
Particle-level Compression for New Physics Searches

Yifeng Huang

Department of Physics and Astronomy
University of California, Irvine
yifengh3@uci.edu

Jack Collins

SLAC National Accelerator Laboratory
jackadsa@gmail.com

Simon Knapen

Physics Division
Lawrence Berkeley National Laboratory, Berkeley
Berkeley Center for Theoretical Physics
University of California, Berkeley smknapen@lbl.gov

Daniel Whiteson

Department of Physics and Astronomy
University of California, Irvine
daniel@uci.edu

Benjamin Nachman

Physics Division
Lawrence Berkeley National Laboratory, Berkeley
Berkeley Institute for Data Science
University of California, Berkeley
bpnachman@lbl.gov

Abstract

In collider-based particle and nuclear physics experiments, data are produced at such extreme rates that only a subset can be recorded for later analysis. Typically, algorithms select individual collision events for preservation and store the complete experimental response. A relatively new alternative strategy is to additionally save a partial record for a subset of events, allowing for later specific analysis of a larger fraction of events. We propose a strategy that bridges these paradigms by compressing entire events for generic offline analysis but at a lower fidelity. An optimal-transport-based β Variational Autoencoder (VAE) is used to automate the compression and the hyperparameter β controls the compression fidelity. We introduce a new approach for multi-objective learning functions by simultaneously learning a VAE appropriate for all values of β through parameterization. We present an example use case, a di-muon resonance search at the Large Hadron Collider (LHC), where we show that simulated data compressed by our β VAE has enough fidelity to distinguish distinct signal morphologies.

1 Introduction

The rate and size of interaction events at modern particle and nuclear physics experiments typically prohibit storage of the complete experimental dataset and require that many interaction events be discarded in real time by a *trigger* system. For example, at the Large Hadron Collider (LHC), collisions occur at a rate of 40 MHz, but the ATLAS and CMS experiments recording rates are typically $\mathcal{O}(\text{kHz})$ [1, 2]. For selected events, the complete experimental response is preserved for later analysis. When the scientific goals only require identifying events which contain rare and easy-to-identify objects, such as high energy photons, the trigger system is highly efficient. However, this strategy leaves unexamined the vast majority of the events, including many with complex features that are hard to quickly identify online and/or may not be rare.

An alternative approach to fully recording a small fraction of the events is to preserve a partial record of a larger fraction [3–5]. This strategy has allowed access to lower energy phenomena which occur at higher rates, but the utility of these partial data records is limited. For example, a recent partial-event analysis targets di-muon resonances [6], only recording the four-momenta of the two muons and a small number of additional event properties for low-mass events that would otherwise be too high rate for the full-event trigger system. This approach has the potential to make a major discovery, but the lack of a full event record makes it nearly impossible to *diagnose* such a discovery. To distinguish between several competing hypotheses which might generate a peak in the di-muon spectrum would require recording new data with a dedicated trigger, which is both time consuming and expensive.

We propose an approach that bridges the full and partial event paradigms automatically with machine learning. This is accomplished by training a neural network to learn a lossy event compression with a tunable resolution parameter. An extreme version of this approach would be to save every event at the highest resolution allowable by hardware (see e.g. Ref. [7] for autoencoders in hardware). We present a more modest version in which we envision full event compression which could run alongside partial event triggers to expand their utility for a larger range of offline analyses. Our approach uses an optimal transport-based Variational Autoencoder (VAE) following Ref. [8].

In a proof-of-concept study, we compress and record a sample of simulated interactions which are similar to those analyzed in Ref [6], preserving information which would otherwise be lost. We show that this additional information can be used to effectively discriminate between two signal models which are difficult to distinguish with only the muon kinematics. The overall structure of the proposal is that first, a signal is discovered in a trigger-level analysis such as this dimuon resonance search. Subsequently, a compressed version of the hadronic event data, which has been stored alongside the muons, can be used to rule out or favour candidate signal models.

2 Related Work

An alternative to compressing individual events is compressing the entire dataset online [9], which is methodologically and practically more challenging. An alternative to saving events for offline analysis is to look for new particles automatically with online anomaly detection [10–13]. While we build our VAE on the setup from Ref. [8] using the Sinkhorn approximation [14, 15] to the Earth Movers Distance, other possibilities have been explored, such as using graph neural networks [16]. We leave a comparison of the power of different approaches to future work.

