DIRAC-INTERCONNECTED NEURAL ELEMENTS: DISCOVERING MODULARITY IN PHYSICAL SYSTEMS WITHOUT REDUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning has shown remarkable success in the data-driven modeling of dynamical systems. Much of its success is attributed not to the flexibility of neural networks but to inductive biases based on physical prior knowledge, such as energy conservation and symplecticity. However, existing methods do not fully exploit the fact that real-world physical systems are interconnections of components. Some methods require the interconnection to be known a priori, while others assume the system to be reducible to an ordinary differential equation (ODE) and learn only the reduced ODE, discarding the algebraic constraints imposed by the interconnection. Here, we propose Dirac-interconnected neural elements (DINEs), a neural network model that represents a physical system as a differential-algebraic equation (DAE), whose algebraic constraints are given by a Dirac structure in kernel representation. With DINEs, we simultaneously identify from data the interconnection among the components as a Dirac structure and learn the characteristics of the components as neural networks. This allows us to keep the learned subsystems in unreduced form and isolate or compose them to make a new system without retraining. Moreover, DINEs can handle partially observable systems. Experimental results demonstrate these capabilities on physical systems beyond the reach of existing methods.

1 INTRODUCTION

Deep learning has been successful in a wide range of domains (Bishop & Bishop, 2024; Noor & Ige, 2025), including the data-driven modeling of dynamical systems (Wang et al., 2023; Legaard et al., 2023). Many approaches approximate the dynamics directly from observed trajectories, as in neural ordinary differential equations (Neural ODEs) (Chen et al., 2018). Since such black-box models often exhibit physically inconsistent behavior in long-term predictions, physical prior knowledge has been embedded into models as an inductive bias. In particular, Hamiltonian, constrained Hamiltonian, and port-Hamiltonian neural networks are inspired by energy-based modeling, where an energy function is defined on the state space, and a single vector field is derived from it, ensuring the energetic behavior of the system by construction (Greydanus et al., 2019; Finzi et al., 2020; Matsubara et al., 2020; Desai et al., 2021; Moradi et al., 2026).

However, real-world physical systems are interconnections of components, such as masses, springs, and dampers in mechanical systems, and inductors, capacitors, and resistors in electrical circuits. Their power exchange has often been described by a graphical notation, bond graphs (Paynter, 1961; Karnopp et al., 2012; Tabatabaian, 2021). The model becomes interpretable and its components become reusable if it identifies from observed data this internal structure, namely the characteristics of the components (i.e., their constitutive relations) and the interconnection among them. Beyond prediction, such models also support the identification of physical parameters, diagnosis, and design. Because the interconnection among the components gives rise to algebraic constraints, interconnected systems are in general described by differential-algebraic equations (DAEs) (Duindam et al., 2009; van der Schaft & Jeltsema, 2014). Neural port-Hamiltonian DAEs (N-pHDAEs) (Neary et al., 2025) and port-Hamiltonian DAE (pH-DAE) identification (Hagelaars et al., 2026) approximate interconnected systems with the interconnection known and fixed in advance. Poisson-Dirac neural networks (PoDiNNs) (Khosrovian et al., 2025) identify the interconnection from observed data, but

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Comparison with related methods.

Model	Ensuring Energetic Behavior	Identifying Interconnection	Learning Constraints	Learning from Partial Observation
Neural ODEs (Chen et al., 2018)	–	–	–	–
HNNs, LNNs, SymODENs, pHNNs [†]	✓	–	–	–
OE-pHNNs (Moradi et al., 2026)	✓	–	–	✓
CHNNs, CLNNs (Finzi et al., 2020)	✓	–	given [§]	–
Compositional pHNNs (Neary & Topcu, 2023)	✓	given	–	–
N-pHDAEs (Neary et al., 2025)	✓	given	given	–
pH-DAE identification (Hagelaars et al., 2026)	✓	given	given	✓
Latent ODEs, CT-SUBNET [‡]	–	–	if reducible	✓
PoDiNNs (Khosrovian et al., 2025)	✓	✓	if reducible	–
DINEs (ours)	✓	✓	✓	✓

[†] Greydanus et al. (2019); Cranmer et al. (2020); Zhong et al. (2020b;a); Desai et al. (2021). [‡] Rubanova et al. (2019); Beintema et al. (2023).
[§] Available only for holonomic constraints.

only in a reduced form. They assume that the system is reducible to an ODE and that all component states are observable, thereby discarding the algebraic constraints and excluding partially observable systems. Table 1 summarizes the comparison: no existing model both learns the interconnection and keeps its algebraic constraints.

Given the above, we propose Dirac-interconnected neural elements (DINEs), a neural network model that represents a physical system as a DAE whose algebraic constraints are given by a Dirac structure in kernel representation (see Figure 1) (Duindam et al., 2009; van der Schaft & Jeltsema, 2014). Our contributions are as follows: (i) By minimizing the DAE residual, DINEs simultaneously learn the interconnection as a Dirac structure and the constitutive relations of the components as neural networks. (ii) Since the system is not reduced to an ODE, DINEs can isolate or compose the learned subsystems without retraining. (iii) DINEs also handle partially observable systems, in which some component states are unobserved. (iv) Experimental results demonstrate these capabilities on physical systems beyond the reach of existing methods.

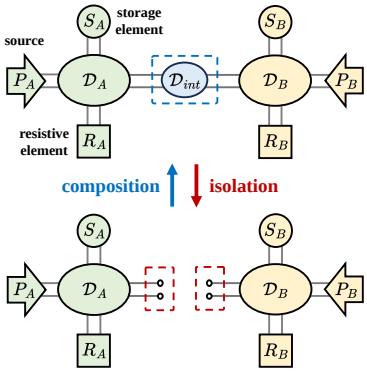


Figure 1: Conceptual diagram of DINEs.

2 Background

2.1 Port-based modeling and Dirac structures

Port-based modeling. Port-based modeling represents a physical system as an interconnection of elements, each of which idealizes a basic physical behavior (Duindam et al., 2009; van der Schaft & Jeltsema, 2014). Each element has *ports*, through which it exchanges energy with the others. The exchange is described by a pair of *port variables*, a *flow* f (e.g., velocity or current) and an *effort* e (e.g., force or voltage), whose duality product gives the power flowing out of the element through the port; their physical nature depends on the domain. A C-type storage element (C; e.g., a spring or a capacitor) has a state q and stores energy $H(q)$ with $f = -\dot{q}$ and $e = \partial H/\partial q$, and an I-type storage element (I; e.g., a mass or an inductor) has a state p and stores energy $H(p)$ with $e = -\dot{p}$ and $f = \partial H/\partial p$; in both cases $\langle e \mid f \rangle = -\dot{H}$ holds, and these equations are called the energy-storage relations. A resistive element (R; e.g., a damper or a resistor) is defined by a resistive relation $\mathcal{R}(f, e) = 0$ that satisfies $ef \leq 0$ and thus dissipates energy. A source (Sf or Se) prescribes a flow or an effort as a function of time. The interconnection is a multiport element that merely routes energy, that is, it is *power continuous* with zero net power at all times: in bond-graph theory, it is called the junction structure and is composed of 0-junctions and 1-junctions, which generalize Kirchhoff's current and voltage laws, transformers (TF), and gyrators (GY) (Karnopp et al., 2012). See Appendix A for details.

Dirac structure. The interconnection of port-based modeling is formalized as a Dirac structure (van der Schaft & Maschke, 1995; Dalsmo & van der Schaft, 1998), a geometric object in-

troduced by Courant (1990) and Dorfman (1993). Let $\mathcal{F}$ be an n-dimensional linear space, whose element f is the vector of the flows at the ports. Let $\mathcal{E} := \mathcal{F}^*$ be its dual space, whose element e is the vector of the efforts at the ports. For $(f, e) \in \mathcal{F} \times \mathcal{E}$, the total power is the duality product $\langle e \mid f \rangle$, which is equal to $e^\top f$ when f and e are represented as column vectors.

Definition 1 (Dirac structure; Courant, 1990; Duindam et al., 2009; van der Schaft & Jeltsema, 2014)**.** A Dirac structure on $\mathcal{F} \times \mathcal{E}$ is a subspace $\mathcal{D} \subset \mathcal{F} \times \mathcal{E}$ such that (i) $\langle e \mid f \rangle = 0$ for all $(f, e) \in \mathcal{D}$, and (ii) $\dim \mathcal{D} = \dim \mathcal{F}$.

