

Integrating particle flavor into deep learning models for hadronization

Jay Chan¹, Xiangyang Ju¹, Adam Kania², Benjamin Nachman^{3,4}, Vishnu Sangli^{5,3} and Andrzej Siodmok²

¹Scientific Data Division, *Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*

²*Jagiellonian University, Krakow, Poland*

³Physics Division, *Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*

⁴*Berkeley Institute for Data Science, University of California, Berkeley, California 94720, USA*

⁵*Department of Physics, University of California, Berkeley, California 94720, USA*



(Received 26 January 2025; accepted 19 May 2025; published 17 June 2025)

Hadronization models used in event generators are physics-inspired functions with many tunable parameters. Since we do not understand hadronization from first principles, there have been multiple proposals to improve the accuracy of hadronization models by utilizing more flexible parametrizations based on neural networks. These recent proposals have focused on the kinematic properties of hadrons, but a full model must also include particle flavor. In this paper, we show how to build a deep learning-based hadronization model that includes both kinematic (continuous) and flavor (discrete) degrees of freedom. Our approach is based on generative adversarial networks and we show the performance within the context of the cluster hadronization model within the Herwig event generator.

DOI: [10.1103/hgbg-k7js](https://doi.org/10.1103/hgbg-k7js)

I. INTRODUCTION

Despite the extensive predictability of quantum chromodynamics, the theory of the strong force, we cannot yet calculate how the fundamental degrees of freedom (quarks/gluons) combine to form the observable states (hadrons). At the same time, it is essential that we be able to model this transition in order to connect first-principles, perturbative calculations with data. Event generators in wide use are based on one of two physically inspired parametric models with many tunable parameters: the cluster model [1] (default in Herwig [2–5] and Sherpa [6,7]) and the string model [8,9] (default in Pythia [10,11]). These models have enabled a wide array of physics results across particle and nuclear physics. However, it is also well known that these models do not describe all regions of phase space. As the science requires more precision and examines more extreme regions of phase space, new and systematically improvable hadronization models are needed.

Deep generative neural networks are promising tools to enhance the precision of hadronization models due to their flexibility [12]. There is a long history of neural networks for modeling nonperturbative functions [13] and recent studies have shown that deep generative models can emulate the string and cluster hadronization when trained

on paired sets of partons and hadrons [14,15]. These techniques have also been extended to the realistic setting where a pairing is not known and only hadrons are observed [16,17]. However, all of the studies so far have trained on simplified simulations without any parton/hadron flavor.

In this paper, we extend the HadML [15,16] setup to include parton and hadron flavor. This is challenging because both continuous (kinematic) and discrete (flavor) information must be generated at the same time. HadML is based on a generative adversarial network (GAN) [18,19] because it naturally accommodates the realistic case mentioned above¹ [16].

This paper is organized as follows. Section II introduces the machine learning methods of the new HadML model and how they are implemented in practice. The dataset we use to stress-test this model, based on Herwig 7 (H7) [5], is described in Sec. III and numerical results are presented in Sec. IV. The paper ends with conclusions and outlook in Sec. V.

II. METHODS

A. Generative adversarial network framework

The overall setup is similar to that in the previous work [15]. A conditional generator function $G(z, \lambda; \omega_G)$ with the

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

¹In fact, one can also view the MLHad approach [17] as a GAN where the generator is parametrized as a normalizing flow [20,21] and the discriminator is similar to the Wasserstein GAN setup [22]. They also proposed a clever variation to avoid regenerating events in each epoch of training through reweighting.

