
Learning Symbolic Physics with Graph Networks

Miles D. Cranmer
Princeton University
Princeton, NJ, USA
mcranmer@princeton.edu

Rui Xu
Princeton University
Princeton, NJ, USA
ruix@princeton.edu

Peter Battaglia
DeepMind
London, UK
peterbattaglia@google.com

Shirley Ho
Flatiron Institute
New York City, NY, USA
shirleyho@flatironinstitute.org

Abstract

We introduce an approach for imposing physically motivated inductive biases on graph networks to learn interpretable representations and improved zero-shot generalization. Our experiments show that our graph network models, which implement this inductive bias, can learn message representations equivalent to the true force vector when trained on n-body gravitational and spring-like simulations. We use symbolic regression to fit explicit algebraic equations to our trained model’s message function and recover the symbolic form of Newton’s law of gravitation without prior knowledge. We also show that our model generalizes better at inference time to systems with more bodies than had been experienced during training. Our approach is extensible, in principle, to any unknown interaction law learned by a graph network, and offers a valuable technique for interpreting and inferring explicit causal theories about the world from implicit knowledge captured by deep learning.

1 Introduction

Discovering laws through observation of natural phenomenon is the central challenge of the sciences. Modern deep learning also involves discovering knowledge about the world but focuses mostly on *implicit* knowledge representations, rather than *explicit* and interpretable ones. One reason is that the goal of deep learning is often optimizing test accuracy and learning efficiency in narrowly specified domains, while science seeks causal explanations and general-purpose knowledge across a wide range of phenomena. Here we explore an approach for imposing physically motivated inductive biases on neural networks, training them to predict the dynamics of physical systems and interpreting their learned representations and computations to discover the symbolic physical laws which govern the systems. Moreover, our results also show that this approach improves the generalization performance of the learned models.

The first ingredient in our approach is the “graph network” (GN) [Battaglia et al., 2018], a type of graph neural network [Scarselli et al., 2009, Bronstein et al., 2017, Gilmer et al., 2017], which is effective at learning the dynamics of complex physical systems [Battaglia et al., 2016, Chang et al., 2016, Sanchez-Gonzalez et al., 2018, Mrowca et al., 2018, Li et al., 2018, Kipf et al., 2018]. We impose inductive biases on the architecture and train models with supervised learning to predict the dynamics of 2D and 3D n-body gravitational systems and a hanging string. If the trained models can accurately predict the physical dynamics of held-out test data, we can assume they have discovered some level of general-purpose physical knowledge, which is implicitly encoded in their

weights. Crucially, we recognize that the forms of the graph network’s message and pooling functions have correspondences to the forms of force and superposition in classical mechanics, respectively. The message pooling is what we call a “linearized latent space:” a vector space where latent representations of the interactions between bodies (forces or messages) are linear (summable). By imposing our inductive bias, we encourage the GN’s linearized latent space to match the true one. Some other interesting approaches for learning low-dimensional general dynamical models include Packard et al. [1980], Daniels and Nemenman [2015], and Jaques et al. [2019].

The second ingredient is using symbolic regression — we use eureka from Schmidt and Lipson [2009] — to fit compact algebraic expressions to a set of inputs and messages produced by our trained model. eureka works by randomly combining mathematical building blocks such as mathematical operators, analytic functions, constants, and state variables, and iteratively searches the space of mathematical expressions to find the model that best fits a given dataset. The resulting symbolic expressions are interpretable and readily comparable with physical laws.

The contributions of this paper are:

1. A modified GN with inductive biases that promote learning general-purpose physical laws.
2. Using symbolic regression to extract analytical physical laws from trained neural networks.
3. Improved zero-shot generalization to larger systems than those in training.