Every Dirac structure $\mathcal{D} \subset \mathcal{F} \times \mathcal{E}$ can be represented in the kernel representation (Dalsmo & van der Schaft, 1998; van der Schaft & Jeltsema, 2014, Sec. 5.1). With linear coordinates for $\mathcal{F}$ and the dual basis for $\mathcal{E}$, it is given by

$$\mathcal{D} = \{(f, e) \in \mathcal{F} \times \mathcal{E} \mid Ff + Ee = 0\}, \tag{1}$$

where F and E are $n \times n$ matrices satisfying

$$EF^\top + FE^\top = 0 \text{ and } \operatorname{rank}[\, F \mid E \,] = n. \tag{2}$$

Conversely, any pair of matrices satisfying these two conditions defines a Dirac structure by Eq. (1).

Dynamics on a Dirac structure. Each of the 0-junction, the 1-junction, the transformer, and the gyrator defines a Dirac structure, and the composition of Dirac structures is again a Dirac structure. Hence, the junction structure is a Dirac structure (Duindam et al., 2009; van der Schaft & Jeltsema, 2014), which we denote by $\mathcal{D}$. Each row of $Ff + Ee = 0$ is a linear constraint that the interconnection imposes on the port variables. The entries of F and E represent the interconnection, including the ratios of the transformers and gyrators.

The dynamics is defined by the condition that the port variables belong to $\mathcal{D}$ at all times (Duindam et al., 2009; van der Schaft & Jeltsema, 2014):

$$(f(t), e(t)) \in \mathcal{D} \text{ for all } t, \quad \text{that is,} \quad Ff(t) + Ee(t) = 0 \text{ for all } t. \tag{3}$$

Here, $f = (f_S, f_R, f_P)$ and $e = (e_S, e_R, e_P)$ are the port variables of the storage elements, the resistive elements, and the sources, called the storage, resistive, and source ports, respectively. The numbers of these ports are n_S, n_R, and n_P, respectively, with $n = n_S + n_R + n_P$. We partition F and E in Eq. (1) in the same way as f and e, that is, $F = [\, F_S \mid F_R \mid F_P \,]$ and $E = [\, E_S \mid E_R \mid E_P \,]$. The port variables are oriented out of the elements and thus into the Dirac structure, so that condition (i) of Definition 1 states that the net power flowing into the interconnection is zero.

Let z_i be the state of the i-th storage element, which is q_i for C-type and p_i for I-type, and let $z = (z_1, \ldots, z_{n_S}) \in \mathbb{R}^{n_S}$; only the storage elements have states, so z is the state of the entire system. The Hamiltonian $H : \mathbb{R}^{n_S} \to \mathbb{R}$ is the total stored energy, that is, the sum of the energies of all the storage elements, $H(z) = \sum_{i=1}^{n_S} H_i(z_i)$, where H_i is the energy of the i-th storage element. The resistive ports satisfy a resistive relation between f_R and e_R such that $\langle e_R \mid f_R \rangle \leq 0$, that is, the resistive elements only absorb power. For each source port, the prescribed variable (the flow of an Sf element or the effort of an Se element) is an external input, and its conjugate variable is determined by the system as an output; we denote the vectors of the inputs and the outputs by u and y, respectively.

The constitutive relations are thus given by the Hamiltonian H and the resistive relation $\mathcal{R}$. Substituting the constitutive relations into Eq. (3) yields, in general, a differential-algebraic equation (DAE) for the state z and the remaining port variables with the input u.

Reduction to an ODE. Under certain conditions, the DAE can be reduced to an ODE in z. Exchanging the flow and the effort of every I-type port, every Sf port, and some resistive ports preserves the conditions of Eq. (2) (van der Schaft & Jeltsema, 2014, Sec. 5.3); we denote the resulting variables with $\tilde{\cdot}$. Then, the flows of all the storage ports are the time derivatives of the states up to sign, $\dot{z} = -\tilde{f}_S$. If $\tilde{F}$ is invertible, Eq. (1) is equivalent to $\tilde{f} = B\tilde{e}$ with the skew-symmetric matrix $B := -\tilde{F}^{-1}\tilde{E}$, which is called the input-output representation of the Dirac structure. This form is an ODE in z once the remaining port variables can be eliminated through the constitutive relations, and is what PoDiNNs employ (Khosrovian et al., 2025).

Under further assumptions, eliminating the resistive port variables yields the input-state-output form of port-Hamiltonian systems (Maschke & van der Schaft, 1992; van der Schaft & Maschke, 1995;

van der Schaft & Jeltsema, 2014),

$$\dot{z} = \big(J - R\big)\,\nabla H(z) + G\,u, \qquad y = G^\top \nabla H(z), \qquad R := G_R \hat{R} G_R^\top, \tag{4}$$

where $u = \tilde{e}_P$, $y = \tilde{f}_P$, $\hat{R}$ is the coefficient of the linear resistive relations, and a skew-symmetric matrix $J = -J^\top$, the resistive connection G_R, and the gain G are blocks of B up to sign. This form is what many existing methods employ (as discussed below).

We refer to these transformations and eliminations of variables as *reduction*, which has two costs. (1) It mixes the interconnection with the elements, since R combines the resistive relations $\hat{R}$ with the blocks G_R that connect them and cannot be uniquely decomposed into the two. (2) It assumes that an ODE exists, but it does not in general. For example, Eq. (4) cannot describe nonlinear frictions, and both forms are not applicable to the balanced force of several nonlinear springs, and a capacitor connected with a voltage source in parallel (see also Appendix B).

2.2 Related work

Learning the dynamics as a single ODE. Neural ODEs (Chen et al., 2018) approximate a general ODE of the form $\dot{z} = \varphi_\theta(z)$ with a neural network, where no mechanism prevents the learned model from creating or destroying energy in long-term predictions. Hamiltonian neural networks (HNNs) (Greydanus et al., 2019) and their extensions, Symplectic ODE-Nets (SymODENs) (Zhong et al., 2020b), neural symplectic forms (Chen et al., 2021), Dissipative SymODENs (Zhong et al., 2020a), port-Hamiltonian neural networks (pHNNs) (Desai et al., 2021), pseudo-Hamiltonian neural networks (Eidnes et al., 2023), and output-error pHNNs (OE-pHNNs) (Moradi et al., 2026), are special cases of Eq. (4) that learn H_θ with J, R, and G fixed or learned. Lagrangian neural networks (LNNs) and deep Lagrangian networks (Lutter et al., 2019; Cranmer et al., 2020), constrained HNNs and LNNs (CHNNs and CLNNs) (Finzi et al., 2020), which add holonomic constraints given in advance, are considered also as the special cases with designed J. See Appendix B for the correspondence. In the methods, H_θ is a single function of the whole state z, or at most separated into the kinetic and potential energies but not into the energies of the individual elements. The resistive elements appear only in R, so that neither the elements nor their interconnection can be identified from the learned model. The constraints imposed by the interconnection are either eliminated manually before training by the choice of z or given in advance.

Learning the components under a given interconnection. Several methods learn the subsystems or elements separately, while the interconnection is specified before training, or at most fitted as a composition between ODE subsystems. Compositional pHNNs (Neary & Topcu, 2023) learn each subsystem in the form of Eq. (4) for a given J_i and compose them through an additional skew-symmetric matrix J_{int}, which is given or learned from data of the composite system as a matrix or a neural network; the composite system is again an ODE of the same form with $H = \sum_i H_{\theta,i}$. N-pHDAEs (Neary et al., 2025) employ the DAE of an electrical network in the descriptor form (Beattie et al., 2018; Mehrmann & Unger, 2023) with the circuit topology known and all of its variables observable, that is, F and E of Eq. (3) are given by Kirchhoff's laws and only the constitutive relations are learned; separately trained N-pHDAEs can be composed without additional training if the additional interconnection is given as a matrix. Hagelaars et al. (2026) identify a linear pH-DAE of the same family from input-output data with unobserved states.

