communications physics

A Nature Portfolio journal

Article

https://doi.org/10.1038/s42005-025-02329-1

Physics-consistent machine learning with output projection onto physical manifolds

Matilde Valente [1], Tiago C. Dias[2], Vasco Guerra [1] ✉ & Rodrigo Ventura[3]

Data-driven machine learning models often require extensive datasets, which can be costly or inaccessible, and their predictions may fail to comply with established physical laws. Current approaches for incorporating physical priors mitigate these issues by penalizing deviations from known physical laws, as in physics-informed neural networks, or by designing architectures that automatically satisfy specific invariants. However, penalization approaches do not guarantee compliance with physical constraints for unseen inputs, and invariant-based methods lack flexibility and generality. We propose a *physics-consistent* machine learning method that directly enforces compliance with physical principles by projecting model outputs onto the manifold defined by these laws. This procedure ensures that predictions inherently adhere to the chosen physical constraints, improving reliability and interpretability. Our method is demonstrated on two systems: a spring-mass system and a low-temperature reactive plasma. Compared to purely data-driven models, our approach reduces errors in physical law compliance, enhances predictive accuracy of physical quantities, and outperforms alternatives when working with simpler models or limited datasets. The proposed projection-based technique is versatile and can function independently or in conjunction with existing physics-informed neural networks, offering an interpretable, general, and scalable solution for developing fast and reliable surrogate models of complex physical systems, particularly in resource-constrained scenarios.

The numerical simulation of physical models is prevalent in science and engineering. These models mathematically represent a physical system, typically by partial differential equations (PDEs) or a set of coupled ordinary differential equations (ODEs). Their development aims at capturing the essential physics of the system, predict its behavior, clarify the principles underlying key observations, and guide experiment design, process optimization and scientific discovery. Computer simulations explicitly solve the differential equations that result from model development. However, they often require solving complex systems of equations and become computationally expensive. Data-driven models have emerged as a promising complement to direct simulations, due to their ability to handle intricate nonlinear input–output relationships.

Machine learning methods are increasingly being used to construct surrogate models for complex physical systems, in disciplines as varied as materials science[1–3], fluid dynamics[4–6], and low-temperature plasmas[7–9]. The reduced computational cost of surrogate models enables real-time control, as quick predictions can be made without execution of the original model. However, these surrogates often require large datasets, which may not be available due to the high cost of data acquisition in practical applications, while their predictive power degrades in the presence of noisy, sparse, or dynamic data[10]. Moreover, being solely dependent on the data provided during the model training, the predictions may fail to comply with known physical laws.

The introduction of physics-informed neural networks (PINNs) by Raissi et al.[11] blends the causality and extrapolation capabilities of physics-based models with the speed, flexibility, and high-dimensional capabilities of neural networks (NNs). By incorporating physical priors described by differential equations into the NN's loss function, PINNs proved to be effective in addressing a variety of practical engineering and scientific challenges[12–16]. Still, as the physical constraints are introduced directly into the NN during training, this approach does not guarantee that the outputs for unseen inputs will satisfy the physical laws after the training process[17–20]. If some properties of the solutions are known, such as, e.g. energy conservation, it is possible to encode them in the network architecture[18,21–24]. However, the need to design specialized network architectures for each system and a specific set of constraints makes it difficult to attain a general

[1]Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal. [2]Department of Electrical Engineering, University of Michigan, Ann Arbor, MI, USA. [3]Instituto de Sistemas e Robotica, Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal.
✉e-mail: vguerra@tecnico.ulisboa.pt

formulation enforcing adherence to the physical laws. Therefore, although PINNs are currently used with success to solve PDEs, fractional equations, integral-differential equations, and stochastic PDEs[19], approaches that introduce general but robust physical laws into machine learning (ML) models remain limited.

In this paper, we present an approach for *physics-consistent* machine learning, motivated by the current limitations in both purely data-driven methods and physics-informed models. Specifically, we focus on studying the effect of projecting the output of an artificial NN onto the manifold defined by a set of chosen physical constraints of the system. In this way, our method leverages fundamental physical principles, such as energy or charge conservation, to correct a posteriori the predictions of an ML model. Consequently, this method ensures physically consistent predictions and improved accuracy, provided the outputs are reasonably close to the target value.

The projection operation is formulated as a constrained optimization problem,

$$\begin{aligned} &\underset{p}{\text{minimize}} \parallel p - f(x;\Theta)\parallel_W^2 \\ &\text{s.t.} \;\; g(x,p) = 0 \end{aligned} \tag{1}$$

where the ML parametric model is defined as $\mathbf{y} = f(\mathbf{x};\boldsymbol{\Theta})$ with $\mathbf{x}$ the input, $\mathbf{y}$ the output, and $\boldsymbol{\Theta}$ is the model parameter vector, the physical laws are defined by the constraint vector $\mathbf{g}(\mathbf{x},\mathbf{y}) = \mathbf{0}$, and W is a symmetric positive definite weighting matrix, i.e., $\parallel \mathbf{v}\parallel_W^2 = \mathbf{v}^{\mathrm{T}} W\mathbf{v}$, being essentially the metric of the output space. The result of the optimization problem in Eq. (1) is denoted by $\mathbf{p}$, corresponding to the projection of the ML output onto the physical manifold.

The projection method constitutes a highly flexible tool to develop fast and reliable surrogate models of complex physical systems that comply with an arbitrary set of physical laws. It enables the use of simpler ML models and/or smaller datasets, while maintaining the same level of predictive accuracy and requiring smaller computational times. Notice that the approach is not restricted to NNs but can also be applied with generality to other ML models, such as support vector regression, decision trees, or even simple linear regression models.

We apply the projection method and study its performance in two different physical systems, used as case studies. In the first one, we build an artificial NN model to predict the time evolution of a spring-mass system given an arbitrary initial condition. The predictions are then corrected by projecting the output onto the manifold defined by energy conservation. The second system is a highly complex and non-linear low-pressure oxygen reactive plasma, created by a DC glow discharge. This system was recently modeled and simulated in ref. 25 and provides an ideal testbed for the proposed approach. In this case, we develop an artificial NN surrogate model according to the reaction mechanism proposed in ref. 25, and project the output onto the manifold defined by charge conservation, plasma quasi-neutrality, and constant operating pressure.

Results demonstrate that the proposed approach ensures strict compliance with general conservation laws while simultaneously improving predictive accuracy. In the spring-mass system, the projection method reduced the energy conservation error by more than four orders of magnitude and improved state variable predictions by up to 72%. In the plasma system case study, compliance errors with physical laws decreased by over 9 orders of magnitude. The projection step introduced a modest ~4% increase in computation time when compared to the NN model, while achieving an error reduction of up to ~64% in weaker models with simpler architectures. Moreover, when comparing models trained with datasets of different sizes, the computational cost was reduced by a factor of ~3.7 while maintaining a nearly identical predictive accuracy. Consequently, these findings establish projection-based physics-consistent machine learning as an interpretable and general method for developing fast and reliable surrogate models in data- and resource-constrained scenarios.

Results and discussion

Framework and workflow

A schematic workflow of our approach is illustrated in Fig. 1. Both systems under study are described by a set of coupled ODEs, and their numerical solution is used as the ground truth. We then train two models: an artificial NN data-driven model (Fig. 1a), and a model where the physics priors are included in the loss function, hereafter denoted as loss-based PINN (Fig. 1b). Specifically, our method is applied as follows: (i) one input vector in provided to the already trained NN model and a prediction vector $f(\mathbf{x};\boldsymbol{\Theta})$ is obtained; (ii) $\mathbf{p}$ is initilized with the output vector $f(\mathbf{x};\boldsymbol{\Theta})$; (iii) the constrained optimization problem is solved by Eq. (1) and the result of the optimization is the corrected output vector $\mathbf{p}$. This way, $\mathbf{p}$ is a point that lies on the physical manifold and thus, by enforcing the physics constraint post-training at inference time, we guarantee that the final prediction is physics-consistent (Fig. 1c). Figure 1c also makes clear the intuition behind the projection method. The same process is applied to each point prediction of the loss-based PINN.

The NN and the loss-based PINN models are used here as base models to demonstrate the proposed projection method. Note, however, that our approach is model-agnostic and can be applied to the predictions of any ML model (e.g., support vector regression or linear regression), as it requires only the physical laws, defined by $g(\mathbf{x},\mathbf{p})$, and a point prediction from the base model, denoted by $f(\mathbf{x};\boldsymbol{\Theta})$. Moreover, it is worth emphasizing that our loss-based PINN, where the physical constraints are very general physical principles, corresponds to problems where the physics laws are more general than in the usual application of PINNs and where PINNs have seldom been tested.

Our analysis focuses on comparing the accuracy of the results before and after the projection of the models' predictions. Additionally, we analyze the robustness of our method across varying model complexity and training, and discuss the physical insight into the underlying physics.

Case study 1: spring-mass system

This example highlights the ability of the projection method to handle sequential predictions and how it corrects the trajectories as they gradually deviate from the target values.