2 Model

Graph networks are a type of deep neural network which operates on graph-structured data. The format of the graphs on which GNs operate is defined as 3-tuples¹, $G = (\mathbf{u}, V, E)$, where:

$$\begin{aligned} \mathbf{u} &\in \mathbb{R}^{L^u} \text{ is a global attribute vector of length } L^u, \\ V &= \{\mathbf{v}_i\}_{i=1:N^v} \text{ is a set of node attribute vectors, } \mathbf{v}_i \in \mathbb{R}^{L^v} \text{ of length } L^v, \text{ and} \\ E &= \{(\mathbf{e}_k, r_k, s_k)\}_{k=1:N^e} \text{ is a set of edge attribute vectors, } \mathbf{e}_k \in \mathbb{R}^{L^e} \text{ of length } L^e, \text{ and indices} \\ &\quad r_k, s_k \in \{1:N^v\} \text{ of the “receiver” and “sender” nodes connected by the } k\text{-th edge.} \end{aligned}$$

Our GN implementation is depicted in fig. 1. Note: it does not include global and edge attributes. This GN processes a graph by first computing pairwise interactions, or “messages”, \mathbf{e}'_k , between nodes connected by edges, with a “message function”, $\phi^e : \mathbb{R}^{L^v} \times \mathbb{R}^{L^v} \times \mathbb{R}^{L^e} \rightarrow \mathbb{R}^{L^{e'}}$. Next, the set of messages incident on each i -th receiver node are pooled into $\bar{\mathbf{e}}'_i = \rho^{e \rightarrow v}(\{\mathbf{e}'_k\}_{r_k=i, k=1:N^e})$, where $\bar{\mathbf{e}}'_i \in \mathbb{R}^{L^{e'}}$, and $\rho^{e \rightarrow v}$ is a permutation-invariant operation which can take variable numbers of input vectors, such as elementwise summation. Finally, the pooled messages are used to compute node updates, \mathbf{v}'_i , with a “node update function”, $\phi^v : \mathbb{R}^{L^v} \times \mathbb{R}^{L^{e'}} \rightarrow \mathbb{R}^{L^{v'}}$. Our specific architectural implementation is very similar to the “interaction network” (IN) variant [Battaglia et al., 2016].

The forms of ϕ^e , $\rho^{e \rightarrow v}$, ϕ^v , and the associated input and output attribute vectors have correspondences to Newton’s formulation of classical mechanics, which motivated the original development of INs. The key observation is that \mathbf{e}'_k could learn to correspond to the force vector imposed on the r_k -th body due to its interaction with the s_k -th body. In our examples, the force vector is equal to the derivative of the Lagrangian: $\frac{\delta \mathcal{L}}{\delta \mathbf{q}}$, and this could be generally imposed if one knows $\frac{d}{dt}(\frac{\delta \mathcal{L}}{\delta \dot{\mathbf{q}}})$ and manually integrates the ODE with the output of the graph net. In a general n -body gravitational system in n dimensions, note that the forces are minimally represented in an \mathbb{R}^n vector space. Thus, if $L^{e'} = n$, we exploit the GN’s “linearized latent space” for physical interpretability: we encourage \mathbf{e}'_k to be the force.

We sketch a non-rigorous proof-like demonstration of our hypothesis. Newtonian mechanics prescribes that force vectors, $\mathbf{f}_k \in \mathcal{F}$, can be summed to produce a net force, $\sum_k \mathbf{f}_k = \bar{\mathbf{f}} \in \mathcal{F}$, which can then be used to update the dynamics of a body. Our model uses the i -th body’s pooled messages, $\bar{\mathbf{e}}'_i$, to update the body’s velocity via Euler integration, $\mathbf{v}'_i = \mathbf{v}_i + \phi^v(\mathbf{v}_i, \bar{\mathbf{e}}'_i)$. If we assume our GN is trained to predict velocity updates perfectly for any number of bodies, this means $\bar{\mathbf{f}}_i = \sum_{r_k=i} \mathbf{f}_k = \phi_i^v(\sum_{r_k=i} \mathbf{e}'_k) = \phi_i^v(\bar{\mathbf{e}}'_i)$, where $\phi_i^v(\cdot) = \phi^v(\mathbf{v}_i, \cdot)$. We have the result for a single interaction: $\bar{\mathbf{f}}_i = \mathbf{f}_{k, r_k=i} = \phi_i^v(\mathbf{e}'_{k, r_k=i}) = \phi_i^v(\bar{\mathbf{e}}'_i)$. Thus, we can substitute into the multi-interaction case: $\sum_{r_k=i} \phi_i^v(\mathbf{e}'_k) = \phi_i^v(\bar{\mathbf{e}}'_i) = \phi_i^v(\sum_{r_k=i} \mathbf{e}'_k)$, and so ϕ_i^v has to be a linear