Identifying the interconnection from data. The closest to our work is PoDiNNs (Khosrovian et al., 2025), which employ the Dirac structure in the input-output representation $\tilde{f} = B_\theta \tilde{e}$. Given the list of elements, they learn B_θ as a constant matrix, the resistive relations as explicit maps $\tilde{e}_R = \hat{\mathcal{R}}_\theta(\tilde{f}_R)$, and H_θ under the assumption that all states are observed. However, after reduction, the interconnection represented by B_θ depends on the elements' states or parameters in general. Khosrovian et al. (2025) themselves acknowledge in their Appendix B.2 that, to represent the equal velocities of two rigidly coupled masses, B_θ depends on the masses and is no longer determined by the interconnection alone. Also, PoDiNNs are inapplicable to some situations, such as the balanced force of several nonlinear springs (see Task 1 in Section 4).

Learning with unobserved states. When only an observation y of the system is available, the state has to be inferred along with the model. The continuous-time subspace encoder network (CT-SUBNET) (Beintema et al., 2021; 2023) learns a black-box state-space model, $\dot{z} = \varphi_\theta(z, u)$ and $y = h_\theta(z)$ with a predetermined state dimension n_z, where an encoder estimates the state at the

start of each window of the time series from the past outputs and inputs and the prediction error over the window is minimized; Latent ODEs (Rubanova et al., 2019) are similar with a probabilistic encoder and decoder. OE-pHNNs and the pH-DAE identification replace the black-box model by Eq. (4) and by the linear DAE, respectively, but limit the observation y to the output conjugate to the input, so that other quantities, such as the currents of the resistors inside a circuit, are not available as observations. In all of these methods, the unobserved state is inferred by the encoder only at the start of each window and is obtained after integrating the model forward in time.

3 DIRAC-INTERCONNECTED NEURAL ELEMENTS

Learnable elements and interconnection. Here, we propose DINEs as a learnable model of Eq. (3), which is defined by constitutive relations and a Dirac structure. For each storage element, we learn a differentiable function $H_{\theta,i}$ of its state z_i by a neural network, and thus the total energy is $H_\theta(z) = \sum_{i=1}^{n_S} H_{\theta,i}(z_i)$. For each resistive element, we learn the resistive relation $\mathcal{R}_{\theta,j}(f_{R,j}, e_{R,j}) = 0$ by a neural network as an explicit map $e_{R,j} = \hat{\mathcal{R}}_{\theta,j}(f_{R,j})$ from the flow to the effort. Dissipativity $\langle e_R \mid f_R \rangle \leq 0$ is not imposed for simplicity, while it is possible with a sign constraint on $\hat{\mathcal{R}}_{\theta,j}$. We suppose sources are prescribed as functions of time.

For the Dirac structure, the matrices F, E should satisfy the two conditions of Eq. (2) for every value of the parameters. To avoid constrained optimization, we suppose that the interconnection is separable and constant, and that the number n_f of flow constraints (and thus the number $n_e = n - n_f$ of effort constraints) is given in advance. A Dirac structure is *separable* if the rows of Eq. (1) can be chosen so that each involves only flows or only efforts; it holds if the system has no gyrators, since junctions and transformers are separable and so is their composition. The interconnection is constant if the ratios of its transformers do not depend on the state (van der Schaft & Jeltsema, 2014, Sec. 2).

Theorem 1 (Exact parameterization of separable Dirac structures)**.** Let n_f and n_e be nonnegative integers with $n_f + n_e = n$, and let $W_0 \in \mathbb{R}^{n \times n}$ be any fixed orthogonal matrix. For $\Omega \in \mathbb{R}^{n_f \times n_e}$, define $W(\Omega) = \exp\big(S(\Omega)\big) W_0$ with $S(\Omega) = \left[\begin{smallmatrix} 0 & \Omega \\ -\Omega^\top & 0 \end{smallmatrix}\right]$, let $W_f \in \mathbb{R}^{n_f \times n}$ and $W_e \in \mathbb{R}^{n_e \times n}$ consist of the first n_f and last n_e rows of $W(\Omega)$, respectively, and set $F = \left[\begin{smallmatrix} W_f \\ 0 \end{smallmatrix}\right]$ and $E = \left[\begin{smallmatrix} 0 \\ W_e \end{smallmatrix}\right]$. Denote the resulting kernel by $\mathcal{D}(\Omega) := \{(f, e) \in \mathcal{F} \times \mathcal{E} \mid Ff + Ee = 0\}$. Then, (i) For every Ω, (F, E) satisfies the conditions of Eq. (2), and $\mathcal{D}(\Omega)$ is a separable Dirac structure with exactly n_f independent flow constraints. (ii) For every separable Dirac structure $\mathcal{D}_\star$ on $\mathcal{F} \times \mathcal{E}$ with exactly n_f independent flow constraints, there exists $\Omega \in \mathbb{R}^{n_f \times n_e}$ such that $\mathcal{D}_\star = \mathcal{D}(\Omega)$ for the same fixed W_0.

Proposition 1 (Energy balance)**.** For every Ω, θ, input u, and solution $(z(t), f(t), e(t))$ of the DAE of DINEs,

$$\tfrac{\mathrm{d}}{\mathrm{d}t} H_\theta(z(t)) = \langle e_R(t) \mid f_R(t) \rangle + \langle e_P(t) \mid f_P(t) \rangle. \tag{5}$$

This means that the total energy is changed outside the storage elements. If, in addition, a sign constraint $f_{R,j} \hat{\mathcal{R}}_{\theta,j}(f_{R,j}) \leq 0$ is imposed on resistive relations, $\tfrac{\mathrm{d}}{\mathrm{d}t} H_\theta(z(t)) \leq \langle e_P(t) \mid f_P(t) \rangle$ holds, which means that the total energy is dissipated.

See Appendix C for the proofs. This parameterization requires n_f as a hyperparameter, the number of independent constraints among the flows. For a connected electrical network with one element per branch, it is the number of nodes minus one, since the row space of W_f is spanned by Kirchhoff's current law at the nodes (van der Schaft & Jeltsema, 2014, Secs. 2.2.2 and 12.1). A constant W does not require the constraints to be linear in the state: the force balance of several nonlinear springs at a massless junction is linear in the efforts and is inside this setting.

Learning and prediction. Given a set of the trajectories of inputs $u(t)$ and observations $y(t_k)$ at sample times t_k for $k = 0, 1, \ldots$, we train DINEs to fit them. The observations y are a given subset of the port variables and states; unlike Eq. (4), they need not be the variables conjugate to the inputs. We prepare a neural network χ_η called trajectory network with parameters η that takes the time t and the index of the trajectories and returns, as continuous-time functions of t, the state $z(t)$ of every storage element, the flow f_R of every resistive element, and the conjugate variable of every source port, whether observed or not. The remaining port variables follow from the constitutive relations. For storage elements, $\dot{z}_i$ and $\partial H_i / \partial z_i$ are computed by the automatic differentiation, and the effort

of a resistive element follows from $e_{R,j} = \hat{\mathcal{R}}_{\theta,j}(f_{R,j})$. The observations $\hat{y}(t)$ are reconstructed from these states and port variables. We suppose that the number of elements of each type (C, I, R, Sf, Se) is given, which fixes the numbers of ports of each type and hence n.

We train this neural network by minimizing the sum of the data-assimilation loss at the sample times t_k and the DAE residual of Eq. (3) at collocation points s_j:

$$\lambda \mathbb{E}_k \left\| \hat{y}(t_k) - y(t_k) \right\|^2 + \mathbb{E}_j \left\| F f(s_j) + E e(s_j) \right\|^2, \tag{6}$$