System description. The system consists of two masses, m_1 and m_2, connected in series by two springs with spring constants k_1 and k_2, and natural lengths L_1 and L_2, respectively. The first spring is connected to a fixed wall at one end and to mass m_1 at the other, while the second spring connects the two masses. The masses are restricted to moving along the x-axis, and there is no friction. The positions of m_1 and m_2 along the x-axis are denoted by x_1 and x_2, and their velocities by v_1 and v_2, respectively. The forces exerted by the springs on the masses are determined by the displacements from their equilibrium positions, following Hooke's law.

The differential equations of motion are given by

$$\begin{cases} m_1\ddot{x}_1 = -k_1(x_1 - L_1) + k_2(x_2 - x_1 - L_2) \\ m_2\ddot{x}_2 = -k_2(x_2 - x_1 - L_2) \end{cases}, \tag{2}$$

where $\ddot{x}_1$ and $\ddot{x}_2$ denote the accelerations of masses m_1 and m_2. The time-evolution of the system is determined from the solution of these equations, with the initial conditions

$$\begin{cases} x_1(0) = x_{0,1}, \quad x_2(0) = x_{0,2} \\ v_1(0) = v_{0,1}, \quad v_2(0) = v_{0,2} \end{cases}. \tag{3}$$

Since there is no friction, the mechanical energy, E, defined by the sum of the kinetic energy of the masses and the elastic potential energy, is conserved. Hereafter we consider $m_1 = 1$ kg, $m_2 = 1$ kg, $k_1 = 5\ \mathrm{N\,m^{-1}}$, $k_2 = 2\ \mathrm{N\,m^{-1}}$, $L_1 = 0.5$ m, and $L_2 = 0.5$ m.

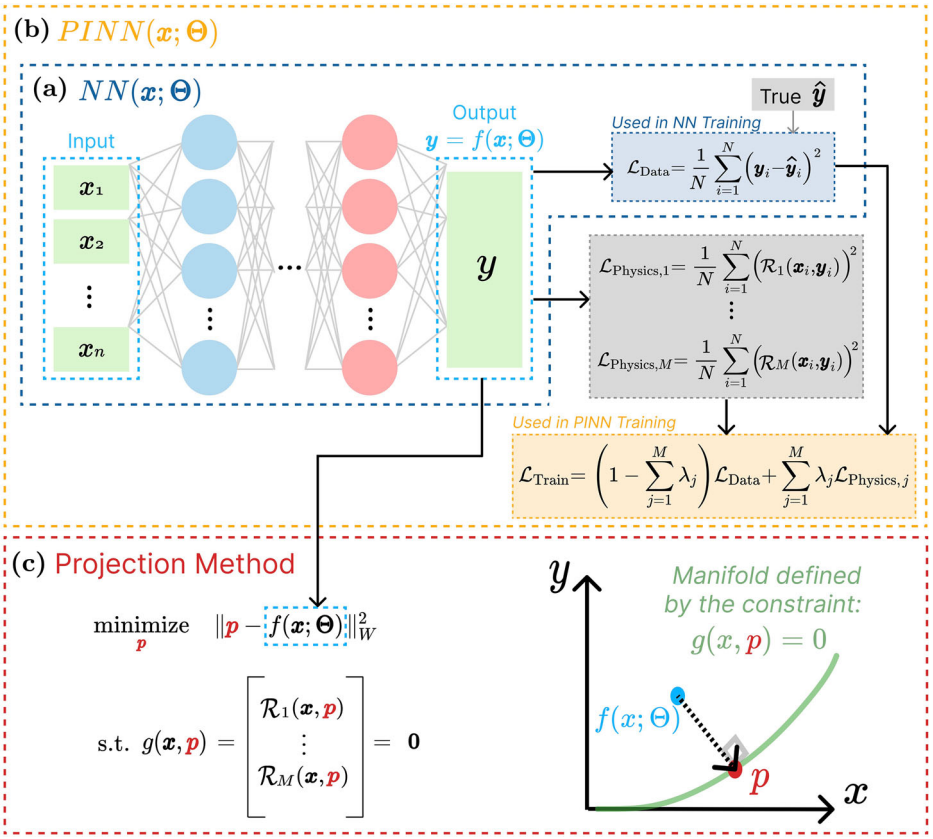

Fig. 1 | Schematic workflow of the approach in this study. a Artificial NN data-driven model $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\Theta})$, where $\mathbf{x}$ represents the input vector, $\mathbf{y}$ is the predicted output vector, and $\boldsymbol{\Theta}$ denotes the model parameter vector. **b** Loss-based PINN model with a regularization term in the loss function, where $\hat{\mathbf{y}}_\mathbf{i}$ represents the target output vector, $\mathbf{y}_\mathbf{i}$ represents the predicted output vector, $\mathcal{L}_{\text{Data}}$ represents the data loss term between the PINN's predicted outputs and the target data, $\mathcal{R}_j$ represents the residual associated with the jth physical law, $\mathcal{L}_{\text{Physics},j}$ represents the physics loss term regarding the jth physical law imposed to the system, $\mathcal{L}_{\text{Train}}$ represents the total training loss function, which is a weighted sum balancing data fitting and adherence to physical constraints, and λ_j represents the weight factor given to the jth physical law, with $\lambda = \sum_{j=1}^{M} \lambda_j \leq 1$. **c** Formulation and visualization of the projection operation method as a constraint optimization problem, with the projection of the output $\mathbf{y}$ of the model onto the manifold defined by the constraint vector $\mathbf{g}(\mathbf{x}, \mathbf{y}) = 0$, where W is a symmetric positive weight matrix.

Data generation. The artificial NN data-driven model takes as input a *state vector* characterizing the state of the system at the current instant t (*i.e.*, defining the positions and velocities of both masses at time t), and predicts the state vector at time $t + \Delta t$, for fixed Δt. This procedure enables the recursive determination of the complete trajectory from a given initial condition, with time-resolution $\Delta t = 50$ ms, as a sequence of transitions between states. We used an energy threshold $E_{\max} = 5$ J to create a set of allowed states for the system, corresponding to the range of positions and velocities the masses can have with $E < E_{\max}$. Eq. (2) was solved using the classical fourth-order Runge–Kutta method (RK4), with arbitrary initial conditions (3) ensuring that $E < E_{\max}$.

Except otherwise noted, the dataset consists of $N = 100{,}000$ arbitrarily generated input state vectors within the energy threshold and the corresponding next states, determined by performing a single Runge–Kutta computation up to time Δt (not to be confused with the Runge–Kutta time-step). To reduce the impact of the difference in feature magnitude on the model and make the training process more stable, we applied the min–max normalization. Consequently, all features were scaled to the range $[-1, 1]$.

Trajectory prediction. The trajectory of the system is defined by the prediction of its state along 165 sequential time steps, i.e., 8.25 s. The general physical law considered in the loss-based PINN and in the projection method is energy conservation, which translates into the residual $\mathcal{R}_1 = E(t) - E(0)$. The results of the four models on the four state-variables and energy conservation are presented in Fig. 2a, b, respectively, for the arbitrary initial condition $x_{0,1} = -0.16$ m, $x_{0,2} = 0.09$ m, $v_{0,1} = -2.18$ m s^{-1} and $v_{0,2} = -0.16$ m s^{-1}. The projection method applied to the outputs of the NN reduces the root mean squared error (RMSE) between the normalized predictions and normalized target values on the four state variables by 49.5%, 71.7%, 21.7%, 42.6% for x_1, v_1, x_2, and v_2, respectively, when compared with the purely data-driven NN model.

The predictions are not only more accurate, but also physically consistent, showing a reduction in the RMSE for the energy conservation law by more than four orders of magnitude when compared with the NN model, reducing the error from $(7.99 \pm 0.35) \times 10^{-1}$ J for the NN to $(1.96 \pm 0.11) \times 10^{-5}$ J after the projection operation, as shown in Fig. 2c. Additionally, the loss-based PINN shows a RMSE for energy conservation of $(8.16 \pm 0.36) \times 10^{-1}$ J, of the same order of magnitude as the data-driven model alone.

To evaluate the robustness of the projection method, we tested 100 different arbitrary initial conditions from the test set and computed the corresponding trajectories for 10 s. The performance metrics across state variables and energy conservation for all four models are represented as violin plots in Fig. 3. Clearly, the projected outputs consistently outperform their non-projected counterparts. When comparing the four models, the NN and PINN projections achieve similar RMSEs for all state variables, with the average RMSEs reduced by approximately 33%, 42%, 17%, and 29% for x_1, v_1, x_2, and v_2, respectively, compared with the non-projected counterparts. Moreover, both projections improve energy conservation, with a reduction in associated error by more than four orders of magnitude, from $\sim 4 \times 10^{-1}$–2×10^{-5} J, showcasing the ability of the method to achieve physics-consistent predictions. Finally, to quantify the dispersion of the RMSEs of the 100 trajectories around its mean value, their standard deviation was computed and corresponds to the error bars in Fig. 3. Although the distribution may deviate from a normal distribution, we conclude the projection method also enhances the prediction stability, as evidenced by consistently smaller standard deviations across most parameters (the only exception is the standard error of x_2 when comparing the PINN with its projected counterpart, which slightly increases from 0.51×10^{-2} to 0.52×10^{-2}).