parameters ω_G is learned to map the initial cluster properties onto the properties of the two² hadrons from each cluster decay $\{h_1, h_2\} \in \mathbb{R}^{2N_h}$. In addition to the four-momenta, the generator function should also output the particle types of the two hadrons. Here, $z \in \mathbb{R}^{N_z}$ is the input noise variable sampled from the prior $p(z)$, and $\lambda \in \mathbb{R}^{N_\lambda}$ is the conditional variable. In Ref. [15], the generator function was conditioned on the cluster four-momentum (E, p_x, p_y, p_z) . In this paper, we consider additional conditional variables from the two incoming cluster-forming quarks, including their four-momenta and particle types. Since two hadrons from a cluster decay must be back-to-back in the rest frame of cluster, the generator G can output the polar angles θ and ϕ of the “first hadron” in the cluster rest frame instead of the 4-momenta of both hadrons. Similarly, the incoming quarks are back-to-back in the cluster rest frame. Therefore, we parametrize their four-momentum as the polar angles θ and ϕ of the “first quark” in the cluster rest frame. Note that here ϕ is defined in the range of $(-\pi/2, \pi/2)$, and the hadron (quark) with ϕ in this range is defined to be the first hadron (quark).

A discriminator function $D(\theta_{h_1}, \phi_{h_1}, \text{PID}_{h_1}, \text{PID}_{h_2}; \omega_D)$, parametrized with ω_D , is learned to represent the probability that $\{\theta_{h_1}, \phi_{h_1}, \text{PID}_{h_1}, \text{PID}_{h_2}\}$ came from cluster fragmentation rather than the generator G . Note that θ_{h_1}, ϕ_{h_1} are the polar angles of the first hadron, and $\text{PID}_{h_1}, \text{PID}_{h_2}$ are the particle types of the two hadrons. G and D are trained alternately where G is trained to maximize the loss function

$$L_G = -\mathbb{E}_{\lambda \sim H7, z \sim p(z)} (\log(D(\tau(\lambda))) + \log(1 - D(G(z, \lambda)))), \quad (2.1)$$

where τ is the cluster fragmentation, and D is trained to minimize the loss function

$$L_D = -\mathbb{E}_{\lambda \sim H7, z \sim p(z)} (\log(D(\tau(\lambda))) + \log(1 - D(G(z, \lambda)))) + \gamma R_1(\omega_D), \quad (2.2)$$

where γ is a regularization weight, which we set to 200. We use R_1 regularization [23] on real data points

$$R_1(\omega_D) = \mathbb{E}_{\lambda \sim H7} [\|\nabla_{\tau(\lambda)} D_{\omega_D}(\tau(\lambda))\|^2]. \quad (2.3)$$

B. Machine learning implementation

The generator (G) and discriminator (D) functions are both parametrized as neural networks. Each of them is a fully connected network with four hidden layers, each with

²The decay of heavy clusters can produce more than two hadrons, but in most cases the collisions we consider produce mainly light clusters that decay into two hadrons. Therefore, in this study, we have limited ourselves to the case of decay into two hadrons, and we will investigate more complex decays in future work.

a width of 1,000 neurons. All intermediate layers in these networks use a LeakyReLU [24] activation function.

The nondiscrete conditional inputs of G are normalized to the range of $(-1, 1)$, whereas the noise prior p is a Gaussian distribution with a mean of 0 and width of 1. The noise dimension N_z is set to 64. The last layer of G is divided into the variables $\kappa \in \mathbb{R}^2$, which correspond to hadron kinematics, and the variables $\pi \in \mathbb{R}^{2N_t}$, which correspond to hadron types. Here, N_t is the number of hadron types considered. For simplicity, we consider only the 40 most common hadron types (i.e., $N_t = 40$).³ The hadron kinematics θ_{h_1} and ϕ_{h_1} are extracted from κ with a tanh activation function, as in the previous work [15]. The hadron type, on the other hand, is a categorical variable. In order to avoid zero gradients when using *argmax* in training, we use the Gumbel-Softmax [25] distribution to approximate the distribution of hadron types:

$$y_i = \frac{\exp((\log \pi_i + g_i)/\tau)}{\sum_i \exp((\log \pi_i + g_i)/\tau)}, \quad (2.4)$$

where g_i are independent and identically distributed samples drawn from $\text{Gumbel}(0, 1)$. τ is a temperature parameter and as it approaches 0 the Gumbel-Softmax distribution becomes identical to the categorical distribution. We anneal τ by linearly decreasing it from 1.0 to 0.1 during training. The hadron type distributions y from the Gumbel-Softmax distribution are then taken as the inputs for D , in addition to the hadron kinematics θ_{h_1} and ϕ_{h_1} . During inference, the generated hadron types are obtained from the Gumbel-Softmax distribution with the *argmax* operation. The last layer of D uses a sigmoid activation function.

