez-Gómez, The Cat's Cradle: deforming the higher rank $E_1$ and $\tilde{E}_1$ theories, J. High Energy Phys. 02 (2021) 122, arXiv:2011.05125 [hep-th].
[40] S. Cabrera, A. Hanany, F. Yagi, Tropical geometry and five dimensional Higgs branches at infinite coupling, J. High Energy Phys. 01 (2019) 068, arXiv:1810.01379 [hep-th].
[41] M. van Beest, A. Bourget, J. Eckhard, S. Schafer-Nameki, (Symplectic) leaves and (5d Higgs) branches in the poly(go)nesian tropical rain forest, J. High Energy Phys. 11 (2020) 124, arXiv:2008.05577 [hep-th].
[42] O. DeWolfe, A. Hanany, A. Iqbal, E. Katz, Five-branes, seven-branes and five-dimensional E(n) field theories, J. High Energy Phys. 03 (1999) 006, arXiv:hep-th/9902179.
[43] A. Iqbal, Selfintersection number of BPS junctions in backgrounds of three-branes and seven-branes, J. High Energy Phys. 10 (1999) 032, arXiv:hep-th/9807117.
[44] V. Saxena, Rank-two 5d SCFTs from M-theory at isolated toric singularities: a systematic study, J. High Energy Phys. 04 (2020) 198, arXiv:1911.09574 [hep-th].
[45] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: large-scale machine learning on heterogeneous systems, Software available from https://www.tensorflow.org/, 2015.
[46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[47] Ekeany, Siamese-network-with-triplet-loss, https://github.com/Ekeany/Siamese-Network-with-Triplet-Loss, 2019.
[48] L. van der Maaten, G.E. Hinton, Visualizing high-dimensional data using t-sne, J. Mach. Learn. Res. 9 (2008) 2579–2605.
[49] C. Tralie, N. Saul, R. Bar-On, Ripser.py: a lean persistent homology library for python, J. Open Sour. Softw. 3 (29) (Sep 2018) 925, https://doi.org/10.21105/joss.00925.
[50] M. Cirafici, Persistent homology and string vacua, J. High Energy Phys. 03 (2016) 045, arXiv:1512.01170 [hep-th].
[51] A. Cole, G. Shiu, Topological data analysis for the string landscape, J. High Energy Phys. 03 (2019) 054, arXiv:1812.06960 [hep-th].