3 β -parameterized Variational Autoencoder

We represent each collider event x as a point cloud of 3-vectors $\{p_{\text{T}}/H_{\text{T}}, \eta, \phi\}$, where η and ϕ are the geometric coordinates of particles in the detector, and p_{T} their transverse momenta which correspond to the weights in the point cloud. These are normalized for each event using $H_{\text{T}} = \sum_i p_{\text{T},i}$. We build an EMD-VAE [8, 17, 18] trained to minimize a reconstruction error given by an approximation to the 2-Wasserstein distance between collider events x and reconstructed examples x' , with loss function

$$L = \langle S(x, x'(z))/\beta + D_{\text{KL}}(q(z|x)||p(z)) \rangle_{p(x)}. \quad (1)$$

An encoder network maps the input x to a Gaussian-parameterized distribution $q(z|x)$ on 256-dimensional latent coordinates z . This network is built as a Deepsets/Particle Flow Network (PFN) [19, 20]. A decoder $x'(z)$ maps latent codes z to jets x' , parameterizing a posterior probability $\log p(x|z) \propto S(x, x'(z))/\beta$, where $S(x, x'(z))$ is a sharp Sinkhorn [15, 21–23] approximation to the 2-Wasserstein distance between event x and its decoded x' with ground distance given by $M_{ij} = \Delta R_{ij}^2 \equiv (\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2$, and calculated using the same algorithm and parameters as in Ref [8]. This decoder network is built as a dense neural network. $D_{\text{KL}}(q(z|x)||p(z))$ is the KL divergence between the encoder probability $q(z|x)$ and the prior $p(z)$, which we take to be a standard Gaussian. This KL divergence can be expressed as a sum of contributions from each of the 256 latent space directions. The details of the architecture setup is described in the Appendix.

The quantity β is typically taken to be a fixed hyperparameter of the network [24] which controls the balance between reconstruction fidelity and degree of compression in the latent space. In this work, we make a novel contribution by elevating β from a fixed hyperparameter to an input of both the encoder and decoder networks. Training events are each accompanied by a value for β generated

from some arbitrary distribution which is provided as an additional input to both the encoder and decoder, and included in the loss calculation for each event. The encoder and decoder then become dependent on the input value for β , and can be written as $q(z|x; \beta)$ and $x'(z; \beta)$, respectively. During testing, events can be input with whatever value for β is deemed useful for a particular application. In this work we take advantage of this property purely for its usefulness in prototyping. However, it is conceivable that the ability to vary β on the fly could prove valuable for an online application that has time-varying constraints on data transmission rates or requirements on reconstruction fidelity. Furthermore, we have found that β -annealing is neither required nor helpful in improving our VAE performance, significantly reducing the complexity and time required for training and experiments.

4 Numerical Results

In this section we will illustrate how a VAE trained to compress the full event at trigger level might be used to augment a trigger-level dimuon resonance search and allow physicists to distinguish between different potential hypothetical signal models without having to record new data using the full-event trigger system.

We train our β -parameterized VAE using simulated Standard Model events, specifically an inclusive b -quark jet sample ($p_T \gtrsim 4 \text{ GeV}/c$), weighted as $\sim 1/p_T^{-3}$. We then consider two potential signals which might be found in a trigger-level dimuon resonance search: first, a light scalar decaying to muons ($S \rightarrow 2\mu$) produced in an exotic B -meson decay ($B \rightarrow KS$) and secondly, a dimuon resonance produced in a hidden valley model [25], from an example in Ref. [26]. The latter model has more hadronic activity than the former, as well as a small amount of invisible momentum, but has otherwise very similar muon kinematics. More details can be found in the Appendix. All signal and background events were generated with the PYTHIA event generator [27]. A simple binary classifier (see Appendix) trained on only the two highest- p_T muons taken from the two signal event samples achieves an AUC of 0.71. The dimuon spectra are different but not convincingly so, and any attempt to distinguish the two signals based on the dimuons alone would be additionally complicated by the uncertainties on details of the showering model for the hidden valley scenario. However, the hadronic activity in the two signal classes is very distinctive, and even a relatively low-fidelity VAE reconstruction may tell them apart.

The structure that is learnt by the VAE as a function of β is qualitatively illustrated in Figs. 1 and 2. In Fig. 1, the red points represent an event from the $b\bar{b}$ test sample, and the blue its maximum likelihood estimate (MLE) reconstruction (defined using the the maximum likelihood latent code for reconstruction rather than a random sample) at various values for β . Each point represents a particle in the event, with area proportional to the momentum p_T of the corresponding particle. We see that for large β , the VAE learns an uninformative average over all training events, but for smaller values of β it begins to learn a more precise reconstruction.