Additional information is given in Table 1, where we indicate the percentage of initial conditions in which the projection method leads to an improvement in the RMSE averaged over the four state variables (R_{mean}) and in the RMSE of all of the state variables simultaneously (R_{all}). Both the NN and PINN projections produce identical improvement rates compared to their respective base models, with ~96% of the trajectories showing

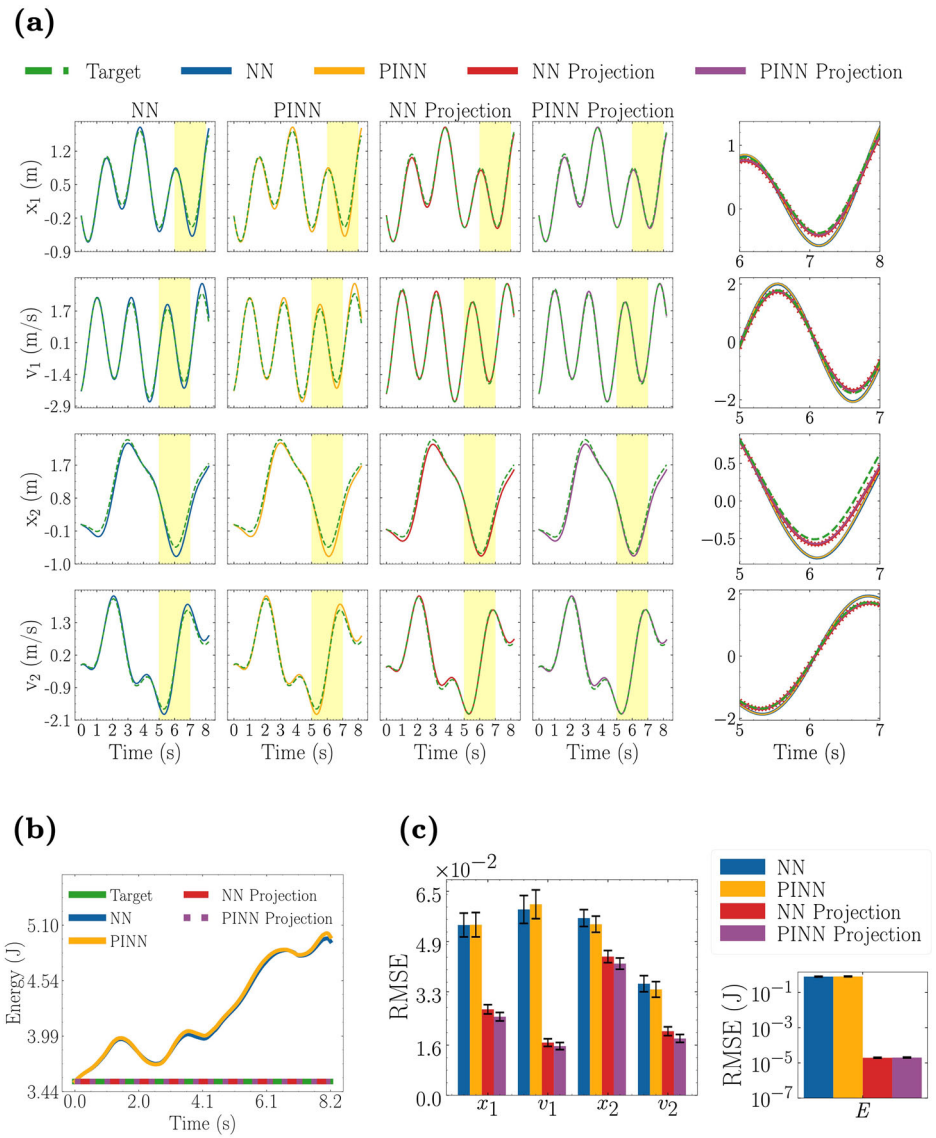

Fig. 2 | Comparative analysis of the models' performance for a given initial condition of the spring-mass system. a Predicted and target positions (x_1, x_2) and velocities (v_1, v_2) over time using the four models considered: NN (first column), loss-based PINN (second column), projection method applied to the NN outputs (third column), and projection method applied to the PINN outputs (fourth column). The yellow-shaded areas highlight regions where the NN and loss-based PINN deviate from the target, demonstrating the projection method's corrective performance. The right column provides zoomed-in views of the yellow-shaded regions. **b** Predicted and target total energy (E) over time using the four models. The energy has units of Joule (J). **c** Bar plot comparing the RMSE for positions, velocities, and energy for the four models considered. The RMSE values are calculated using normalized positions and velocities, while the energy RMSE is computed in Joules, with the initial energy as the target. The error bars represent the error dispersion along the predicted trajectory for each variable. Different colors refer to the predictions of different models: data-driven NN (blue), loss-based PINN (yellow), projection applied to the NN predictions (red), projection applied to the PINN predictions (purple).

enhanced R_{mean} performance and ~60% showing improvement in R_{all}. These results indicate that the projection method consistently provides better predictions than non-projected ones and suggest its benefits are comparable regardless of the underlying base model (NN or loss-based PINN).

Several aspects of this case study are worth highlighting. First, the purely data-driven NN and the loss-based PINN yield similar results in the prediction of the four state variables. Second, the loss-based PINN does not reduce the error in energy conservation compared to the purely data-driven NN, despite having the corresponding constraint in the loss function. Third, the projection method consistently provides better predictions than the non-projected ones, and its benefits are comparable regardless of the underlying base model (NN or loss-based PINN). These trends are evident in Fig. 3 and by comparing the first and second columns in Table 1. These findings suggest that incorporating very general physical principles, such as energy conservation, directly into the loss function as regularization terms may not provide sufficient physical guidance to the model post-training during testing. Somewhat counterintuitively, such an approach likely expands rather than constrains the accessible output parameter space, introducing optimization challenges that impair the model's ability to generalize to new observations. Consequently, while the data loss gradient pushes the model parameters towards the ground truth output values, the inclusion of a physics loss associated to a general law will likely perturb the gradient direction away from the ground truth. Finally, the projection enforces energy conservation, reducing the associated error by 4 orders of magnitude.

Case study 2: low-temperature reactive plasma

This example demonstrates the ability of the projection method to handle complex, high-dimensional, and strongly nonlinear systems. Additionally, we draw on the underlying knowledge of the plasma system to interpret in physical terms the impact of the projection operation on the model outputs.

System description. The system under study is an oxygen (O_2) low-temperature reactive plasma (LTP) created by a continuous DC glow discharge operating at pressures in the range $P \in [0.1, 10]$ Torr, discharge current $I \in [5, 50]$ mA, in a long cylindrical tube of radius $R \in [4, 20]$ mm. As is typical for low-temperature molecular plasmas, the system exhibits a variety of coupled energy pathways and elementary processes, such as electron impact excitation and de-excitation, gas phase and heterogeneous reactions, dissociation, molecule formation, ionization, and charge transfer. A detailed set of reactions and corresponding rate coefficients, validated against benchmark experiments, the so-called

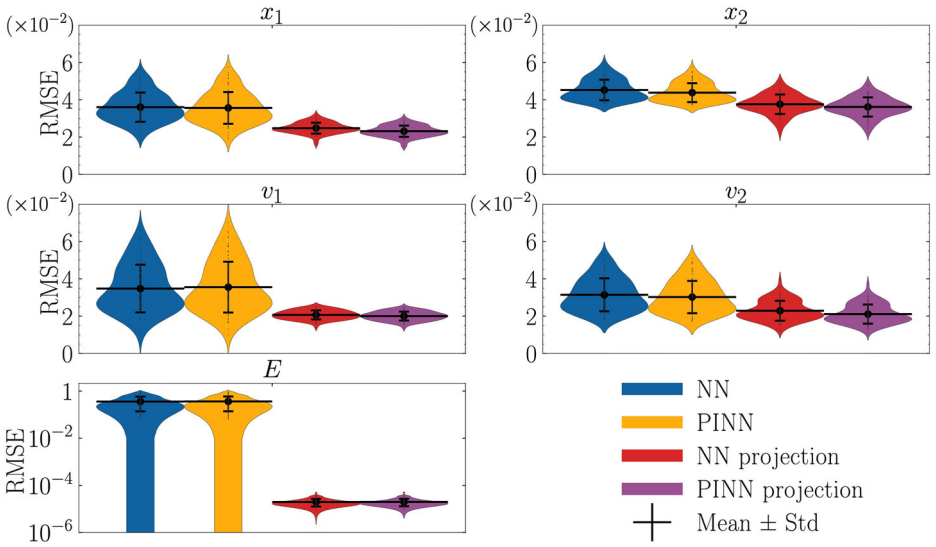

Fig. 3 | Comparative analysis of the models' performance for several initial conditions of the spring-mass system. Distributions of root mean square error (RMSE) of the four models for each normalized state variable—positions (x_1, x_2) and velocities (v_1, v_2)—and for energy conservation (E). Each violin plot contains 100 trajectories and each error bar corresponds to the standard deviation of their RMSEs. Different colors refer to the predictions of different models: data-driven NN (blue), loss-based PINN (yellow), projection applied to the NN predictions (red), projection applied to the PINN predictions (purple).

