

Generalized parton distributions from symbolic regression

Andrew Dotson,^{1,*} Zaki Panjsheeri,^{2,†} Anusha Reddy Singireddy,³ Douglas Q. Adams,² Marija Čuić,²
Emmanuel Ortiz-Pacheco,⁴ Saraswati Pandey,² Marie Boër,⁵ Gia-Wei Chern,²
Michael Engelhardt,¹ Gary R. Goldstein,⁶ Yaohang Li,³ Huey-Wen Lin,⁴
Simonetta Liuti^{2,‡} and Matthew D. Sievert¹

(EXCLAIM Collaboration)

¹*Department of Physics, New Mexico State University, Las Cruces, New Mexico 88003, USA*

²*Department of Physics, University of Virginia, Charlottesville, Virginia 22904, USA*

³*Department of Computer Science, Old Dominion University, Norfolk, Virginia 23529, USA*

⁴*Department of Physics and Astronomy, Michigan State University, East Lansing, Michigan 48824, USA*

⁵*Department of Physics, Virginia Tech, Blacksburg, Virginia, USA*

⁶*Department of Physics and Astronomy, Tufts University, Medford, Massachusetts 02155, USA*



(Received 2 July 2025; accepted 26 May 2026; published 29 September 2026)

Artificial intelligence/machine learning–informed symbolic regression is the next stage of scientific modeling. We utilize a highly customizable symbolic regression package “PySR” (Python symbolic regression) to model the x and t dependence of the flavor isovector combination $H_{u-d}(x, t, \xi)$ at $\xi = 0$. These PySR models were trained on generalized parton distribution (GPD) results provided by both lattice QCD and phenomenological sources Goldstein Gonzalez Liuti, Goloskokov Kroll, and Vanderhaeghen Guichon Guidal. We demonstrate, for the first time, the consistency and systematic convergence of symbolic regression by quantifying the disparate models through their Taylor expansion coefficients. In addition to PySR penalizing models with higher complexity and mean-squared error, we implement schemes that test specific physics hypotheses, including force-factorized x and t dependence and Regge behavior in PySR GPDs. We show that PySR can identify factorizing GPD sources based on their response to the force-factorized model. Knowing the precise behavior of the GPDs, and their uncertainties in a wide range in x and t , crucially impacts our ability to concretely and quantitatively predict hadronic spatial distributions and their derived quantities.

DOI: [10.1103/ynm6-znr4](https://doi.org/10.1103/ynm6-znr4)

I. INTRODUCTION

Recent developments in the phenomenology of QCD extend the reach of our current understanding of the internal structure of hadrons, as generated from quark and gluon dynamics. In particular, it is believed that one can map out the spatial distributions of quarks and gluons in hadronic systems by studying the correlations measured in exclusive scattering processes with more than one particle detected in the final state. This class of experiments sensitive to 3D

subatomic spatial structure encompasses all deeply virtual exclusive scattering (DVES) processes. DVES experiments are also believed to be sensitive to the quark and gluon orbital angular momentum—the missing component in the proton spin puzzle—and they are an important part of the physics program at the upcoming Electron Ion Collider, a state-of-the-art facility specifically planned for visualizing the internal structure of the proton, as well as of both light and heavy nuclei [1].

Extracting QCD matrix elements from DVES cross section and asymmetries is, understandably, a complex problem where the dynamic quark and gluon observables for the process, the generalized parton distributions (GPDs), are multidimensional functions of four kinematic invariants: x , the parton momentum fraction of the proton, ξ the fractional longitudinal momentum transfer between the initial and scattered proton, t , the four-momentum transfer squared proportional to the transverse component, and Q^2 , the virtual photon four-momentum squared. GPDs enter the

* Contact author: adots004@nmsu.edu

† Contact author: zap2nd@virginia.edu

‡ Contact author: sl4y@virginia.edu

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). 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.

cross section in integrated forms known as Compton form factors (CFFs) (we refer the reader to Refs. [2–4] for detailed reviews on the subject).

It is becoming increasingly clear that, even in the simplest leading-order case, the extraction of GPDs from data is intractable if the range of computational methods is not extended beyond the standard ones, i.e., including resources granted by artificial intelligence (AI) and machine learning (ML) based on pattern recognition, statistical analysis, and algorithmic decision making. On the other hand, an alternative source of information on GPDs is given by lattice QCD (LQCD) calculations, providing a framework for the direct computation of GPDs from fundamental QCD principles, that can be compared with experimental observables as they become available.

Regardless of whether GPDs will be ultimately inferred from experiment and/or from the calculation of LQCD matrix elements, possibly with the aid of ML methods, the physical information they carry through their dependence in the multiple kinematic variables they depend on, will remain challenging to disentangle. It is therefore appropriate to ask the question:

Can we identify specific regularities, factorization, or other symmetric behaviors in the multidimensional GPD data, that lead to the interpretation of specific physics content of the data?

Addressing the latter includes, in particular, the extraction of information on the spatial distributions of quarks and gluons inside the proton. Addressing the question of making progress toward the simultaneous extraction from data and interpretation of GPDs brought us to explore new developments in *physics explainable AI methods* that are designed to provide a new handle to overcome the inherent, game-stopping computational hurdles of the analysis. Toward this goal, we introduce here a symbolic regression (SR)–based analysis of GPDs.

SR is a relatively new AI technique that is ideal for physics applications in that it allows us to identify concise and interpretable mathematical expressions directly from data. Rooted in genetic programming, it has recently become a favored data-driven deep learning model discovery tool, in various sectors of theoretical physics. For instance, the SR package NeSymReS [5] has recently been used to extract fragmentation functions from identified charged hadron data in semi-inclusive deep inelastic scattering at COMPASS [6], and PySR (Python symbolic regression) was used in Ref. [7] to fit the proton structure functions in the Drell-Yan process.

We apply SR to the x and t dependence of GPDs from LQCD calculations, at a fixed value of the other kinematic variables, ξ and Q^2 . SR replaces the traditional

regression procedure of starting with a fixed functional form and optimizing a set of parameters, with constructing an arbitrary functional form by assembling a string of elementary operations and mutating the expression dynamically to adapt to the data. Like neural networks and other ML algorithms, SR has the advantage of removing the systematic bias due to a frozen or truncated functional form. But unlike other approaches, SR produces simple analytic expressions that optimize both the fit quality and the simplicity of the expression. As such, SR has a distinct advantage with regard to *interpretability*. This results not only in a greater ease of interpretation of the range of behaviors preferred by the fit, but also in the interpretation of what types of analytic forms are more physically sound behaviors.

The flexible analytic forms are particularly well suited to imposing physical constraints, which often take the form of integral or differential equations. Because of its ease of mathematical interpretation, SR can be used to greatly expedite the process of narrowing the search space of functional forms only to those which are physically reasonable. The simplest demonstration of this is through the post-facto selection of fit functions which obey a physical constraint such as a sum rule. But far more powerful is to directly modify the loss function used by the SR algorithm to determine the optimum functional form. Implementing a custom loss function in SR allows us to bias the optimization process to only construct functional forms which satisfy a given physical constraint. This allows one to reasonably explore the *space of all possible fits consistent with the physical constraint*, limited by the hyperparameters of the algorithm. Through its ability to robustly incorporate physical constraints, SR exhibits a remarkable potential for improvement in *extrapolation* by balancing fit quality, complexity, and physical constraints. In turn, controlling these features is a mandatory step for evaluating the Fourier transforms that will ultimately yield the quark and gluon spatial distributions of the proton. For these reasons, SR constitutes a profound new tool for discovery of the hadron structure.

Our analysis of GPDs was first motivated to find the physics in LQCD results that come straight out the Lagrangian with no physics in between. The initial scan of LQCD results with SR led to a first surprising outcome that the LQCD results that we had been analyzing vary independently in the x and t variables (or they display a factorized dependence in x and t): this question becomes important for visualizing the internal proton structure, since t is related to the spatial distance of a given quark and gluon from the proton’s relativistic center of momentum. A factorized result would imply that quark and gluon configurations carrying a different x momentum fraction all have the same radial dependence, a structure contradicting many scenarios in the phenomenology of QCD. In a second phase of our study, we addressed how to make this finding

rigorous, including the study of the kinematic settings under which factorization happens, and we devised a *convergence criterion*, that we propose as a method that can be applied generally in SR, beyond the scope of interpreting QCD results. The convergence criterion, named expansion coefficient clustering (ECC), is explained in detail in this manuscript. The symbolic expressions that we found are not in the format that one would directly expect from phenomenology, but they hint at new possible ways of interpreting results. All of the stages of our analysis have led us to devise methods to enable the exploration of the physics scenarios that are under wraps in the data or LQCD results, as it was in the first attempt, only that now we added layers of stringency and self-consistency.

This paper is organized as follows: in Sec. II we give an overview of GPDs and the steps involved toward their experimental observation. In Sec. III we introduce our

approach for SR, using, specifically PySR. In Sec. IV we present our results for the PySR best fit of the LQCD flavor nonsinglet GPDs, discussing various scenarios for the x and t dependence and their impact on the spatial density distributions. Finally, in Sec. V we present our conclusions and an outlook on our future program development.

II. GENERALIZED PARTON DISTRIBUTIONS: DEFINITIONS AND PROPERTIES

Generalized parton distributions are matrix elements of collinear, lightlike separated operators evaluated between hadronic states with different momentum and helicity (for reviews, see Refs. [2,3]):

- (i) In the vector sector, the GPDs H and E parametrize the leading order in QCD matrix element as (we adopt the same notation as in Ref. [8])

$$W_{\Lambda\Lambda'}^{[\gamma^+]}(p, p', q) = \int \frac{dz^-}{4\pi} e^{ixp^+z^-} \langle p', \Lambda' | \bar{q} \left(-\frac{z}{2} \right) \gamma^+ \mathcal{U} \left[-\frac{z}{2}, \frac{z}{2} \right] q \left(\frac{z}{2} \right) | p, \Lambda \rangle \Big|_{z^+=z_T=0} \\ = \frac{\bar{u}(p', \Lambda') \gamma^+ u(p, \Lambda)}{2P^+} H(x, \xi, t, Q^2) + \frac{\bar{u}(p', \Lambda') i\sigma^{+\nu} \Delta_\nu u(p, \Lambda)}{2M(2P^+)} E(x, \xi, t, Q^2), \quad (1)$$

where $p(p')$ and $\Lambda(\Lambda')$ are the incoming (outgoing) proton momenta and helicities, respectively, $P = (p + p')/2$, $\mathcal{U}$ is a collinear gauge link (Wilson line), and we employ light-front coordinates $p^\pm \equiv (2^{-1/2})(p^0 \pm p^3)$. There exist two independent GPDs for the two different proton helicity states (same, H , and flip, E), and similar definitions can be given for GPDs in the axial-vector sector, $\tilde{H}$, $\tilde{E}$, as well as in the gluon and chiral odd sectors (see, e.g., Ref. [2] for a detailed review). The relevant kinematic variables, Fig. 1, are defined as

$$Q^2 = -(k_e - k'_e)^2, \quad x_{Bj} = \frac{Q^2}{2(pq)}, \quad t = \Delta^2 = -\frac{4M^2\xi^2}{1-\xi^2} - \frac{\Delta_T^2}{1-\xi^2}, \quad \xi = -\frac{(\Delta q)}{2(Pq)} = \frac{x_{Bj}(1 + \frac{t}{2Q^2})}{2 - x_{Bj} + \frac{x_{Bj}t}{Q^2}}, \quad x = \frac{kq}{(Pq)}, \quad (2)$$

where $q = k_e - k'_e$, and k is the struck quark momentum.

- (ii) GPDs can be measured in experiments since they enter the parametrization of the amplitude for DVES. Within a QCD factorized picture [9–12], they enter the amplitude multiplied by Wilson coefficient functions calculable in perturbative QCD. GPDs define the integrated quantities over the longitudinal momentum fraction, x , known as CFFs, which can be schematically written as

$$CFF(\xi, t, Q^2) = \int dx K(x, \xi, Q^2) GPD(x, \xi, t, Q^2), \quad (3)$$

where K is a perturbative QCD calculable kernel. This factorization theorem is illustrated in the Feynman diagram for deeply virtual meson production shown in Fig. 1. To leading power in Q^2 in the factorization theorem, all DVES processes, including

deeply virtual Compton scattering [13], deeply virtual meson production, and timelike Compton scattering, are parametrized in terms of the same

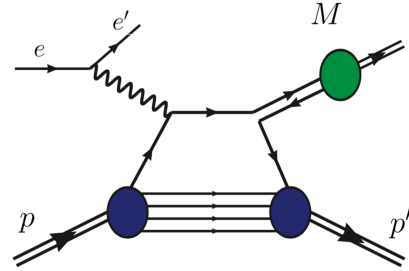


FIG. 1. Kinematics for a DVES process, $ep \rightarrow e'p'M$, where M is a meson. The relevant kinematic variables are given by the hard scale, or the four-momentum difference squared between the initial and final electron, $Q^2 = -(k_e - k'_e)^2$, the initial (p) and final (p') proton momenta, their difference, $\Delta = p - p'$, with $\Delta^2 = t$, and the meson momentum, q' .

building blocks—the GPDs—with different kinematic coefficients.

- (iii) The GPDs H and E , in the vector sector, respectively, define the integrands for the Dirac and Pauli form factors of the nucleon,

$$\begin{aligned} A_{10}(t) &\equiv \int_{-1}^1 dx H_q(x, \xi, t, Q^2) = F_1(t), \\ B_{10}(t) &\equiv \int_{-1}^1 dx E_q(x, \xi, t, Q^2) = F_2(t), \end{aligned} \quad (4)$$

where the notation on the lhs, A_{n0}, B_{n0} , with n integer, adopted in most of the LQCD literature, follows from the property of polynomiality, which is in turn a reflection of the QCD operator product expansion [14]. The variable ξ drops out of the $n = 1$ integrated forms as a consequence of Lorentz covariance. The $n = 2, 3$ Mellin moments in x are defined below:

$$A_{20}(t, Q^2) + (2\xi)^2 C_{20}(t, Q^2) \equiv \int_{-1}^1 dx x H_q(x, \xi, t, Q^2), \quad B_{20}(t, Q^2) - (2\xi)^2 C_{20}(t, Q^2) \equiv \int_{-1}^1 dx x E_q(x, \xi, t, Q^2), \quad (5a)$$

$$A_{30}(t, Q^2) + (2\xi)^2 A_{32}(t, Q^2) \equiv \int_{-1}^1 dx x^2 H_q(x, \xi, t, Q^2), \quad B_{30}(t, Q^2) + (2\xi)^2 B_{32}(t, Q^2) \equiv \int_{-1}^1 dx x^2 E_q(x, \xi, t, Q^2). \quad (5b)$$

A_{20} and B_{20} give access to angular momentum, through the sum rule [15]

$$\int_{-1}^1 dx x \sum_q (H_q(x, \xi, t, Q^2) + E_q(x, \xi, t, Q^2)) + H_g(x, \xi, t, Q^2) + E_g(x, \xi, t, Q^2) = J_q + J_g = \frac{1}{2}, \quad (6)$$

where the subscripts q and g run over the different flavor quarks and gluons, respectively.

- (iv) Through GPDs, one can evaluate the 2D quark and gluon spatial distributions in the transverse plane with respect to the proton motion, by performing a Fourier transform in the transverse momentum transfer Δ_T [Eq. (2)], for any given x . A physical interpretation is most transparent for $\xi = 0$, where we can define the spatial distribution for the unpolarized quark and gluon density in the transverse plane, $\rho^{q,g}$, as [16]

$$\rho_{q,g}(x, \mathbf{b}_T) = \int \frac{d^2 \Delta_T}{(2\pi)^2} e^{-i\mathbf{b}_T \cdot \Delta_T} H_{q,g}(x, 0, t, Q^2). \quad (7)$$

By studying the combined x and t (or $\mathbf{b}_T$) dependence, one can, in principle, infer the location, or radial distribution, of a given parton species for any given longitudinal momentum x , directly from data [16–19]. Depending on the dominance of different QCD mechanisms, it has been surmised that quarks and gluons with low momentum x will be further spread out in the proton volume with respect to quarks with large x , which, based on one-gluon exchange mechanisms, tend to concentrate in point-like configurations (see, e.g., the review in Ref. [20], as well as Ref. [21] and Ref. [22] and references therein).

A quantitative measure of these behaviors is given by the quark root mean square (RMS) transverse radius $\langle \mathbf{b}_T^2(x) \rangle^{1/2}$ weighted by the density in Eq. (7),

$$\mathbf{b}_{\text{RMS}}(x) = \langle \mathbf{b}_T^2(x) \rangle_{q,g}^{1/2} = \left(\frac{\int d^2 \mathbf{b}_T \mathbf{b}_T^2 \rho_{q,g}(x, \mathbf{b}_T)}{\int d^2 \mathbf{b}_T \rho_{q,g}(x, \mathbf{b}_T)} \right)^{1/2}. \quad (8)$$

It is important to notice that, if the x and t dependence factorizes in $H_{q,g}$, then all x configurations will have the same radial distribution, or $\mathbf{b}_{\text{RMS}}$ will be constant with x .

- (i) Finally, the GPDs H and $\tilde{H}$ also possess connections to ordinary parton distribution functions (PDFs) in the forward limit $t = \xi = 0$,

$$\begin{aligned} H_q(x, 0, 0, Q^2) &\equiv f_1^q(x, Q^2), \\ \tilde{H}_q(x, 0, 0, Q^2) &\equiv g_1^q(x, Q^2), \end{aligned} \quad (9)$$

where f_1 and g_1 are the standard spin-independent and spin-dependent density distributions.

- (ii) The flavor structure of GPDs is analogous to the PDFs and is summarized in the Appendix A.

Here we notice that we may also decompose the quark flavor dependence into isovector and isoscalar components,

$$H_{u-d} \equiv H_u - H_d, \quad (10a)$$

$$H_{u+d} \equiv H_u + H_d. \quad (10b)$$

Analogous expressions can be written for the GPD, $E_{q,g}$.