All neural networks are implemented and trained using PyTorch [26]. The generator and discriminator are optimized alternately with Adam [27] with a learning rate of 3×10^{-4} for both networks. The training uses a batch size of 40,000 and is performed for 25 epochs. The hyperparameters are optimized with weights and biases [28].

III. DATASET

The dataset was generated by the Herwig 7.2.1 Monte Carlo generator, which by default uses a cluster hadronization model. In the first step of the cluster model, partons are grouped into colorless objects called clusters (exited prehadrons), which then decay into two hadrons (or lighter clusters⁴). Since in our study, we wanted to integrate the

³The most common hadron types are identified from an independent and slightly different simulation sample generated by H7. This is why the frequencies reported in Fig. 2 are not strictly decreasing.

⁴The heavier clusters can also decay into lighter clusters before decaying into hadrons. However, since in this publication, we are mainly interested in the generation of individual hadron flavor. We leave the decays of the heavy clusters for a future follow-up paper.

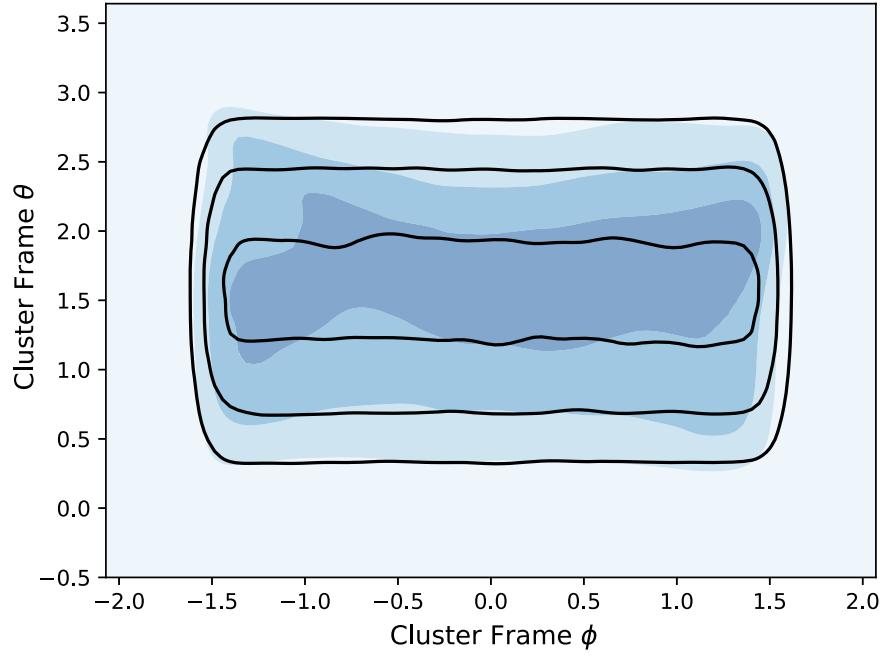


FIG. 1. Flavor-inclusive density contour of ϕ and θ in the cluster frame for the nominal dataset (solid line) as well as for the dataset created from the fitted HadML model (filled colors).

particle flavor into the HadML model, in addition to the kinematic information (the four momenta of all light clusters in the event and their decay products, hadrons), our dataset also contains information about the type of primary hadrons⁵ (PID_{h_1} , PID_{h_2}), as well as the Particle Data Group [29] identification (PDG ID) of the partons that make up a given light cluster. All datasets were generated in electron-positron collisions at an energy of 91.2 GeV, which corresponds to events recorded by LEP experiments at CERN. Data from LEP is crucial for fitting hadronization models, therefore such a sample is the most natural for the development of new hadronization approaches. To test whether the HadML model can adapt to different flavor compositions, we prepared two datasets with different settings of the cluster model parameters responsible for the generation of hadron types. To be more precise, the nominal dataset was generated using H7's default settings. For the variational dataset, we have maximized the weights for producing charm quark-antiquark pairs,⁶ strange quark-antiquark pairs and diquark-antiquark pairs as well as the relative weight SngWt for the production of singlet baryons and

the relative weight DecWt for the production of decuplet baryons in cluster hadronization.⁷