Table 1 | Improvement rates of the projection method over base models for the spring-mass system

Rates	NN → NN projection	PINN → PINN projection
R_{mean} (%)	96	97
R_{all} (%)	60	61

Percentage of arbitrary initial conditions where the mean RMSE across the four state variables (R_{mean}) improved, and where the RMSE improved simultaneously for all four state variables (R_{all}), for: projected NN compared with NN and projected PINN compared with PINN.

reaction mechanism, was recently developed by T.C. Dias et al.[25], where the experimental data for validation are also given. Herein, we consider the kinetic scheme from ref. 25 without vibrational excitation, which accounts for 12 species—electrons, ground-state molecules $O_2(X)$ and atoms $O(^3P)$, electronically excited states $O_2(a^1\Delta_g, b^1\Sigma_g^+$, Hz) and $O(^1D)$, ground-state ozone O_3 and vibrationally excited ozone $O_3^\star$, negative ions O^-, and positive O_2^+ and O^+ ions—and more than 85 elementary processes. With the exception of the electron density, n_e, the steady-state concentrations of each species n_s are obtained from the solution to the coupled system of ODEs

$$\frac{dn_s}{dt} = \sum_i \left[(a_{si} - b_{si}) k_i \prod_j n_j^{b_{ji}} \right], \tag{4}$$

where a_{si} and b_{si} are the stoichiometric coefficients of species s, as they appear on the left- and right-hand sides of a reaction i, respectively. In addition, the average gas temperature T_g and the gas temperature near the tube wall T_{nw} are calculated from the gas thermal balance equation, the reduced electric field E/N, where N is the gas density, from the quasi-neutrality condition, the electron drift velocity v_d and temperature T_e as integrals over the electron energy distribution function, and the electron density from the discharge current. Further details are given in ref. 25.

Data generation. The models include three input features (P, I, and R) and the 17 outputs just described. We generated the datasets with the LisbOn KInetics Chemistry+Boltzmann (LoKI-B+C) simulation tool[25,26]. Except otherwise noted, the dataset comprises $N = 1000$ uniformly distributed values across the three input features within their specified boundaries. Similarly to the previous case, we applied the min-max normalization to the features, scaling them to the range $[-1, 1]$. Moreover, we applied a log-transformation to the features demonstrating skewness in their distribution. Further details on data generation and preprocessing are given in the "Data pre-processing" subsection in the "Methods" section.

Prediction of the steady-state plasma properties. The general physical laws constraining the system include: the ideal gas law, relating the gas pressure (input) with the species densities and the gas temperature (outputs); the imposed discharge current (input), expressing electric charge conservation and relating its input value with the tube radius (input), and with the drift velocity and electron density (outputs); and the quasi-neutrality law, relating the electron density (output) with the positive and negative ion densities (outputs). These laws can be represented by the residuals in Eqs. (5)–(7), respectively,

$$\mathcal{R}_1 = P - \sum_i [X_i] k_B T_g \tag{5}$$

$$\mathcal{R}_2 = I - e n_e v_d \pi R^2 \tag{6}$$

$$\mathcal{R}_3 = n_e - \sum_i [X_i^+] + \sum_j [X_j^-] \tag{7}$$

where $[X_i]$ is the density of species i, $\sum_i [X_i]$ is the total gas density given by the sum of all the species densities, k_B is the Boltzmann constant, e is the charge of the electron, $\sum_i [X_i^+]$ is the sum of the positively charged ion population, and $\sum_j [X_j^-]$ is the sum of the negatively charged ion population.

The test-set results of the four models on the prediction of the 17 output variables and on the three physical laws are shown in Fig. 4a, b, respectively, expressed as root mean squared errors (RMSE). Applying the projection method to NN predictions reduces the RMSE of compliance with physical laws by more than 9 orders of magnitude, while improving predictive accuracy of three of the output variables: $O_2(X)$, O_2^+, and n_e. As it was the case in the spring-mass system, imposing very general physical laws is ineffective in guiding the loss-based PINN model, which is not able to reduce the RMSE of compliance with physical laws (Fig. 4b) nor of the predicted outputs, when compared with the NN model.

Figure 4c, d provides insight into the physical interpretation of the results, by representing the main species in the mixture and examining the effects of each physical constraint individually. Ground-state $O_2(X)$ molecules constitute 88.3% of the total mixture, and the prediction of their density improves primarily due to the ideal gas law constraint (5), which explicitly accounts for species densities. Similarly, the electron density n_e

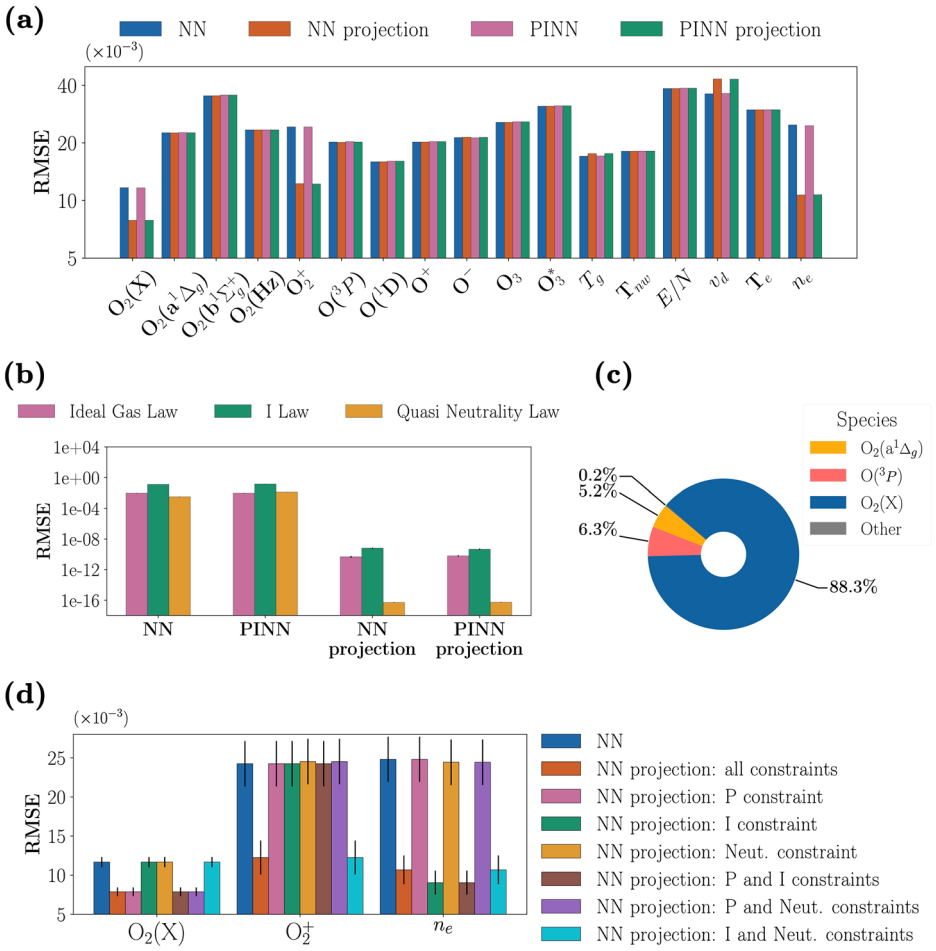

Fig. 4 | Steady-state feature estimation in the low-temperature reactive plasma system. a Test set results of the four models (NN (blue), loss-based PINN (pink), projected NN (red), and projected PINN (green)) when predicting the 17 outputs. The predicted outputs include 11 species densities ($O_2(X)$ and atoms $O(^3P)$, electronically excited states $O_2(a^1\Delta_g, b^1\Sigma_g^+$, Hz) and $O(^1D)$, ground-state ozone O_3 and vibrationally excited ozone $O_3^\star$, negative ions O^-, and positive O_2^+ and O^+ ions), the average gas temperature T_g and the gas temperature near the tube wall T_{nw}, the reduced electric field E/N, where N is the gas density, the electron drift velocity v_d and temperature T_e, and the electron density n_e. **b** Test set results of the four models when evaluating the compliance with physical laws (the ideal gas law (pink), electric charge conservation (green), and the quasi-neutrality law (yellow)). **c** Relative density of the main species in the mixture at steady-state. **d** Bar plot of the three outputs that improved the most when the projection method was applied to the NN predictions. For each output is presented the RMSE of the NN (blue), the projection when the three constraints are applied simultaneously (red), the projection when the constraints are applied individually (pink, green, yellow), and the projection when the constraints are applied in pairs (brown, purple, cyan). `$P$ constraint' refers to the ideal gas law, `I constraint' refers to the electric charge conservation, and `Neut. constraint' refers to the quasi-neutrality law.

benefits from the imposition of the discharge current constraint (6), as it appears directly in the expression of this law. Finally, O_2^+ constitutes <0.0001% of the gas mixture and, as expected, remains unaffected by the ideal gas law (5). It is the dominant positive ion by at least 2 orders of magnitude[25], making its prediction inherently sensitive to the quasi-neutrality condition (7). However, this constraint alone is insufficient to correct the prediction, as the electron density sets an "anchor" for the total ion density. Therefore, the accuracy of the O_2^+ density in the projection method is coupled with the prediction of n_e, and both the discharge current (6) and quasi-neutrality (7) laws are required to improve the model's performance regarding this output.

