Symmetries and self-supervision in particle physics

Barry M. Dillon
Institut für Theoretische Physik
Universität Heidelberg
dillon@thphys.uni-heidelberg.de

Gregor Kasieczka
Institut für Experimentalphysik
Universität Hamburg
gregor.kasieczka@cern.ch

Hans Olischläger
Institut für Theoretische Physik
Universität Heidelberg

Tilman Plehn
Institut für Theoretische Physik
Universität Heidelberg
plehn@uni-heidelberg.de

Peter Sorrenson
Institut für Theoretische Physik
Heidelberg Collaboratory for Image Processing
Universität Heidelberg
peter.sorrenson@iwr.uni-heidelberg.de

Lorenz Vogel
Institut für Theoretische Physik
Universität Heidelberg

Abstract

A long-standing problem in the design of machine-learning tools for particle physics applications has been how to incorporate prior knowledge of physical symmetries. In this note we propose contrastive self-supervision as a solution to this problem, with jet physics as an example. Using a permutation-invariant transformer network, we learn a representation which outperforms hand-crafted competitors on a linear classification benchmark.

1 Introduction

The impact of machine-learning tools in particle physics has been far-reaching, with many of the traditional phenomenological tools being upgraded or replaced with more powerful and more efficient alternatives. However, the traditional tools and techniques have been constructed based on physical insight and symmetries that have been studied for decades, and to make the most of deep-learning tools this knowledge should be embedded in the network architectures we use. A common choice is to embed this knowledge in the data through aggressive preprocessing steps, however this limits the power of the machine-learning tools whose strength lies in extracting non-trivial information from low-level raw data. This is a long-standing problem which affects all applications of machine-learning in particle-physics. We discuss *self-supervision* as the solution to this problem.

Self-supervision refers to optimization tasks that use pseudo-labels in the loss function. The pseudo-labels are not class labels, but instead are generated from the data in an unsupervised manner. In our work we consider one specific form of self-supervised learning called *contrastive learning* [2]. Here the pseudo-labels are generated by taking a single sample from the dataset and applying a set of transformations (known as augmentations) to this sample. Both the original sample and the augmented sample are then given the same pseudo-label. We choose to use physically-motivated augmentations which do not change the properties of the underlying physical object, but will change its expression in the measurement space, e.g. resulting in different values in a detector. The neural network can then learn that such pairs of observations represent the same physical object and can discard irrelevant information accordingly.

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

To demonstrate this idea we focus on particle jets. Jets are produced in large quantities at hadron colliders, and are some of the most complex objects analysed there. Each jet consists of $\mathcal{O}(10-100)$ particles whose substructure contains non-trivial kinematic correlations that can be used to study the origin of the jet. However there are a number of symmetries and augmentations of the jet under which these non-trivial correlations should be invariant. The goal therefore is to use self-supervised contrastive learning to map the raw jet constituent data to a new representation which is invariant to the pre-defined symmetries and augmentations, but which retains information on these non-trivial kinematic correlations. This new representation can then be used for downstream tasks.

2 Contrastive learning

In contrastive learning we sample a batch of jets $\{x_i\}$ from the dataset during training, and generate an augmented batch $\{x'_i\}$ by applying a set of physics-inspired augmentations to each jet in the original batch. The pseudo-labels are generated by considering pairs of original and augmented jets:

- positive pairs: $\{(x_i, x'_i)\}$
- negative pairs: $\{(x_i, x_j)\} \cup \{(x_i, x'_j)\}$ for $i \neq j$.

The task is to teach the network to map positive pairs close together and negative pairs far apart in the new representation space. The mapping to this space is defined by $f(x_i) = z_i$, with f representing a transformer network that we will define in Sec. 4. The contrastive loss function [2] used in the optimization is:

$$\mathcal{L}_i = -\log \frac{e^{s(z_i, z'_i)/\tau}}{\sum_{j \neq i \in \text{batch}} \left[e^{s(z_i, z_j)/\tau} + e^{s(z_i, z'_j)/\tau} \right]}, \quad s(z_i, z_j) = \frac{z_i \cdot z_j}{|z_i||z_j|} = \cos\theta_{ij} \tag{1}$$

where the cosine-similarity function $s(z_i, z_j)$ is the measure the network uses to determine how similar two jet representations are. Note that due to the definition of the similarity measure, the new representations are constrained to the surface of a unit hyper-sphere. The minimization of Eq. 1 ensures that the positive-pairs (in the numerator) should be close together in the new representation space, while the negative-pairs (in the denominator) should be far apart. Bringing positive pairs close together ensures that the new representation is invariant to the set of symmetries and augmentations inherited from our prior physics knowledge, while forcing negative pairs apart ensures that the non-trivial correlations present in the raw data are retained in this new representation.