The various layers of convolution between the GPDs, CFFs [Eq. (3)], and the exclusive cross section (see Ref. [23] for a detailed description) illustrate the nested inverse problems confronting the extraction of the multidimensional GPDs. Characterizing spatial densities, Eq. (7), requires a systematic strategy to solve this inverse problem in a challenging multidimensional space. Even doing so, it is clear that in order to extract physically meaningful quantities from data/LQCD calculations, one needs to go beyond accepting the analysis outcome as a “black box” prediction, and understand the “why,” or an existing general rule behind it.

III. SYMBOLIC REGRESSION AND TRAINING CRITERIA

SR is a machine learning technique which generates and adapts symbolic mathematical expressions to fit a given dataset. Unlike traditional regression techniques, in which a functional form is specified *a priori*, thereby introducing a systematic bias, in SR the ultimate functional form is flexible. By iteratively adapting symbolic expressions using a given set of mathematical operations, SR optimizes a loss function which takes into account not only the fit quality, or its accuracy—minimizing the mean square error—but also the complexity of the expression which ultimately provides the best “interpretable” relation. By allowing for changes in the functional forms, SR eliminates much of the arbitrary subjectivity involved in fitting a dataset. In essence, the otherwise *ad hoc* human-led choices implicit in the standard regression procedure are replaced by the systematic *hyperparameters* and algorithmic choices of the machine learning. At the same time, differently from a standard neural network (NN) model, by returning results cast in a symbolic form, SR provides insight in the behavior of data that is directly readable, and that can, therefore, conduce to uncovering the physics relations underlying the data. Our analysis uses the package PySR which provides the advantage of being both versatile and highly customizable (see Sec. III A and Refs. [24,25] for a complete description of the algorithms and inputs that drive the evolution of PySR models). The SR Loss function is built with both *mean square error* (MSE) and complexity (Sec. III B).

A major hurdle in SR analyses is the *convergence*: with every SR run on the same dataset one can obtain ostensibly different functional forms of the selected models, which can all be considered to be equivalently accurate and of similar complexity. To extract a physics relation from data it, therefore, becomes mandatory to be able to discriminate one model from another. In Sec. III C we introduce a criterion for the *symbolic convergence* (SC) of SR models by casting model replicas in a common basis of Taylor

polynomials, and by subsequently dividing them into clusters based on the density of replicas with comparable expansion coefficients. This procedure is defined as ECC. ECC further develops the notion of SC beyond what is currently implemented in the literature.

As a specific case, we address the application of SR to LQCD results. LQCD enables calculations in the strong coupling regime of the theory—which is otherwise intractable—to perform simulations of the hadronic system on a discrete spacetime lattice, in a framework similar to the ones used in statistical mechanics. Analogously to NNs, albeit in a different context, LQCD predictions are, therefore, similar to a “black box”: while predicting with a quantitative error many of the properties of hadrons and their interactions, understanding and interpreting the physics mechanisms underlying these results still requires additional theoretical and computational efforts. In particular, LQCD calculations of GPDs, reviewed in Sec. III D, provide tables of values as a function of the quark and gluon longitudinal momentum fraction, x , similar to PDFs, as well as the invariant momentum transfer squared, t , between the initial and final protons. As explained in Sec. I, Eq. (8), GPDs give access through Fourier transformation, to the location of quarks and gluons inside the proton with respect to a relativistic center of momentum. It is important to understand whether and how the x and t dependence are correlated, for instance, whether or not increasing the momentum x of partons gives origin to a shrinking of their radial distributions, the specific quark flavor dependence of the radial distributions, and the relation between quark and gluon spatial configurations.

A. PySR: Symbolic regression for Python

PySR is a multipopulation evolutionary algorithm for generating these symbolic expressions which applies a classic evolutionary algorithm in parallel to a collection of separate, disconnected populations of expressions (“islands”) evolving independently. Other codes with different implementations are available to the community, and could be equivalently adopted in our analysis, in particular, the physics-inspired AI Feynman [26], and various other expression tree-based algorithms following Eureka [27]. These SR methods, as well as several others that have been generated recently, are summarized in the living review of Ref. [28]. Our choice of PySR is motivated by the ultimate goal of interfacing the SR part of our code with the other components within the broader goal of our analysis, which includes solving the inverse problem of the extraction of GPDs and spatial structure information from data (see Sec. I). By default, PySR creates and evolves $n_p = 40$ islands containing $L = 1000$ expressions each. These expressions are strings composed of elementary mathematical operations. In this work, we use the simple binary operations

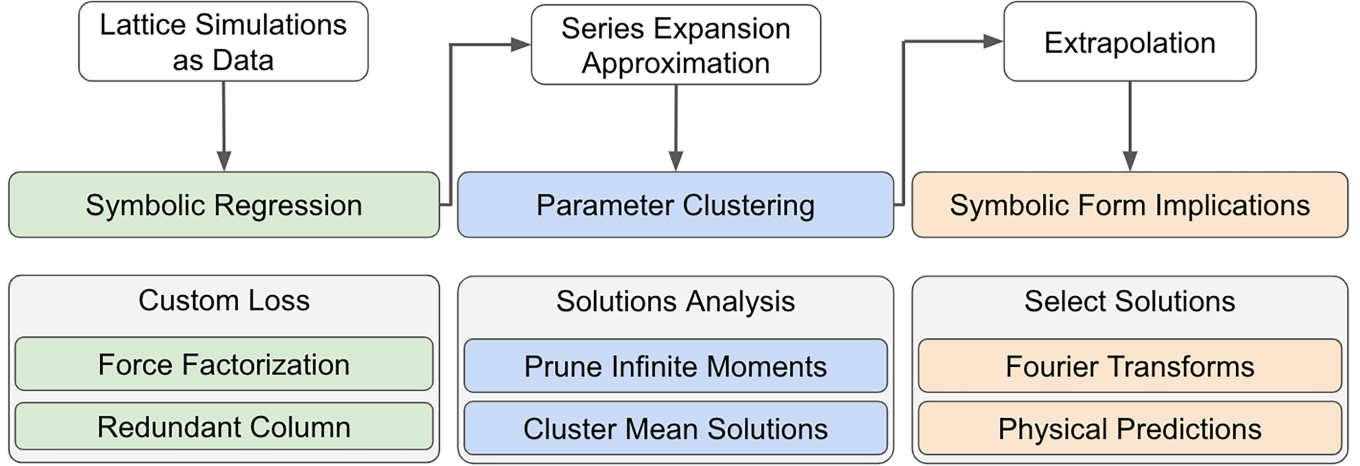


FIG. 2. Illustration of the procedure we used to investigate the properties of GPDs. To gain insight into the less-explored x -region of GPDs from lattice QCD, we consider both either lattice computations or existing GPD models as “data sources” to train machine learning algorithms. This approach extends the kinematic coverage of GPDs beyond the discrete lattice data, allowing us exploration of regions of x where direct lattice calculations are challenging. We attempt both NNs as well as SR as regression tools, leveraging their strengths to uncover underlying structures and potential parametrizations of GPDs in this region.

$$\begin{aligned}
 \text{Add}[x, y]: & \quad x + y, \\
 \text{Subtr}[x, y]: & \quad x - y, \\
 \text{Mult}[x, y]: & \quad x * y, \\
 \text{Div}[x, y]: & \quad x \div y, \quad \text{and} \\
 \text{Pow}[x, y]: & \quad x \wedge y.
 \end{aligned} \tag{11}$$

PySR also allows for the implementation of unary operators such as $\text{Log}[x]$, $\text{sin}[x]$, or more complex special functions like $\Gamma[x]$, but we do not include these in the present work. We choose to include the “Pow” binary operator (with constraint $\{“^”:(-1, 1)\}$) which allows more flexibility than default. The $(-1, 1)$ constraint allows unlimited complexity of the base, and exponent complexity limited to 1, which is either a constant or a variable. We have a “semi-Reggeized” approach which is done separately (Sec. IV A 3) and uses the fully default hyperparameters.

We choose the hyperparameters to be close to default, and list them entirely in the Appendix (Table III) for reproducibility. Notice that our analysis is mainly focused on showing that through physics-augmented PySR techniques, we can obtain functional forms which extrapolate in a controlled, interpretable way, independently from their particular, exact form. Therefore, at this level, to optimize the hyperparameters does not impact the results of our work. Future explorations of this framework may involve modifying them further.

PySR’s multipopulation nature allows for efficient parallelization, and it also enables “migration” between the islands: the incorporation of a successful expression from one island as a subexpression within another island. PySR allows a fixed probability $\alpha_M = 5\%$ by default to replace an expression with one of the best expressions from the other

islands, as well as a fixed probability $\alpha_H = 5\%$ by default to replace an expression with one of the best expressions so far generated. The main loop of PySR proceeds for $n_{\text{iterations}}$ number of steps, with $n_{\text{iterations}} = 40$ by default, however, as cautioned in the documentation [25], this value is not adequate for fully converged evolution. As such, we use $n_{\text{iterations}} = 1000$ for our main results, and we illustrate the approach to convergence through comparisons with lower values of $n_{\text{iterations}}$. Further details about the version of PySR used in this study are given in the Appendix B.

The purpose of this work is to apply the new tool PySR to generate flexible fits to the quark isovector GPD $H_{u-d}(x, t, 0)$ at zero skewness $\xi = 0$ computed in lattice QCD from Ref. [29]—these data are shown in Fig. 5. We will also benchmark the performance of PySR and its interpretability by applying it to various well-understood microscopic models of the GPDs, as well as to compare its performance with other established ML techniques like NNs. The various benchmarks and comparisons we perform in this work are outlined in the flowchart in Fig. 2.

B. Complexity and loss

In addition to a defined search space, search size, and training dataset, one must define a loss function (fitness function) which provides a metric for discriminating good models from bad ones and guides the overall evolution. The winning PySR model for a dataset is selected from a “Hall of Fame” that consists of optimal expressions at each complexity. What constitutes the best model is also somewhat user-defined: one may choose to simultaneously optimize the loss function (fitness function) and complexity of candidate models, or choose against considering the complexity and simply optimize the fitness function. In the

interest of attaining physically interpretable models without overfitting, we elect to consider the complexity in the selection criteria of all PySR models explored in this work, optimizing both the loss function and complexity of these models. For this reason, a candidate PySR model can be selected over a more complex model with marginally better fitness. It has been the case that convergence is not particularly well defined in SR due to the nature of genetic algorithms [24], and so one expects the symbolic form of these expressions to vary, run to run. However, as discussed in Sec. III C, we implement a convergence testing procedure on top of the PySR framework that further develops the notion of symbolic convergence quantitatively.

When fitting an expression to a dataset, it is important to give the expression enough flexibility that it can describe the general trends of the data, but not so much flexibility that *overfitting* becomes a problem. Even in traditional regression approaches, an expression with too many free parameters and too few data points results in a fit which incorporates not just the “true” overall behavior of the data, but also the random fluctuations it contains as well. In the extreme case, any set of n distinct data points (x_i, y_i) can always be fit exactly with an $(n - 1)^{\text{th}}$ degree polynomial. While such an “exact” fit to the data is always possible, it is far from desirable—an “optimal” fit is one that avoids overfitting and can successfully predict/reproduce data that were not included in determining the fit. In addition to being less reliable at prediction, overfit models are also more difficult to interpret.

For symbolic regression, in which the length and form of the expression is not fixed, overfitting is avoided by incorporating the *complexity* of the generated expression into the so-called loss function which the algorithm optimizes. Complexity is defined in PySR as the number of nodes in an expression tree. This definition assigns a complexity of “1” to all variables, constants, and operations. Equivalently, the complexity of an expression is then the sum of the number of occurrences of any constant, operator, or variable in the expression. For example, if α, β are constants, then the common PDF expression

$$x^\alpha(1-x)^\beta = \text{Mult}[\text{Pow}[x, \alpha], \text{Pow}[\text{Subtr}[1, x], \beta]] \quad (12)$$

has a complexity of 9: one instance of “Mult,” two instances of “Pow,” one instance of “Subtr,” two instances of x , and once instance each of 1, α , and β .

By default, the total loss $l(M)$ of a model expression M in PySR is a combination of the $\text{MSE}(M)$ and the complexity $C(M)$, defined as

$$l(M) \equiv l_{\text{pred}}(M) \exp f[C(M)], \quad (13)$$

where $C(M)$ is the complexity of the model and the “predictive loss” $l_{\text{pred}}(M)$ is the MSE. The MSE is a

weighted average of squared residuals for a given set of data and predictions. For a set of N data points $(x_i, y_i \pm \sigma_i)$ and a model $M(x)$ which evaluates to a set of predictions $(x_i, M(x_i))$ at the positions of the data, the MSE is given by

$$\text{MSE}(M) \equiv \frac{1}{N} \sum_{i=1}^N (M(x_i) - y_i)^2, \quad (14)$$

$$\text{WMSE}(M) \equiv \frac{1}{N} \sum_{i=1}^N w_i(x_i) (M(x_i) - y_i)^2, \quad (15)$$

where, for datasets with varying point-by-point uncertainties, the weights are given by $w_i = 1/\sigma_i^2$. For datasets with constant point-by-point uncertainties, we choose $w_i = 1$. Also in Eq. (13), $f[C(m)]$ is an *adaptive parsimony* term called *frecency* which allows for flexibility when penalizing complex expressions based on both the frequency and recency that models with that complexity have been considered by the genetic algorithm.

For phenomenological models, we set the uncertainties to 1 for the reasons described in the Appendix. M_i^{pred} and y_i^{data} are the model-predicted values and data being learned, respectively. In addition to providing best-fit models for $H^{u-d}(x, t; \zeta = 0, Q^2 = 4 \text{ GeV}^2)$ that consider only the MSE and complexity, which we call “Default” selection criteria as well, we also provide PySR models that are trained on a custom loss function $l_{\text{custom}}(M)$ that, in addition to the MSE and complexity, also penalize models that do not factorize in x and t . This is done by introducing a non-factorization penalty $F(M)$ to the loss function [see Eq. (27)]. For expressions that factorize, $F(M) = 0$ and the custom loss is simply the MSE for that factorized model. During the training, expressions that do not factorize are assigned such a large penalty that nonfactorized models are forced to have lower fitness compared to the factorized ones, and thus stop being suggested by PySR. Therefore, all final losses returned by PySR are still the MSE, even in cases when a nonfactorization custom loss is implemented.

C. Consistency and convergence

In a hypothetical world where Euler never existed, functions $f_1(x)$ and $f_2(x)$,

$$\begin{aligned} f_1(x) &= e^{ix}, \\ f_2(x) &= \cos(x) + i \sin(x), \end{aligned} \quad (16)$$

could be mistaken for two different functions. These two functions are equivalent, which one could deduce from their all-orders Taylor expansions,

$$e^{ix} = \sum_{n=0}^{\infty} \frac{(i)^n}{n!} x^n, \quad (17)$$

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}, \quad (18)$$

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}. \quad (19)$$

That is, once we put these functions in a common basis (in this case, a basis of Taylor polynomials), one may make more quantitative comparisons between functions by studying any variation in the expansion coefficients.

Because the output of PySR results are symbolic expressions, we require such a procedure in order to make quantitative statements of the “symbolic convergence” of PySR models. In this section, we put model replicas in a common basis of Taylor polynomials and then cluster model replicas based on the density of replicas with comparable expansion coefficients. This procedure defines what we call ECC, further developing the notion of symbolic convergence beyond what is currently implemented in the literature.

As a demonstration of ECC, we initialize an ensemble of 1000 unique instances of PySR and run them independently using the same initial training information. We denote these instances as PySR *replicas*. These PySR replicas were trained on a particular slice of the multidimensional LQCD data analyzed later in this paper, with identical hyperparameters. In this example, the LQCD results are for a one-dimensional function, $H^q(x)$, with $q = u - d$, representing a PDF calculated as defined in Eq. (9).

To study the differences of these models quantitatively, we compute the $\mathcal{O}(x - c_x)^5$ Taylor expansion for each replica in the PDF limit near the point $c_x = 0.555$. We choose $c_x = 0.555$ because it is the midpoint of the x -region which contains data used to train our SR models. For convenience of notation let $H(x) \equiv H_{u-d}$. The coefficients are then defined via the Taylor expansion near $x = c_x = 0.555$,

$$H(x)|_{\text{near } x=c_x} = \sum_{n=0}^{\infty} a_n (x - c_x)^n, \quad (20)$$

where a_n are the usual Taylor coefficients,

$$a_n \equiv \frac{1}{n!} H^{(n)}(x) \Big|_{x=c_x}, \quad (21)$$

which involve the n th derivative with respect to x of $H(x)$ evaluated at $x = c_x$. These Taylor coefficients obtained for each PySR replica can be viewed as a sample from an underlying multivariate probability distribution. The Taylor coefficients for a single PySR replica are also associated with an achieved MSE, and complexity. It should be remarked that the Taylor expansion basis explored here is not a unique choice: other orthogonal

bases over which we can expand the function can also be considered, and could in principle provide an optimization of our proposed ECC method, to be explored in future work.

A comprehensive visual picture of the coefficients, MSE values, and complexities can be seen in Fig. 3 using a corner plot [30]¹ for each of the 1000 PySR replicas, as well as a spaghetti plot² of all 1000 unexpanded PySR models in the PDF limit. We use a corner plot as convenient way to visualize the relevant eight-dimensional space and also provide some human intuition for correlations. Each sample/point in the corner plot corresponds to one curve in the spaghetti plot represented in the inset.

The MSEs and complexities of each PySR replica can be systematically improved by varying initial hyperparameters, including, but not limited to, *niterations* (number of generations) and *maxsize* (a complexity cap). From PySR documentation, *niterations* = 40 is the default and chosen for speed and not performance purposes. Default hyperparameter replicas are in excellent agreement in the interpolation region, and, even in the extrapolated region where there is no training data, many models converge into potential clusters of solutions. Marginal coefficient distributions for a_0 and a_1 are clearly Gaussian, whereas higher-order coefficients $[a_2, a_3, a_4, a_5]$ exhibit multimodal behavior. We find such multimodal behavior is related to the behavior of these PySR replicas in the extrapolation region.