Robustness of the projection method

With sufficient training time and a sufficiently complex architecture, a neural network can produce predictions that closely approximate the target values, reducing (or even eliminating) the need for post-training corrections. Conversely, a poorly trained model may yield predictions that deviate significantly from the manifold defined by the physical laws, making the projection operation ineffective or leading to ambiguities associated with multiple possible solutions. In this section, we analyze the robustness of the projection method with respect to model complexity, quantified by the number of NN parameters (i.e., weights and biases) and the size of the training dataset. The results focus on the low-temperature reactive plasma system described in the "Case study 2: low-temperature reactive plasma" subsection in the "Results and discussion" section.

Ablation study. We start by comparing the errors before and after applying the projection operation to the NN predictions, considering models with different complexities. In order to guarantee a comparable analysis between the models, we considered 18 different architectures, each with 2 hidden layers and a number of neurons ranging from 1 to 1000 in the hidden layers. Both hidden layers have the same number of neurons. For each architecture, 1 NN model was trained with a dataset consisting of $N = 1000$ data points.

Figure 5a.i shows the RMSE of the predictions of the NN and of the projection method as a function of the number of parameters in the model. These values are calculated by averaging the errors across the 17 outputs. The "RMSE variation rate" is defined as the relative change in RMSE after performing the projection and is considered negative if the RMSE decreases. Figure 5a.ii provides a similar analysis, but for the 3 outputs that improved the most after applying the projection operation, as opposed to the mean of all outputs. The graphs show that the NN projection consistently achieves lower RMSE than the standard NN. The improvement is particularly pronounced for $O_2(X)$, O_2^+ and n_e predictions, in simpler models with fewer parameters (~10^2). As the number of parameters increases to 10^5, both methods reach a performance plateau, and the advantage of the projection method becomes marginal, though it still yields slightly better predictions. These results suggest that the projection method is particularly beneficial for more constrained, and hence computationally more efficient, NN architectures.

It is instructive to analyze the predictions of pressure-dependent trends for fixed discharge current and tube radius. Figure 5b.i–iii illustrates how NN architectures of increasing complexity predict n_e as a function of pressure for $I = 30$ mA and $R = 12$ mm, and benchmarks these predictions to LoKI simulation values. In the simplest architecture (Fig. 5b.i), the predictions of the NN deviate significantly from the targets in both magnitude and trend (RMSE = 2.55×10^{-2}). Despite this poor initial performance, the

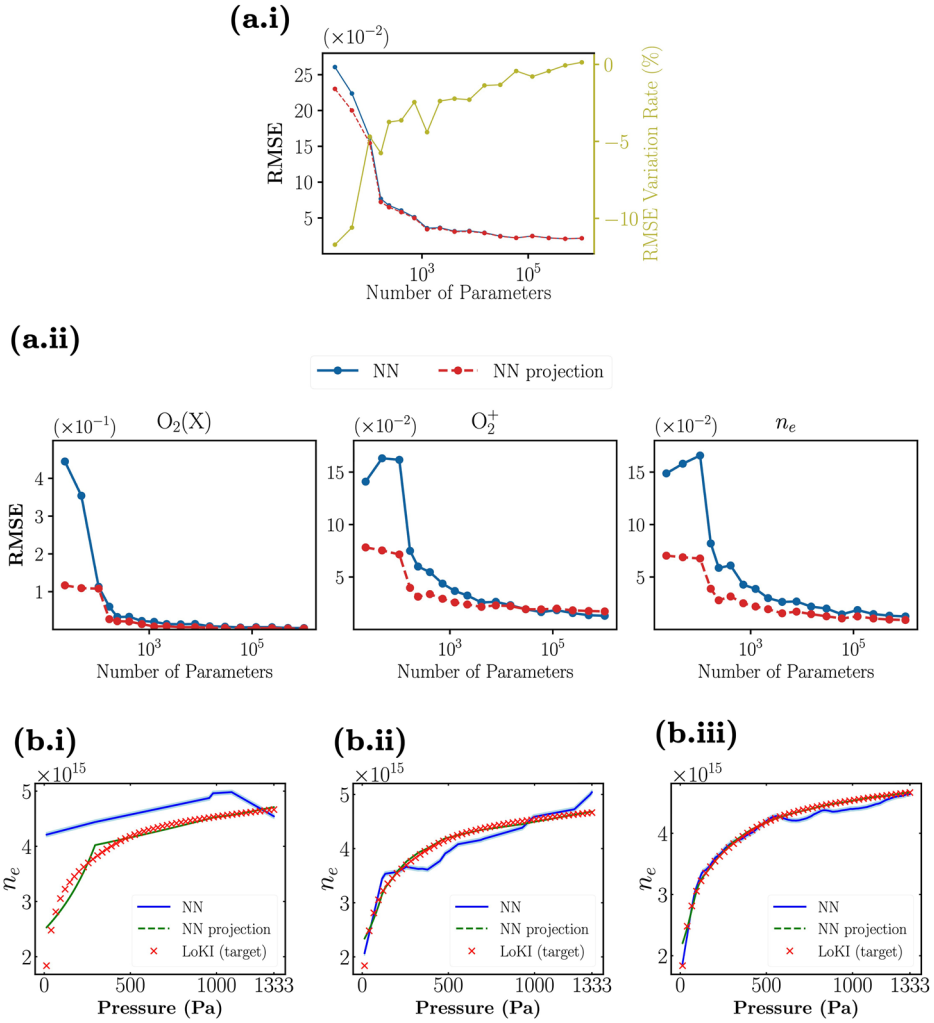

Fig. 5 | Comparative analysis of the low-temperature reactive plasma system's model performance before and after applying the projection operation to the NN outputs. a RMSE of the NN and the NN projection as a function of the model complexity (number of weights and biases in the NN architecture). Blue solid lines with circle markers represent the NN predictions, red dashed lines with circle markers represent the projected NN predictions. The green solid line with circle markers in (i) represents the error variation rate before and after applying the projection to the NN outputs. **b** NN and NN projection pressure related trends of the electron density, n_e, for $I = 30$ mA, $R = 12$ mm, and the following NN architectures: (i) [8,8]; (ii) [26,26]; (iii) [1000,1000]. The target values (red crosses) are obtained from LisbOn KInetics (LoKI) simulation tool. Blue solid lines represent the NN predictions, green dashed lines represent the projected NN predictions.

projection successfully aligns the predictions with the LoKI targets (RMSE = 0.53×10^{-2}), albeit with a visible lack of smoothness. In the intermediate case (Fig. 5b.ii), the projection method smooths the noisy NN curve, reducing the RMSE in ~57%. Finally, in the most complex architecture (Fig. 5b.iii), the NN starts to align with the target values, indicating that sufficient model complexity can compensate for the absence of explicit physical information, though at the cost of increased computational resources.

Small samples. We now compare the errors before and after applying the projection operation to the NN predictions, considering training datasets with different sizes. In order to guarantee a comparable analysis between the models, an architecture with 2 hidden layers and 50 neurons in each layer was used. To mitigate randomness associated with a specific dataset sample, 20 random samples are drawn for each dataset size from a larger dataset. The NN model is then trained on each sample, and the trained models are evaluated on a test set to obtain errors for each of the 17 outputs. The results are shown in Fig. 6a.i, where each point represents the mean error across the 20 samples for each dataset size, considering the average performance on the 17 output predictions. Furthermore, 11 different dataset sizes were analyzed, ranging from 20 to 2500 observations. The aggregate RMSE across all 17 output quantities decreases consistently as the dataset size grows, with both methods showing similar rates of improvement. The projection method maintains a persistent advantage of approximately −2.4% to −2.8% in RMSE variation rate compared with the NN.

Figure 6a.ii depicts a similar RMSE analysis as in Fig. 6a.i, but for the three most relevant individual outputs. A similar conclusion as in the previous section can be drawn: the projected predictions outperform the non-projected counterparts, particularly for smaller dataset sizes, where the model benefits the most from the addition of physical constraints. This is further confirmed by inspection of Fig. 6b, representing the predictions of the electron density as a function of pressure, for the same discharge current and tube radius as in Fig. 5b. With limited training data (Fig. 6b.i), the NN predicts a nonphysical behavior at higher pressures, with a sharp decrease above 800 Pa that contradicts the expected saturation pattern. In contrast, the projection method maintains physically sound predictions across the entire pressure range, effectively compensating for the scarce training data. As expected, the accuracy of the NN progressively improves when trained on larger datasets (Fig. 6b.ii, iii), while the projection method's predictions remain more accurate and consistently aligned with physical expectations. These results demonstrate the role of the projection method as an efficient procedure to include physical constraints into the predictions in scenarios where scarce data is available, maintaining physically meaningful predictions and substantial error reductions with limited training samples.