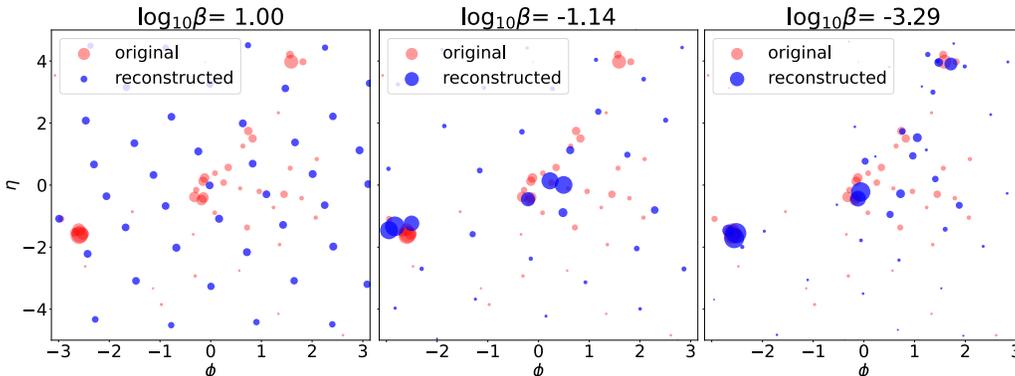


Figure 1: Reconstruction of an SM event as a function of $\log_{10} \beta$. Red points: Example test event. Blue points: MLE reconstruction of the same test event. Each point represents a particle in the event, with area proportional to the particle's p_T . The number of active latent dimensions (defined as those with $D_{\text{KL},i} > 0.1$) is 0, 9, and 48 at each of these β values respectively, and the compression rates D_{KL} are 0, 12, and 145 bits, respectively.

Some properties of the information content of the learnt representations are reflected in Fig. 2. In the left plot, we show the individual KL divergences associated with each of the 256 latent space directions as a function of β . We see that for $\beta > 1$, all latent space directions are uninformative. Latent directions start to become sequentially informative with $D_{KL} > 0$, for $\beta < 1$. For any value of β , there is a clear hierarchy of information content in different latent directions, which can be utilized for an efficient encoding: those values which are encoded with low resolution (large variance for $q(z|x)$) can safely be represented with lower precision numbers than those having higher resolution. Note that the $\langle D_{KL}(q(z|x)||p(z)) \rangle_{p(x)} > \mathcal{I}(x; z)$, with \mathcal{I} being the mutual information, and therefore the KL divergence places an upper bound on the minimum number of bits required for the lossy compression algorithm with average distortion $\langle S(x, x') \rangle$.

The overall structure of the left plot is summarized by the two heat capacities in the right plot, defined in analogy with the thermodynamic heat capacity [8, 28, 29] by

$$C_S = \frac{d \langle S(x, x') \rangle}{d\beta}, \quad C_{KL} = -\frac{d \langle D_{KL}(q(z|x)||p(z)) \rangle}{d \log \beta}. \quad (2)$$

Because the lines in the left plot have gradient close to -0.5 , the heat capacities are related to the effective number of degrees of the system by $\dim \simeq 2C'$, similarly to a thermodynamic system with quadratic Hamiltonian. There is a plateau for $\beta \lesssim 10^{-2}$, indicating that the VAE is unable to learn additional informative structure at smaller scales than this. For $\beta \lesssim 10^{-4}$, the VAE begins to overfit the data and the two heat capacities diverge.