A natural way to characterize multimodal behavior of samples is to understand how the samples are clustered. That is, we wish to assign clusters to the replicas which extrapolate into the different bands at low x visible in the spaghetti plot in Fig. 3. One may first attempt to cluster replicas based on their MSE, however, because the replicas converge to comparable MSEs regardless of their behavior in the extrapolation region, MSE is a poor parameter to control the clustering. Instead, we perform an initial clustering analysis of Taylor expansion coefficients $[a_0, \dots, a_5]$ using the well-accepted K-means clustering algorithm [31] in the six-dimensional coefficient space. Because K-means cannot deduce the optimal number of clusters automatically, we then construct a plot of the within-cluster sum of squares (WCSS) [32] as a function of the number of clusters k (Fig. 4). The WCSS is defined as follows [32,33]:

$$\text{WCSS}(k) = \arg \min_{C_1, \dots, C_k} \sum_{j=1}^k \sum_{x_i \in C_j} |x_i - \mu_j|^2, \quad (22)$$

¹Corner plots are multipanel scatter plots used in statistics that allow us to visualize the correlations among the various variables in a high-dimensional parameter space.

²A spaghetti plot is a visual representation of the replicas using a continuous line tracing their path.

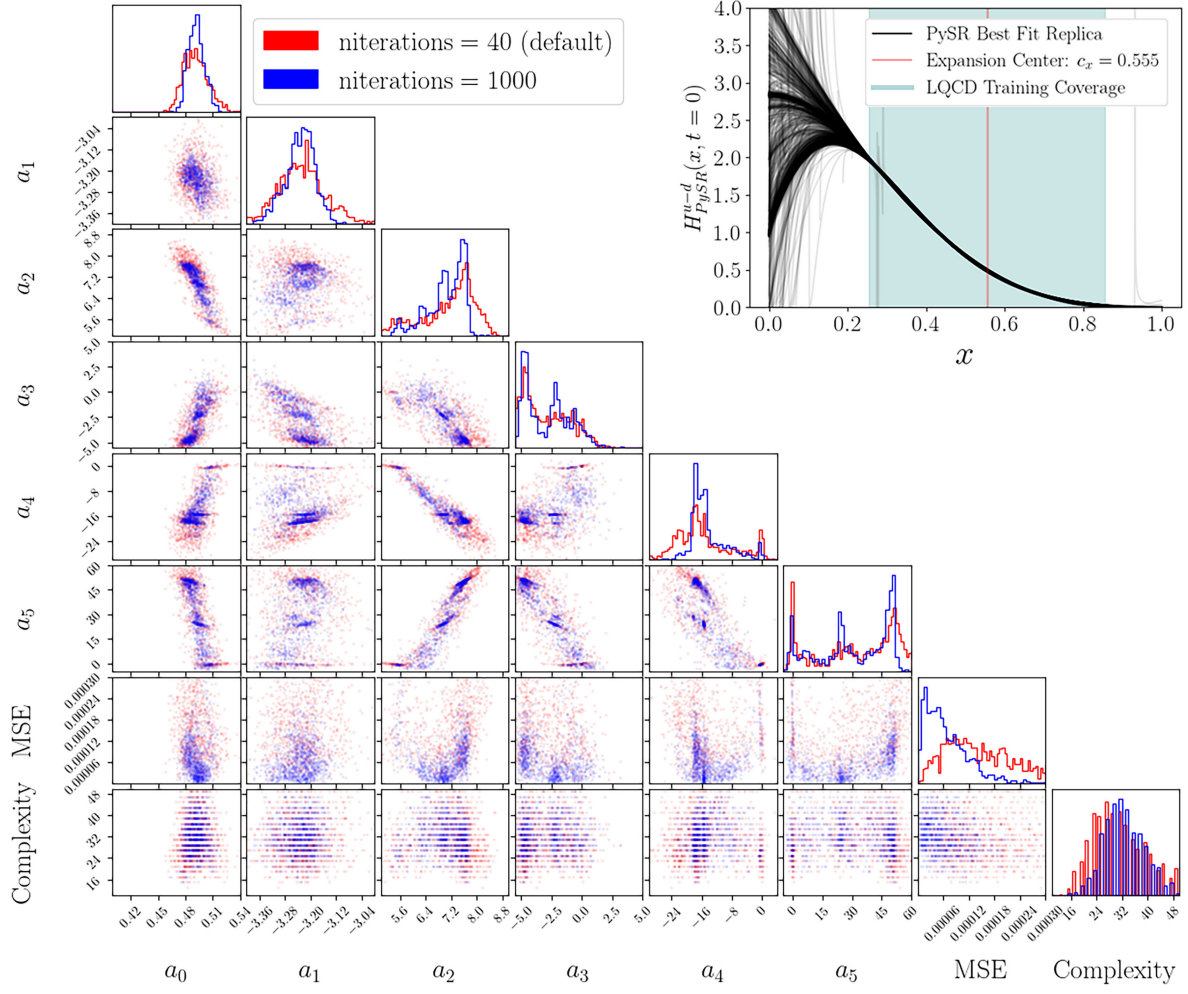


FIG. 3. Ensemble of 1000 unbiased fits to the lattice GPD data using PySR for $n_{\text{iterations}} = 40$ (default PySR setting, red) versus $n_{\text{iterations}} = 1000$ (blue). The GPDs themselves for all 1000 replicas are plotted in the inset/spaghetti plot, in the top-right corner (for $n_{\text{iterations}} = 1000$). The array of smaller plots shows the distribution and correlations of Taylor coefficients $a_0 \dots a_5$, weighted MSE, and complexity. Clear clusters of preferred solutions are visible by eye, and increasing $n_{\text{iterations}}$ selects those preferred solutions more precisely.

where C is the set k clusters, μ_j is the average of the samples within the j th cluster ($1 \leq j \leq k$), and x_i is the i th replica within cluster C_j . The selected clustering is then the ensemble of k clusters that minimize $\text{WCSS}(k)$. We use the default implementation in scikit-learn [34] for calculating the WCSS from K-means as a function of the number of clusters. The value of k corresponding to the location of maximum curvature of the WCSS is referred to as the “elbow” or “knee” of the WCSS. The location of the knee was found using the Kneedle algorithm [35] using the Python package kneed [36], which uses the standard definition of the curvature $K_f(k)$ of a function $f(k)$ at any point k ,

$$K_f(k) = \frac{f''(k)}{(1 + f'(k)^2)^{3/2}}. \quad (23)$$

The value of k that maximizes $K_f(k)$ corresponds to the location of the Kneedle knee. After implementing K-means and Kneedle on the set of 1000 PySR best-fit replica Taylor coefficients, we find that these replicas are best described by three clusters of solutions (Fig. 4).

1. Interpolation and extrapolation

We can see from Fig. 3 that not all replicas extrapolate to one of three clusters. There are spurious solutions that extrapolate in between bands of overlapping models. To account for PySR models that do not fall into any cluster of solutions, we considered other clustering algorithms which do not require every single replica to lie inside a cluster [32]. We settled on using HDBSCAN to recluster the replicas (also using scikit-learn [34]). When using HDBSCAN we must set the minimum number of points per cluster, “min_samples.” We found that assigning

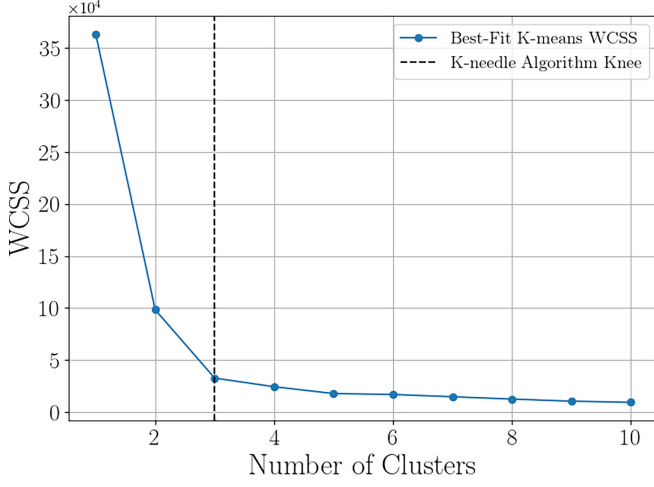


FIG. 4. K-means within-cluster sum of squares as a function of the number of clusters. Clustering is simultaneously performed on distributions $a_0 - a_5$ using K-means. The location of the elbow, selected by the Kneedle algorithm, suggests that these replicas are best classified into three clusters.

$\text{min_samples} = 42$ is the lowest number of points that can define a cluster while still resulting in three clusters (+1 clusterless category) as an output. Taylor coefficients are just one set of parameters one could use to perform clustering. There are of course other possible approximations with other associated expansion basis functions, as well as different parameters one could consider while clustering.

Note that clustering solely based on the MSEs of the replicas does not result in the same clusters as those achieved by clustering their Taylor coefficients. That is, MSE-based clusters are not mappable to the clusters found by ECC. The ECC clusters are associated with visually consistent spaghetti lines in the extrapolation region, whereas MSE clusters are not. This can be seen as a feature (not a bug) of our novel ECC method. Each Taylor coefficient cluster provides a valid MSE in the data region, but a different consistent band of predictions in the extrapolation region.

D. Lattice results

Lattice QCD calculations provide a numerical framework for studying hadronic structure directly from QCD. The present work incorporates lattice QCD data for the nucleon isovector unpolarized GPD extracted from numerical simulations performed in Ref. [29] at physical quark masses, ensuring a realistic description of the nucleon. In particular, the calculations were carried out using a $N_f = 2 + 1 + 1$ ensemble with the highly improved staggered quark [37,38] action for the sea quarks and the Wilson-clover action for the valence quarks. The lattice spacing is $a \approx 0.09$ fm, with a spatial (temporal) extent of 5.8 (8.6) fm, at a physical pion mass of $m_\pi \approx 135$ MeV. The nucleon matrix elements relevant for the quasi-GPDs were computed using

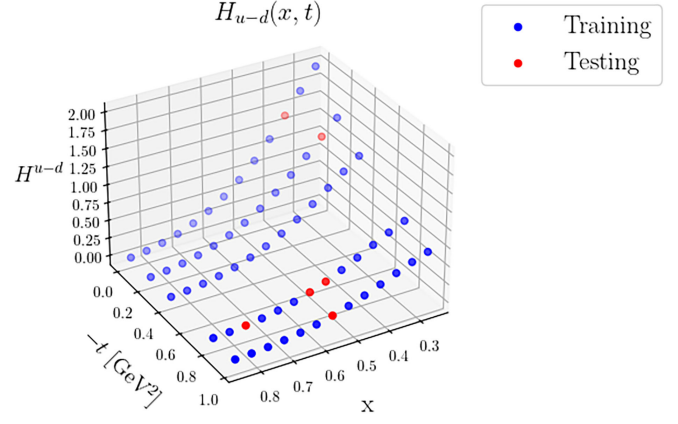


FIG. 5. H_{u-d} results [29] used as input source data for training and testing SR. We can use SR to find a closed form symbolic expression for H_{u-d} using the lattice result points shown in this figure. Using computationally costly lattice calculations places us in a regime of “small data” where we only have about 100 points to work with.

momentum-smeared nucleon interpolating fields, and renormalization was performed in the regularization-independent momentum-subtraction scheme [29,39,40]. These lattice results serve as inputs for the analysis presented across this contribution.

We start with a grid of 13×5 lattice QCD data points in (x, t) , from Ref. [29] for the nucleon isovector (flavor nonsinglet) GPD, $H_{u-d}(x, t, \xi = 0)$, Eq. (10a), at fixed skewness parameter, $\xi = 0$. The relevant renormalization scales (see [29] and references therein) are $\mu_R = 3.8$ GeV, $p_z^R = 2.2$ GeV, and $\mu^{\overline{\text{MS}}} = 3$ GeV. For brevity, we use the short-hand notation $H_{u-d}(x, t) \equiv H_u - H_d$ throughout. The 13 points in x are linearly spaced between $x = 0.255$ and $x = 0.855$, while the five points in t , fixed by the external momenta used in [29], are $|t| \in 0.0, 0.193, 0.387, 0.774, 0.967$ GeV². We randomly remove 10% of this data to reserve for testing the PySR models, while the remaining 90% is used to train the models (Fig. 5).

In addition to LQCD we consider phenomenological models where the functional form being learned by SR is already known. Our sources are chosen from three different models from the literature that are built according to different phenomenological criteria: a spectator model-based parametrization (GGL (Goldstein Gonzalez Liuti)) [41,42], a double-distribution GPD (GK (Goloskokov Kroll)) [43], and a phenomenology-based one with built-in factorized x and t dependence (VGG (Vanderhaeghen Guichon Guidal)) [44,45]. For the model parametrizations we evaluate the sources on the grid of points in (x, t) for which we have lattice data available and apply the same partitioning into training/testing data.

E. Selection criteria and training regions

In addition to training PySR on a variety of sources for the flavor nonsinglet GPD combination, $H_{u-d}(x, t)$, we also

provide PySR models with various functional features in mind, which we refer to as *selection criteria*. All models consider at least a default selection criteria, which optimizes the mean squared error and complexity of the models. The selection criteria we consider are (1) best fit, (2) force-factorized, and (3) semi-Reggeized.

1. Best fit

This selection criterion corresponds to a loss function where the objective is the training MSE of the PySR model, optimized against the complexity. There is no biasing toward specific functional forms or satisfying constraints, aside from the model building blocks, as earlier defined.

2. Force-factorized

Force-factorized (FF) corresponds to a modification of the custom loss function to assign a penalty in the training to expressions that do not factorize in x and t . A function of two variables x, t factorizes if it can be written as

$$H_{u-d}(x, t) = f_1(x)f_2(t), \quad (24)$$

then forming ratios at fixed t_i, t_j gives

$$\frac{H_{u-d}(x, t)}{H_{u-d}(x, t_j)} = \frac{f_1(x)f_2(t)}{f_1(x)f_2(t_j)} = \frac{f_2(t)}{f_2(t_j)}, \quad (25)$$

where the ratio depends on the choices of t_i and t_j , but is constant in x . We define the fixed- t ratio

$$R(x, |t|) \equiv \frac{H_{u-d}(x, |t| = 0)}{H_{u-d}(x, |t|)}, \quad (26)$$

where the t_i are values of $t \neq 0$ defined by the lattice grid in Sec. III D. For factorized forms for H_{u-d} , R_i should be constant for all x .

In the FFcustom loss, a factorization penalty is assigned through an additive term if the model for H_{u-d} does not have a factorized form, Eq. (24), for two expressions f_1 and f_2 joined at a node through either multiplication or division. In other words, in addition to minimizing model MSEs, there is a penalty throughout the training if candidate models for $H_{u-d}(x, t)$ do not factorize in x and t . A custom loss is a fitness function that allows one to nudge PySR models toward functional forms that optimize user-defined objectives. Here, similar to [46], we define the FFcustom loss which is implemented at each iteration of the training,

$$l_{\text{custom}}(M) = \text{MSE}(M) + F(M), \quad F(M) \begin{cases} = 0, & \text{if } H_{u-d}(x, t) = f_1(x)f_2(t), \\ + = \gamma & \text{if root degree} \neq 2, \\ + = \gamma & \text{if root operator} \neq \times \text{ or } \div, \\ + = \gamma^2 & \text{if not of form } f_1(x)f_2(t), \end{cases} \quad (27)$$

where f_1 and f_2 are functions of either x or t and $\gamma = 10$ is a scale much greater than the typical MSE. The degree of the root node specifies whether the expressions are related via binary operation (degree = 2) or unary operation (degree = 1). It is impossible for an expression to factorize without two expressions at the root node being related via binary operation, and so a small penalty is assigned if that is not true for that current iteration in order to begin nudging the PySR model evolution toward the desired factorized form. For the same reasoning, the binary operator relating the two expressions at the root node must be either multiplication ($\times$) or division ($\div$), and so another small penalty is assigned if this is not satisfied. If those two criteria are satisfied, the algorithm then checks if the two expressions joined at the root node via multiplication or division are univariate and that neither expression contains the variable contained in the other expression, or else a larger penalty is assigned. Splitting up the penalty into three smaller penalties allows the models to evolve toward increasingly fit functional forms without being heavily penalized for not immediately having the desired factorized form.

Using the custom loss function in Eq. (27), all PySR replicas trained on the sources in this work were successfully factorized in x and t . Note that this is not a statement that the training source data itself factorizes, but is rather a constraint on the space of models that the algorithm is permitted/encouraged to explore.