IV. RESULTS

The two datasets described in Sec. III have the same distributions of θ_h and ϕ_h , which are nearly independent of hadron type. As in our previous works, these distributions are well modeled by HadML (Fig. 1) and are essentially uniform in ϕ and Gaussian-like in θ .

The lab-frame spectra of energies and angles differ between the nominal and alternative samples because of the differences in hadron masses. Since the masses are known, the lab-frame properties are therefore determined by how well we model the frequency of the various flavor types. Figure 2 shows the frequencies for H7 and HadML, inclusive in the flavor of the quark types composing the decaying cluster. As expected, the pions are the most frequent (PDG IDs 111 and ± 211), followed by the ρ (PDG IDs 113 and ± 213) and ω (PDG ID 223). Next are the kaons (PDG ID 3xx), protons (PDG ID ± 2212), and neutrons (PDG ID ± 2112). A series of other intermediate hadronic resonances follow the lightest baryons. The nominal and alternative H7 models significantly differ in these rates, most notably for the pion versus ω production and in the rate of baryons. HadML captures these

⁵Primary hadrons are hadrons produced directly from cluster decays.

⁶Even though the weights for producing charm quark-antiquark pairs are maximized, the rate of charm hadrons is still very low compared to other hadron types. Therefore, they do not show up in the hadron type distributions where only the 40 most common hadron types are selected.

⁷To achieve this, we used the following H7's settings: HadronSelector: PwtDIquark = 10, HadronSelector: PwtBquark = 10, HadronSelector: PwtCquark = 10, HadronSelector: SngWt = 10 and HadronSelector: DecWt = 10. For details, please see H7's manual [3].

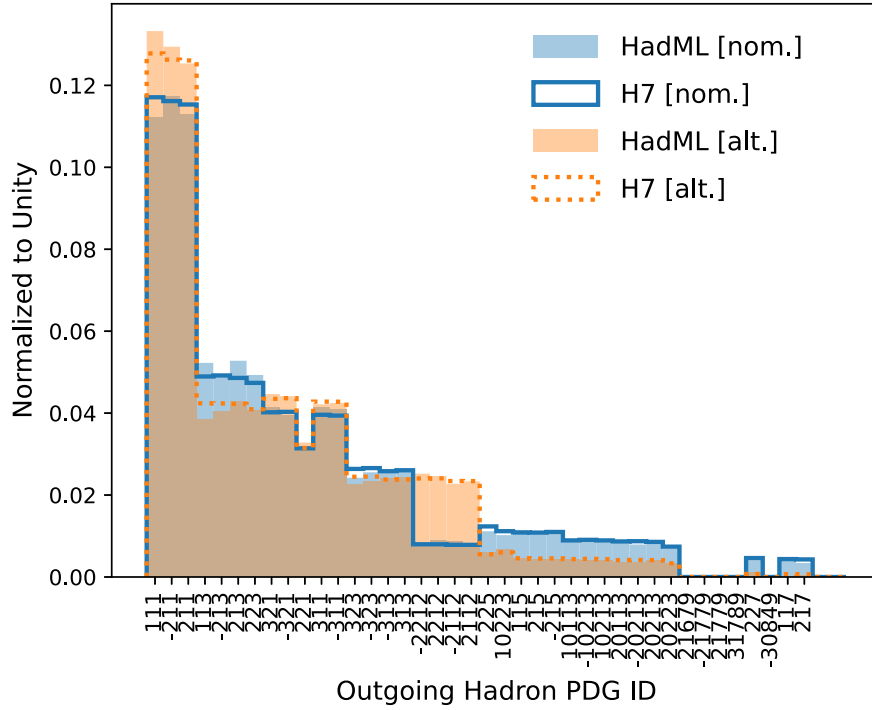


FIG. 2. The Particle Data Group Identification (PDG ID) [29] of hadrons generated by the nominal and alternative H7 datasets as well as for datasets created from the fitted HadML models.

trends across the full spectrum at $\mathcal{O}(1\%)$ precision. This is true even for the large drop in frequency between the pions and all other hadrons as well as between the large raise in baryon production between the nominal and alternative H7 models.