Computation time. To evaluate the computational performance of our approach, we compare the total computation time required by both the NN and the NN projection methods. Figure 7a depicts the mean RMSE of the predictions for $O_2(X)$, O_2^+, and n_e as a function of the time required to train the models of varying complexity (described in the "Ablation study" subsection in the "Results and discussion" section) and to evaluate their

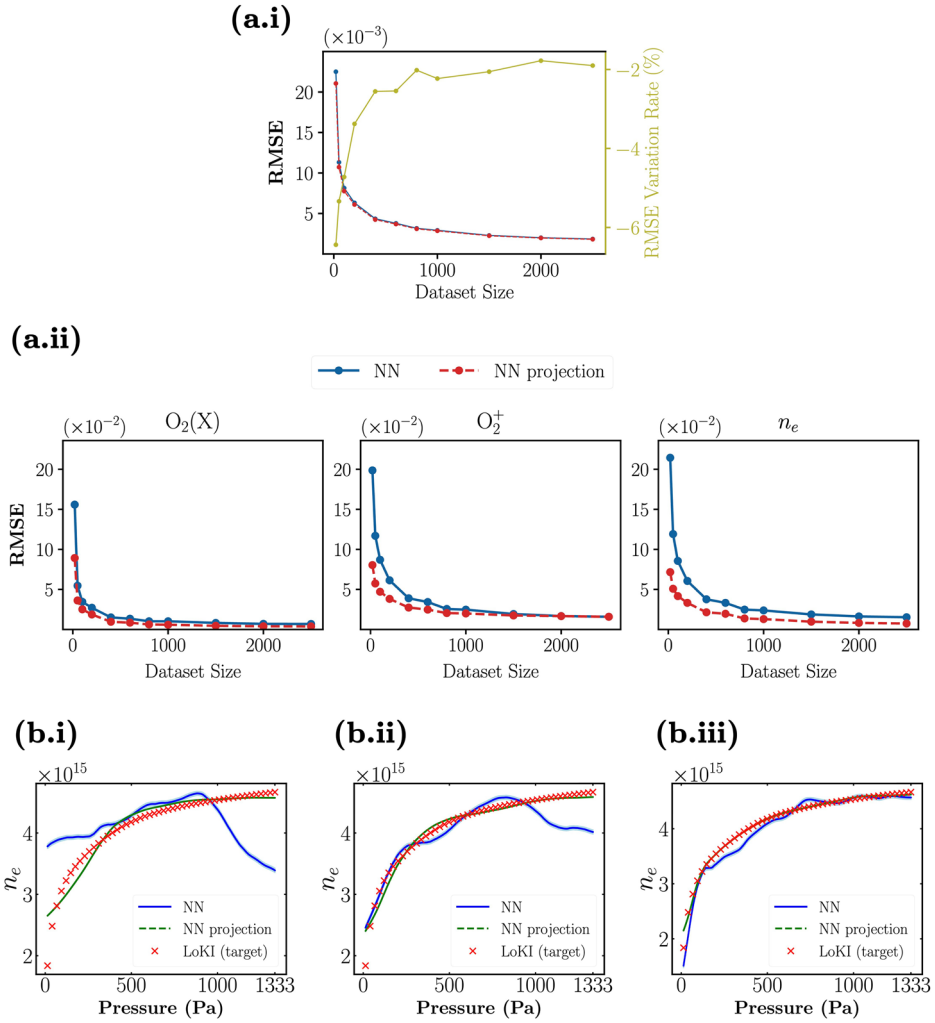

Fig. 6 | Comparative analysis of the low-temperature reactive plasma system's model performance before and after applying the projection operation to the NN outputs. a RMSE of the NN and the NN projection as a function of the dataset size. Blue solid lines with circle markers represent the NN predictions, red dashed lines with circle markers represent the projected NN predictions. The green solid line with circle markers in (i) represents the error variation rate before and after applying the projection to the NN outputs. **b** NN and NN projection pressure related trends of the electron density, n_e, for $I = 30$ mA, $R = 12$ mm, and the following dataset sizes: (i) 200; (ii) 600; (iii) 2500. The target values (red crosses) are obtained from LisbOn KInetics (LoKI) simulation tool. Blue solid lines represent the NN predictions, green dashed lines represent the projected NN predictions.

predictions in a test set. For the NN projection method, the total computation time includes the NN training and inference times, as well as the additional time required to project the outputs onto the physically consistent manifold. The curves correspond to an exponential fitting function to the data and are merely a guide to the eye. We observe that models with longer training times benefit the most from the projection method. Moreover, on average, the projection step takes only 1.75 s for a dataset of 500 observations, which corresponds to a modest 4.33% increase in the total computation time when compared with the NN model. This slight increase in computation time of the projection method comes with a reduction of the RMSE by an average of 32.08%, with improvements reaching up to 63.86% in simpler architectures that require more epochs to converge.

Figure 7b illustrates the mean RMSE of the predictions as a function of the total time required to generate data sets of different sizes (described in the "Small samples: subsection in the "Results and discussion" section), train the models, and evaluate their predictions in a test set. As expected, increasing the dataset size leads to lower RMSE values for both models, but at the cost of significantly longer computation times. Clearly, the NN projection is computationally more efficient than the purely data-driven NN, despite the additional projection step. As an example, the NN trained with 200 observations achieved an RMSE of 0.050, requiring a total computation time of 4913 s. In contrast, applying the projection method to the predictions of the NN trained with only 50 observations resulted in a comparable RMSE of 0.048, but with a significantly reduced computation time of 1338 s. This represents a reduction in computational cost by a factor of ~3.7 while maintaining a nearly identical predictive accuracy.

Conclusions

We introduced a flexible and effective mechanism to improve the quality of the predictions of machine learning surrogate physical models. The approach is based on projecting the model output onto the manifold defined by an arbitrary number of general physical laws, correcting the predictions after the training process, at prediction time. We established the proof-of-concept of our method using neural networks and two case studies, a spring-mass system and a low-temperature reactive plasma. We consistently confirmed the hypothesis that our technique produces physically consistent and scientifically sound results, improves generalizability, and increases efficiency by reducing the computational and data resources necessary to build the surrogate models. Moreover, adding or removing physical constraints merely alters the projection manifold for the post-training predictions, without requiring any modifications to the initial model. This flexibility allows for deeper physical insight and clearer interpretation of the physical results.

Our physics-consistent machine learning projection method can be used independently or as a complement to physics-informed neural networks. For the specific systems under study, PINN's approach of introducing physical information directly into the loss function during training was ineffective in reducing predictive errors. This result indicates that inclusion in the loss function of physically strong but very general laws, such as energy and charge conservation, can bring additional optimization challenges and make post-training projections more appropriate. The difficulty is likely associated with an expansion (rather than a restriction) of accessible output parameter space related to these universal laws. This may occur because, when the training data already adheres to these laws, incorporating them as

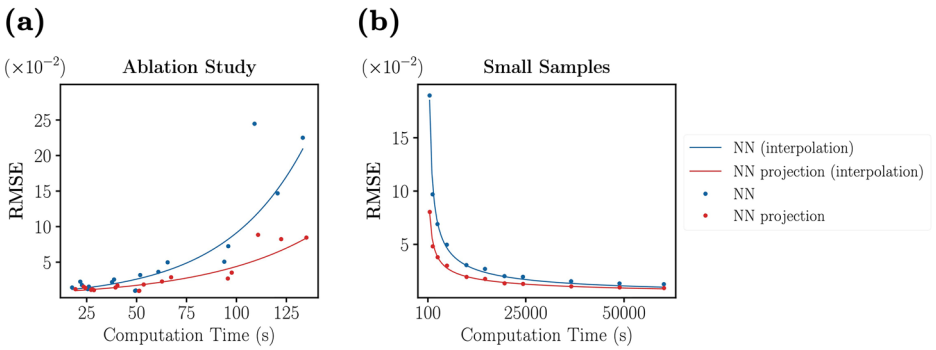

Fig. 7 | Comparative analysis of the mean RMSE of the predictions of $O_2(X)$, O_2^+ and n_e before and after applying the projection operation to the NN outputs as a function of the total computation time. These results refer to the low-temperature reactive plasma system case study. a Analysis for models with ranging complexity. **b** Analysis for models trained with datasets of different sizes.