3. Semi-Reggeized

To motivate the last selection criteria considered in this work, we recall that the sum rule for up and down valance quark PDFs,

$$\int dx(u_v(x, Q^2) - d_v(x, Q^2)) = 1, \quad (28)$$

implies that the isovector combination $H_{u-d}(x, t)$ approaches a finite value as $x \rightarrow 0$. This is related to the scaling predicted by Regge theory [47], in which hadronic cross sections scale as $\sigma \sim s^{2(j-1)}$, where s is the center-of-mass energy (squared) and j is the effective angular momentum of the t -channel mediator [47,48]. Collinear factorization relates cross sections like deep inelastic

scattering to PDFs with $x \approx Q^2/s$, leading to the general expectation that $f(x) \sim x^\alpha$ at small $x \ll 1$. While the gluon and sea quark distributions grow at small x ($\alpha < 0$), the valence quark distributions fall at small x ($\alpha > 0$), and the isovector component even more so. The behavior of this GPD near the end point $x \rightarrow 1$ is similarly governed by the probability to find the other two valence quarks at $x \rightarrow 0$ in the proton wave function. This is accordingly governed by the Brodsky-Farrar quark counting rules [49], which predict the vanishing of PDF at large $x \rightarrow 1$ as $f(x) \sim (1-x)^\beta$, with the exponent $\beta > 0$ depending on the number of valence spectators. In essence, the probability that a struck valence quark carries 100% of the protons momentum should vanish at a predictable rate. We therefore also include a *redundant column approach* (RCA), which nudges PySR evolution toward models that have the form

$$H_{u-d}(x, t) = x^\alpha (1-x)^\beta P(x) g(t), \quad (29)$$

where the x -dependent form inspired by both Regge theory [47] and the Brodsky-Farrar quark counting rules [49] is commonly used in parton distribution function analyses (see, e.g., Ref. [50] and references therein). Here also α and β are parameters, $g(t)$ describes the unconstrained t dependence of H_{u-d} , and $P(x)$ describes any residual x dependence. This parametrization should give a corresponding PDF $f_{u-d}(x)$ that is finite at the end point regions $x \in 0, 1$. In general, the exponents such as $\alpha = \alpha(t)$ can also depend on t in a nontrivial way (the “Regge trajectory”). For simplicity, in this work we consider the case of constant (t -independent) quantities α and β and refer to such a fit as being *semi-Reggeized*. In RCA we introduce the redundant variable $y \equiv 1-x$ so that $H_{u-d} = H_{u-d}(x, y, t)$. PySR models are then selected post-facto that have the functional form

$$H_{u-d}(x, y, t) = x^\alpha y^\beta P(x) g(t). \quad (30)$$

Taking the log of both sides gives³

$$\begin{aligned} \ln H_{u-d}(x, y, t) &= \ln(x^\alpha y^\beta P(x) g(t)) \\ &= \alpha \ln x + \beta \ln y + \ln P(x) + \ln g(t). \end{aligned} \quad (31)$$

Numerical results for the three different frameworks are presented and discussed in Sec. IV.

³The transformation to log format increases the probability of the regressor to find the desired form as a linear combination of terms due to the fact that PySR-selected models that utilize the binary operator *Pow* manifestly break the semi-Reggeized functional form. Therefore, dropping that operator from consideration increases the probability of finding a suitably Reggeized form compared to performing SR on the original source data.

IV. INTERPRETING THE TRENDS OF LATTICE QCD RESULTS

We now present results illustrating, in particular, how the use of SR for GPD analyses enables the extraction of physics information, from the spatial radii of quark and gluon distributions, to the orbital angular momentum content of the nucleon, as obtained directly from data.

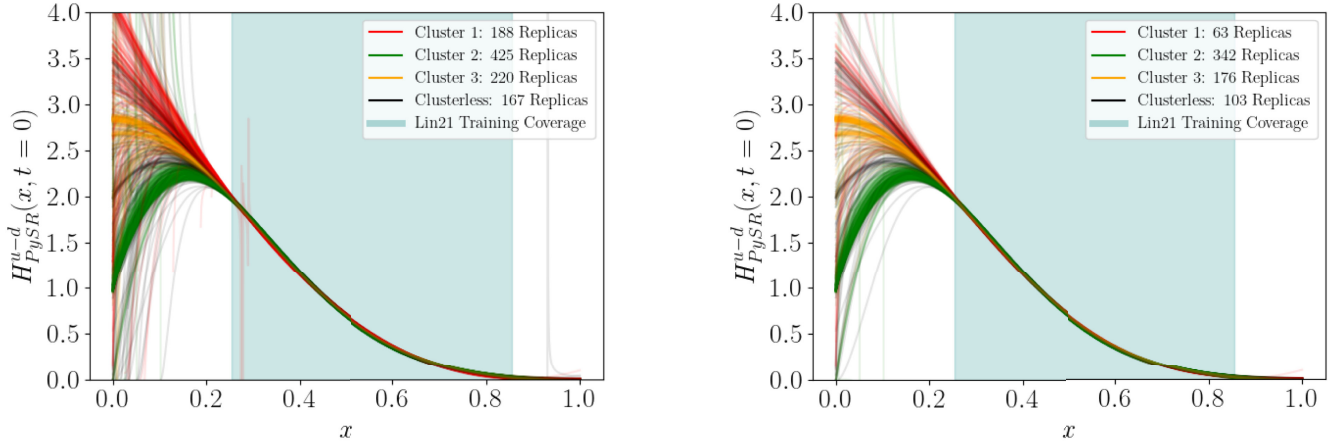
A. Fit results

We first show numerical results of the fits for the various selection criteria described in Sec. III, in particular for best fit (BF) and FF, and we analyze the structures appearing in results as distinct clusters of curves. Since we do not impose a normalization criterion, namely the functional forms are not required to integrate to the same quantity, we illustrate and discuss the distribution around the nominal integrated value obtained from the form factor, A_{10} , in Eq. (4).

1. Best fit

We expand the PDF limit of 10^3 best-fit PySR replicas for $H_{u-d}(x, t)$ near $x = c_x = 0.555$ to $\mathcal{O}(x - c_x)^5$ (Fig. 3). Analyzing this case we encountered the issue that not all PySR replicas appear as free of pathologies: poles are visible in both the interpolation region (marked in gray in the figure) and in the extrapolation region. Some PySR curves introduce poles either in between training data points or in the extrapolation region in order for the model to fit the source data. These poles can be attributed to the model building blocks in this paper consisting of the binary operators $(+, -, \times, \div, ^)$. From these operators, one may construct polynomials, ratios of polynomials, powers of polynomials, and any combination.

Our procedure to deal with pole solutions is to filter them out by checking whether or not the PySR model has a well-defined integral $\int_0^1 dx H_{u-d, \text{PySR}}(x, t=0) \equiv A_{10}(t=0)$, using numerical integration. PySR replicas with finite $A_{10}(t=0)$ define what we refer to as *good replicas* throughout due to the value of this moment being connected to the sum rule in Eq. (4). The latter represents a physical constraint, in that we identify $A_{10}(t=0)$ as the quarks $u-d$ charge given by the value of the corresponding flavor-dependent Dirac form factor, Eq. (4), measured in Refs. [51,52]. It is also a self-consistent check from LQCD since this quantity can be calculated in an independent framework from Ref. [53] (see, e.g., Refs. [54–56]). Out of the initial set of 1000 best-fit replicas, 684 survived the finite-moment filter and were classified as good replicas. Cluster color-coded spaghetti plots of the BF replicas before and after imposing the finite-moment filter are shown in Fig. 6(a). The finite-moment filter can also be implemented into the training at the level of the custom loss function. We see from Fig. 6(b) that 342/684 good BF replicas are categorized into cluster 2, 176/684 into cluster 3, and only 63/684 into cluster 1. Many of the cluster 1 replicas diverge



(a) HDBSCAN clustering of 1000 Best-fit PySR replicas into 3 clusters + 1 clusterless category based on location of K-needle knee.

(b) Remaining 684 Best-Fit PySR replicas (clustered with HDBSCAN and K-needle) after imposing Finite-Moment-Filter.

FIG. 6. Cluster color-coded spaghetti plot of best-fit PySR replicas before (left) and after (right) selecting onto replicas with finite $A_{10}(t=0)$. The 1000 PySR replicas trained on LQCD source data [29] were generated and their PDF limit ($t=0$) was expanded near $x=0.555$ to $\mathcal{O}(x-0.555)^5$. K-means clustering was run on the distributions of expansion coefficients simultaneously, with three clusters of replicas found based on the location of the Kneedle knee. These coefficients were then reclustered using HDBSCAN for the additional “clusterless” category at fixed $N=3$ clusters.

near $x=0$, forcing them to be discarded after imposing the finite-moment filter. Cluster 2 and cluster 3 replicas tend toward a finite value for $H_{u-d}(x, t=0)$ near $x=0$, resulting in a considerably higher ratio of surviving replicas for those clusters.

Finally, although the finite-moment filter requires that good replicas have finite $A_{10}(t=0)$, we placed no restriction that the good replicas satisfy exactly the sum rule $A_{10}(t=0)=1$. Interestingly, as shown in Fig. 7, most of these surviving replicas still satisfy $A_{10}(t=0)=1$ to within 10%. Because the sum rule involves integrating over the full range in x , rather than just the training region $0.255 \leq x \leq 0.855$, we view this as a success in the

extrapolation capability of PySR-selected models. The physical constraint provided by integration to the elastic form factors, along with additional ones, including for instance positivity of the solutions, can be inserted directly in the custom loss function as a means of gaining additional predictive power in the extrapolation region.

2. Forced factorization

For the FF criterion, following the procedure described in Sec. III C, we generated 10^3 FF PySR replicas and we performed an ECC (Sec. III B) on the distribution of replica Taylor coefficients in the PDF limit near $x=c_x=0.555$ to

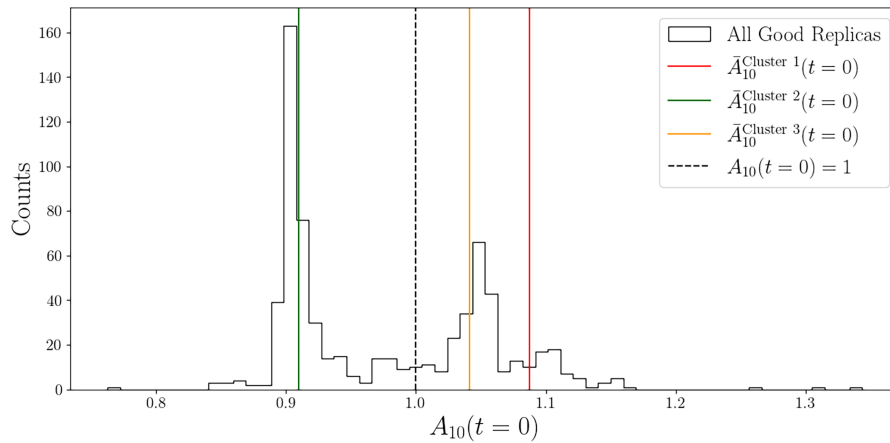


FIG. 7. Distribution of finite-moment PySR best-fit replicas $A_{10}(t=0)$. Each replica is an independent run of PySR upon the same LQCD computations. The moments of the SR replicas are centered near 1 on average, which appears to have arisen without being enforced by construction. Each cluster also has a mean moment which is color-coded based on HDBSCAN clustering.

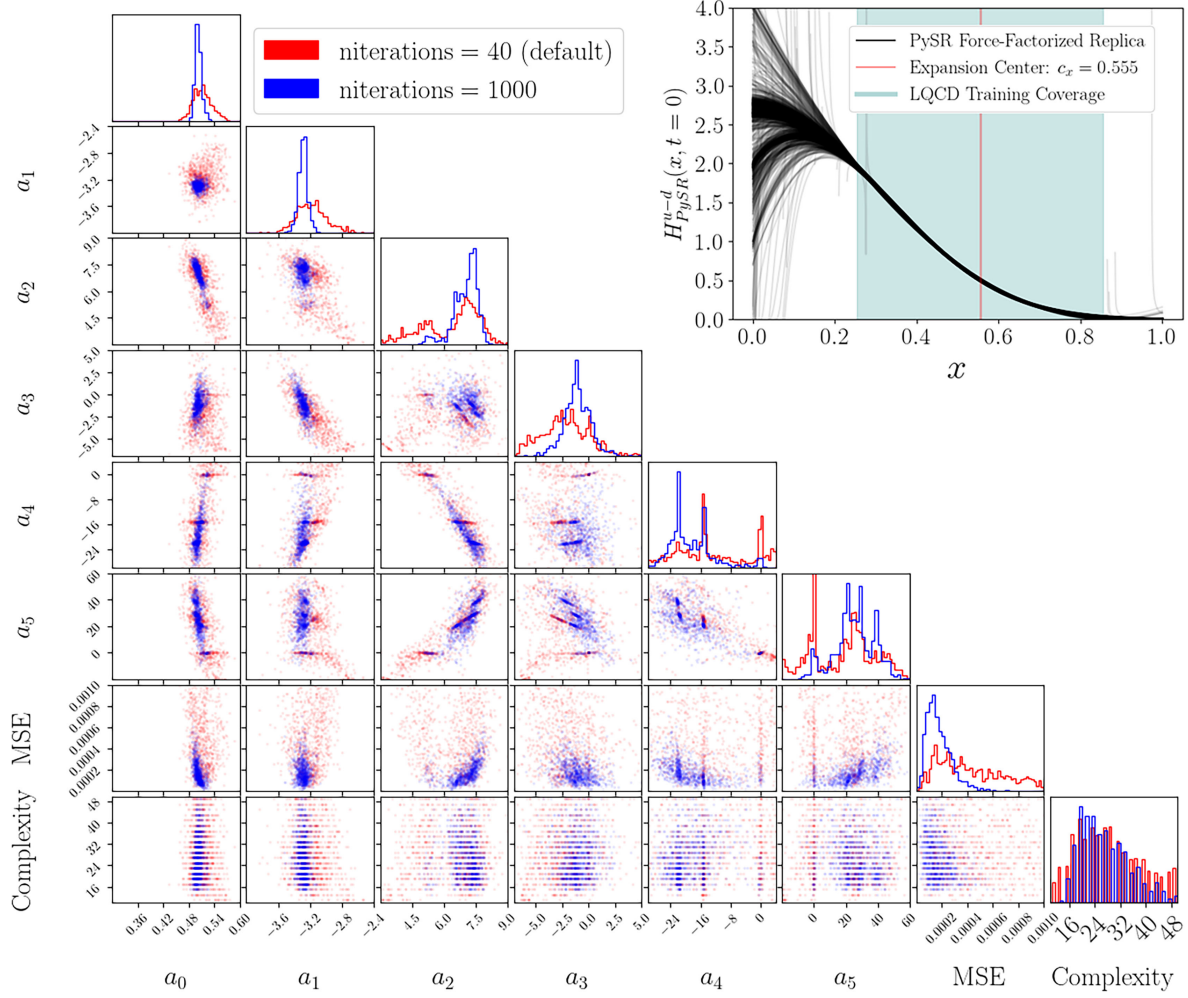


FIG. 8. Ensemble of 1000 factorized fits to the LQCD GPD data using PySR for $n_{\text{iterations}} = 40$ (default PySR setting, red) versus $n_{\text{iterations}} = 1000$ (blue). The GPDs themselves for all 1000 replicas are plotted in the inset in the top-right corner (for $n_{\text{iterations}} = 1000$). The array of smaller plots shows the distribution and correlations of Taylor coefficients $a_0 \dots a_5$, MSE, and complexity. Clear clusters of preferred solutions are visible by eye, and increasing $n_{\text{iterations}}$ selects those preferred solutions more precisely.

$\mathcal{O}(x - c_x)^5$. We see from Fig. 8 that most FF replicas describe the training region well, with some replicas containing poles in either the interpolation or extrapolation regions. In the low- x extrapolation region, we see the familiar clustering of replicas into bands, demonstrating that even in the absence of data at low x , PySR replicas largely extrapolate to only a few different behaviors. This is to say that, similar to the BF case, generating 1000 replicas does not result in 1000 different models, but only a few to within small deviations in the expansion coefficients.

Similar to the BF replicas, MSE alone was not sufficient for clustering these FF replicas. This is because PySR is converging toward replicas with comparable MSE, and therefore ECC or other choices for higher-dimensional clustering are necessary in order to adequately cluster the replicas. As in Sec. III C, we apply ECC on the FF replicas to assign clusters to replicas that fall into the visible bands

shown in Fig. 8. We again use the K-means algorithm to compute the WCSS per number of clusters, and fix the number of FF clusters using the Kneadle algorithm (Fig. 9). We see from Fig. 9 that the FF replicas are best classified into three clusters.

Due to the existence of FF replicas that extrapolate in between the bands visible in Fig. 8, we recluster the replicas using HDBSCAN, fixing $N_{\text{FF Clusters}} = 3$, and include a clusterless category [Fig. 10(a)].

Though the behaviors of FF replicas are similar to those of the BF replicas, the clustering of FF replicas is quite different. This is best understood after selecting onto FF replicas with finite $A_{10}(t=0)$ [Fig. 10(b)]. Both sets of replicas (BF and FF) were best described by three clusters using ECC, however, what was called cluster 1 for the best-fit replicas [Fig. 6(a)] is not present in the force-factorized replicas. Manually fixing $N_{\text{FF Clusters}} = 4$ produces a cluster

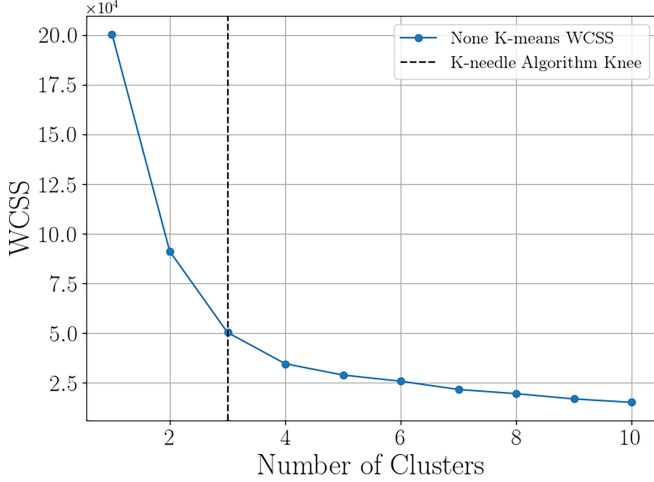


FIG. 9. K-means within-cluster sum of squares as a function of the number of clusters. The location of the elbow, selected by the Kneedle algorithm, suggests that these FF replicas are best classified into three clusters.

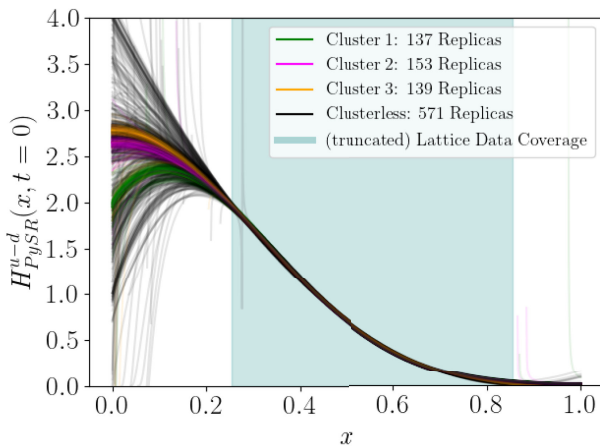
of FF replicas that does extrapolate similar to the BF cluster 1 replicas, but this only results in eight FF replicas after selecting onto replicas with finite $A_{10}(t=0)$ and so they are not considered here. That is, after filtering out models containing poles in x at $t=0$, there is no visible FF cluster that corresponds to the low- x behavior of BF cluster 1. Another interesting result of ECC applied to FF replicas is that though replicas within cluster 2 and cluster 3 extrapolate similarly at low x , they are considered members of different clusters. This is no mistake, as these two clusters correspond to replicas with different a_5 . That is, these clusters describe genuinely different models that are degenerate at small x .