¹We adhere closely to the notation used in Battaglia et al. [2018] to formalize our model.

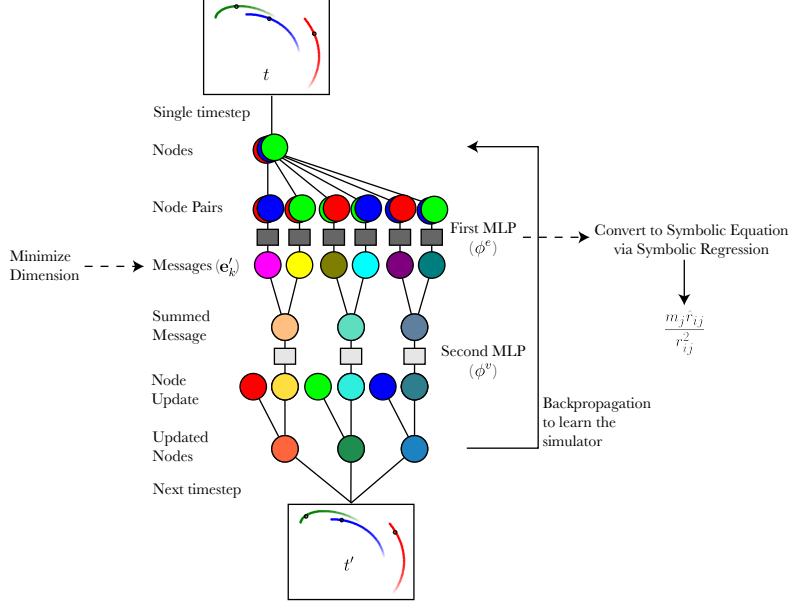


Figure 1: A schematic depicting how we extract physical knowledge from a GN.

transformation. Therefore, for cases where ϕ_i^v is invertible (mapping between the same dimensional space), $\mathbf{e}'_k = (\phi_i^v)^{-1}(\mathbf{f}_k)$, and so the message vectors are linear transformations of the true forces when $L^{e'} = D$. We demonstrate this hypothesis on trained GNs in section 3.

3 Experiments

We set up 100,000 simulations with random masses and initial conditions for both a $1/r$ and $1/r^2$ force law in 2D, a $1/r^2$ law in 3D, and a string with an r^2 force law between nodes in 2D with a global gravity, for 1000 time steps each. These laws are chosen arbitrarily as examples of different symbolic forms. The three n-body problems have six bodies in their training set, and the string has ten nodes, of which the two end nodes are fixed. We train a GN on each of these problems where we choose $L^{e'} = D$, i.e., the length of the message vectors in the GN matches the dimensionality of the force vectors: 2 for the 2D simulations and 3 for the 3D simulations. Our GN, a pure TensorFlow [Abadi et al., 2015] model, has both ϕ^e and ϕ^v as three-hidden-layer multilayer perceptrons (MLPs) with 128 hidden nodes per layer with ReLU activations. We optimize the L1 loss between the predicted velocity update and the true velocity update of each node.