Since HadML is conditioned on the flavor of the constituent quarks, we can also investigate the relationship between the cluster composition and the resulting hadron types. We expect that if one of the quarks is strange, then the

hadrons should be biased toward strange hadrons (e.g., kaons). This is what we see in Fig. 3, which compares the inclusive hadron flavor distribution with the spectrum after requiring that at least one of the incoming quarks is strange. There are large changes between the inclusive and conditional distributions, which are well reproduced by HadML. The neural network is also able to learn that when both quarks are strange, then both outgoing hadrons must be strange.

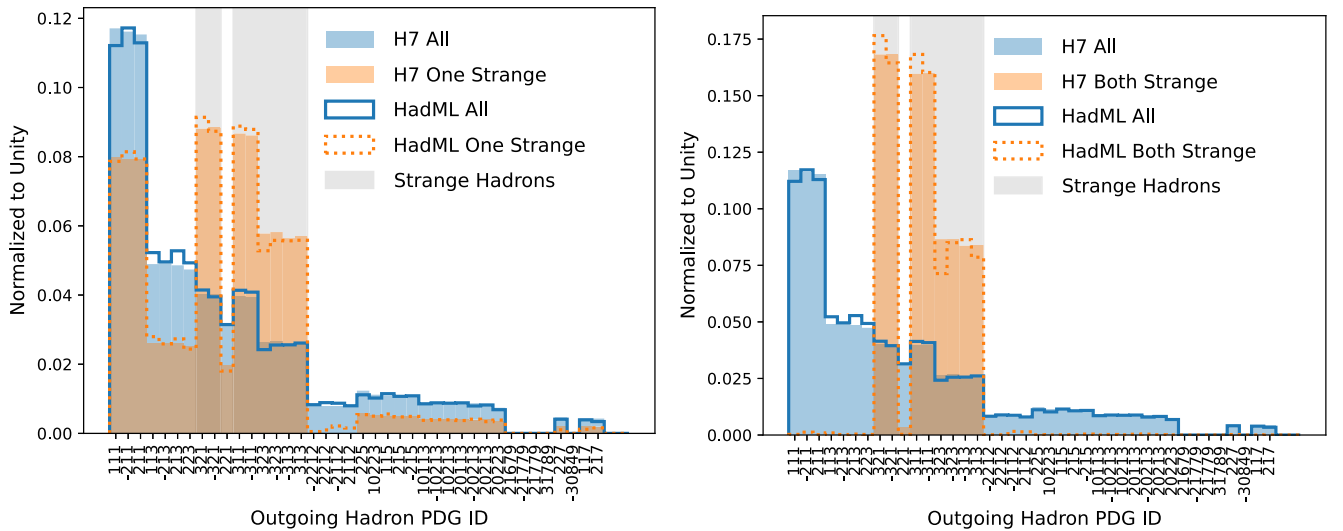


FIG. 3. Left (Right): The PDG ID of hadrons generated inclusively and requiring that at least one (both) of the quarks composing the cluster are strange for H7 and for the HadML model.