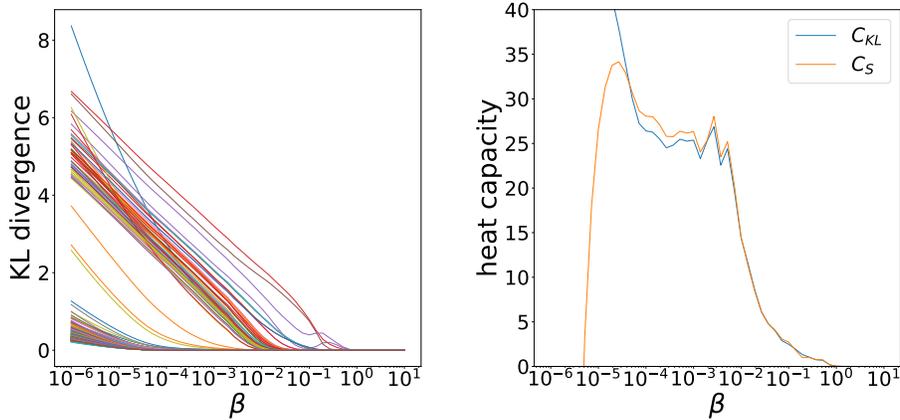


Figure 2: Left: KL divergences of the 256 individual latent space directions as a function of β , averaged over the test SM data. Right: Heat capacities on same data, defined in the text.

Having trained a VAE on SM data, we want to explore the distinguishability of our signal models after being compressed and reconstructed. To do this we trained a PFN classifier on the H_T -normalized VAE reconstructions of the two signal categories. The details of this architecture and training parameters are described in the Appendix.

In Fig. 3, we see the AUC for classification between reconstructed signal events as a function of β . For $\beta > 1$, the VAE reconstruction is uninformative for signal classification, but it contains increasingly useful information for classification for $10^{-2} < \beta < 1$, before a plateau is reached for $\beta < 10^{-2}$. This corresponds with the features in Fig. 2 right, which shows a plateau in the effective number of degrees of freedom of the learnt representation below 10^{-2} . These features could be combined with the dimuon resonance information (which alone provides an AUC of 0.71), and the overall H_T of the events (which

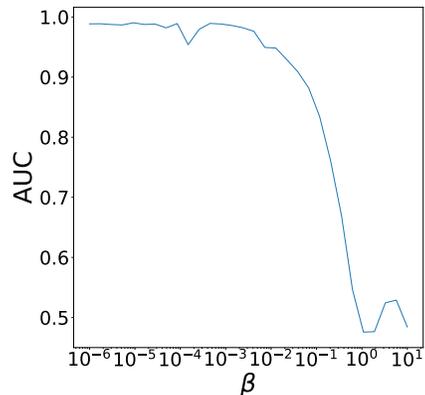


Figure 3: AUC for classification between the two signal models as a function of β , the parameter that controls the compression fidelity.

alone provides an AUC of 0.84) to further improve separation between the signals.

5 Conclusions and Outlook

Particle and nuclear physics experiments produce data at such volumes that analysis is impossible without some kind of sacrifice. The traditional approach is to keep full records of a small subset of the data, while newer strategies store partial information about a larger fraction. In some cases, however, neither strategy is sufficient to discover and characterize the physics behind evidence of a new kind of particle or interaction. We propose an alternative approach to address this limitation by automatically compressing the full experimental information, relying on an optimal-transport-based Variational Autoencoder with a parameter β that controls the bandwidth. In a benchmark test, we are able to distinguish between new physics scenarios at a level significantly beyond what is available from the partial event system. We expect in the coming run of the Large Hadron Collider, such a system may be able to help discover and diagnose new physics.

Acknowledgments and Disclosure of Funding

We thank Jesse Thaler for suggesting the parameterized loss function as an alternative to annealing. DW/EH, BN/SK, JC were supported by the U.S. Department of Energy, Office of Science under grant DE-SC0009920 and contracts DE-AC02-05CH11231 and DE-AC02-76SF00515, respectively. Part of this work was performed at the Aspen Center for Physics, which is supported by National Science Foundation grant PHY-1607611. This research made use of the Matplotlib [30], Jupyter [31], NumPy [32], and SciPy [33] software packages.