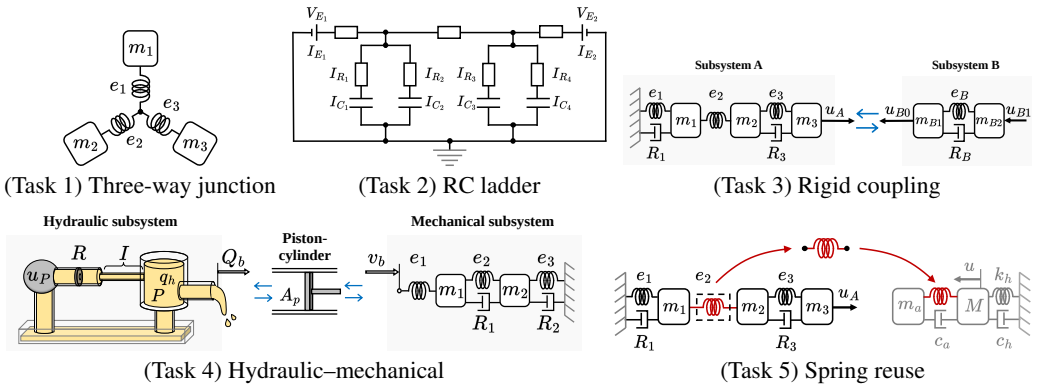
for a weight λ. Each term is averaged over a minibatch of trajectories, times t_k, and collocation points s_j, which are randomly sampled at every update, and the parameters θ of H_θ and $\mathcal{R}_\theta$, the matrix Ω, and the parameters η of χ_η are optimized jointly by gradient descent. χ_η takes the role of a numerical integrator during training, solving the learned DAE by minimizing its residual, and is discarded after training. This approach is similar to physics-informed neural networks (PINNs) (Raissi et al., 2019), except that the equation itself is learned. For prediction, initial values are estimated from the observations available at the start of the prediction so that they are consistent with the algebraic rows of the learned DAE, and the learned DAE in Eq. (3) is integrated by the three-stage Radau IIA method of order five (Hairer & Wanner, 1996) with Newton's method (Nocedal & Wright, 2006). See Appendix D for the details of implementation.

Isolation and composition. The learned DINEs contain the constitutive relations of the elements and the Dirac structure (F, E) separately, so subsystems can be isolated or composed by operations on their ports, without eliminating variables, deriving reduced equations, or retraining, up to scale calibrations. A subpart of the system can be isolated by removing the constitutive relations of the elements outside it, while keeping their ports in (F, E) as boundary ports, i.e., ports that are not connected to any element. When considering boundary ports to be connected to non-prescribed sources (Sf or Se), the isolated system is again DINEs. In particular, when the subpart is a single element, its own port becomes the boundary port, and the element is isolated with its learned constitutive relation alone and can be reused in another system. Two isolated systems S_A, S_B can be composed by adding a boundary port to each of their Dirac structures and connecting these ports to each other. This process stacks their matrices (F_A, E_A) and (F_B, E_B) block-diagonally and adds two rows that impose $f_A + f_B = 0$ and $e_A - e_B = 0$ on the connected port variables, with signs depending on the orientation of the ports. These rows correspond to $\mathcal{D}_{int}$ in Fig. 1. The composed DAE has $n_A + n_B + 2$ constraints in the $n_A + n_B$ ports, which can be integrated directly. Alternatively, the connected ports are eliminated algorithmically before integration, after which the composed DAE has again a separable and constant Dirac structure on the remaining ports and hence a DINE (van der Schaft & Jeltsema, 2014, Sec. 6).

Discussions for comparisons and limitations. Thanks to the kernel representation and the trajectory network χ_η, DINEs do not require reduction and admit partial observation. In contrast, PoDiNNs require the input-output representation to exist, possibly after reduction, and every state to be observed. OE-pHNNs and the pH-DAE identification suppose the observation to be the output conjugate to the input. Our parameterization of (F, E) requires n_f, which has to be swept in practice because the interconnection is unknown. Nevertheless, a wrong model is not identified silently, because a misspecification of n_f leads to a catastrophic failure; this is expected from the implication of Theorem 1 that a misspecified n_f excludes the true Dirac structure from the model family.

Gyrators, which typically bridge physical domains as in DC motors and hydraulic motors, are outside our setting. Constraints whose coefficients depend on the state, which lead to modulated Dirac structures as for a multi-link robot in Cartesian coordinates, and a kinetic energy that depends on the momenta and the positions together, which violates the form $H_\theta(z) = \sum_i H_{\theta,i}(z_i)$ as for the same robot in generalized coordinates, are also outside our setting.

When reading the learned interconnection, it is beneficial to transform W_f and W_e to reduced row echelon forms, so that each row states one constraint among a minimal set of ports. Depending on the interconnection and on which elements are observed, the scales of port variables, constitutive relations, and states are not unique. This gauge freedom is removed in most cases by normalizing the entries of the rows corresponding to junctions (but not to transformers) to 0 or ± 1; otherwise, a few additional measurements fix the remaining scales, before or after isolation and composition.



(Task 1) Three-way junction (Task 2) RC ladder (Task 3) Rigid coupling

(Task 4) Hydraulic–mechanical (Task 5) Spring reuse

Figure 2: Diagrams of the systems of Tasks 1–5.

4 Experiments and Results

4.1 Experimental Settings

Tasks. We evaluate DINEs on the five tasks shown in Figure 2. Appendix E gives their detailed settings, equations, parameters, and data generation. In Task 1, a *three-way junction*, we examine whether the constraints are learned. The force balance at the junction leads to an algebraic constraint, which PoDiNNs cannot learn. In Task 2, an *RC ladder*, we examine whether the system can be learned from observed currents when the states (that is, capacitor charges) are unobserved. Its node voltages are determined by nonlinear algebraic relations that arise from the constitutive relations of the resistors and Kirchhoff's laws. In Tasks 3 and 4, we examine the composition of learned subsystems, and in Task 5 the isolation. Each subsystem has one boundary port, which is connected to a source during learning, and then composed to another. In Task 3, a *rigid coupling*, the sources are external forces, applied to two masses. In Task 4, a *hydraulic–mechanical coupling*, the boundary inputs are the flow leaving the reservoir and the velocity imposed at the free end of the first spring. Because of the multi-domain nature of the system, this composition is represented by a transformer, whose constant ratio (that is, the piston area) is unknown and identified from a few measurements of the composed system. In Task 5, a *spring reuse*, we examine whether the spring learned in a subsystem (i.e., a nonlinear spring) of Task 3 can be isolated and reused in a new mechanical system without retraining.

Baselines and prior knowledge. None of the existing methods introduced in Section 2.2 is applicable to Tasks 1–5 as it is, except Neural ODEs to Task 1. On Task 1, the other methods reduce to Neural ODEs, require the constraints to be given in advance, or derive the state-dependent coefficient matrices. We nevertheless evaluate PoDiNNs on Task 1 to confirm this limitation. For Task 2, we extend the observation map of CT-SUBNET from $y = h_\theta(z)$ to $y = h_\theta(z, u)$ because the observed currents y depend directly on the source voltages u. CT-SUBNET is given its state dimension n_z, which is swept. Recall that OE-pHNNs and the pH-DAE identification assume the observation to be conjugate to the input, and N-pHDAEs require the circuit topology to be known and all of its variables observable. For Tasks 3 and 4, we construct Coupled Neural ODEs as a baseline without physical structure, where a Neural ODE learns each subsystem and a separate network learns the variable conjugate to the boundary input, and then these networks are composed. For Task 4, the piston area is identified from a few observations of the composed system. For DINEs, we assume that the I-type elements, the capacitors of Task 2, and the reservoir of Task 4 are linear with their coefficients learned. For comparison, we also evaluate DINEs (soft), which replace the parameterization of Theorem 1 by penalty terms of the conditions of Eq. (2) in the loss (see Appendix D.4 for details). We summarize the prior knowledge and assumptions available for each model in Appendix F.

Training and prediction. DINEs are trained as stated above, whereas Neural ODEs, PoDiNNs, and the subsystem models of Coupled Neural ODEs are trained by one-step prediction with the classical fourth-order Runge–Kutta method, and CT-SUBNET by the multi-step prediction error over windows whose initial state is given by its encoder. The learned DAEs are integrated by the Radau IIA

Table 2: Experimental results.