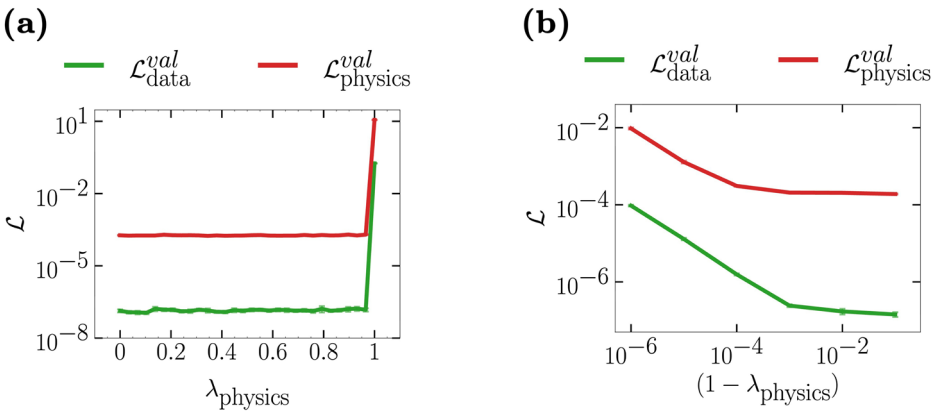

Fig. 8 | Data and physics validation losses as a function of the weighting parameter λ_{physics}. Each point represents the mean MSE across 20 PINNs initialized with different seeds, and the error bars show the standard deviations of the MSEs across the 20 models. $\mathcal{L}_{\text{data}}^{\text{val}}$ and $\mathcal{L}_{\text{physics}}^{val}$ represent, respectively, the data and physics losses on the validation set. a Analysis for $\lambda_{\text{physics}} \in [0, 1)$. **b** Detailed view of the critical region near $\lambda_{\text{physics}} = 1$.

regularization terms adds no new meaningful information to the system. Further research is needed to clarify this issue. Notwithstanding, synergy between PINNs and the projection method is expected. While remaining largely unexplored, it holds the promise of further leveraging artificial intelligence to address scientific and practical engineering problems, especially in contexts of limited resources.

In this work, the projection method was applied to the output of the ML model post-training, rather than being integrated into the training process. Incorporating this approach in a unified manner could potentially enhance the model's generalizability and convergence speed, representing a promising direction for further research. Additionally, concerning certain data features that are known to present challenges for artificial neural networks, such as high-frequency signals, future work will focus on evaluating how our methodology can improve learning under such conditions.

Methods

Data pre-processing

For both the spring-mass and the low-temperature reactive plasma (LTP) systems, we used min-max scaling normalizing features to the $[-1, 1]$ interval, thereby ensuring equal contribution from all features to the loss function and mitigating magnitude disparities. In addition, the LTP system exhibited skewness, particularly evident in low-pressure regimes. We quantified this skewness for each output variable and applied logarithmic transformations when the skewness exceeded a predetermined threshold. This transformation was necessary for several output parameters, specifically $O(^1D)$, O^+, E/N, and T_e.

Model training and optimization

In both case studies, fully connected feed-forward NNs were implemented in PyTorch. Moreover, a standard train-test-validation split of the dataset was performed to evaluate model performance, with 80% allocated for training, 10% for validation, and 10% for testing. For the relatively straightforward spring-mass system, we set a fixed training duration of 60 epochs. In turn, an early stopping criterion is used in the modeling of the highly complex LTP system. The implemented early stopping criterion follows Prechelt's PQ_α early stopping method[27], an approach that aims to balance the trade-off between training time and generalization performance. The implementation handles special cases such as insufficient epochs and zero training progress scenarios, with the threshold parameter α controlling stopping sensitivity. Xavier's initial value[28] is used as the initial NN parameters, which are updated using the gradient-based algorithm Adam[29].

Spring-mass system. In the spring-mass system from the "Case study 1" subsection in the "Results and discussion" section, the NN takes the system's current state variables (x_1, v_1, x_2, v_2) and predicts the state variables in the following state. We use a standard mean squared error (MSE) loss function for the NN training, while for the training of the loss-based PINN an additional loss function is defined as the residuals of the energy conservation law (cf. Fig. 1b). In this way, the loss terms are written as

$$\mathcal{L}_{\text{physics}} = \text{MSE}(E_{\text{out}}, E_{\text{in}}) \tag{8}$$

$$\mathcal{L}_{\text{data}} = \text{MSE}(y_{\text{out}}, \hat{y}) \tag{9}$$

$$\mathcal{L}_{\text{total}} = (1 - \lambda_{\text{physics}}) \cdot \mathcal{L}_{\text{data}} + \lambda_{\text{physics}} \cdot \mathcal{L}_{\text{physics}} \tag{10}$$

where E_{out} and E_{in} are the system energies calculated from the output and input state variables, respectively, and λ_{physics} is the positive weighting parameter given to the energy conservation constraint defined in $[0, 1]$.

To determine λ_{physics} we conducted a grid search over more than 10 orders of magnitude to systematically assess its influence on the loss-based PINN model performance on a validation set. By covering the entire search space, this approach provides a transparent method for studying the trade-off between $\mathcal{L}_{\text{data}}$ and $\mathcal{L}_{\text{physics}}$. As shown in Fig. 8, both $\mathcal{L}_{\text{data}}$ and $\mathcal{L}_{\text{physics}}$

exhibit limited sensitivity to $\lambda_{physics}$, except when its value is ~1, where increasing $\lambda_{physics}$ leads to a degradation in performance. Specifically, no clear minimum is observed across the scanned range, and neither metric shows consistent improvement relative to the standard NN (i.e., $\lambda_{physics} = 0$). These results suggest that, for the physical system under study and the particular formulation of the loss-based PINN, the inclusion of a physics regularization term does not improve either data generalization or physical compliance on new observations. Accordingly, for the analysis in the "Case study 1: spring-mass system" subsection in the "Results and discussion" section, we selected $\lambda_{physics} = 0.005$, a value that ensures that $\mathcal{L}_{data}$ and $\mathcal{L}_{physics}$ are balanced in terms of order of magnitude, so that they make similar contributions to the training process.

The same NN structure was used for all examples. Leaky ReLU activation functions were applied between the fully connected layers of sizes [22, 98, 9]. The learning rate in the Adam algorithm[29] was set to $\eta = 0.0001$. Finally, a maximum number of 60 training epochs was defined to allow the comparison between models.

Low-temperature reactive plasma. In the system studied in the "Case study 2: low-temperature reactive plasma" subsection in the "Results and discussion" section, we use a standard mean squared error (MSE) loss function for the NN training, as before. Regarding the training of the loss-based PINN, we define a loss function as the residuals of each of the three physical laws (5)–(7). The corresponding terms are written as

$$\mathcal{L}_{\mathrm{P}} = \mathrm{MSE}(P_{\mathrm{out}}, P_{\mathrm{in}}) \tag{11}$$

$$\mathcal{L}_{\mathrm{I}} = \mathrm{MSE}(I_{\mathrm{out}}, I_{\mathrm{in}}) \tag{12}$$

$$\mathcal{L}_{\mathrm{n_e}} = \mathrm{MSE}(n_{e,\mathrm{out}}, n_{i,\mathrm{out}}) \tag{13}$$

where P_{in} and I_{in} are the input pressure and discharge current, respectively, P_{out} and I_{out} are calculated from the output predictions and input features (see Eqs. (5) and (6)), $n_{\mathrm{e,out}}$ is the predicted electron density and $n_{\mathrm{i,out}}$ is the predicted difference between the total positive and negative ion densities (see Eq. (7)). Consequently, the total loss function associated with the physical constraints is given by

$$\lambda_{\mathrm{physics}} = \lambda_{\mathrm{P}} + \lambda_{\mathrm{I}} + \lambda_{n_e} \tag{14}$$

$$\mathcal{L}_{\mathrm{physics}} = \lambda_{\mathrm{P}} \cdot \mathcal{L}_{\mathrm{P}} + \lambda_{\mathrm{I}} \cdot \mathcal{L}_{\mathrm{I}} + \lambda_{n_e} \cdot \mathcal{L}_{n_e} \tag{15}$$

where λ_{P}, λ_{I}, and λ_{n_e} are positive weighting parameters given to each physical constraint defined in $\lambda_{\mathrm{physics}} \in [0, 1)$, while $\mathcal{L}_{\mathrm{data}}$ and $\mathcal{L}_{\mathrm{total}}$ are given as before, by Eqs. (9) and (10). We have carried out a study on the effect of $\lambda_{\mathrm{physics}}$ on the loss function, similar to the one discussed in the "Spring-mass system" subsection in the "Methods" section. As before, there was no clear correlation between the value of $\lambda_{\mathrm{physics}}$ and the performance of the loss-based PINN model in the validation set. Therefore, $\lambda_{\mathrm{physics}}$ was selected to balance in terms of orders of magnitude the values of $\mathcal{L}_{\mathrm{data}}$ and $\mathcal{L}_{\mathrm{physics}}$ in $\mathcal{L}_{\mathrm{total}}$. Moreover, the same weight was given to each of the three laws, being $\lambda_{\mathrm{physics}} = 0.015$.

The same NN structure, with Leaky ReLU activation functions applied between 2 fully connected layers of sizes [50, 50], is used for all examples, except for the analysis in the "Ablation study" subsection in the "Results and discussion" section. The learning rate is updated dynamically to ensure a more efficient training process. In this way, by monitoring the convergence of the validation loss during the training process, the learning rate is reduced by a factor of 10 when the validation loss reaches a plateau.