3 Symmetries and augmentations

The results discussed here use the top-tagging dataset generated for the challenge outlined in [1]. The dataset consists of jets with the transverse momentum $p_T \in [550, 650]$GeV and a radius $R = \sqrt{\Delta\eta^2 + \Delta\phi^2} = 0.8$, originating from either a top-quark or a gluon parton. Details on how this data was generated are contained within the reference [1]. To a good approximation we can assume that the constituents of the jets are all massless, meaning we can describe the kinematics of each constituent with just the transverse momentum p_T, the pseudo-rapidity η, and the azimuthal angle ϕ. Thus a jet can be written as a collection of constituents $x_i = \{(p_T, \eta, \phi)_k\}_i$ with k labelling the constituent in the jet and i labelling the jet in the dataset. We applying the following augmentations to the jets during contrastive learning:

Rotations: The centre of the jet is defined as the p_T-weighted centroid of the constituents, and with $R < 1$ the jet mass is approximately invariant to rotations around this point. There is also no preferred orientation of the jet substructure as measured in the collider, so these rotations are an approximate symmetry of the system.

Translations: Rotations in the azimuthal angle ϕ are of course a symmetry of the system. In addition, in the limit of massless constituents, translations in the pseudo-rapidity η are also symmetries. This means we use translations in the $\eta - \phi$ plane as a symmetry of the system.

Collinear splittings: The angular resolution of a detector is unable to distinguish between two constituents with transverse momentum $p_{T,a}$ and p_T, b which are very close together, i.e. $\Delta R_{ab} \ll 1$. To encode this knowledge into the new representations, collinear augmentations are introduced which

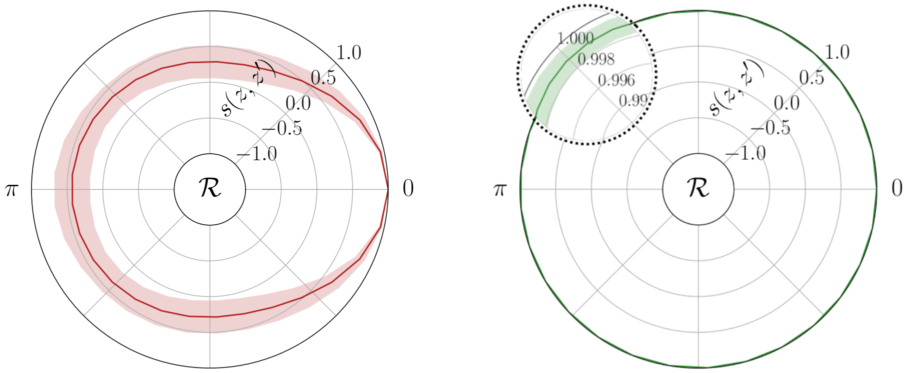

Figure 1: Visualization of the rotational invariance in representation space, keeping in mind that $s(z, z') = 1$ indicates identical representations. We show JetCLR representation trained without (left) and with (right) rotational transformations, from the authors of Ref. [5].

randomly select some number of constituents in the jet and split them into two constituents such that

$$p_{T,a} + p_{T,b} = p_T \qquad \eta_a = \eta_b = \eta$$
$$\phi_a = \phi_b = \phi \; . \tag{2}$$