Model	(Task 1) Three-way junction MSE↓	VPT↑
Neural ODEs	$(2.871_{\pm 1.014}) \times 10^4$	$0.062_{\pm 0.015}$
PoDiNNs*	$(6.129_{\pm 0.089}) \times 10^4$	$0.000_{\pm 0.000}$
DINEs (soft)†	(not converged in 3 of 5 seeds)	
DINEs (ours)	$\mathbf{3.319}_{\pm 1.221}$	$\mathbf{0.860}_{\pm 0.043}$
DINEs ($n_f = 1$)‡	(not converged in all seeds)	
DINEs ($n_f = 3$)‡	$(4.975_{\pm 0.062}) \times 10^4$	$0.000_{\pm 0.000}$
DINEs ($n_f = 4$)‡	$(6.697_{\pm 0.912}) \times 10^4$	$0.000_{\pm 0.000}$
DINEs ($n_f = 5$)‡	$(6.347_{\pm 0.058}) \times 10^4$	$0.000_{\pm 0.000}$
	$\times 10^{-5}$	$\varepsilon = 10^{-4}$

Model	(Task 2) RC ladder MSE↓	VPT↑
CT-SUBNET ($n_z = 4$)	$11.248_{\pm 2.962}$	$0.039_{\pm 0.011}$
CT-SUBNET ($n_z = 8$)	$3.300_{\pm 0.899}$	$0.096_{\pm 0.084}$
CT-SUBNET ($n_z = 12$)	$3.267_{\pm 0.869}$	$0.157_{\pm 0.023}$
CT-SUBNET ($n_z = 16$)	$3.683_{\pm 1.077}$	$0.205_{\pm 0.055}$
CT-SUBNET ($n_z = 20$)	$3.864_{\pm 1.846}$	$0.241_{\pm 0.018}$
DINEs (soft)	(not converged in all seeds)	
DINEs (ours)	$\mathbf{1.003}_{\pm 0.113}$	$\mathbf{0.711}_{\pm 0.002}$
	$\times 10^{-5}$	$\varepsilon = 10^{-4}$

Model	(Task 3) Rigid coupling MSE↓	VPT↑	(Task 4) Hydraulic–mechanical MSE↓	VPT↑	(Task 5) Spring reuse MSE↓	VPT↑
Coupled Neural ODEs	$189.508_{\pm 101.995}$	$0.002_{\pm 0.001}$	$73.893_{\pm 10.636}$	$0.004_{\pm 0.001}$		
DINEs (ours)	$\mathbf{1.206}_{\pm 0.492}$	$\mathbf{0.970}_{\pm 0.068}$	$\mathbf{1.068}_{\pm 0.837}$	$\mathbf{0.831}_{\pm 0.321}$	$6.963_{\pm 4.625}$	$0.968_{\pm 0.050}$
	$\times 10^{-7}$	$\varepsilon = 10^{-6}$	$\times 10^{-7}$	$\varepsilon = 10^{-6}$	$\times 10^{-8}$	$\varepsilon = 10^{-6}$

Means ± standard deviations over five trials. The scale of the MSE and the threshold ε of the VPT are given in the last row of each block.
* For reference only, since PoDiNNs are theoretically inapplicable to Task 1. † 3 of 5 seeds did not converge. ‡ n_f is misspecified.

method and the ODE models by an adaptive Runge–Kutta method. On Tasks 3 and 4, the learned subsystems are composed without retraining: the scale of the boundary variable of each subsystem of DINEs is calibrated once from the learned interconnection or from boundary measurements in its training data (see Appendix D.3), the piston area of Task 4 is identified from a few states of the composed system, and the composed system is integrated as a DAE whose boundary variables are unknowns.

For prediction, DINEs estimate the initial state to be consistent with the observation at the first sample (or the history of the inputs and observations on Task 2) under the learned algebraic constraints. Neural ODEs and PoDiNNs use the observed initial state. CT-SUBNET estimates the initial state from the history using its encoder. Coupled Neural ODEs do the same as DINEs.

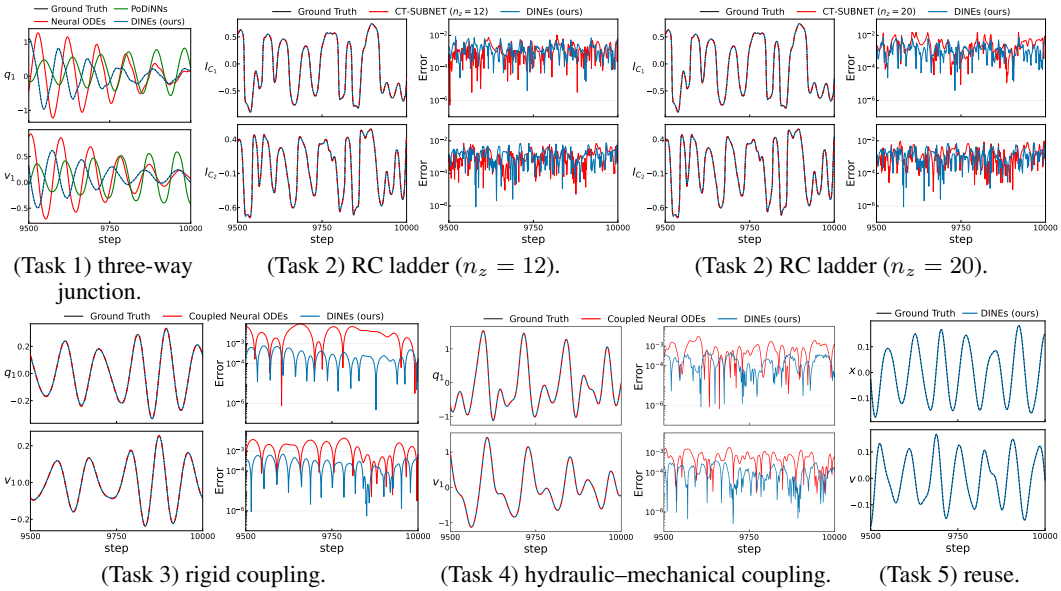
Evaluation metrics. Each model predicts the trajectory of K steps of every test trajectory, and the squared error between the predicted and true observations, averaged over the d observed channels, $e_k = \frac{1}{d}\|\hat{y}(t_k) - y(t_k)\|^2$, is computed at every sample $k = 0, \ldots, K$. The MSE is its average over k. The valid prediction time (VPT) (Vlachas et al., 2020) is the duration until the step at which the error first exceeds a threshold ε, normalized by the length of the test trajectory. Both metrics are averaged over the ten test trajectories for each seed, and we report the mean and the standard deviation over the five seeds. See Appendix E.1 for the detailed definitions.

4.2 Results

Results and visualizations. Table 2 summarizes the numerical performances, and Figure 3 visualizes example predictions. DINEs achieved the lowest MSEs and the highest VPTs in all Tasks 1–4, and also showed high prediction accuracy on Task 5.

Neural ODEs can represent the system in Task 1 as a reduced ODE in principle, but their prediction errors grew quickly. This is because the training trajectories lie on the manifold defined by the constraints of the interconnection, and a prediction becomes out of distribution once it leaves the manifold. DINEs learn the interconnection and thus suppress such errors. PoDiNNs cannot learn this kind of constraint and also perform poorly. A variant with a state-dependent matrix $B_\theta(q)$ is not evaluated here as not examined in the original paper. On Task 2, we compared CT-SUBNET with the latent state dimension n_z ranging from the true number 4 to a redundant 20. The MSE improved from $n_z = 4$ to $n_z = 8$ and deteriorated slightly beyond $n_z = 12$, whereas the VPT kept improving up to $n_z = 20$ with decreasing gains. None of these settings reaches the level of DINEs.

On Tasks 3 and 4, the predictions of Coupled Neural ODEs look reasonable in Figure 3, but DINEs achieved MSEs nearly two orders of magnitude smaller and much longer VPTs. Coupled Neural



(Task 1) three-way junction. (Task 2) RC ladder ($n_z = 12$). (Task 2) RC ladder ($n_z = 20$).

(Task 3) rigid coupling. (Task 4) hydraulic–mechanical coupling. (Task 5) reuse.

Figure 3: Visualizations of example predictions. Error denotes the absolute error.

ODEs can be composed only when they learn the relation between the boundary port variables given as inputs and their conjugate variables as outputs. In contrast, DINEs have an advantage that the Dirac structure provides such relation up to the gauge freedom.

Task 5 has no baseline because DINEs are the only method applicable to it. DINEs achieved the MSE and VPT comparable to (or better than) those on the other tasks. Although this is partly because every component other than the spring reused from Task 3 is known in this setting, the result shows that the isolation and the composition were performed with high accuracy.

Necessity of hard constraints. For DINEs (soft), which impose only soft constraints, the Newton method for initialization or prediction did not converge in most seeds on Tasks 1 and 2. This is presumably because a solution of the DAE may not exist when the constraints are only soft. This result shows the necessity of the parameterization derived in Theorem 1.