Lastly, we further interrogate the model’s ability to respect physical symmetries associated with particle flavor. Unlike symmetries associated with kinematic properties, we do not explicitly enforce flavor symmetries. However, we can examine them *post hoc*. Many data analyses are not sensitive to flavor violation at the level of $\mathcal{O}(1\%)$ because they marginalize over flavor and/or their detector is not sensitive to differences in particle types below that level. If the violation is at the $\mathcal{O}(1\%)$ and it is necessary to enforce them exactly, one can resample from the GAN without much computational cost. In the future, it would be interesting to explore explicit flavor conservation in the model or to allow for some small amount of flavor violation for approximate symmetries (e.g., the total number of π^+ and π^- is not exactly twice the number of π^0). For example, we find that with our trained model, $n_{\pi^+}/n_{\pi^-} = 1.0(002)$, $n_{\rho^+}/n_{\rho^-} = 0.93(002)$, $n_{K^+}/n_{K^-} = 1.1(0.003)$, $n_p/n_{\bar{p}} = 0.93(0.005)$, where the uncertainties are statistical. Additionally, we find charged-to-neutral ratios of 0.49(0.001) for pions, 0.51(0.001) for rho mesons, 1.0(0.002) for kaons, and 0.97(0.004) for protons/neutrons. In the Appendix, we examined our model’s ability to preserve strangeness. All of these follow the expected trends and are consistent with the exact laws to $\mathcal{O}(1\%)$ even though the model was not explicitly penalized for nonpreservation of these laws (it was indirectly through the properties of the training dataset).

V. CONCLUSIONS

This paper marks an important milestone in the development of surrogate models for hadronization: the inclusion of hadron flavor. Hadronization is not understood from first principles so it is a natural candidate for the flexible modeling afforded by deep generative models. Previous works on machine learning-based hadronization had focused on the case of only pions and we extend the GAN-based HadML approach to include other hadron types.⁹ This required us to combine continuous (e.g., angles/momenta) with discrete

(particle types) in our generator. We accomplished this goal using the Gumbel-Softmax [25] distribution for hadron types, to enable differentiability.

The insights of this paper could be combined with our previous paper [16] to fit the HadML model with flavor to data in the lab frame. Additional work is also required to integrate all hadron types and to go beyond two-body decays of hadrons. Ultimately, we hope to create a model flexible enough to accommodate the cluster model, the string model, and nature.

The nominal and alternative H7 samples used for training can be found on Zenodo at [31]. Software for reproducing the plots can be found on GitHub at [32,33].

ACKNOWLEDGMENTS

The work of A. S. is funded by Grant No. 2019/34/E/ST2/00457 of the National Science Centre, Poland. A. K. acknowledges support by the Priority Research Area Digiworld under the program Excellence Initiative—Research University at the Jagiellonian University in Cracow. J. C., B. N., and X. J. are supported by the U.S. Department of Energy (DOE), Office of Science under Contract No. DE-AC02-05CH11231. Support for J. C. and X. J. was also provided through the Scientific Discovery through Advanced Computing (SciDAC) program funded by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research and High Energy Physics.

DATA AVAILABILITY

The data that support the findings of this article are openly available [31].

APPENDIX: STRANGENESS CONSERVATION

Since we currently do not impose explicit constraints on the HadML model to avoid flavor violation, it is possible the generative events can violate flavor conservation. Figure 4, as an example, shows how often strangeness conservation is violated. Note that it is allowed to violate by 2 since an s -quark pair ($s\bar{s}$) can be created out of the vacuum. Therefore, the errors the HadML model has made mainly occur in the first and the third bins in the figure, which is $\sim 2\%$ of the time. This indicates that HadML is able to preserve the flavors at certain degree even though there is no explicit constraint. Explicit constraints on flavor conservation can be added to the HadML model in the future.

⁸We checked this explicitly with our Herwig dataset.

⁹While this work was being finalized, Ref. [30] became the first paper to combine continuous and discrete outputs for generative modeling in HEP. They used a diffusion model with only a few discrete labels (particle types), both of which differ from (and are not directly applicable to) our setup. However, it would be interesting to explore possible connections between approaches in the future.