3. Semi-Reggeized

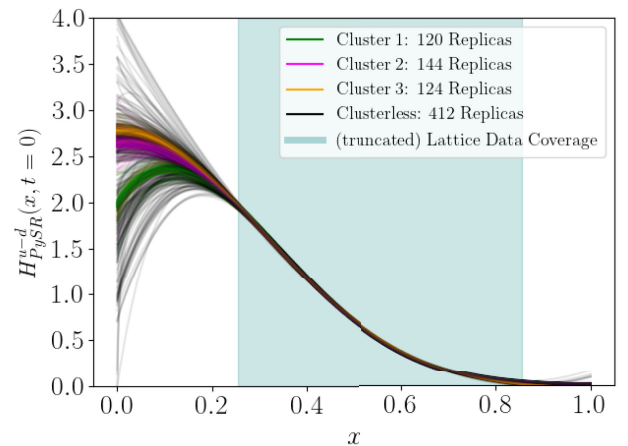
Semi-Reggeized forms, or RCA, were analyzed by introducing the redundant variable $y \equiv 1 - x$ so that $H_{u-d} = H_{u-d}(x, y, t)$, Eqs. (29) and (30). For the semi-Reggeized approach, we did not apply the series expansion and clustering. Instead we selected an exemplar (shown in Table I). We did, however, run SR many times, and write an algorithm to check which results if any contained a Reggeized form without defining a custom loss that favors such forms. The exemplar selected is the first SR result obtained which satisfied the Reggeized format,

$$H_{u-d}^{\text{Regge}}(x, \mathbf{t}) = 2.74e^{-\mathbf{t}}x^{0.087}(1-x)^{[1.57-0.67\ln(1-x)]}, \quad (32)$$

where $\mathbf{t} \equiv -t/\Lambda^2$, and $\Lambda^2 = 1 \text{ GeV}^2$. The selected Reggeized exemplar has $\text{MSE} = 1.9 \times 10^{-4}$ relative to the testing dataset. Though the MSE is notably larger than that of the BF or FF LQCD exemplars, the MSE for the Reggeized exemplar is the same order of magnitude as the average MSE of FF LQCD replicas (Table II), a testament to the convergence of PySR with well-chosen hyperparameters. We know from Eq. (32) that along with the characteristic Regge behavior $x^\alpha(1-x)^\beta$ for $H_{u-d}^{\text{Regge}}(x, \mathbf{t})$, there is residual x dependence in the exponential that vanishes as $x \rightarrow 0$. Though $\ln^2(1-x)$ diverges at $x=1$, $\exp(-\ln^2(1-x))$ vanishes as $x \rightarrow 1$, recovering the characteristic Regge behavior at both end points. In future work, we may devise a “force-Reggeized” (FR) custom loss function and carry out a procedure such as we did in ECC as a way to test the consistency, convergence, and clustering of Reggeized SR replicas. The Kullback-Leibler (KL) divergence of an FR custom loss $\text{KL}(\text{MSE}_{BF} || \text{MSE}_{FR})$ would allow one to make more quantitative statements



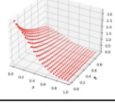
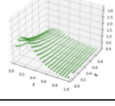
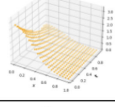
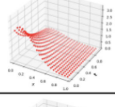
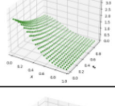
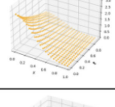
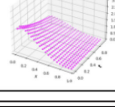
(a) HDBSCAN clustering of 1k FFPySR replicas into 3 clusters based on location of K-needle knee.



(b) Remaining 800 FFPySR replicas (clustered with HDBSCAN and K-needle) after imposing Finite-Moment-Filter.

FIG. 10. Cluster color-coded spaghetti plot of FF before (left) and after (right) selecting onto replicas with finite $A_{10}(t=0)$ based on the Kneedle algorithm.

TABLE I. SR functional forms for all three models: best fit (BF), force-factorized (FF), and semi-Reggeized (Regge). For BF and FF, the functional forms corresponding to the three different clusters determined in this analysis are displayed and numbered, with $K = 1, 3$. The fitness metric and complexity values associated with each determination are discussed in the text. The boldface $t = -t/\Lambda^2$, with $\Lambda^2 = 1 \text{ GeV}^2$, is used to ensure that all numerical coefficients in the symbolic expressions are dimensionless.

Selection criteria	Predictions over full phase space	Symbolic Forms	Fitness metrics WMSE MSE	Complexity
H_{BF}^1		$\frac{3.3(1 - 0.8x^2)^{5.2}}{x + 2t + (1.52t)^t}$	3.3×10^{-3} 2.1×10^{-6}	21
H_{BF}^2		$\frac{(0.034x^{-4.52})^x}{1.89t + 0.9451x + 0.94\{[(0.86(t+1)-x)t]t\}^{-0.031}}$	2.5×10^{-3} 2.3×10^{-6}	31
H_{BF}^3		$\frac{1.02\{0.0047[x+(t-0.82)(t-0.29)]^t\}^x - 0.018}{1.77t - 0.64}$	1.9×10^{-3} 2.1×10^{-6}	29
H_{FF}^1		$1.55 \left(-0.483 + \frac{1}{(0.42 + t)^{0.75}} \right) \left(\frac{-0.0019 + 0.0043x}{x} \right)^x$	6.6×10^{-2} 4.4×10^{-5}	23
H_{FF}^2		$\frac{-0.01 + 1.45(323^x x^x)^{-x}}{0.56 + t^{1.14}}$	4.3×10^{-2} 5.5×10^{-5}	19
H_{FF}^3		$\frac{1.19\{-0.048 + 2.34[(0.063x)^x]^{1.85}\}}{(0.913 + t^t)t}$	7.1×10^{-3} 2.0×10^{-5}	29
H_{Regge}		$2.74 e^{[-t - 0.67 \ln^2(1-x)]} x^{0.087} (1-x)^{1.57}$	1.5×10^{-2} 1.9×10^{-4}	26

regarding the truth of the Reggeized hypothesis for a given dataset. It is of course reassuring that the *redundant column approach* in conjunction with post-facto selection (based on functional form, not MSE) resulted in a Reggeized model that performs comparably well to the average FF replica.

4. Uncertainty treatment

The uncertainty in SR determinations can appear in two distinct instances in this work: (i) point-by-point data uncertainty and (ii) global uncertainty introduced by the nature of SR dependent upon genetic annealing evolution (Sec. III). Both sources of uncertainty are reflected in the

TABLE II. Average MSE of BF and FF PySR replicas calculated on the testing dataset. The KL divergence being computed is $KL(\text{Best-Fit}||\text{Force-Factorized})$ [Eq. (35)].

PySR fit statistics				
	LQCD	VGG	GGL	GK
MSE (BF)	6.86×10^{-5}	1.55×10^{-6}	5.18×10^{-5}	8.35×10^{-6}
MSE (FF)	1.77×10^{-4}	5.10×10^{-6}	1.87×10^{-3}	1.70×10^{-3}
MSE Ratio: FF/BF	2.25×10^0	3.29×10^0	3.61×10^1	2.03×10^2
KL	3.0×10^{-2}	2.68×10^{-2}	1.86×10^{-1}	6.17×10^{-1}

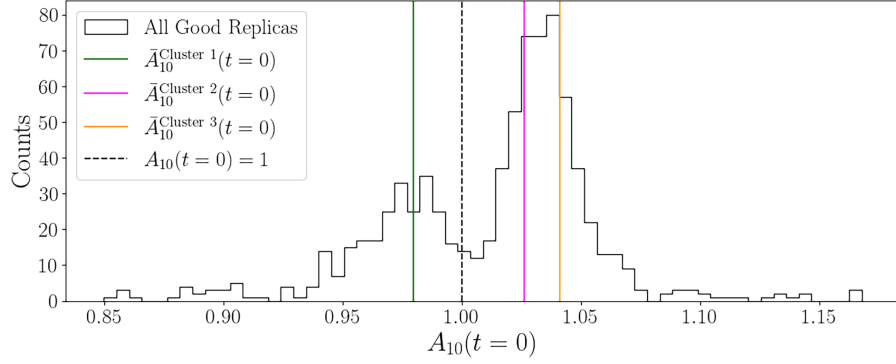


FIG. 11. Distribution of remaining $A_{10}(t=0)$ from FF PySR replicas. The average contribution to $A_{10}(t=0)$ from each cluster using Kneede is also shown by the color-coded vertical lines. The locations of the peaks of $A_{10}(t=0)$ appear to be heavily correlated with the cluster label, due to the location of the average value of $A_{10}(t=0)$ for a given cluster overlapping closely with the locations of the peaks of $A_{10}(t=0)$ for all 800/1000 remaining replicas.

final choice of each predicted curve shown in our spaghetti plots [Figs. 6(a) and 10(a)].

To handle the point-by-point uncertainty, in this work we adopt MSE Eq. (14) for the “mean of squared residuals” and WMSE Eq. (15) as the “weighted mean squared residuals,” where residuals are defined as the difference between the variables’ observed and predicted values, for each given input data point. However, it should be remarked that the LQCD errors are highly correlated but unaccounted for. LQCD correlations are visibly apparent from their trend in both the kinematic variables, x and t , and are due to the dominance of systematic errors over the statistical ones (for an explanation of LQCD systematics for this calculation, see [29]). Therefore, in our analysis, while adopting the “MSE” and “WMSE” terminology from the SR literature, it is understood that these quantities do not carry statistical significance. They are, however, used as “fitness metrics” to gauge the goodness of the various symbolic expressions approaching the LQCD curves. Values of fitness metrics for exemplar curves are presented in Table I.

To handle global uncertainty introduced by SR, we assess the aggregate of many SR attempts using the same data, and find that SR converges in the training region and introduces clusters of solutions in the extrapolation region. These clusters of solutions are introduced by SR while attempting to perform a global optimization of the fitness metric, which is not changing. The same global optimum yields multiple clusters of solutions using the same SR algorithm many times.

In this work, we see that clustering of SR results show tension between two effects. The space of possible curves widens, and the number of clusters can go up, as the uncertainty on the data increases, and SR has access to more composite functions and physical constraints are ignored. Conversely, as physical constraints are applied, data uncertainty shrinks and the number of composite functions is reduced, then the space of possible result curves from SR shrinks, and the number of viable clusters decreases. Specific examples of physical constraints include but are

not limited too polynomiality, positivity, matching onto form factors/PDFs, and normalization. We reserve treating the correlated uncertainties, more statistical Markov-Chain-Monte-Carlo SR approaches, and additional physical constraints for future work.

Because a single SR run produces a global minimum curve and parameters, we do not have a statistically sound way of producing a spaghetti plot. Running SR many times produces many best-fit curves, and we can assess how SR introduces an intrinsic spread of its own results, in other words, we can perform, in this case, a statistical analysis, and define an error band for the results. This is illustrated in Fig. 12 where we plot the integrated over x GPD, H_{u-d} , namely the Dirac form factor, F_1^{u-d} , Eq. (4), vs $-t \equiv |t|$. The figure clearly shows the spread in the spaghetti plot that can eventually be cast into a band around the most likely result,

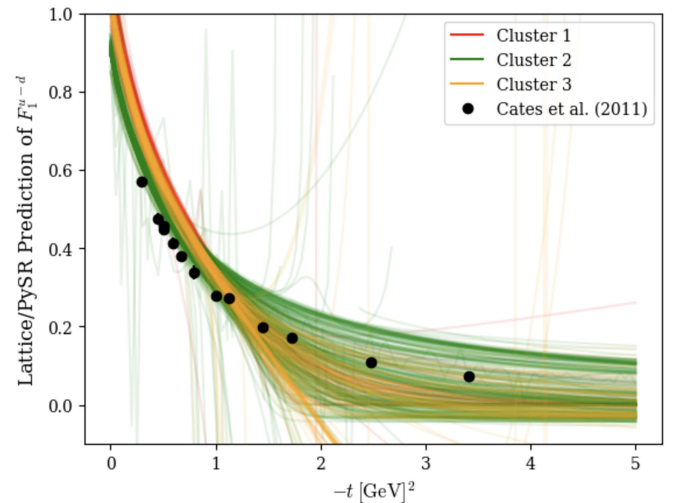


FIG. 12. BF SR results for the flavor separated electromagnetic form factor, F_1^q , Eq. (4), with $q = u - d$, calculated from Fig. 6 (a). Experimental data from Ref. [51]. Within the region of the training data, $0.1 \text{ GeV}^2 \leq |t| \leq 1 \text{ GeV}^2$, the symbolic expressions track the behavior of the form factor.

however, the band does not correspond to a clearly defined probability. Note that the integration over x extends beyond the region of the training data. Therefore, the agreement with data demonstrates that our model extrapolates well in both x and t .

B. PySR selected model results

As shown in Sec. IV A, we were able to map the various extrapolating behaviors of PySR replicas to clusters using ECC. In Table I, we present a typical result for each cluster. The selected LQCD $H_{u-d}(x, t)$ PySR forms are organized as $H_{BF}^i(x, \mathbf{t})$, $H_{FF}^i(x, \mathbf{t})$, $i = 1, K$ taking a Kneedle number of clusters $K = 3$ (Secs. III C and IV A 2), and H_{Regge} . The specific forms of $H_{BF}^i(x, \mathbf{t})$ (BF = best fit), or $H_{FF}^i(x, \mathbf{t})$ (FF = force-factorized), are chosen as the ones that give the lowest MSE in each category, where $\mathbf{t}$ is a dimensionless quantity $\mathbf{t} \equiv |t|/\Lambda^2$, with parameter $\Lambda^2 = 1 \text{ GeV}^2$. Furthermore, by selecting the replica with the lowest MSE for that cluster, it also has finite $A_{10}(t=0)$ (replicas shown in Fig. 10(b), moments shown in Fig. 11). The second column of the table illustrates their 3D graphic forms, while the math expressions are shown in column 3.

The last two columns illustrate the fitness metric, explained in Sec. IV A 4, and the complexity, respectively.

The table also illustrates the PySR exemplar complexities obtained before simplifying with sympy.

H_{Regge} was determined with the RCA method described in Sec. IV A 3. We see that along with the characteristic Regge behavior $x^\alpha(1-x)^\beta$, there is residual x dependence in the exponential. Eulers constant e showing up as the base of the exponential is an artifact of the fit being performed in log space and then exponentiated, as opposed to PySR serendipitously discovering e . Though the MSE of this Reggeized exemplar is notably larger than that of the BF or FF LQCD exemplar, it is the same order of magnitude as the average MSE of FF LQCD replicas (Table II). In future work, we will devise a “force-Reggeized” custom loss function and carry out the procedure of ECC in order to test the consistency and convergence of Reggeized SR replicas, analogous to the treatment of the best-fit and FF replicas.

C. How reliably can we test factorization?

As noted in Sec. I, one can glean information on the nucleon quark and gluon spatial substructure by studying

Fixed $|t|$ ratios $R(|t|)$

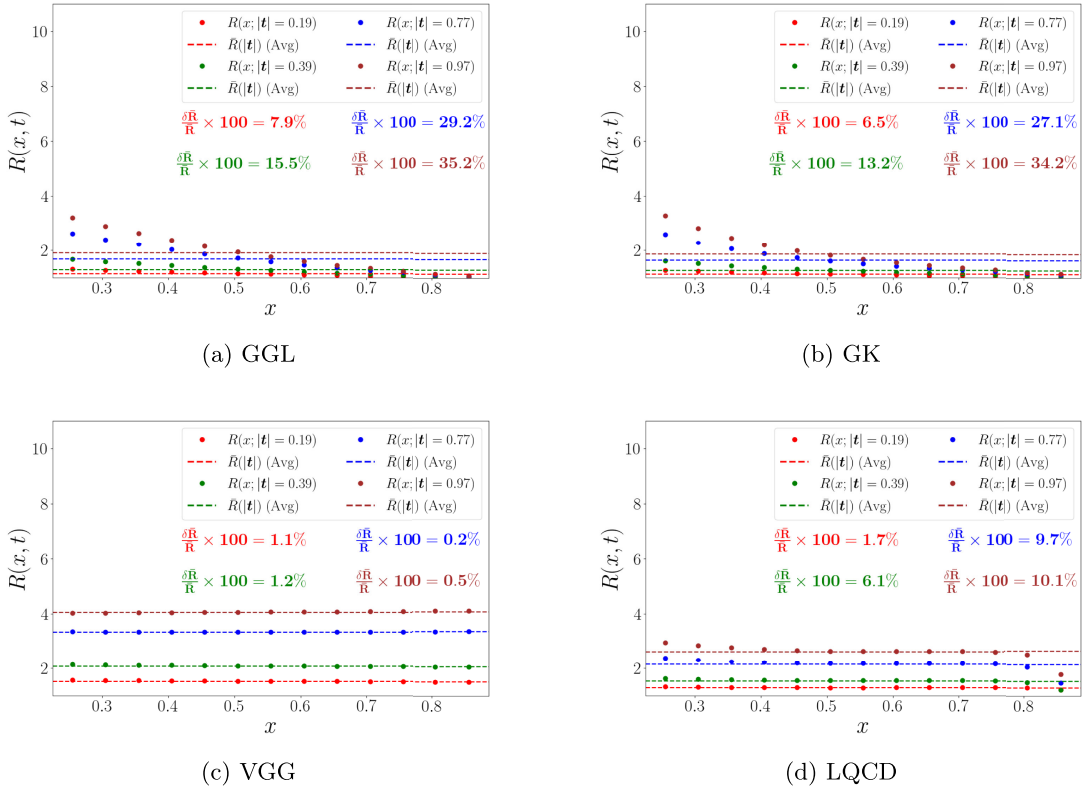


FIG. 13. Ratios $R(x, |t|)$ of GPD parametrizations VGG, GGL, GK, as well as LQCD results [53] plotted vs x for different values of $|t|$: 0.19, 0.39, 0.77, 0.97 GeV^2 . The x dependence of the GGL and GK ratios demonstrates the factorization breaking in these parametrizations. VGG, on the other hand, explicitly factorizes for any quark flavor, and approximately factorizes in the flavor isovector combination. Similarly, we see approximate factorization in the LQCD in this region, in particular in the interval $0.45 \leq x \leq 0.75$. No SR is used to produce the content of this figure.

how configurations of a given x depend on t , and consequently, by Fourier transformation, how they are distributed in the spatial variable $\mathbf{b}$ around the proton's relativistic center of momentum. An obvious case is when the x and t behaviors are independent of each other, describing the GPD in a factorized form that leads to a constant RMS radius for all values of x [see Eq. (8)].