Impact of the number of constraints. DINEs require the number n_f of flow constraints, which is unknown in general. To investigate the impact of its misspecification, we examine $n_f = 1, \ldots, 5$ on Task 1. With $n_f = 1$, the Newton method for prediction did not converge. Note that this problem does not arise in training, where the trajectory network is used. With $n_f = 3, 4, 5$, the Newton method converged, but the MSEs and VPTs were catastrophic. Since the number of effort constraints is $n_e = n - n_f$, n_f determines not the total number of constraints but their allocation between flows and efforts. Because the MSE and VPT differ so clearly, n_f can be tuned easily by holding out a small validation set from the training data, and a wrong model is not identified silently.

Learned interconnection. We confirmed that the matrices F and E learned the interconnection correctly in Tasks 1–4. Note that, since the Dirac structure depends only on the row spaces of F and E, the learned rows are not unique, and a readout procedure is needed for interpretation. See Appendix F for examples of the readout of the learned interconnection.

5 Conclusion

We proposed DINEs, which learn a physical system as a DAE whose algebraic constraints are given by a separable Dirac structure parameterized exactly by an orthogonal matrix, without reducing the system to an ODE. Our experiments showed that DINEs learn constraints and unobserved states that existing methods cannot, and that learned subsystems and elements are composed and reused across physical domains without retraining. The automatic determination of n_f and the extension to non-separable Dirac structures are left for future work.

AI USE STATEMENT

We used large language models to translate, edit, and proofread the manuscript text and to check LaTeX formatting and bibliographic details. We also used these models to survey related work. All AI-assisted text was reviewed and revised by the authors, and all derivations were checked manually by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work concerns the data-driven modeling of physical systems and uses only simulated data generated from known equations of motion. It involves no human subjects, personal data, or data releases with privacy implications, and we do not foresee direct harmful applications.

REPRODUCIBILITY STATEMENT

The construction of the interconnection matrices F and E and the training loss, Eq. (6), are specified in Section 3, and the derivation of the reduced form used for comparison is given in Appendix B. Appendix D provides the implementation of DINEs and its ablated variants, and Appendix E shows the equations of motion, parameters, observed variables, sampling settings, and optimizer settings of Tasks 1–5, together with the architectures and training of the baselines. Source code and the data-generation scripts will be made public after refactoring.

REFERENCES

Christopher Beattie, Volker Mehrmann, Hongguo Xu, and Hans Zwart. Linear port-Hamiltonian descriptor systems. *Mathematics of Control, Signals, and Systems*, 30(4):17, 2018. doi: 10.1007/s00498-018-0223-3.

Gerben I. Beintema, Roland Tóth, and Maarten Schoukens. Nonlinear state-space identification using deep encoder networks. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control (L4DC)*, volume 144 of *Proceedings of Machine Learning Research*, pp. 241–250. PMLR, 2021.

Gerben I. Beintema, Maarten Schoukens, and Roland Tóth. Continuous-time identification of dynamic state-space models by deep subspace encoding. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

Christopher M. Bishop and Hugh Bishop. *Deep Learning: Foundations and Concepts*. Springer, 2024.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 6572–6583, 2018.

Yuhan Chen, Takashi Matsubara, and Takaharu Yaguchi. Neural symplectic form: Learning Hamiltonian equations on general coordinate systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 16659–16670, 2021.

Theodore James Courant. Dirac manifolds. *Transactions of the American Mathematical Society*, 319(2):631–661, 1990.

Miles Cranmer, Samuel Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations (DeepDiffEq)*, 2020. arXiv:2003.04630.

Morten Dalsmo and Arjan J. van der Schaft. On representations and integrability of mathematical structures in energy-conserving physical systems. *SIAM Journal on Control and Optimization*, 37 (1):54–91, 1998.

Shaan A. Desai, Marios Mattheakis, David Sondak, Pavlos Protopapas, and Stephen J. Roberts. Port-Hamiltonian neural networks for learning explicit time-dependent dynamical systems. *Physical Review E*, 104(3):034312, 2021.

Irene Dorfman. *Dirac Structures and Integrability of Nonlinear Evolution Equations*. John Wiley & Sons, 1993.

John R. Dormand and Peter J. Prince. A family of embedded Runge–Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, 1980.

Vincent Duindam, Alessandro Macchelli, Stefano Stramigioli, and Herman Bruyninckx (eds.). *Modeling and Control of Complex Physical Systems: The Port-Hamiltonian Approach*. Springer, 2009.

Sølve Eidnes, Alexander J. Stasik, Camilla Sterud, Eivind Bøhn, and Signe Riemer-Sørensen. Pseudo-Hamiltonian neural networks with state-dependent external forces. *Physica D: Nonlinear Phenomena*, 446:133673, 2023.

Marc Finzi, Ke Alexander Wang, and Andrew Gordon Wilson. Simplifying Hamiltonian and Lagrangian neural networks via explicit constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 13880–13889, 2020.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 15353–15363, 2019.

N. Hagelaars, G. J. E. van Otterdijk, S. Moradi, R. Tóth, N. O. Jaensson, and M. Schoukens. Identification of port-Hamiltonian differential-algebraic equations from input-output data, 2026. arXiv:2601.16827.

Ernst Hairer and Gerhard Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer, Berlin, 2 edition, 1996.

Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, Berlin, 2 edition, 1993.

Dean C. Karnopp, Donald L. Margolis, and Ronald C. Rosenberg. *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems*. John Wiley & Sons, 5th edition, 2012.

Razmik Arman Khosrovian, Takaharu Yaguchi, Hiroaki Yoshimura, and Takashi Matsubara. Poisson-Dirac neural networks for modeling coupled dynamical systems across domains. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

Christian Møldrup Legaard, Thomas Schranz, Gerald Schweiger, Ján Drgoňa, Basak Falay, Cláudio Gomes, Alexandros Iosifidis, Mahdi Abkar, and Peter Gorm Larsen. Constructing neural network based models for simulating dynamical systems. *ACM Computing Surveys*, 55(11):236:1–236:34, 2023.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

Michael Lutter, Christian Ritter, and Jan Peters. Deep Lagrangian networks: Using physics as model prior for deep learning. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

Bernhard M. Maschke and Arjan J. van der Schaft. Port-controlled Hamiltonian systems: Modelling origins and system-theoretic properties. In *Proceedings of the 2nd IFAC Symposium on Nonlinear Control Systems Design (NOLCOS)*, pp. 282–288, Bordeaux, France, 1992.

Takashi Matsubara, Ai Ishikawa, and Takaharu Yaguchi. Deep energy-based modeling of discrete-time physics. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 13100–13111, 2020.

Volker Mehrmann and Benjamin Unger. Control of port-Hamiltonian differential-algebraic systems and applications. *Acta Numerica*, 32:395–515, 2023.

Francesco Mezzadri. How to generate random matrices from the classical compact groups. *Notices of the American Mathematical Society*, 54(5):592–604, 2007.

Sarvin Moradi, Gerben I. Beintema, Nick O. Jaensson, Roland Tóth, and Maarten Schoukens. Port-Hamiltonian neural networks with output error noise models. *Automatica*, 187:112892, 2026. doi: 10.1016/j.automatica.2026.112892.

Cyrus Neary and Ufuk Topcu. Compositional learning of dynamical system models using port-Hamiltonian neural networks. In *Proceedings of the 5th Annual Learning for Dynamics and Control Conference (L4DC)*, volume 211 of *Proceedings of Machine Learning Research*, pp. 679–691. PMLR, 2023.

Cyrus Neary, Nathan Tsao, and Ufuk Topcu. Neural port-Hamiltonian differential algebraic equations for compositional learning of electrical networks. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, pp. 2097–2103. IEEE, 2025.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

Mohd Halim Mohd Noor and Ayokunle Olalekan Ige. A survey on state-of-the-art deep learning applications and challenges. *Engineering Applications of Artificial Intelligence*, 159:111225, 2025.

Henry M. Paynter. *Analysis and Design of Engineering Systems*. MIT Press, 1961.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Yulia Rubanova, Ricky T. Q. Chen, and David K. Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Mehrzad Tabatabaian. *Engineering Systems Dynamics, Modelling, Simulation, and Design: Lagrangian and Bond Graph Methods*. BCcampus, 2021. `https://pressbooks.bccampus.ca/engineeringsystems/`.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Arjan J. van der Schaft and Dimitri Jeltsema. Port-Hamiltonian systems theory: An introductory overview. *Foundations and Trends in Systems and Control*, 1(2–3):173–378, 2014.