Finally, to reduce the dependence on the seed selection and to obtain statistically significant results, a bootstrapped model was trained. In particular, in the results in the "Prediction of the steady-state plasma properties" subsection in the "Results and discussion" section, $N = 30$ different NN models initialized with different seeds are trained. The final model prediction is given by the mean of the individual predictions of the N models, and a quantification of the uncertainty of the prediction is obtained by computing the corresponding standard deviation.

The projection method

Our projection method can be formalized as follows. Consider an ML parametric model $y = f(x; \Theta)$, where x is the input, y the output, and Θ is the model parameter vector. Given a set $\mathcal{D} = \{(x_k, y_k)\}$ of datapoints for training and assuming a loss function $\mathcal{L}(y_1, y_2)$, training the model amounts to solving the optimization problem

$$\underset{\Theta}{\text{minimize}} \sum_{(x,y)\in\mathcal{D}} \mathcal{L}(y, f(x;\Theta)). \tag{16}$$

Additionally, the requirement that a set of physical laws relating both $\mathbf{x}$ and $\mathbf{y}$ must be satisfied is considered. This requirement can be expressed as a vector-valued constraint function $\mathbf{g}(\mathbf{x}, \mathbf{y})$ that is zero if, and only if, those physical laws are satisfied, i.e., $\mathbf{g}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$. Even though the input dataset $\mathcal{D}$ might satisfy this constraint, there is no a priori guarantee that the ML model f will output values that satisfy these laws.

One common approach to add physical information to an ML model is to include in the loss function $\mathcal{L}$ a regularization term that penalizes violations of the physical constraints. However, once the model is trained, there is still no guarantee that the outputs for unseen inputs satisfy those constraints. Moreover, if the physical laws are too general, they may fail to guide the NN and improve its predictions, as shown in this work. Here, we follow an alternative procedure that explores the idea of projecting the output y of the model onto the manifold defined by the constraint $g(x, y) = 0$.

The projection operation is formulated as the constraint optimization problem defined by Eq. (1) and was implemented using CasADI opti stack[30]. Specifically, the nonlinear programming solver IPOPT was used to solve the optimization problem with tolerances set to 10^{-8} for the LTP system and 10^{-3} for the spring-mass system. In addition, all along the paper we performed the projection with the identity matrix, $W = \mathbb{1}$. Strategies for optimizing the projection with a different weighting matrix can be explored and are left for future work.

Data availability

The datasets generated in this study are available on Github [https://github.com/matildevalente/physics_consistent_machine_learning].

Code availability

The code that supports the findings in this study is available on Github [https://github.com/matildevalente/physics_consistent_machine_learning].

Received: 25 March 2025; Accepted: 25 September 2025;
Published online: 17 November 2025

References

1. Nyshadham, C. et al. Machine-learned multi-system surrogate models for materials prediction. *npj Comput. Mater.* **5**, 51 (2019).
2. Schmidt, J., Marques, M., Botti, S. & Marques, M. Recent advances and applications of machine learning in solid-state materials science. *npj Comput. Mater.* **5**, https://doi.org/10.1038/s41524-019-0221-0 (2019).
3. Jiang, J., Chen, M. & Fan, J. A. Deep neural networks for the evaluation and design of photonic devices. *Nat. Rev. Mater.* **6**, 679–700 (2021).
4. Pache, R. & Rung, T. Data-driven surrogate modeling of aerodynamic forces on the superstructure of container vessels. *Eng. Appl. Comput. Fluid Mech.* **16**, 746–763 (2022).
5. Dhakane, V. & Yada, A. Computational fluid dynamics-deep neural network (CFD-DNN) surrogate model with graphical user interface (GUI) for predicting hydrodynamic parameters in three-phase bubble column reactors. *Ind. Eng. Chem. Res.* **63**, 11670–11685 (2024).

6. Zhang, W. et al. Convolutional neural networks-based surrogate model for fast computational fluid dynamics simulations of indoor airflow distribution. *Energy Build.* **326**, 115020 (2025).
7. Bonzanini, A. D. et al. Foundations of machine learning for low-temperature plasmas: methods and case studies. *Plasma Sources Sci. Technol.* **32**, 024003 (2023).
8. Liu, P., Wu, Q., Ren, X., Wang, Y. & Ni, D. A deep-learning-based surrogate modeling method with application to plasma processing. *Chem. Eng. Res. Des.* **211**, 299–317 (2024).
9. Zhao, Y., Chen, W., Miao, Z., Yang, P. & Zhou, X. Deep learning-assisted magnetized inductively coupled plasma discharge modeling. *Plasma Sources Sci. Technol.* **33**, 125013 (2024).
10. Diaw, A. et al. Efficient learning of accurate surrogates for simulations of complex systems. *Nat. Mach. Intell.* **6**, 568–577 (2024).
11. Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comp. Phys.* **378**, 686–707, (2019).
12. Wong, H. S. et al. Strategies for multi-case physics-informed neural networks for tube flows: a study using 2D flow scenarios. *Sci. Rep.* **14**, 11577 (2024).
13. Pun, G. P. P. et al. Physically informed artificial neural networks for atomistic modeling of materials. *Nat. Commun.* **10**, 2339 (2019).
14. Huang, B. & Wang, J. Applications of physics-informed neural networks in power systems—a review. *IEEE Trans. Power Sys*. **38**, 572–588 (2023).
15. Moschou, S. P. et al. Physics-informed neural networks for modeling astrophysical shocks. *Mach. Learn.: Sci. Technol.* **4**, 035032 (2023).
16. Seo, J., Kim, I. H. & Nam, H. Leveraging physics-informed neural computing for transport simulations of nuclear fusion plasmas. *Nucl. Eng. Technol.* **56**, 5396–5404 (2024).
17. Kashinath, K. et al. Physics-informed machine learning: case studies for weather and climate modelling. *Philos. Trans. R. Soc. A* **379**, 20200093 (2021).
18. Karniadakis, G. E. et al. Physics-informed machine learning. *Nat. Rev. Phys.* **3**, 422–440 (2021).
19. Cuomo, S. et al. Scientific machine learning through physics–informed neural networks: where we are and what's next. *J. Sci. Comput* **92**, 88 (2022).
20. Seo, J. Solving real-world optimization tasks using physics-informed neural computing. *Sci. Rep.* **14**, 202 (2024).
21. Bolton, T. & Zanna, L. Applications of deep learning to ocean data inference and subgrid parameterization. *J. Adv. Model. Earth Syst.* **11**, 376–399 (2019).
22. Jin, P. et al. SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems. *Neural Netw.* **132**, 166–179 (2020).
23. Beucler, T. et al. Enforcing analytic constraints in neural networks emulating physical systems. *Phys. Rev. Lett.* **126**, 098302 (2021).
24. Tong, Y. et al. Symplectic neural networks in Taylor series form for Hamiltonian systems. *J. Comp. Phys.* **437**, 110325 (2021).
25. Dias, T. C. et al. A reaction mechanism for oxygen plasmas. *Plasma Sources Sci. Technol.* **32**, 084003 (2023).
26. Tejero-del-Caz, A. et al. The LisbOn KInetics Boltzmann solver. *Plasma Sources Sci. Technolo.* **28**, 043001 (2019).
27. Prechelt, L. Early stopping—but when? In *Neural Networks: Tricks of the Trade*, *Lecture Notes in Computer Science* Vol. 7700 (eds Orr, G. B. & Müller, K. R.) (Springer, 2012).
28. Glorot, X. & Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proc. of the 13th International Conference on Artificial Intelligence and Statistics*, *PMLR* Vol. 9 (eds Teh, Y. W. & Titterington, M.) 249–256 (JMLR Workshop and Conference Proceedings, 2010).
29. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. arXiv preprint https://arxiv.org/abs/1412.6980 (2017).
30. Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B. & Diehl, M. *CasADi—A Software Framework for Nonlinear Optimization and Optimal Control* (Mathematical Programming Computation, 2018).

Acknowledgements

This work was supported by the Portuguese FCT—Fundação para a Ciência e a Tecnologia, under funding to LARSyS (DOI: 10.54499/LA/P/0083/2020, 10.54499/UIDP/50009/2020, and 10.54499/UIDB/50009/2020), to IPFN (DOI: 10.54499/UIDB/50010/2020, 10.54499/UIDP/50010/2020, and 10.54499/LA/P/0061/2020), and to project PTDC/FIS-PLA/1616/2021 (DOI: 10.54499/PTDC/FISPLA/1616/2021). We would like to thank Dr. Alcides Fonseca for the very helpful comments and suggestions to improve this paper.

Author contributions

Matilde Valente implemented the software, generated the results, and co-drafted the manuscript. Tiago C. Dias contributed to the conceptualization and methodology of the study and generated the dataset of the low-temperature reactive plasma system. Vasco Guerra contributed to the conceptualization and methodology of the study, supervised the work, and co-drafted the manuscript. Rodrigo Ventura proposed the theoretical framework for the projection method and supervised the work. Tiago C. Dias and Rodrigo Ventura reviewed and edited the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42005-025-02329-1.

Correspondence and requests for materials should be addressed to Vasco Guerra.

Peer review information *Communications Physics* thanks the anonymous reviewers for their contribution to the peer review of this work. [A peer review file is available].