To study this problem, phenomenological parametrizations in the variables x and t have been constructed using known theoretical information on the symmetry constraints and specific functional behavior. For instance, due to the constraint, $H_q(x, 0, 0) = q(x)$, Eq. (9), the GPD x behavior in regions, $x \rightarrow 0$ and $x \rightarrow 1$, can be assumed to be similar to the $x^{-\alpha}$ or Regge behavior, and $(1-x)^\beta$, powerlike behavior obtained from quark counting rules, respectively (we refer the reader to Refs. [57] and [58] for reviews). At intermediate x , interpolated forms can then be derived, preserving the smooth shape of the distributions. The t dependence for these parametrizations is also modeled along similar lines, in this case, using, besides the predicted Regge behavior in t , the constraint from Eq. (4).

For our study of LQCD results, we use three phenomenological GPD parametrizations for the quantity, $H_{u-d} = H_u - H_d$, as benchmarks for the performance of our ansatz in PySR. The chosen parametrizations are labeled as VGG [59], GGL [42], and GK [60], respectively. GGL and GK are constructed so that they explicitly break factorization in x and t , whereas VGG exhibits approximate factorization. The behavior of the distributions concerning factorization can be shown by plotting the ratio $R(x, |t|)$,

$$R(x, |t|) = \frac{H_{u-d}(x, 0)}{H_{u-d}(x, |t|)}, \quad (33)$$

where, if the source/model factorizes, R is constant at fixed $|t|$.

The ratio R is plotted vs x for various values of $|t|$, in Fig. 13, for the phenomenological parametrizations as well as the LQCD results. The approximate factorized behavior of the VGG model is clearly visible as compared to the behavior of GGL and GK which deviate from constant by over 30% at $|t| \approx 0.97 \text{ GeV}^2$, suggesting that in an SR analysis they would be poorly described by a FF model. For the LQCD source data—for which no analytic expression is known—we find that they approximately factorize to within $\approx 10\%$ in the training region. However, the percent deviation of LQCD from constant is larger than that of VGG.

In Figs. 14 and 15 we show the exemplars from each cluster presented in Sec. IV A plotted against LQCD results and the phenomenological models, respectively. The curves in the figures were chosen based on which replica had the lowest MSE per cluster, representing the BF, FF, and semi-Reggeized models. In each panel we plot the various functional forms vs x , at the t values calculated in

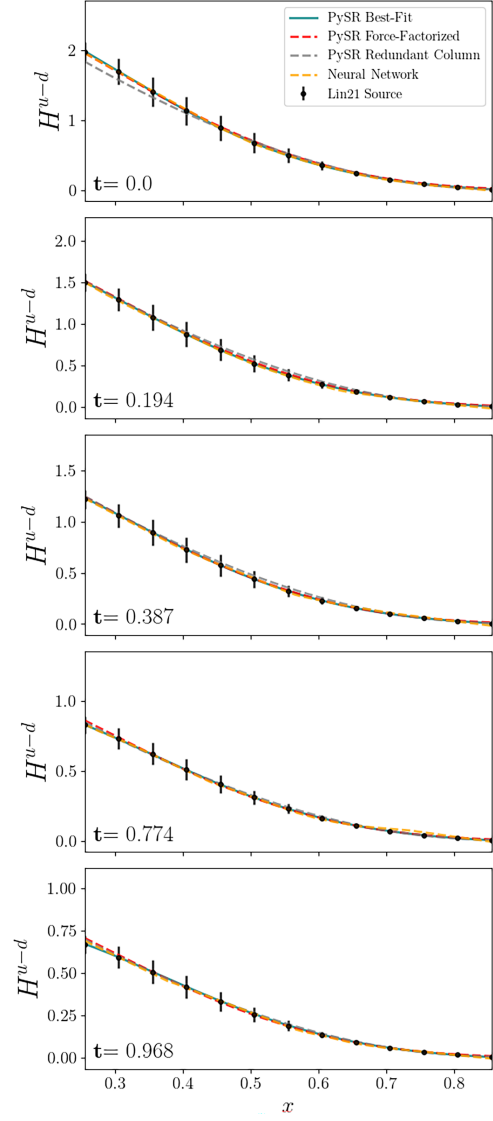


FIG. 14. Array of fits to the lattice source data [29] for $H^{u-d}(x, \mathbf{t})$ projected onto fixed $\mathbf{t} \equiv -t/\Lambda^2$, with $\Lambda^2 = 1 \text{ GeV}^2$. The PySR best-fit, force-factorized, and redundant column models shown correspond to H_{BF}^3 , H_{FF}^2 , and H_{Regge} in Table I, respectively.

LQCD. In the LQCD case (Fig. 14), we also compare our SR results to that of a feed-forward neural network, implemented in the PyTorch library [61]. Considering our dataset is relatively small and low-dimensional, we only require the use of one hidden layer. The popularity of the use of ML for scientific discovery owes much of its rise to the effectiveness of neural networks, which have constituted a large part of the hadronic physics community's embrace of ML techniques, e.g., [62–66]. Although a powerful tool for regression and classification, NNs are plagued with issues of interpretability due to the “black box” nature of their hidden layers. On the other hand, the major advantage of symbolic regression is that the learned model is interpretable—as we demonstrate throughout this

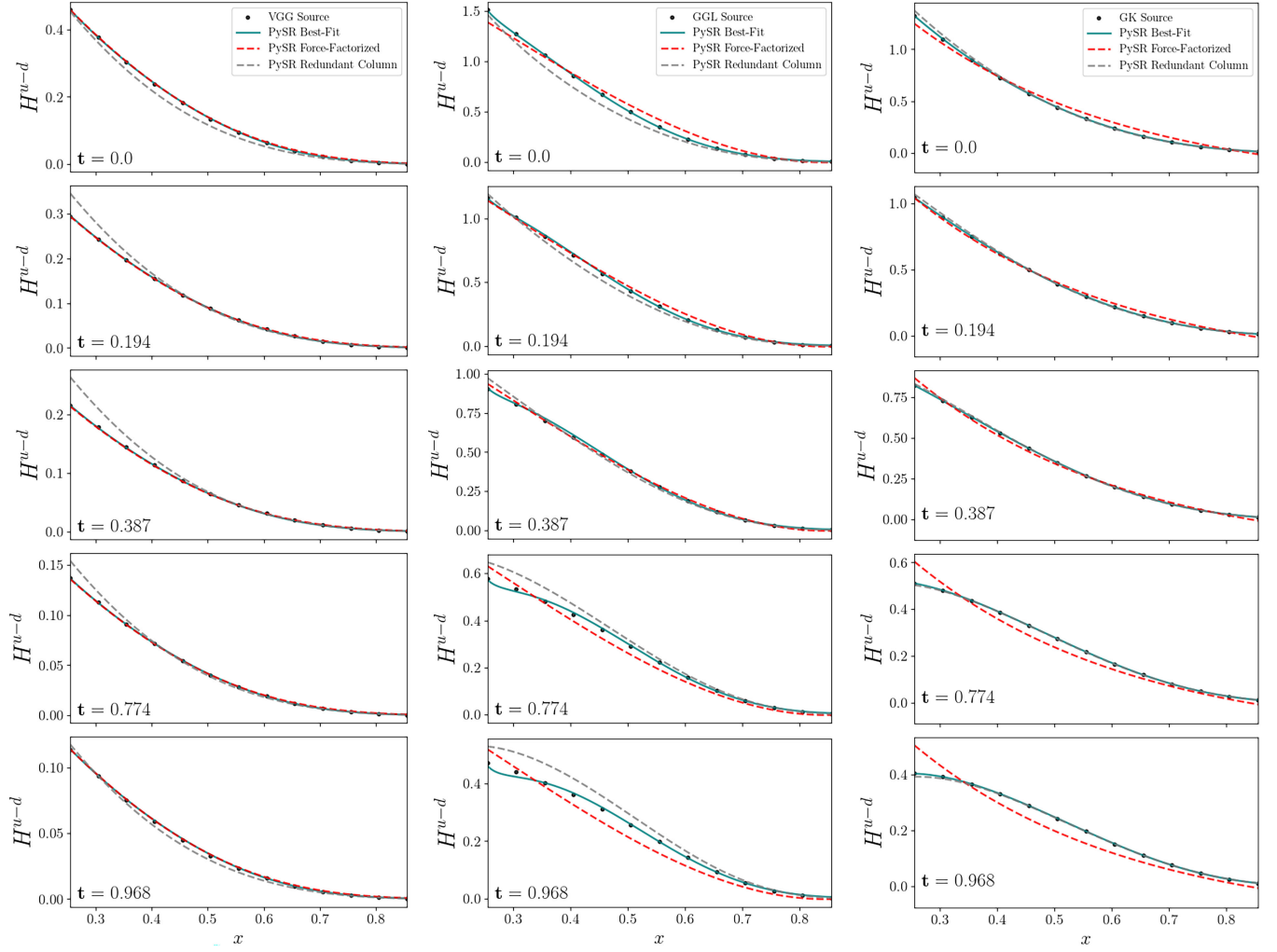


FIG. 15. Array of fits to the VGG [59] (left), GGL [42] (center), and GK [60] (right) source data for $H^{u-d}(x, t)$ projected onto fixed $t \equiv -t/\Lambda^2$, with $\Lambda^2 = 1 \text{ GeV}^2$. Best-fit and force-factorized PySR replicas with the lowest WMSE were selected as exemplars without applying the clustering procedure from Sec. III C and are omitted from Table I.

work, SR oftentimes gives human-readable expressions from which one can immediately extract physics content. While symbolic regression and neural networks both provide excellent fits to the data, as is evident from Fig. 14, we can readily interpret the output of the former and not the latter.

To test PySRs capacity to falsify various physics assumptions in the SR calculated functional forms' behavior, such as factorization, we first define the KL divergence [67] for two probability distributions P and Q evaluated at points $i \in X$,

$$\text{KL}(P||Q) = \sum_{i \in X} P_i \log \frac{P_i}{Q_i}. \quad (34)$$

The KL divergence is a statistic for quantifying the amount of information lost when replacing distribution P with Q . Bound between $0 \leq \text{KL}(P||Q) \leq \infty$, the lower limit is

saturated when the two distributions are identical ($P_i = Q_i \forall i \in X$), and the upper limit is saturated when the same measurements from two distributions are incompatible (e.g., $P_1 \neq 0$ and $Q_1 = 0$).

We use Eq. (34) to compare the MSE distributions from our PySR analysis, applied to all four sources: LQCD and the three models (VGG, GGL, and GK), for both BF and FF custom loss functions. We generate 10^2 PySR replicas for each source and we extract the distribution of MSEs relative to the testing data selected using the equivalent procedure shown in Fig. 5 for each set of replicas. For each source we then compute the KL divergence of the BF MSEs from the FF MSEs,

$$\text{KL}(\text{MSE}_{\text{BF}}||\text{MSE}_{\text{FF}}), \quad (35)$$

through the Python package SciPy.stats.entropy [68]. Our baseline for defining two compatible distributions is

provided by the VGG source that factorizes to $\approx 1\%$ (Fig. 13),

$$KL(MSE_{BF}^{VGG} || MSE_{FF}^{VGG}).$$

In Table II we compare the separation between VGG KL divergence and the ones for LQCD, GGL, and GK source MSEs.

We see from Table II that for all datasets for which the MSE was calculated, the corresponding KL divergences are consistent with VGG and LQCD approximately factorizing, while both GGL and GK are described poorly with a FF model.

VGG and LQCD

Interestingly, the LQCD results outperformed the “truth” case of VGG, with lower values for both the MSE ratio and KL divergence. Though the LQCD source data itself breaks factorization more than VGG (Fig. 13), LQCD results for the MSE ratio and KL divergence still show that LQCD is compatible with factorization on the region. There are many factors that could cause LQCD to outperform the truth case, such as the source data for VGG and LQCD not being identically easy for PySR to learn. This is suggested

by the near order-of-magnitude larger value of MSE_{BF}^{LQCD} compared to that of VGG. That increase in BF MSE closes the gap more between MSE_{BF}^{LQCD} and MSE_{FF}^{LQCD} , artificially making it appear as though LQCD factorizes better than VGG, when in reality BF LQCD results simply converged less than BF VGG.

GGL and GK

Both GGL and GK KL divergences suggest incompatibility with factorization, however, the difference between the two KL divergences naively suggests that GGL is more compatible with factorization than GK. This seemingly contradicts the percent deviations provided in Fig. 13, which show that GGL source data break factorization slightly more than GK. However, as was the case when comparing VGG and LQCD, MSE_{BF}^{GGL} is notably larger than that of VGG, resulting in the apparent contradiction. For this reason, we are unable to establish a hierarchy of source factorization breaking based on these results, but we are able to falsify the assumption that a given source factorizes in general.

In Fig. 16, we present histograms of the MSEs relative to the testing dataset for the BF and FF replicas trained on LQCD, VGG, GGL, and GK, to illustrate how the VGG

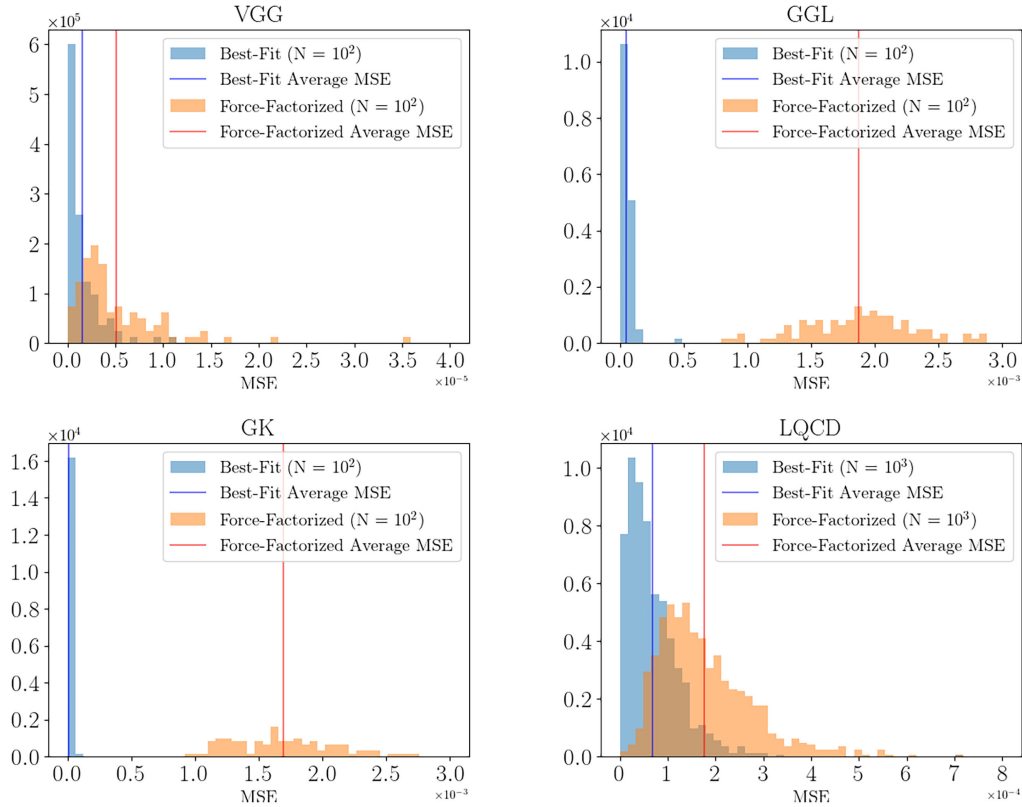


FIG. 16. Histogram of the MSEs relative to the testing dataset for 100 BF and FF replicas trained on phenomenological sources (VGG, GGL, and GK) as well as lattice source data [29]. All MSE distributions are normalized to 1. The overlap in MSEs for the VGG case is consistent with VGG being known to approximately factorize, while the clear separation of MSE for GGL and GK between BF and FF replicas is consistent with these sources being incompatible with the factorization ansatz. Note that the areas shown in these histograms are not to scale, because of the log-scale MSE axis.

and LQCD sources are compatible with factorization. This is shown in the graph by the overlap of MSEs for both BF and FF replicas. On the other side, we also see a clear lack of overlap between BF and FF MSEs for GGL and GK sources, implying that they are not compatible with the factorization hypothesis.

In future work, one might consider generating two sets of BF replicas for each source instead of one, and computing the corresponding KL divergence $\text{KL}(\text{MSE}_{\text{BF}}^1 || \text{MSE}_{\text{BF}}^2)$ as a baseline for compatible distributions. This may help eliminate bias due to comparing KL divergences across sources, rather than the comparison being across different assumptions for the same source.