Arjan J. van der Schaft and Bernhard M. Maschke. The Hamiltonian formulation of energy conserving physical systems with external ports. *Archiv für Elektronik und Übertragungstechnik (AEÜ)*, 49(5–6):362–371, 1995.

Pantelis R. Vlachas, Jaideep Pathak, Brian R. Hunt, Themistoklis P. Sapsis, Michelle Girvan, Edward Ott, and Petros Koumoutsakos. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, 126:191–217, 2020.

Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, Anima Anandkumar, Karianne Bergen, Carla P. Gomes, Shirley Ho, Pushmeet Kohli, Joan Lasenby, Jure Leskovec, Tie-Yan Liu, Arjun Manrai, Debora Marks, Bharath Ramsundar, Le Song, Jimeng Sun, Jian Tang, Petar Veličković, Max Welling, Linfeng Zhang, Connor W. Coley, Yoshua Bengio, and Marinka Zitnik. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021.

Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Dissipative SymODEN: Encoding Hamiltonian dynamics with dissipation and control into deep learning. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations (DeepDiffEq)*, 2020a. arXiv:2002.08860.

Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ODE-Net: Learning Hamiltonian dynamics with control. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020b.

A ELEMENTS AND JUNCTION STRUCTURE OF PORT-BASED MODELING

This appendix supplements Section 2.1. Table 3 summarizes the power-conjugate variables, the state variables, and the elements of port-based modeling in the mechanical, electromagnetic, and hydraulic domains. For example, the resistive relation of a linear damper or resistor is $e_R = -\hat{R} f_R$ with a positive coefficient $\hat{R}$. A source (Sf or Se) supplies energy from, or drains it to, the outside of the system by prescribing a flow (e.g., a current source) or an effort (e.g., a voltage source or an external force) as a function of time, independently of the conjugate variable. In this paper, each of the storage elements, resistive elements, and sources is a one-port, that is, it has a single pair of port variables (f, e), and energy is stored, dissipated, or supplied only inside these elements. A storage element whose energy depends on several states, such as the inertia of a multi-link robot whose kinetic energy depends on the momenta and the positions together, is a multiport and is outside our setting. The junction structure is an assemblage of the following four types of elements (Karnopp et al., 2012). A 0-junction imposes a common effort with balanced flows, and a 1-junction imposes a common flow with balanced efforts, which generalize Kirchhoff's current and voltage laws, respectively. In a mechanical system, a spring between two masses applies a common force to both and is connected to them by a 0-junction, whereas the springs and dampers that connect a mass to a fixed wall have a common velocity and are connected by a 1-junction. A transformer (TF) is a two-port element that relates the efforts of its two ports, and likewise the flows, by a common ratio n, that is, $f_2 = n f_1$ and $e_1 = -n e_2$, as in a lever, a gear box, or a piston–cylinder pair. A gyrator (GY) is a two-port element that relates the effort of each port to the flow of the other by a common ratio r, that is, $e_1 = -r f_2$ and $e_2 = r f_1$, as in a DC motor pump. The two-ports are power continuous as $e_1 f_1 + e_2 f_2 = 0$ holds, where the power at both ports is counted as flowing into the element.

Table 3: Power-conjugate variables, state variables, and elements in each domain

	Bond graph	Mechanical	Electromagnetic	Hydraulic
f	flow	velocity	current	volume flow
e	effort	force	voltage	pressure
q	time integral of flow	displacement	charge	volume
p	time integral of effort	momentum	flux linkage	flow tube momentum
C	C-type storage element	spring	capacitor	reservoir
I	I-type storage element	mass	inductor	fluid inertia
R	resistive element	damper	resistor	fluid resistor
Sf	flow source	moving boundary	current source	flow rate source
Se	effort source	external force	voltage source	pressure source

Note that the flow or the effort of a port can be the negative of the listed quantity as $\langle e \mid f \rangle$ is the power flowing out of the element. For example, the effort of a mass is the negative of the force acting on it.

B REDUCTION OF EQ. (3) TO THE INPUT-STATE-OUTPUT FORM

This appendix supplements Section 2: it derives the input-output representation and Eq. (4) from Eq. (3), following van der Schaft & Jeltsema (2014, Secs. 4 and 5), gives the correspondence of existing methods to Eq. (4), and states the two costs of reduction in equations.

Exchange of port variables. Exchanging the flow and the effort of a port exchanges the corresponding columns of F and E, which preserves the condition $EF^\top + FE^\top = 0$ and the rank condition (van der Schaft & Jeltsema, 2014, Sec. 5.3). We exchange them for every I-type port, every Sf port, and a subset of the resistive ports, and denote the resulting matrices, flows, and efforts with $\tilde{}$. Every storage port then satisfies $\tilde{f}_S = -\dot{z}$ and $\tilde{e}_S = \nabla H(z)$, and every source port satisfies $\tilde{e}_P = u$, where $\nabla H(z) \in \mathbb{R}^{n_S}$ is the vector of the partial derivatives $\partial H / \partial z_i$. Moreover, $\tilde{E}\tilde{F}^\top + \tilde{F}\tilde{E}^\top = 0$ and $\operatorname{rank}[\, \tilde{F} \mid \tilde{E} \,] = n$ hold.

Input-output representation. If $\tilde{F}$ is invertible for the chosen subset of the resistive ports, Eq. (1) is equivalent to $\tilde{f} = B\tilde{e}$ with $B := -\tilde{F}^{-1}\tilde{E}$, that is, $\mathcal{D}$ is the graph of B in these variables. This is the special case without constraints of the constrained input-output representation of Dalsmo & van der Schaft (1998) and van der Schaft & Jeltsema (2014, Sec. 5.2). Here, B is skew-symmetric because

multiplying $\tilde{E}\tilde{F}^\top + \tilde{F}\tilde{E}^\top = 0$ by $\tilde{F}^{-1}$ from the left and by $\tilde{F}^{-\top}$ from the right gives $B + B^\top = 0$. Since $\tilde{f}_S = -\dot{z}$, Eq. (3) becomes an ODE in z once the remaining port variables are eliminated through the constitutive relations, which is possible, for example, when $B_{RR} = 0$ and the resistive relations are explicit maps in these variables; this is the form that PoDiNNs employ (Khosrovian et al., 2025).

Input-state-output form. We can decompose B into submatrices corresponding to the storage, resistive, and source ports, and write $J := -B_{SS}$, $G_R := -B_{SR}$, and $G := -B_{SP}$. Suppose, as is typical, that B_{RR}, B_{RP}, and B_{PP} vanish, that is, no resistive or source port is connected directly to another resistive or source port. Also, suppose that the resistive relation is linear, $\tilde{e}_R = -\hat{R}\tilde{f}_R$ with $\hat{R} = \hat{R}^\top \succeq 0$, where the coefficient of an exchanged resistive port is the inverse of that in the original variables. The storage rows of $\tilde{f} = B\tilde{e}$ give $\dot{z} = -\tilde{f}_S = J\nabla H(z) + G_R\tilde{e}_R + Gu$, the resistive rows give $\tilde{f}_R = B_{RS}\nabla H(z) = G_R^\top \nabla H(z)$ by the skew-symmetry of B, and the source rows give $\tilde{f}_P = B_{PS}\nabla H(z) = G^\top \nabla H(z)$. Substituting $\tilde{e}_R = -\hat{R}\tilde{f}_R = -\hat{R}G_R^\top \nabla H(z)$ into the storage rows and setting $y := \tilde{f}_P$ yields Eq. (4) with $R = G_R\hat{R}G_R^\top$. If a resistive or source port were connected directly to another resistive or source port, B_{RP} or B_{PP} would not vanish, and the output y would acquire a feedthrough term in u (van der Schaft & Jeltsema, 2014, Sec. 4.2).