IR smearing: From Quantum Field Theory (QFT) we know that the hard-process in a collision factorizes from the universal soft-gluon emissions, meaning that stochastic augmentations to the soft radiation in the jet should not alter the physical information contained within it. The augmented jet is created by smearing the (η, ϕ) positions of all constituents in the jet by sampling from Normal distributions as:

$$\eta' \sim \mathcal{N}\left(\eta, \frac{\Lambda_{\text{soft}}}{p_T}\right) \qquad \text{and} \qquad \phi' \sim \mathcal{N}\left(\phi, \frac{\Lambda_{\text{soft}}}{p_T}\right) \; , \tag{3}$$

where η' and ϕ' are the new smeared coordinates, and $\Lambda_{\text{soft}} = 100\text{MeV}$ is a scale that determines the strength of the smearing relative to the p_T. As well as encoding detector- and QFT-related knowledge, the IR and collinear augmentations will provide an approximate infrared and collinear (IRC) safety to the new representations. This is a technical requirement of observables that allow them to be calculated analytically in a perturbative expansion.

4 JetCLR

The contrastive learning is implemented with the following training loop:

1. sample batch of jets $\{x_i\}$ from dataset
2. create augmented set of jets $\{x'_i\}$ by applying each of the augmentations in sequence
3. pass both $\{x_i\}$ and $\{x'_i\}$ through the network to obtain $\{z_i\}$ and $\{z'_i\}$
4. calculate the contrastive loss and update the network weights

The mapping from raw data to the new observable representation is parameterized by a permutation-invariant transformer-encoder network. This tool is called JetCLR [5] (Contrastive Learning of Jet Representations).

The network has the structure described in Ref. [5], consisting of a linear embedding layer, followed by a transformer-encoder network [10, 11, 13], then summation along the constituent dimension, and finally a fully-connected head network. The embedding increases the dimension of each constituent from three to some higher dimension, which is also the dimension of the final representation space. We found 1000 to perform best in our experiments. Since transformers are equivariant to permutations of the constituents, and since we subsequently sum over the constituent dimension, our network is permutation invariant, similar to the Set Transformer [9] and other similar models used in particle

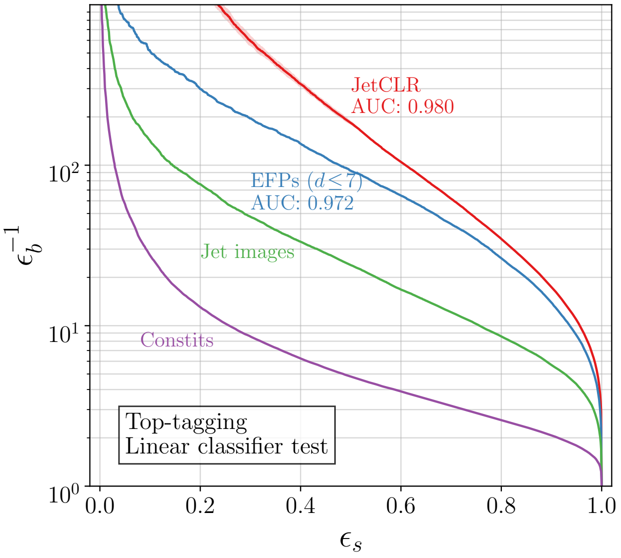

Figure 2: Comparison of JetCLR with standard hand-crafted representations, from the authors of Ref. [5]. The solid curve and shaded bands on the JetCLR curve show the mean and standard deviation over 4 different runs.

physics [8, 12]. The head network is necessary to increase the representational power of the total network but, following standard practice in contrastive learning [2], it is found that the pre-head output performs best as a representation for downstream tasks. The description of the hyper-parameters used can be found in [5].

In addition to permutation invariance, physically-motivated IR-safety can be built into the network. This is achieved by a combination of p_T-dependent masking in the attention layers of the transformer, as well as a p_T-dependent weighting in the summation layer, as in Ref. [5].

5 Results

Firstly, it can be established that the representations obtained with JetCLR are invariant to the symmetries used during training. As an example, we show in Fig. 1 how the cosine similarity changes as a jet is rotated. Displayed are two different representations, one of which was trained with rotation augmentations to the data, and the other without. We clearly see that the addition of rotation augmentations gives the representation space invariance to rotations of the input jet.