Once we have a trained model, we record the messages, \mathbf{e}'_k , for all bodies over 1000 new simulations for each environment. We fit a linear combination of the vector components of the true force, \mathbf{f}_k , to each component of \mathbf{e}'_k , as can be seen in fig. 2 for $1/r$. The results for each system show that the \mathbf{e}'_k vectors have learned to be a linear combination of the components when $L^{e'} = D$. We see similar linear relations for all other simulations.

We are also able to find the force law when it is unknown by using symbolic regression to fit an algebraic function that approximates ϕ^e . We demonstrate this on the trained GN for the $1/r$ problem using eureka from Schmidt and Lipson [2009] to fit algebraic equations that fit the message. We allow it to use algebraic operators $+$, $-$, \times , $/$, as well as input variables (Δx and Δy for component separation, r for distance, and m_2 for sending body mass) and real constants. Complexity is scored by counting the number of occurrences of each operator, constant, and input variable. This returns a list of the models with the lowest mean square error at each complexity. We parametrize Occam’s razor to find the “best” algebraic model by first sorting the best models by complexity, and then taking the model that maximizes the fractional drop in mean square error (MSE) over the next simplest model: $\arg \max_c (-\Delta \log(\text{MSE}_c) / \Delta c)$, where c is the complexity. The best model found by the symbolic regression for the first output element of ϕ^e is $(0.46m_2\Delta y - 1.55m_2\Delta x)/r^2$,

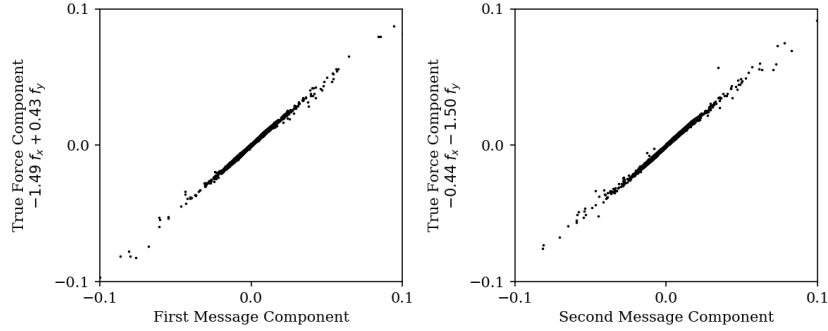


Figure 2: These plots demonstrate that the graph network’s messages have learned to be linear transformations of the two vector components of the true force: f_x and f_y , for the $1/r$ law in 2D.

which is a linear combination of the components of the true force, $m_2 \hat{r}/r$. We can see this is approximately the same linear transformation as the components in the left plot of fig. 2, but this algebraic expression was learned from scratch.

We now test whether the GN will generalize to more nodes better than a GN with a larger $L^{e'}$. This is because it is possible for a GN to “cheat” with a high dimension message-passing space, trained on a fixed number of bodies. One example of cheating would be for ϕ^e to concatenate each sending node’s properties along the message, and ϕ^v to calculate forces from these and add them. When a new body is added, this calculation might break. While it is still possible for ϕ^e to develop an elaborate encoding scheme with $L^{e'} = D$ to cheat at this problem, it seems more natural for ϕ^e to learn the true force when $L^{e'} = D$ and therefore show improved generalization to a greater number of nodes.

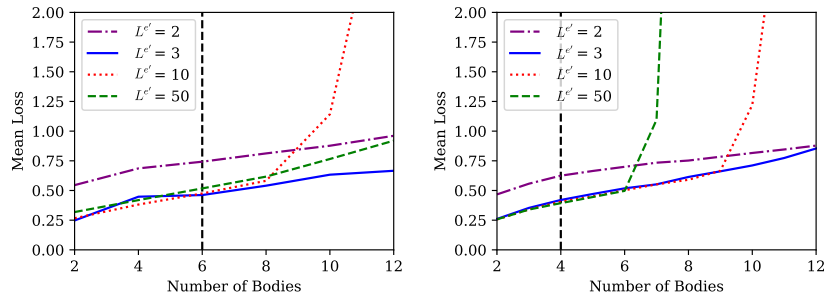


Figure 3: These plots demonstrate the improvement in generalization from minimizing the message passing space. The loss of GNs with different message-passing space dimension ($L^{e'}$), trained on a 6-body and 4-body system, in the left and right plots, respectively (indicated by the vertical line), are tested on a variable number of bodies in a $1/r^2$ simulation in 3D.