References

- [1] M. Aaboud *et al.*, “Performance of the ATLAS Trigger System in 2015,” *Eur. Phys. J. C*, vol. 77, no. 5, p. 317, 2017. DOI: [10.1140/epjc/s10052-017-4852-3](https://doi.org/10.1140/epjc/s10052-017-4852-3). arXiv: [1611.09661](https://arxiv.org/abs/1611.09661) [hep-ex].
- [2] V. Khachatryan *et al.*, “The CMS trigger system,” *JINST*, vol. 12, no. 01, P01020, 2017. DOI: [10.1088/1748-0221/12/01/P01020](https://doi.org/10.1088/1748-0221/12/01/P01020). arXiv: [1609.02366](https://arxiv.org/abs/1609.02366) [physics.ins-det].
- [3] R. Aaij *et al.*, “Tesla : an application for real-time data analysis in High Energy Physics,” *Comput. Phys. Commun.*, vol. 208, pp. 35–42, 2016. DOI: [10.1016/j.cpc.2016.07.022](https://doi.org/10.1016/j.cpc.2016.07.022). arXiv: [1604.05596](https://arxiv.org/abs/1604.05596) [physics.ins-det].
- [4] V. Khachatryan *et al.*, “Search for narrow resonances in dijet final states at $\sqrt{s} = 8$ TeV with the novel CMS technique of data scouting,” *Phys. Rev. Lett.*, vol. 117, no. 3, p. 031 802, 2016. DOI: [10.1103/PhysRevLett.117.031802](https://doi.org/10.1103/PhysRevLett.117.031802). arXiv: [1604.08907](https://arxiv.org/abs/1604.08907) [hep-ex].
- [5] M. Aaboud *et al.*, “Search for low-mass dijet resonances using trigger-level jets with the ATLAS detector in pp collisions at $\sqrt{s} = 13$ TeV,” *Phys. Rev. Lett.*, vol. 121, no. 8, p. 081 801, 2018. DOI: [10.1103/PhysRevLett.121.081801](https://doi.org/10.1103/PhysRevLett.121.081801). arXiv: [1804.03496](https://arxiv.org/abs/1804.03496) [hep-ex].
- [6] A. M. Sirunyan *et al.*, “Search for a Narrow Resonance Lighter than 200 GeV Decaying to a Pair of Muons in Proton-Proton Collisions at $\sqrt{s} = \text{TeV}$,” *Phys. Rev. Lett.*, vol. 124, no. 13, p. 131 802, 2020. DOI: [10.1103/PhysRevLett.124.131802](https://doi.org/10.1103/PhysRevLett.124.131802). arXiv: [1912.04776](https://arxiv.org/abs/1912.04776) [hep-ex].
- [7] G. Di Guglielmo *et al.*, “A Reconfigurable Neural Network ASIC for Detector Front-End Data Compression at the HL-LHC,” *IEEE Trans. Nucl. Sci.*, vol. 68, no. 8, pp. 2179–2186, 2021. DOI: [10.1109/TNS.2021.3087100](https://doi.org/10.1109/TNS.2021.3087100). arXiv: [2105.01683](https://arxiv.org/abs/2105.01683) [physics.ins-det].
- [8] J. H. Collins, “An Exploration of Learnt Representations of W Jets,” Sep. 2021. arXiv: [2109.10919](https://arxiv.org/abs/2109.10919) [hep-ph].
- [9] A. Butter *et al.*, “Ephemeral Learning – Augmenting Triggers with Online-Trained Normalizing Flows,” Feb. 2022. arXiv: [2202.09375](https://arxiv.org/abs/2202.09375) [hep-ph].
- [10] O. Cerri, T. Q. Nguyen, M. Pierini, M. Spiropulu, and J.-R. Vlimant, “Variational Autoencoders for New Physics Mining at the Large Hadron Collider,” *JHEP*, vol. 05, p. 036, 2019. DOI: [10.1007/JHEP05\(2019\)036](https://doi.org/10.1007/JHEP05(2019)036). arXiv: [1811.10276](https://arxiv.org/abs/1811.10276) [hep-ex].