Secondly, the JetCLR representations can be compared against other widely-used representations in the literature. Although the technique is entirely self-supervised, to obtain a measure of the effectiveness of a jet representation a linear neural network is trained to classify between labeled top and QCD jets in the representation space. Such a supervised linear classifier test is standard practice in the self-supervised literature. In Fig. 2 we show a ROC curve comparison of the linear classifier test results on various representations. The constituents representation is constructed by ordering all constituents in a jet by their p_T, taking the 20 with the highest p_T, and flattening the $\{(p_T, \eta, \phi)_j\}$ array into a 60-dimensional vector. The jet images representation [3, 4, 6] encodes the jet as a gray scale image. Finally the Energy Flow Polynomials (Ref. [7]) are a more sophisticated, higher-dimensional representation of the jet data that are explicitly IRC safe, and invariant under rotations and translations by construction. It can be seen that JetCLR outperforms all of these hand-crafted representations on the linear classifier test.

6 Conclusions

We have presented, with the explicit example of JetCLR, how self-supervision can solve the problem of embedding prior physics knowledge obtained from theoretical considerations into machine learning tools. We encoded some symmetries through model constraints, such as permutation invariance, where this is easy to do, and encoded other symmetries, such as rotations, as augmentations in a contrastive learning framework. We hope that this example will inspire similar work aiming to incorporate prior knowledge into particle physics applications, or other applications in the wider physical sciences.

References

[1] Anja Butter et al. The Machine Learning Landscape of Top Taggers. *SciPost Phys.*, 7:014, 2019.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Everest Hinton. A simple framework for contrastive learning of visual representations. 2020.

[3] Josh Cogan, Michael Kagan, Emanuel Strauss, and Ariel Schwarztman. Jet-Images: Computer Vision Inspired Techniques for Jet Tagging. *JHEP*, 02:118, 2015.

[4] Luke de Oliveira, Michael Kagan, Lester Mackey, Benjamin Nachman, and Ariel Schwartzman. Jet-images — deep learning edition. *JHEP*, 07:069, 2016.

[5] Barry M. Dillon, Gregor Kasieczka, Hans Olischlager, Tilman Plehn, Peter Sorrenson, and Lorenz Vogel. Symmetries, safety, and self-supervision. 2021.

[6] Gregor Kasieczka, Tilman Plehn, Michael Russell, and Torben Schell. Deep-learning Top Taggers or The End of QCD? *JHEP*, 05:006, 2017.

[7] Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler. Energy flow polynomials: A complete linear basis for jet substructure. *JHEP*, 04:013, 2018.

[8] Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler. Energy Flow Networks: Deep Sets for Particle Jets. *JHEP*, 01:121, 2019.

[9] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pages 3744–3753. PMLR, 2019.

[10] Vinicius Mikuni and Florencia Canelli. ABCNet: An attention-based method for particle tagging. *Eur. Phys. J. Plus*, 135(6):463, 2020.

[11] Vinicius Mikuni and Florencia Canelli. Point cloud transformers applied to collider physics. *Mach. Learn. Sci. Tech.*, 2(3):035027, 2021.

[12] Bryan Ostdiek. Deep set auto encoders for anomaly detection in particle physics. 2021.

[13] Alexander Shmakov, Michael James Fenton, Ta-Wei Ho, Shih-Chieh Hsu, Daniel Whiteson, and Pierre Baldi. SPANet: Generalized Permutationless Set Assignment for Particle Physics using Symmetry Preserving Attention. 6 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] see abstract and the conclusions in Section 6.
   (b) Did you describe the limitations of your work? [Yes] one of the advantages of the technique we present is that it is self-supervised and thus not reliant on simulated data, in Section 5 we explain that despite this we are forced to use a supervised linear classifier test as a measure of the representation quality.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
    (a) Did you state the full set of assumptions of all theoretical results? [N/A]
    (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
    (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] both data and code are public, see Ref. [1] and Ref. [5].
    (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] see Sec. 4 and Ref. [5].
    (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] see Fig. 2.
    (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] see Sec. 4 and Ref. [5].
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
    (a) If your work uses existing assets, did you cite the creators? [Yes] see Ref. [1].
    (b) Did you mention the license of the assets? [N/A]
    (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
    (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
    (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
    (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
    (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
    (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]