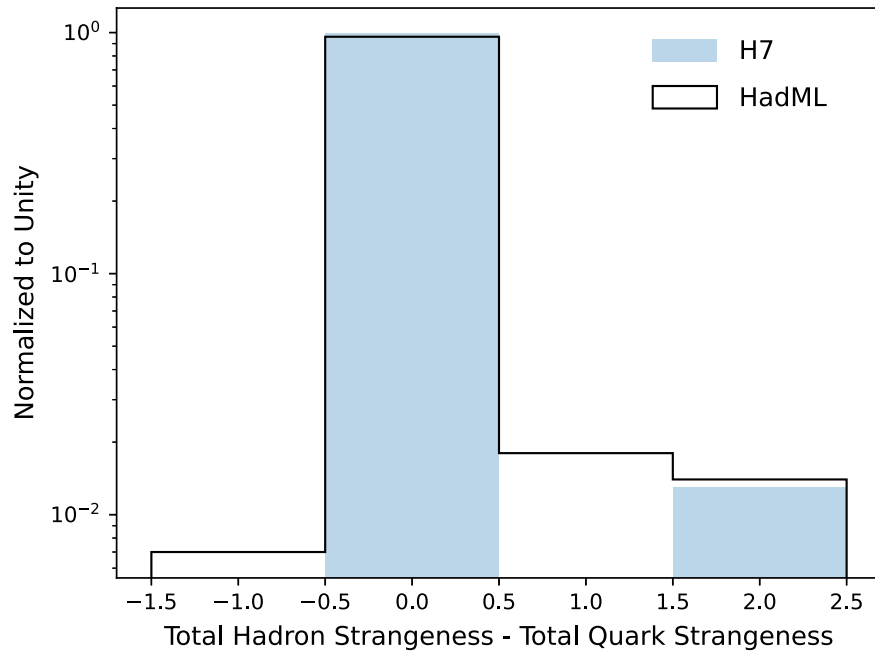


FIG. 4. Violation of strangeness in outgoing hadrons from incoming quarks.

-
- [1] B. R. Webber, A QCD model for jet fragmentation including soft gluon interference, *Nucl. Phys.* **B238**, 492 (1984).
 - [2] G. Corcella, I. G. Knowles, G. Marchesini, S. Moretti, K. Odagiri, P. Richardson, M. H. Seymour, and B. R. Webber, Herwig 6: An event generator for hadron emission reactions with interfering gluons (including supersymmetric processes), *J. High Energy Phys.* **01** (2001) 010.
 - [3] M. Bahr *et al.*, Herwig++ physics and manual, *Eur. Phys. J. C* **58**, 639 (2008).
 - [4] J. Bellm *et al.*, Herwig 7.0/Herwig++ 3.0 release note, *Eur. Phys. J. C* **76**, 196 (2016).
 - [5] J. Bellm *et al.*, Herwig 7.2 release note, *Eur. Phys. J. C* **80**, 452 (2020).
 - [6] T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert and J. Winter, Event generation with Sherpa 1.1, *J. High Energy Phys.* **02** (2009) 007.
 - [7] E. Bothmann *et al.* (Sherpa Collaboration), Event generation with Sherpa 2.2, *SciPost Phys.* **7**, 034 (2019).
 - [8] B. Andersson, G. Gustafson, G. Ingelman, and T. Sjostrand, Parton fragmentation and string dynamics, *Phys. Rep.* **97**, 31 (1983).
 - [9] T. Sjostrand, Jet fragmentation of nearby partons, *Nucl. Phys.* **B248**, 469 (1984).
 - [10] T. Sjostrand, S. Mrenna, and P. Z. Skands, A brief introduction to Pythia 8.1, *Comput. Phys. Commun.* **178**, 852 (2008).
 - [11] T. Sjostrand, S. Mrenna, and P. Z. Skands, Pythia 6.4 physics and manual, *J. High Energy Phys.* **05** (2006) 026.
 - [12] S. Badger *et al.*, Machine learning and LHC event generation, *SciPost Phys.* **14**, 079 (2023).
 - [13] R. D. Ball *et al.* (NNPDF Collaboration), The path to proton structure at 1% accuracy, *Eur. Phys. J. C* **82**, 428 (2022).
 - [14] P. Ilten, T. Menzo, A. Youssef, and J. Zupan, Modeling hadronization using machine learning, *SciPost Phys.* **14**, 027 (2023).
 - [15] A. Ghosh, X. Ju, B. Nachman, and A. Siodmok, Towards a deep learning model for hadronization, *Phys. Rev. D* **106**, 096020 (2022).
 - [16] J. Chan, X. Ju, A. Kania, B. Nachman, V. Sangli, and A. Siodmok, Fitting a deep generative hadronization model, *J. High Energy Phys.* **09** (2023) 084.
 - [17] C. Bierlich, P. Ilten, T. Menzo, S. Mrenna, M. Szewc, M. K. Wilkinson, A. Youssef, and J. Zupan, Towards a data-driven model of hadronization using normalizing flows, *SciPost Phys.* **17**, 045 (2024).
 - [18] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair *et al.*, Generative adversarial nets, in *Proceedings of the 27th International Conference on Neural Information Processing Systems—Volume 2, NIPS'14* (MIT Press, Cambridge, MA, USA, 2014), pp. 2672–2680, <http://dl.acm.org/citation.cfm?id=2969033.2969125>.
 - [19] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, Generative adversarial networks: An overview, *IEEE Signal Process. Mag.* **35**, 53 (2018).