1. LQCD moments

An even more stringent test of factorization that captures a broader range of the phase space covered by LQCD is obtained by inspecting the behavior of the ratios of GPD moments, Eqs. (4) and (5a), for which LQCD results are available,

$$\frac{A_{20}(t)}{A_{10}(t)}, \quad \frac{A_{30}(t)}{A_{10}(t)}. \quad (36)$$

Deviations of the ratios from a constant behavior in $|t|$ signal a breaking of factorization. In Fig. 17, we show these ratios for moments $A_{i0}(t)$ for $i \in \{1, 2, 3\}$ provided by LQCD [54–56], and computed from our 597/1000 PySR BF replicas that display finite behavior in $A_{i0}(t)$ for $i \in \{1, 2, 3\}$.

The SR results, based on lattice data calculated at the physical pion mass, are compared with two available lattice datasets, Fig. 17. The first corresponds to the ratio A_{20}/A_{10} , also computed at the physical pion mass, allowing for a direct and accurate comparison. The second involves the ratio A_{30}/A_{10} obtained from simulations performed at a

heavier pion mass of $m_\pi = 260$ MeV. In particular, for the A_{20}/A_{10} ratio, the first moment A_{10} was constructed from the electric G_E and magnetic G_M form factors of the nucleon from ETMC Collaboration [55], while A_{20} was taken directly from Ref. [54], where the errors were propagated twice, once to estimate the first moment A_{10} and the second time to estimate the ratio A_{20}/A_{10} . It is important to mention that the errors in the ratio of the lattice data were estimated assuming the data from both moments are uncorrelated, which is not strictly true. These estimates are therefore meant as an approximation for a comparison with our PySR results. Mean values and uncertainties from SR were produced at the same t values of the LQCD points after training and extrapolating $H_{u-d}(x, t)$ with the methodologies detailed in Sec. III.

From Fig. 17 one can see that, even if the differences in the lattice input between Refs. [56] and [29] make a fully quantitative comparison challenging, a nontrivial $|t|$ dependence of moment ratios is incompatible with a factorization ansatz for the unintegrated H_{u-d} , the factorization breaking being more pronounced at larger values of t in both cases. Though the BF replicas do not explicitly factorize in x and t symbolically, they do approximately factorize numerically, as indicated by the near constant value of $A_{20}(t)/A_{10}(t)$. This is because the BF replicas were trained on a range of lattice points that factorized to within $\approx 10\%$ [Fig. 13(d)]. These replicas were convergent on the training region $0 \leq |t| \leq 1$ GeV², which resulted in the approximately constant point-by-point uncertainties in Fig. 17.

On the other side, when analyzing PySR results we should keep in mind that, as shown in Table II, we detected no noticeable preference for the BF exemplars over the FF exemplars. This behavior in the (x, t) region is present for the training data as well, and it will be therefore reflected in the replicas' behavior. It is consistent with a framework

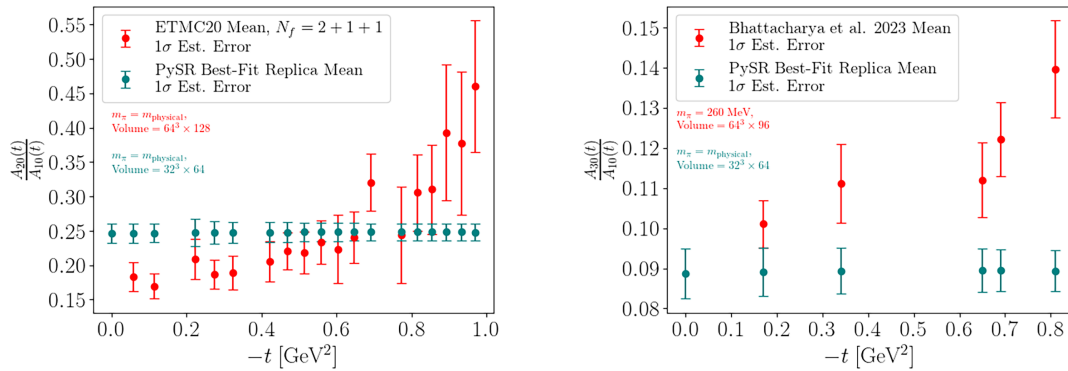


FIG. 17. Left: the reconstructed ratio $A_{20}(t)/A_{10}(t)$ from the produced SR results after training and extrapolating $H_{u-d}(x, t)$ in the x region from the reference lattice dataset [29] (blue) and comparing with available lattice data on $A_{20}(t)$ and $A_{10}(t)$ (red), Refs. [54,55]. Both ratios are based on LQCD data obtained at the physical pion mass for a better and accurate comparison. Right: comparison of ratios $A_{30}(t)/A_{10}(t)$ between lattice data from Ref. [56] computed at higher than the physical pion mass $m_\pi = 260$ MeV (red) and the 636 best-fit PySR replica with finite $A_{10}(t)$ and $A_{30}(t)$ from data at the physical pion mass (blue). The SR ratios in both plots were generated at the same $-t$ values where the lattice data are available.

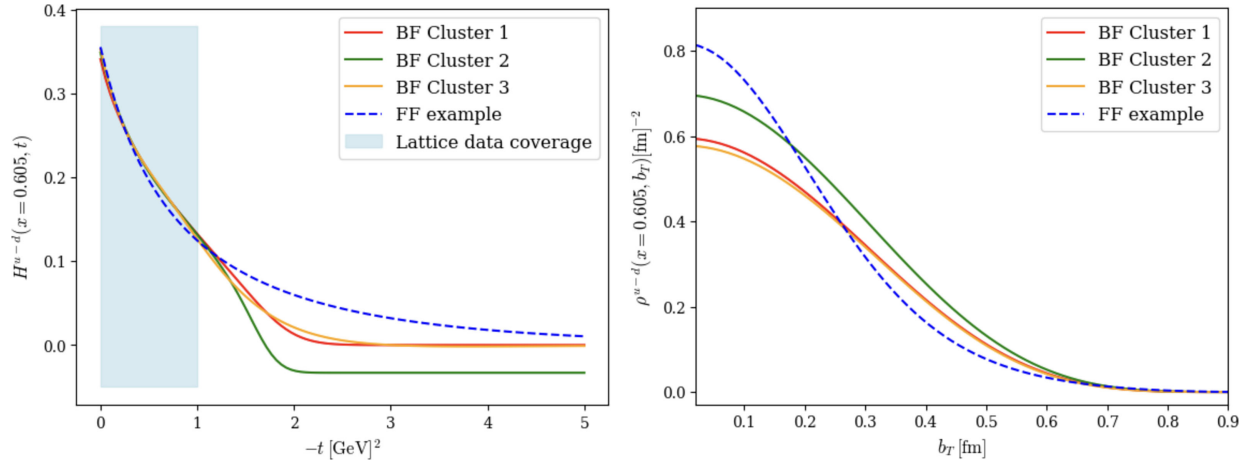


FIG. 18. Studying the t dependence and corresponding $\mathbf{b}_T$ dependence of the BF models from Table I as well as an FF example, Eq. (37). Although all four distributions match within the region of the training data, they differ in the extrapolated region. That difference propagates in the $\mathbf{b}_T$ space and therefore impacts the interpretation of the proton's spatial structure.

where the replicas trained on LQCD x -dependent results approximately factorize, namely, even the BF replicas that were not constrained to factorize still approximately numerically factorize. A further confirmation is also found in the constant behavior on average of both $A_{20}(t)/A_{10}(t)$ and $A_{30}(t)/A_{10}(t)$ with t . This statement is not in tension with current lattice moment results [56] that imply global factorization breaking due to being integrated over x . This could mean that only in regions of (x, t) where $x \leq 0.255$ and $|t| \geq 1 \text{ GeV}^2$ may factorization breaking be demonstrated more quantitatively. In future work, we may impose conditions in the loss function such as requiring that percent deviation of R , defined in Sec. IV C, from constant must be within 10% up to $|t| = 1 \text{ GeV}^2$, beyond which replicas are encouraged to violate factorization.

In summary, from the comparison of LQCD moments with PySR, we can conclude that LQCD data allow for a breaking of factorization in the x, t behavior of GPDs that is entirely focused in the low- x region. Because the moments are integrated over the full range $0 < x < 1$, it is possible that even though factorization is broken globally, $H_{u-d}(x, t)$ still approximately factorize in specific regions in (x, t) , such as the region trained on in this work, without being in tension with other LQCD moment results.

Our results provide an unbiased comparison of different physics scenarios underlying the GPD description. In particular, we can infer from our analysis of LQCD results a picture of the low- x behavior consistent with Reggeized models such as GGL and GK where the x and t dependence appears entangled in the exponential behavior of the parametric form, e.g., $x^{\alpha(1-x)t}$, while it excludes completely factorized forms of the type in Eq. (24).

D. Extracting physical densities

As discussed in Sec. I, one of the key properties of GPDs is their impact parameter space representation $\rho_{q,g}(x, \mathbf{b}_T)$,

which corresponds to spatial densities of the quarks and gluons inside the nucleon. The factorization of the x and t dependence of GPDs becomes crucial in impact parameter space due to its consequences for the quark RMS transverse radius $\langle \mathbf{b}_T^2(x) \rangle^{1/2}$, the “average radii.” One can immediately see from Eq. (8) that if the x and t dependence factorizes, then the x dependence cancels and the radius is constant in x . This would be at variance with the QCD-based prediction for the radius, which requires that as $x \rightarrow 1$, $\langle \mathbf{b}_T(x) \rangle^{1/2} \rightarrow 0$ (for a review of both theory and experiment see Ref. [20]).

In Fig. 18, we present $\rho_{u-d}(x, \mathbf{b}_T)$, the Fourier transform of the GPDs $H_{u-d}(x, 0, t)$, which is obtained from our SR fit to LQCD results: three representative models obtained through the BF method and clustered by WMSE both within and beyond the range of the data are shown along with a FF model, that we choose as

$$H_{\text{FF},u-d}(x, \mathbf{t}) = \frac{[(23.9x + 32.5)^{-1.37}x]^x - 0.016}{1.67t - 0.638}, \quad (37)$$

since it predicts a magnitude for the GPD in the extrapolated region similar to the prediction of the BF models. From the figure, that juxtaposes the GPD as a function of $|t|$ on the lhs, to the density on the rhs for a given value of $x = 0.6$, we can see that GPDs belonging to the range of clusters shown in Figs. 6(b) and 10 produce the largest differences at small $\mathbf{b}$, understandably, since this region is governed by large $|t|$, beyond the reach of LQCD results.

The average radii $\langle \mathbf{b}_T(x) \rangle^{1/2}$, Eq. (8), corresponding to the four densities in Fig. 18, shown in Figs. 18 and 20, illustrate the capability of our SR framework to study different physics scenarios that emerge from the data: the FF one is, of course, constant in x , while the curves from the BF WMSE clustering show different trends for the radii of quark and gluon configurations with different momentum

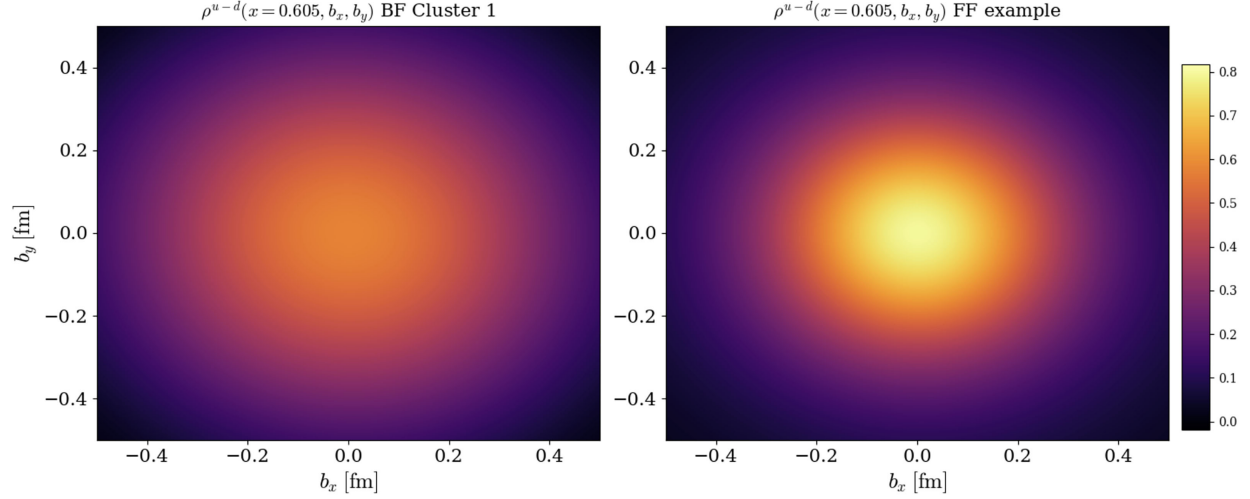


FIG. 19. The BF cluster 1 model compared to an example FF model as a function of b_x and b_y . These densities demonstrate that the u and d quarks are more likely to be located at the proton's center of momentum. While these densities offer a useful visualization for investigating the proton's spatial structure, they do not demonstrate fully the implications of having a factorized versus a nonfactorized GPD, to do so, we need to explore the average radii.

fractions, x , from fast decreasing as $x \rightarrow 1$, to increasing. The latter would rule out the QCD prediction of pointlike configurations [22].

V. CONCLUSIONS AND OUTLOOK: A ROAD MAP TO SR-ASSISTED TOMOGRAPHY

We have demonstrated that SR is a powerful new tool that is capable of new tasks that traditional tools could not accomplish (Table I). Our analysis offers new insights on the hadronic physics side where we have shown that SR can be fruitfully applied to extract new or better information on the proton 3D quark and gluon substructure than traditional methods (Fig. 19). Because SR allows for improved extrapolation capabilities (Fig. 18) beyond the range of LQCD results, we were able to single out a pathway for a quantitative description of the spatial distributions of quarks and gluons on the transverse plane, extracting, for the first time, quantitative trends of their radii (Fig. 20) in different ranges of the momentum fraction, x . These trends emerge in the absence of any known priors, using solely results from LQCD on GPDs and nucleon form factors. On the computational, AI/ML side, we have presented a new approach to systematically benchmark and/or assess the performance of this new tool (Table II), including the ECC convergence criterion (Figs. 3, 6, 8, and 10).

While producing the main results of this work, we have encountered many other instances where our original results can be further elaborated on and/or improved. Listing a few in our outlook on future work includes, e.g., the implementation of additional physics symmetry constraints in the loss function, and considering the full phase space validation, beyond $t = 0$. As information on

the other GPD variables, namely, skewness ξ and the dependence on the renormalization scale μ^2 , become available, we will also be able to impose proper perturbative QCD evolution in the loss function. The SR composite function building blocks can also be further manipulated to understand in more detail the physics underlying the various functional behaviors, and our ECC can be extended to study clustering using other series expansions.

Finally, while community-standard practice generally lacks comprehensive assessment and error analysis of

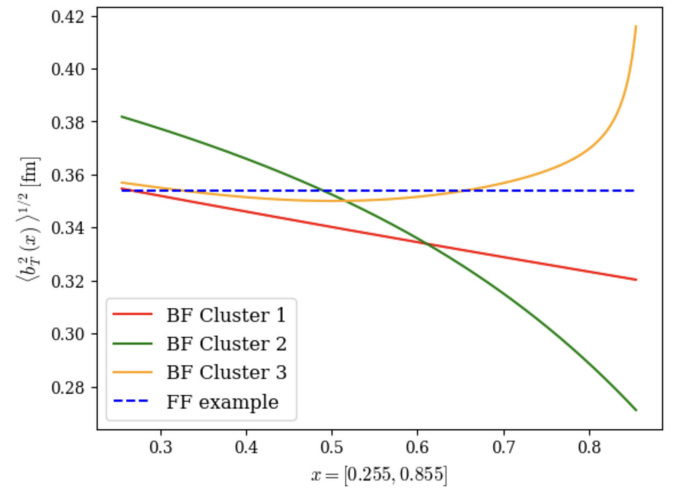


FIG. 20. The average radii $\langle \mathbf{b}_T(x) \rangle^{1/2}$ corresponding to the BF models in Table I as well as an FF example, Eq. (37). The expectation from QCD phenomenology is that as $x \rightarrow 1$, $\langle \mathbf{b}_T(x) \rangle^{1/2} \rightarrow 0$. Notice how the factorized result is a constant in x , and BF cluster 3 turns upwards as $x \rightarrow 1$. We emphasize that all four of these distributions are a prediction of the combined SR and LQCD framework.

SR techniques, even in one dimension, often providing prediction results without associated error bands, we are interested in assessing such errors. A careful uncertainty quantification investigation deserves an entire new study where we will first study SR on a known fixed dataset, and subsequently consider multiobjective optimization with an associated Pareto front analysis. We will also pursue rigorous uncertainty quantification by comparing to uncertainty quantification in Bayesian neural networks, and taking into account the correlations of LQCD results.

This is not yet “the answer” for our extracted GPDs, but this is the tool that will be used to obtain them.

ACKNOWLEDGMENTS

We thank Prasanna Balachandran and Dennis Sivers for fruitful discussions. This work was completed by EXCLAIM Collaboration under the Department of Energy (DOE) Grant No. DE-SC0024644. We also acknowledge DOE Grants No. DE-SC0016286 (D.A., S.L., and Z.P.), No. DE-SC0024560 (M.S.), No. PHY 1653405, No. PHY 2209424, and Research Corporation for Science Advancement through the Cottrell Scholar Award (H.L.), and SURA Center for Nuclear Femtography (S.L., Y.L., and Z.P.).

A. D., Z. P., A. R. S. (listed in alphabetical order), and D. Q. A. contributed substantially to the calculations exhibited in this paper. EXCLUSIVES with AI and Machine Learning (EXCLAIM) Collaboration was recently formed, with the main goal of developing a framework where ML/AI techniques are explored to address problems emerging in the phenomenology of DVES and related processes.

DATA AVAILABILITY

The data that support the findings of this article are publicly available (Ref. [29]). The Python package PySR was used to generate our results with hyperparameters described in the Appendix.