- [11] O. Knapp, O. Cerri, G. Dissertori, T. Q. Nguyen, M. Pierini, and J.-R. Vlimant, “Adversarially Learned Anomaly Detection on CMS Open Data: re-discovering the top quark,” *Eur. Phys. J. Plus*, vol. 136, no. 2, p. 236, 2021. DOI: [10.1140/epjp/s13360-021-01109-4](https://doi.org/10.1140/epjp/s13360-021-01109-4). arXiv: [2005.01598](https://arxiv.org/abs/2005.01598) [hep-ex].
- [12] E. Govorkova, E. Puljak, T. Aarrestad, M. Pierini, K. A. Woźniak, and J. Ngadiuba, “LHC physics dataset for unsupervised New Physics detection at 40 MHz,” Jul. 2021. arXiv: [2107.02157](https://arxiv.org/abs/2107.02157) [physics.data-an].
- [13] V. Mikuni, B. Nachman, and D. Shih, “Online-compatible Unsupervised Non-resonant Anomaly Detection,” Nov. 2021. arXiv: [2111.06417](https://arxiv.org/abs/2111.06417) [cs.LG].
- [14] G. Luise, A. Rudi, M. Pontil, and C. Ciliberto, “Differential properties of sinkhorn approximation for learning with wasserstein distance,” 2018. DOI: [10.48550/ARXIV.1805.11897](https://doi.org/10.48550/ARXIV.1805.11897). [Online]. Available: <https://arxiv.org/abs/1805.11897>.
- [15] R. Sinkhorn and P. Knopp, “Concerning nonnegative matrices and doubly stochastic matrices,” *Pacific J. Math.*, vol. 21, no. 2, pp. 343–348, 1967. [Online]. Available: <https://projecteuclid.org/443/euclid.pjm/1102992505>.
- [16] S. Tsan *et al.*, “Particle Graph Autoencoders and Differentiable, Learned Energy Mover’s Distance,” in *35th Conference on Neural Information Processing Systems*, Nov. 2021. arXiv: [2111.12849](https://arxiv.org/abs/2111.12849) [physics.data-an].
- [17] K. Fraser, S. Homiller, R. K. Mishra, B. Ostdiek, and M. D. Schwartz, “Challenges for unsupervised anomaly detection in particle physics,” *JHEP*, vol. 03, p. 066, 2022. DOI: [10.1007/JHEP03\(2022\)066](https://doi.org/10.1007/JHEP03(2022)066). arXiv: [2110.06948](https://arxiv.org/abs/2110.06948) [hep-ph].
- [18] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *CoRR*, 2014. arXiv: [1312.6114](https://arxiv.org/abs/1312.6114) [stat.ML].
- [19] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, “Deep sets,” *Advances in neural information processing systems*, vol. 30, 2017.
- [20] P. T. Komiske, E. M. Metodiev, and J. Thaler, “Energy Flow Networks: Deep Sets for Particle Jets,” *JHEP*, vol. 01, p. 121, 2019. DOI: [10.1007/JHEP01\(2019\)121](https://doi.org/10.1007/JHEP01(2019)121). arXiv: [1810.05165](https://arxiv.org/abs/1810.05165) [hep-ph].
- [21] M. Cuturi, “Sinkhorn distances: Lightspeed computation of optimal transport,” in *NIPS*, 2013. arXiv: [1306.0895](https://arxiv.org/abs/1306.0895) [stat.ML].
- [22] G. Luise, A. Rudi, M. Pontil, and C. Ciliberto, “Differential properties of sinkhorn approximation for learning with wasserstein distance,” *Advances in Neural Information Processing Systems*, vol. 31, pp. 5859–5870, 2018. arXiv: [1805.11897](https://arxiv.org/abs/1805.11897) [stat.ML].
- [23] G. Patrini *et al.*, “Sinkhorn autoencoders,” in *Uncertainty in Artificial Intelligence*, PMLR, 2020, pp. 733–743.
- [24] I. Higgins *et al.*, “Beta-vae: Learning basic visual concepts with a constrained variational framework,” in *ICLR*, 2017.
- [25] M. J. Strassler and K. M. Zurek, “Echoes of a hidden valley at hadron colliders,” *Phys. Lett. B*, vol. 651, pp. 374–379, 2007. DOI: [10.1016/j.physletb.2007.06.055](https://doi.org/10.1016/j.physletb.2007.06.055). arXiv: [hep-ph/0604261](https://arxiv.org/abs/hep-ph/0604261).
- [26] S. Knapen, J. Shelton, and D. Xu, “Perturbative benchmark models for a dark shower search program,” *Phys. Rev. D*, vol. 103, no. 11, p. 115 013, 2021. DOI: [10.1103/PhysRevD.103.115013](https://doi.org/10.1103/PhysRevD.103.115013). arXiv: [2103.01238](https://arxiv.org/abs/2103.01238) [hep-ph].
- [27] T. Sjöstrand *et al.*, “An introduction to PYTHIA 8.2,” *Comput. Phys. Commun.*, vol. 191, pp. 159–177, 2015. DOI: [10.1016/j.cpc.2015.01.024](https://doi.org/10.1016/j.cpc.2015.01.024). arXiv: [1410.3012](https://arxiv.org/abs/1410.3012) [hep-ph].
- [28] A. A. Alemi and I. Fischer, “Therm1l: Thermodynamics of machine learning,” *arXiv preprint arXiv:1807.04162*, 2018.
- [29] D. J. Rezende and F. Viola, “Taming vaes,” *arXiv preprint arXiv:1810.00597*, 2018.
- [30] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- [31] T. Kluyver *et al.*, “Jupyter notebooks – a publishing format for reproducible computational workflows,” in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt, Eds., IOS Press, 2016, pp. 87–90.
- [32] C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2). [Online]. Available: <https://doi.org/10.1038/s41586-020-2649-2>.