We test the hypothesis of better generalization with $L^{e'} = D$ in fig. 3, by training GNs with different $L^{e'}$ on the 3D $1/r^2$ simulations. The observed trend is that systems with $L^{e'} > D$ see their loss blow up with a larger number of bodies — presumably because they have “cheated” slightly and not learned the force law in ϕ^e but in a combination of ϕ^e and ϕ^v , whereas the $L^{e'} \in \{2, 3\}$ systems’ ϕ^e has learned a projection of the true forces and is able to generalize better for greater number of bodies. A conclusion of this may be that one can optimize GNs by minimizing $L^{e'}$ to the known minimum dimension required to transmit information (e.g., 3 for 3D forces), or, if this dimension is unknown, until the loss drops off.

4 Conclusion

We have demonstrated an approach for imposing physically motivated inductive biases on graph networks to learn interpretable representations and improved zero-shot generalization. We have shown through experiment that our graph network models which implement this inductive bias can learn message representations equivalent to the true force vector for n-body gravitational and spring-like simulations in 2D and 3D. We also have demonstrated a generic technique for finding an unknown force law: symbolic regression models to fit explicit algebraic equations to our trained model’s message function. Because GNs have more explicit sub-structure than their more homogeneous deep learning relatives (e.g., plain MLPs, convolutional networks), we can draw more fine-grained interpretations of their learned representations and computations. Finally, we have demonstrated that our model generalizes better at inference time to systems with more bodies than had been experienced during training.

Acknowledgments: Miles Cranmer and Rui Xu thank Professor S.Y. Kung for insightful suggestions on early work, as well as Zejiang Hou for his comments on an early presentation. Miles Cranmer would like to thank David Spergel for advice on this project, and Thomas Kipf, Alvaro Sanchez, and members of the DeepMind team for helpful comments on a draft of this paper. We thank the referees for insightful comments that both improved this paper and inspired future work.

References

- 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, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from tensorflow.org.
- P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*, pages 4502–4510, 2016.
- P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.
- M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.
- M. B. Chang, T. Ullman, A. Torralba, and J. B. Tenenbaum. A compositional object-based approach to learning physical dynamics. *arXiv preprint arXiv:1612.00341*, 2016.
- B. C. Daniels and I. Nemenman. Automated adaptive inference of phenomenological dynamical models. *Nature Communications*, 6(1):1–8, Aug. 2015. ISSN 2041-1723.
- J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.
- M. Jaques, M. Burke, and T. Hospedales. Physics-as-Inverse-Graphics: Joint Unsupervised Learning of Objects and Physics from Video. *arXiv:1905.11169 [cs]*, May 2019.
- T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel. Neural relational inference for interacting systems. *arXiv preprint arXiv:1802.04687*, 2018.
- Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. *arXiv preprint arXiv:1810.01566*, 2018.
- D. Mrowca, C. Zhuang, E. Wang, N. Haber, L. F. Fei-Fei, J. Tenenbaum, and D. L. Yamins. Flexible neural representation for physics prediction. In *Advances in Neural Information Processing Systems*, pages 8799–8810, 2018.

- N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. Geometry from a time series. *Physical review letters*, 45(9):712, 1980.
- A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia. Graph networks as learnable physics engines for inference and control. *arXiv preprint arXiv:1806.01242*, 2018.
- F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.
- M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.