APPENDIX A: CHARGE CONJUGATION AND ISOSPIN DECOMPOSITION

The GPD H^q for a given quark flavor q can be combined with the corresponding antiquark distribution $H^{\bar{q}}$ to form linear combinations which are even (H_q^+) or odd (H_q^-) under charge conjugation,

$$H_q^+(x, t) = H_q + H_{\bar{q}}, \quad H_q^-(x, t) = H_q - H_{\bar{q}}. \quad (\text{A1})$$

Note that, since

$$H_{\bar{q}}(x) = -H_q(-x), \quad (\text{A2})$$

the C-even distributions H_q^+ are *antisymmetric* with respect to $x = 0$, while the C-odd distributions H_q^- are *symmetric*.

In the forward limit ($\xi = t = 0$), the quark GPDs H_q match onto the corresponding collinear distributions [see Eq. (9)]. As such, the conserved net quantum numbers of the proton $p \simeq (uud)$ carried by the valence quarks are reflected in QCD sum rules for the PDFs, which therefore constrain the GPDs,

$$\begin{aligned} \int_0^1 dx(u(x, Q^2) - \bar{u}(x, Q^2)) &= \int_0^1 dx H_u^-(x, 0, 0, Q^2) = 2, \\ \int_0^1 dx(d(x, Q^2) - \bar{d}(x, Q^2)) &= \int_0^1 dx H_d^-(x, 0, 0, Q^2) = 1. \end{aligned} \quad (\text{A3})$$

Writing separately the contribution from quarks/antiquarks that can be generated from gluon splitting and populate the proton sea, $q_s/\bar{q}_s$, and the primordial valence or intrinsic ones, q_v ,

$$H_q = H_{q_v} + H_{q_s}, \quad H_{\bar{q}} = H_{\bar{q}_v} + H_{\bar{q}_s}, \quad (\text{A4})$$

therefore, taking the Kuti-Weisskopf assumption that quark-antiquark pairs are generated perturbatively through gluon splitting, thus, $H_{q_s} = H_{\bar{q}_s}$, one has

$$H_q^+ = [H_{q_v} + H_{q_s}] + H_{\bar{q}_s} \equiv H_{q_v} + 2H_{q_s}, \quad (\text{A5})$$

$$H_q^- = [H_{q_v} + H_{q_s}] - H_{\bar{q}_s} \equiv H_{q_v}. \quad (\text{A6})$$

Analogous expressions can be written for the GPD, E .

We may also decompose the quark flavor dependence into isovector and isoscalar components,

$$H_{u-d} \equiv H_u - H_d = \frac{H_u^- - H_d^-}{2} + \frac{H_u^+ - H_d^+}{2}, \quad (\text{A7a})$$

$$H_{u+d} \equiv H_u + H_d = \frac{H_u^- + H_d^-}{2} + \frac{H_u^+ + H_d^+}{2}. \quad (\text{A7b})$$

APPENDIX B: INPUTS AND HYPERPARAMETERS OF PYSR

As discussed in Sec. III A, PySR is a multipopulation genetic algorithm that generates symbolic mathematical expressions to optimize a loss function with respect to a set of data. PySR incorporates three unique features into its algorithm: simulated annealing to select the winning expression within an island, an “Evolve-Simplify-Optimize” loop when applying the genetic algorithm to a given island, and a strategy of “adaptive parsimony” to maintain a good population of expressions at each complexity.

Simulated annealing is a well-established global optimization technique [69,70] for avoiding the problem of becoming trapped in a local minimum. Rather than accepting the expression with the minimized loss as the winner

with $p = 100\%$ probability, one takes a Boltzmann form $p = \exp(\Delta l / \alpha T)$ where Δl is the change in the expression's loss and αT plays the role of a temperature. For each mutation step $k = 1 \cdots n_c$, with $n_c = 300,000$ by default, PySR takes the temperature to be $T = 1 - k/n_c$, and this simulates the “cooling” from high to low temperatures when searching for the expression with the globally minimum loss. The overall temperature scale of the simulated annealing is set by α , which is taken to be 0.1 by default.

The “Evolve-Simplify-Optimize” loop of PySR specializes the classic evolutionary algorithm for use on equations. This three-step process first takes each expression in an island through n_c steps of random mutations, only thereafter attempting to analytically simplify equivalent mathematical expressions. Finally, the algorithm optimizes the arbitrary constants in each expression to fit the data.

Finally, whereas a traditional “fixed parsimony” penalty to the loss function (proportional to the complexity of an

expression) results in few candidate expressions at higher complexity, PySR chooses a novel “adaptive parsimony” approach. PySR adaptively tunes the per-complexity penalty so that the number of expressions at each complexity is approximately the same. One can loosely think of this as a “chemical potential for complexity,” with the quantity governing the response to the complexity known as the “frequency.”

The various inputs and hyperparameters governing PySR v0.19.4 which was used for this study are shown in Table III to four significant figures. As recommended in the PySR documentation [25], we have defined the power operator $\text{Pow}[x, y]$ so that base x can have any complexity, but the power y is limited to just complexity 1. That is, we only permit expressions with simple powers like a^x or $(2x + 1)^y$, but we disallow forms with expressions in the exponent, like a^{x+y} . Note that this does still permit *nesting* of simple powers, such as $((e \wedge x) \wedge x) \wedge x$.

TABLE III. Hyperparameters, their descriptions, default values in PySR v0.19.4, and the values used in this study.

Parameter	Default	This Study	Description
Alpha	1.574	1.574	Initial temperature for simulated annealing (requires annealing to be true).
Annealing	False	False	Whether to use annealing.
Constraints	Not Used	{“^”(−1, 1)}	Dictionary of int (unary) or 2-tuples (binary), this enforces maxsize constraints on the individual arguments of operators.
Binary_operators	*, /, +, −,	*, /, +, −, ^	List of strings for binary operators used in the search.
CrossoverProbability	7.110e-05	7.110e-05	Absolute probability of crossover-type genetic operation, instead of a mutation.
FractionReplaced	1.337e-02	1.337e-02	How much of population to replace with migrating equations from other populations.
FractionReplacedHof	3.098e-02	3.098e-02	How much of population to replace with migrating equations from Hall of Fame.
Maxsize	30	50	Max complexity of an equation. You can use either maxsize or maxdepth.
Model_selection	”Accuracy”	’Accuracy’	Model selection criterion when selecting a final expression from the list of best expression at each complexity. Can be “accuracy,” “best,” or “score.”
Ncyclesperiteration	2559	2559	Number of total mutations to run, per 10 samples of the population, per iteration.
Niterations	4.0e1	1.0e03	Number of iterations of the algorithm to run. The best equations are printed and migrate between populations at the end of each iteration.
Npop	35	35	Number of individuals in each population.
Optimize_probability	1.489e-02	1.489e-02	Probability of optimizing the constants during a single iteration of the evolutionary algorithm.
Optimizer_iterations	6.0	6.0	Number of iterations that the constants optimizer can take.
Optimizer_nrestarts	6.0	6.0	Number of time to restart the constants optimization process with different initial conditions.
Parsimony	1.257e-03	1.257e-03	Multiplicative factor for how much to punish complexity.
PerturbationFactor	43.86	43.86	Constants are perturbed by a max factor of $(\text{perturbationFactor} * T + 1)$. Either multiplied by this or divided by this.
Populations	23.0	23.0	Number of populations running.
Topn	32	32	How many top individuals migrate from each population.
Tournament_selection_p	0.8375	0.8375	Probability of selecting the best expression in each tournament. The probability will decay as $p * (1 - p)^n$ for other expressions, sorted by loss.

(Table continued)

TABLE III. (Continued)

Parameter	Default	This Study	Description
Unary_operators	sin, cos, exp, log	Not used	Operators which only take a single scalar as input.
UseFrequency	False	False	Whether to measure the frequency of complexities, and use that instead of parsimony to explore equation space. Will naturally find equations of all complexities.
WarmupMaxsizeBy	3.117e-02	3.117e-02	Whether to slowly increase max size from a small number up to the maxsize (if greater than 0). If greater than 0, says the fraction of training time at which the current maxsize will reach the user-passed maxsize.
WeightAddNode	4.523e-03	4.523e-03	Relative likelihood for mutation to add a node.
WeightDeleteNode	14.48	14.48	Relative likelihood for mutation to delete a node.
WeightDoNothing	1.375e-03	1.375e-03	Relative likelihood for mutation to leave the individual.
WeightInsertNode	27.52	27.52	Relative likelihood for mutation to insert a node.
WeightMutateConstant	3.491	3.491	Relative likelihood for mutation to change the constant slightly in a random direction.
WeightMutateOperator	5.418e-03	5.418e-03	Relative likelihood for mutation to swap an operator.
WeightRandomize	1.629	1.629	Relative likelihood for mutation to completely delete and then randomly generate the equation.
WeightSimplify	0.002	0.002	Relative likelihood for mutation to simplify constant parts by evaluation.

-
- [1] R. A. Khalek *et al.*, [arXiv:2103.05419](#).
- [2] M. Diehl, *Eur. Phys. J. C* **19**, 485 (2001).
- [3] A. Belitsky and A. Radyushkin, *Phys. Rep.* **418**, 1 (2005).
- [4] K. Kumericki, S. Liuti, and H. Moutarde, *Eur. Phys. J. A* **52**, 157 (2016).
- [5] L. Biggio, T. Bendinelli, A. Neitz, A. Lucchi, and G. Parascandolo, [arXiv:2106.06427](#).
- [6] N. Makke and S. Chawla, *Mach. Learn. Sci. Tech.* **6**, 025003 (2025).
- [7] M. Morales-Alvarado, D. Conde, J. Bendavid, V. Sanz, and M. Ubiali, in 38th conference on Neural Information Processing Systems (2024), [arXiv:2412.07839](#).
- [8] S. Meissner, A. Metz, and M. Schlegel, *J. High Energy Phys.* **08** (2009) 056.
- [9] J. Collins, *Foundations of Perturbative QCD* (Cambridge University Press, 2011), Vol. 32, ISBN 978-1-009-40184-5, 978-1-009-40183-8, 978-1-009-40182-1.
- [10] J. Collins and T. Rogers, *Phys. Rev. D* **91**, 074020 (2015).
- [11] X.-D. Ji and J. Osborne, *Phys. Rev. D* **57**, R1337 (1998).
- [12] X.-D. Ji and J. Osborne, *Phys. Rev. D* **58**, 094018 (1998).
- [13] X.-D. Ji, *Phys. Rev. D* **55**, 7114 (1997).
- [14] M. Diehl, *Phys. Rep.* **388**, 41 (2003).
- [15] X.-D. Ji, *Phys. Rev. Lett.* **78**, 610 (1997).
- [16] M. Burkardt, *Phys. Rev. D* **62**, 071503 (2000); *Phys. Rev. D* **66**, 119903(E) (2002).
- [17] M. Burkardt, *Phys. Rev. D* **72**, 094020 (2005).
- [18] A. V. Belitsky, X.-d. Ji, and F. Yuan, *Phys. Rev. D* **69**, 074014 (2004).
- [19] S. Liuti and S. K. Taneja, *Phys. Rev. D* **70**, 074019 (2004).
- [20] D. Dutta, K. Hafidi, and M. Strikman, *Prog. Part. Nucl. Phys.* **69**, 1 (2013).
- [21] P. Jain, B. Pire, and J. P. Ralston, *Phys. Rep.* **271**, 67 (1996).
- [22] S. J. Brodsky and G. F. de Teramond, *MDPI Phys.* **4**, 633 (2022).
- [23] B. Kriesten, S. Liuti, L. Calero-Diaz, D. Keller, A. Meyer, G. R. Goldstein, and J. Osvaldo Gonzalez-Hernandez, *Phys. Rev. D* **101**, 054021 (2020).
- [24] M. Cranmer, [arXiv:2305.01582](#).
- [25] <https://astroautomata.com/PySR/> (2023), accessed:2024-12-20.
- [26] S.-M. Udrescu and M. Tegmark, *Sci. Adv.* **6**, eaay2631 (2020).
- [27] R. Dubcakova, *Genet. Program. Evolvable Mach.* **12**, 173 (2011).
- [28] N. Makke, SR-LivingReview, <https://github.com/nmakke/SR-LivingReview?tab=readme-ov-file>, accessed: 2024-12-01.
- [29] H.-W. Lin, *Phys. Rev. Lett.* **127**, 182001 (2021).
- [30] D. Foreman-Mackey, *J. Open Source Software* **1**, 24 (2016).
- [31] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. (Cambridge University Press, Cambridge, 2007).
- [32] Scikit-learn clustering examples, <https://scikit-learn.org/1.5/modules/clustering.html>, accessed: 2024-12-01.
- [33] S. Lloyd, *IEEE Trans. Inf. Theory* **28**, 129 (1982).
- [34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, *J. Mach. Learn. Res.* **12**, 2825 (2011).

- [35] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, in 2011 31st International Conference on Distributed Computing Systems Workshops (IEEE Computer Society, Los Alamitos, CA, USA, 2011), pp. 166–171.
- [36] kneed 0.8.5, <https://pypi.org/project/kneed/>, accessed: 2025-03-01.
- [37] E. Follana, Q. Mason, C. Davies, K. Hornbostel, G. P. Lepage, J. Shigemitsu, H. Trotter, and K. Wong, *Phys. Rev. D* **75**, 054502 (2007).
- [38] A. Bazavov, C. Bernard, J. Komijani, C. DeTar, L. Levkova, W. Freeman, S. Gottlieb, R. Zhou, U. M. Heller, J. E. Hetrick *et al.* (MILC Collaboration), *Phys. Rev. D* **87**, 054505 (2013).
- [39] X. Ji, Y.-S. Liu, Y. Liu, J.-H. Zhang, and Y. Zhao, *Rev. Mod. Phys.* **93**, 035005 (2021).
- [40] H.-W. Lin, J.-W. Chen, X. Ji, L. Jin, R. Li, Y.-S. Liu, Y.-B. Yang, J.-H. Zhang, and Y. Zhao, *Phys. Rev. Lett.* **121**, 242003 (2018).
- [41] G. R. Goldstein, J. O. G. Hernandez, and S. Liuti, *Phys. Rev. D* **91**, 114013 (2015).
- [42] B. Kriesten, P. Velie, E. Yeats, F. Y. Lopez, and S. Liuti, *Phys. Rev. D* **105**, 056022 (2022).
- [43] S. Goloskokov and P. Kroll, *Eur. Phys. J. C* **74**, 2725 (2014).
- [44] M. Vanderhaeghen, P. A. M. Guichon, and M. Guidal, *Phys. Rev. Lett.* **80**, 5064 (1998).
- [45] M. Vanderhaeghen, P. A. M. Guichon, and M. Guidal, *Phys. Rev. D* **60**, 094017 (1999).
- [46] M. Cranmer, Make PySR look for strictly factorised functions, <https://github.com/MilesCranmer/PySR/discussions/465>.
- [47] T. Regge, *Nuovo Cimento* **14**, 951 (1959).
- [48] Y. V. Kovchegov and E. Levin, *Quantum Chromodynamics at High Energy* (Oxford University Press, Oxford, 2013), Vol. 33, ISBN 978-1-009-29144-6, 978-1-009-29141-5, 978-1-009-29142-2, 978-0-521-11257-4, 978-1-139-55768-9.
- [49] S. J. Brodsky and G. R. Farrar, *Phys. Rev. Lett.* **31**, 1153 (1973).
- [50] R. D. Ball *et al.* (NNPDF Collaboration), *Eur. Phys. J. C* **82**, 428 (2022).
- [51] G. D. Cates, C. W. de Jager, S. Riordan, and B. Wojtsekhowski, *Phys. Rev. Lett.* **106**, 252003 (2011).
- [52] I. A. Qattan and J. Arrington, *Phys. Rev. C* **86**, 065210 (2012).
- [53] H.-W. Lin, *Phys. Lett. B* **824**, 136821 (2022).
- [54] C. Alexandrou *et al.*, *Phys. Rev. D* **101**, 034519 (2020).
- [55] C. Alexandrou, S. Bacchio, M. Constantinou, J. Finkenrath, K. Hadjiyiannakou, K. Jansen, G. Koutsou, and A. Vaquero Aviles-Casco, *Phys. Rev. D* **100**, 014509 (2019).
- [56] S. Bhattacharya, K. Cichy, M. Constantinou, X. Gao, A. Metz, J. Miller, S. Mukherjee, P. Petreczky, F. Steffens, and Y. Zhao, *Phys. Rev. D* **108**, 014507 (2023).
- [57] P. V. Landshoff, J. C. Polkinghorne, and R. D. Short, *Nucl. Phys. B* **28**, 225 (1971).
- [58] D. W. Sivers, *Annu. Rev. Nucl. Part. Sci.* **32**, 149 (1982).
- [59] M. Vanderhaeghen, P. A. M. Guichon, and M. Guidal, *Phys. Rev. D* **60**, 094017 (1999).
- [60] S. V. Goloskokov and P. Kroll, *Eur. Phys. J. C* **42**, 281 (2005).
- [61] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, [arXiv:1912.01703](https://arxiv.org/abs/1912.01703).
- [62] K. Kumericki, D. Mueller, and A. Schafer, *J. High Energy Phys.* **07** (2011) 073.
- [63] E. R. Nocera, R. D. Ball, S. Forte, G. Ridolfi, and J. Rojo, *Nucl. Phys. B* **887**, 276 (2014).
- [64] R. D. Ball *et al.* (NNPDF Collaboration), *J. High Energy Phys.* **04** (2015) 040.
- [65] H. Moutarde, P. Sznajder, and J. Wagner, *Eur. Phys. J. C* **79**, 614 (2019).
- [66] J. Grigsby, B. Kriesten, J. Hoskins, S. Liuti, P. Alonzi, and M. Burkardt, *Phys. Rev. D* **104**, 016001 (2021).
- [67] S. Kullback and R. A. Leibler, *Ann. Math. Stat.* **22**, 79 (1951).
- [68] SciPy.stats. entropy, <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.entropy.html>, accessed: 2025-03-03.
- [69] M. Pincus, *Oper. Res.* **18**, 1225 (1970).
- [70] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Science* **220**, 671 (1983).