- [33] P. Virtanen *et al.*, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2).
- [34] R. S. Willey and H. L. Yu, “The Decays $K^\pm \rightarrow \pi^\pm \ell^+ \ell^-$ and Limits on the Mass of the Neutral Higgs Boson,” *Phys. Rev. D*, vol. 26, p. 3287, 1982. DOI: [10.1103/PhysRevD.26.3287](https://doi.org/10.1103/PhysRevD.26.3287).
- [35] R. Chivukula and A. V. Manohar, “Limits on a light higgs boson,” *Physics Letters B*, vol. 207, no. 1, pp. 86–90, 1988, ISSN: 0370-2693. DOI: [https://doi.org/10.1016/0370-2693\(88\)90891-X](https://doi.org/10.1016/0370-2693(88)90891-X). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/037026938890891X>.
- [36] B. Grinstein, L. J. Hall, and L. Randall, “Do B meson decays exclude a light Higgs?” *Phys. Lett. B*, vol. 211, pp. 363–369, 1988. DOI: [10.1016/0370-2693\(88\)90916-1](https://doi.org/10.1016/0370-2693(88)90916-1).
- [37] M. W. Winkler, “Decay and detection of a light scalar boson mixing with the Higgs boson,” *Phys. Rev. D*, vol. 99, no. 1, p. 015018, 2019. DOI: [10.1103/PhysRevD.99.015018](https://doi.org/10.1103/PhysRevD.99.015018). arXiv: [1809.01876](https://arxiv.org/abs/1809.01876) [hep-ph].
- [38] M. Cacciari, G. P. Salam, and G. Soyez, “The anti- k_t jet clustering algorithm,” *JHEP*, vol. 04, p. 063, 2008. DOI: [10.1088/1126-6708/2008/04/063](https://doi.org/10.1088/1126-6708/2008/04/063). arXiv: [0802.1189](https://arxiv.org/abs/0802.1189) [hep-ph].
- [39] S. Catani, Y. L. Dokshitzer, M. H. Seymour, and B. R. Webber, “Longitudinally invariant K_t clustering algorithms for hadron hadron collisions,” *Nucl. Phys. B*, vol. 406, pp. 187–224, 1993. DOI: [10.1016/0550-3213\(93\)90166-M](https://doi.org/10.1016/0550-3213(93)90166-M).
- [40] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

Appendix

Simulation details

Our first signal benchmark is that of single, real scalar field (S) which mixes with the SM Higgs boson. As a result of this mixing, there exist a penguin diagram which induces an exotic decay mode of B -mesons to S plus a strangeness-carrying final state [34–36]. Despite the loop-suppression of this partial width, the branching ratio may nevertheless be appreciable due to the very small total width of the B -meson. The branching ratio for $S \rightarrow 2\mu$ is rather uncertain, but is thought to be between 10% and 1% for most of the mass range where $2m_\mu < m_S < m_B - m_K$ [37]. Phenomenologically, the main signature of the model is a low mass (displaced) dimuon resonance, which often sits inside a b -jet. For the purpose of our study, we set $m_S = 0.4$ GeV and work with a leading order inclusive unweighted b - b sample subject to $p_T > 10$ GeV at the parton-level.

Our second signal benchmark is a hidden valley model [25], in which the SM Higgs decays to a dark sector with some confining dynamics. This includes a parton shower in the dark sector, followed by a hadronization into dark mesons. Some of these dark mesons are taken to be stable and therefore invisible, while others can decay to a pair of dark photons through a dark sector chiral anomaly. These dark photons may subsequently decay to SM final states, among which dimuons. (See [26] for a detailed description and definition of the model.) We take the dark photon mass to be 0.4 GeV, to match the dimuon invariant mass chosen for our first benchmark. The signature of this model is a low mass dimuon resonance, surrounded by hadronic activity plus a small amount of missing energy. The kinematics of the muon pair is very similar in both models, but the pattern of hadronic activity and missing energy in the event can differ substantially.