- [20] D. Rezende and S. Mohamed, Variational inference with normalizing flows, in *Proceedings of the 32nd International Conference on Machine Learning* (2015), Vol. 37, pp. 1530–1538, [arXiv:1505.05770](https://arxiv.org/abs/1505.05770).
- [21] I. Kobyzev, S. Prince, and M. Brubaker, Normalizing flows: An introduction and review of current methods, *IEEE Trans. Pattern Anal. Mach. Intell.* **43**, C1 (2021).
- [22] M. Arjovsky, S. Chintala, and L. Bottou, Wasserstein generative adversarial networks, in *Proceedings of the 34th International Conference on Machine Learning*, edited by D. Precup and Y. W. Teh, Vol. 70 of Proceedings of Machine Learning Research (PMLR, 2017), pp. 214–223, <https://proceedings.mlr.press/v70/arjovsky17a.html>.
- [23] L. Mescheder, A. Geiger, and S. Nowozin, Which training methods for GANs do actually converge?, in *Proceedings of the 35th International Conference on Machine Learning*, edited by J. Dy and A. Krause, Vol. 80 of Proceedings of Machine Learning Research (PMLR, 2018), pp. 3481–3490, <https://proceedings.mlr.press/v80/mescheder18a.html>.
- [24] B. Xu, N. Wang, T. Chen, and M. Li, Empirical evaluation of rectified activations in convolutional network, [arXiv:1505.00853](https://arxiv.org/abs/1505.00853).
- [25] E. Jang, S. Gu, and B. Poole, Categorical reparameterization with gumbel-softmax, in *International Conference on Learning Representations* (2017), <https://openreview.net/forum?id=rkE3y85ee>.
- [26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan *et al.*, PyTorch: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., Vancouver, 2019), pp. 8024–8035.
- [27] D. Kingma and J. Ba, Adam: A method for stochastic optimization, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [28] L. Biewald, Experiment tracking with weights and biases (2020), <https://www.wandb.com/>.
- [29] R. L. Workman *et al.* (Particle Data Group), Review of particle physics, *Prog. Theor. Exp. Phys.* **2022**, 083C01 (2022).
- [30] J. Birk, E. Buhmann, C. Ewen, G. Kasieczka, and D. Shih, Flow matching beyond kinematics: Generating jets with particle-ID and trajectory displacement information, *Phys. Rev. D* **111**, 052008 (2025).
- [31] J. Chan, X. Ju, A. Kania, B. Nachman, V. Sangli, and A. Siodmok, Herwig dataset for hadml particle gan training, [10.5281/zenodo.10246934](https://zenodo.org/record/10246934) (2023).
- [32] J. Chan, X. Ju, A. Kania, B. Nachman, V. Sangli, and A. Siodmok, Code for hadml particle gan training, [10.5281/zenodo.10275487](https://zenodo.org/record/10275487) (2023).
- [33] <https://github.com/hep-lbdl/hadml/releases/tag/2.0.0>