While analysis in [6] is able to record hadronic activity in a coarse-grained manner, by recording anti- k_T jets [38], our approach is able to retain more fine-grained substructure information, which may ultimately be needed to distinguish similarly looking signatures.

VAE architecture and training details

In our experiments, the encoder network consists of four 1D convolution layers with filter size 1024, kernel size 1, and stride 1, followed by a sum layer, followed by four dense layers of size 1024. β is also provided as an input to these dense layers, in parallel with the sum layer. Unless otherwise specified, all layers have activation function Leaky ReLU with negative slope coefficient of 0.1. 256 latent space $\mu, \log \sigma^2$ are encoded with linear activation. Sampled codes z are used as input for a decoder, alongside β . The decoder consists of five layers with size 1024, followed by a linear dense

layer which outputs fifty particles represented as $\{(p_T/H_T, \eta, \sin \phi, \cos \phi)\}$, and then an arctan function reduces this to $\{(p_T/H_T, \eta, \phi)\}$. The explicit arctan allows the network to avoid learning a discontinuity in ϕ , which is also the motivation for the trigonometric form of the inputs.

In order to reduce the number of particles to less than 50 for computational speed, events that have more than 50 particles are reclustered using the exclusive- k_T algorithm [39] with radius parameter $R = 1.0$ and terminating when there are exactly 50 particles left in the event. The VAE works with p_T -normalized inputs, and so the p_T for each event is rescaled by $H_T = \sum_i p_{T,i}$ before being input into the encoder. This H_T is recorded separately for each event, and can be used to rescale the event reconstructions.

We trained the VAE on the $b\bar{b}$ background events. Each batch of training data consists of an array of shape $(n_{\text{batch}}, 50, 4)$, with $n_{\text{batch}} = 100$, which is input to the VAE encoder, and an array of shape (n_{batch}) samples for β , which is an auxiliary input to the encoder and decoder. These samples are drawn from a log-uniform distribution in the range $\log_{10} \beta \in [-6, 1]$. The loss for each training example is also calculated using this β . Training is performed using the Adam optimizer [40] with learning rate initialized at 3×10^{-5} . Each epoch consists of 1000 batches (as opposed to the entire training dataset), and the learning rate is decreased in factors of $\sqrt{10}$ each time the validation loss has not improved in 10 epochs. Training is stopped after validation loss has not improved in 20 epochs. This entire cycle is repeated 10 times.

The code for this paper can be found at https://github.com/erichuangyf/EMD_VAE/tree/neurips_ml4ps.

Classifier architecture and training details

The PFN model used to classify reconstructed signals consists of three time-distributed dense layers of size 128 followed by a sum layer followed by three dense layers of size 128. Each PFN was trained on events with shape $(n_{\text{batch}}, 50, 3)$, with the batch size $n_{\text{batch}} = 1000$. Training is performed using the Adam optimizer with learning rate initialized at 10^{-3} . The learning rate is decreased in factors of $10^{1/4}$ each time the validation loss has not improved in 5 epochs. Training is stopped after 200 epochs or the validation loss has not improved in 10 epochs.

We also trained the PFN classifier on the hardest two muons taken from the two signal event samples. The training procedure is similar to the PFN on full signal events, except the input shape for each event was truncated to $(n_{\text{batch}}, 2, 3)$, with the batch size $n_{\text{batch}} = 1000$. The muon-only PFN model consists of three time-distributed dense layers of size 128 followed by a sum layer followed by three dense layers of size 128. Training is performed using the Adam optimizer with learning rate initialized at 10^{-3} . The learning rate is decreased in factors of $10^{1/4}$ each time the validation loss has not improved in 5 epochs. Training is stopped after 200 epochs or the validation loss has not improved in 10 epochs.

All training was performed on NVidia Quadro RTX 6000 with CUDA version 11.2 using TensorFlow version 2.4.1. The training time for the VAE model was about 24 hours and the training time for each PFN model was about 5 minutes based on this setup.

Potential Broader Impacts

Compression is used throughout science and industry and so our observations may be useful to setups beyond collider physics.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to **[Yes]**, **[No]**, or **[N/A]**. You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? **[Yes]** See Section 5.

- Did you include the license to the code and datasets? [N/A] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
 - (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
 - (b) Did you describe the limitations of your work? [Yes]
 - (c) Did you discuss any potential negative societal impacts of your work? [Yes]
 - (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? [Yes]
 - (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]
 - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
 - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
 - (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]
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? [N/A]
 - (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]