Existing methods as special cases of Eq. (4). HNNs (Greydanus et al., 2019) use $\dot{z} = J_c\nabla H_\theta(z)$ with a pair of positions and momenta as the state $z = (q, p) \in \mathbb{R}^{2d}$ and the standard symplectic matrix $J_c = \left[\begin{smallmatrix} 0 & I_d \\ -I_d & 0 \end{smallmatrix}\right]$, that is, Eq. (4) with $R = 0$, $G = 0$, and $J = J_c$; due to the skew-symmetry of J_c, they conserve the energy H_θ exactly along the exact solutions. Neural symplectic forms (Chen et al., 2021) further learn $J = J_\theta(z)$ as the inverse of a learned symplectic form. SymODENs (Zhong et al., 2020b) learn H_θ and $G_\theta(q)$ with $J = J_c$ and $R = 0$, and Dissipative SymODENs (Zhong et al., 2020a) add a learned dissipation matrix $R_\theta(q) = \Gamma_\theta(q)\Gamma_\theta(q)^\top \succeq 0$. pHNNs (Desai et al., 2021) and pseudo-Hamiltonian neural networks (Eidnes et al., 2023) employ a predetermined J and learn H_θ and R_θ (typically, its diagonal elements corresponding to the damping coefficients of the resistive elements), and learn an external force in place of Gu, which depends only on time or state. OE-pHNNs (Moradi et al., 2026) learn J_θ, R_θ, G_θ, and H_θ from noisy observations y. Lagrangian neural networks (LNNs) and deep Lagrangian networks (DeLaN) (Lutter et al., 2019; Cranmer et al., 2020) are based on Lagrangian mechanics, but can be considered as Hamiltonian systems with tangent bundles, where H and J are derived from the learned Lagrangian. CHNNs and CLNNs (Finzi et al., 2020) add holonomic constraints given in advance by projecting the vector field onto the constraint manifold.

Further details of related methods. N-pHDAEs (Neary et al., 2025) rewrite the DAE in semi-explicit form by a QR decomposition of the matrices, reduce it to an ODE by index reduction, and integrate it with a Runge–Kutta method, whereas the pH-DAE identification (Hagelaars et al., 2026) avoids reduction and solves the DAE by the backward Euler method; the latter suggests that the coefficient matrices can be learned but learns only the parameters of the elements in the experiments. Latent ODEs (Rubanova et al., 2019) encode the observed window into a distribution over the initial latent state z_0, predict the future trajectory, and decode each observation through a learned likelihood of y given z.

Cost 1: the interconnection is mixed with the elements. In Eq. (4), J and G are blocks of the interconnection, whereas $R = G_R\hat{R}G_R^\top$ combines the constitutive relation $\hat{R}$ of the resistive elements with the blocks G_R that connect them; a learned R therefore cannot be decomposed uniquely into $\hat{R}$ and G_R. For a nonlinear explicit relation $\tilde{e}_R = -\mathcal{R}(\tilde{f}_R)$, the same substitution gives $\dot{z} = J\nabla H(z) - G_R\,\mathcal{R}\big(G_R^\top \nabla H(z)\big) + Gu$, so that the dissipative term depends on the state and no longer factors as a constant matrix times $\nabla H(z)$.

Cost 2: an ODE is assumed to exist. When no choice of the exchanged ports makes $\tilde{F}$ invertible, e.g., for the balanced forces $\sum_i \partial H/\partial q_i = 0$ of multiple nonlinear springs connected to a massless point, an ODE in z exists only after part of the state has been eliminated manually or J is replaced with a rank-deficient matrix that is no longer determined by the interconnection alone but depends on the state and on the constitutive relations of the elements (e.g., the spring coefficients). Task 1 in Section 4 and in Appendix E.2 is an instance. Likewise, for a capacitor connected in parallel with a voltage source, the effort constraint $\partial H/\partial q = u$ has no flow part, so that $\tilde{F}$ is singular and the state q is determined by the input instead of by an ODE.

C PROOFS

Proof of Theorem 1. The theorem states that a subspace $\mathcal{D} \subset \mathcal{F} \times \mathcal{E}$ is a separable Dirac structure with exactly n_f independent flow constraints if and only if $\mathcal{D} = \mathcal{D}(\Omega)$ for some $\Omega \in \mathbb{R}^{n_f \times n_e}$; part (i) is the sufficiency of this representation and part (ii) its necessity.

Sufficiency (i). Since $S(\Omega)^\top = -S(\Omega)$, $W(\Omega) = \exp\big(S(\Omega)\big) W_0$ is orthogonal. Hence, the rows of W_f and W_e are mutually orthogonal, so $W_e W_f^\top = 0$. Moreover, as the rows within each block are orthonormal, W_f and W_e have full row rank. Therefore, $EF^\top + FE^\top = 0$ and $\operatorname{rank}[\, F \mid E \,] = n$. The flow rows and the effort rows are separate by construction, so $\mathcal{D}(\Omega)$ is a separable Dirac structure with exactly n_f independent flow constraints and n_e independent effort constraints.

Necessity (ii). Let $\mathcal{D}_\star$ be any separable Dirac structure with exactly n_f independent flow constraints. By the definition of separability, $\mathcal{D}_\star$ has a kernel representation, Eq. (1), in which every row involves only flows or only efforts. Collecting the n_f flow rows and the remaining n_e effort rows, we can write its matrices as

$$F = \begin{bmatrix} W_f \\ 0 \end{bmatrix}, \qquad E = \begin{bmatrix} 0 \\ W_e \end{bmatrix}, \qquad W_f \in \mathbb{R}^{n_f \times n}, \quad W_e \in \mathbb{R}^{n_e \times n}, \tag{7}$$

up to the recombination of rows within each block, and the two conditions of Eq. (2) reduce to $W_e W_f^\top = 0$ and to W_f and W_e having full row rank, with $n_f + n_e = n$. The Dirac structure is rewritten as $\{(f, e) \mid W_f f = 0,\ W_e e = 0\}$, which depends only on the row spaces of W_f and W_e. The rows can be orthonormalized within each block, and then $W = \left[\begin{smallmatrix} W_f \\ W_e \end{smallmatrix}\right]$ is an orthogonal matrix.

If $n_f n_e = 0$, take $\Omega = 0$. Given a fixed orthogonal matrix W_0, applying the CS decomposition to $W W_0^\top$ gives

$$W W_0^\top = U \exp\big(S(\Theta)\big) V^\top, \qquad U = \operatorname{diag}(U_f, U_e), \qquad V = \operatorname{diag}(V_f, V_e), \tag{8}$$

where $U_f, V_f \in \mathbb{R}^{n_f \times n_f}$ and $U_e, V_e \in \mathbb{R}^{n_e \times n_e}$ are orthogonal, and $\Theta \in \mathbb{R}^{n_f \times n_e}$ is zero except for the diagonal entries $\Theta_{ii} = \theta_i \in [0, \pi/2]$ for $i \le \min(n_f, n_e)$. Here, $\exp\big(S(\Theta)\big)$ is the cosine–sine factor, which rotates each paired flow and effort coordinate and fixes the unpaired coordinates. Set $\Omega := V_f \Theta V_e^\top$. Since $S(\Omega) = V S(\Theta) V^\top$,

$$W = U V^\top \exp\big(S(\Omega)\big) W_0 = \operatorname{diag}(U_f V_f^\top, U_e V_e^\top)\, W(\Omega). \tag{9}$$

The block-diagonal coefficient matrix $\operatorname{diag}(U_f V_f^\top, U_e V_e^\top)$ only recombines the flow rows and the effort rows separately and therefore leaves both constraint kernels unchanged. Thus, $W(\Omega)$ defines $\mathcal{D}_\star$, that is, $\mathcal{D}_\star = \mathcal{D}(\Omega)$. □

Proof of Proposition 1. By Theorem 1 (i), (F, E) defines a Dirac structure for every Ω, so Eq. (3) and condition (i) of Definition 1 give $\langle e_S \mid f_S \rangle + \langle e_R \mid f_R \rangle + \langle e_P \mid f_P \rangle = 0$ along every solution. For the i-th storage element, the energy-storage relation of either type gives $e_{S,i} f_{S,i} = -\frac{\partial H_{\theta,i}}{\partial z_i} \dot{z}_i = -\frac{\mathrm{d}}{\mathrm{d}t} H_{\theta,i}(z_i)$, and summing over i yields $\langle e_S \mid f_S \rangle = -\frac{\mathrm{d}}{\mathrm{d}t} H_\theta(z)$, which proves Eq. (5). The inequality follows from $\langle e_R \mid f_R \rangle = \sum_j f_{R,j} \hat{\mathcal{R}}_{\theta,j}(f_{R,j}) \le 0$. □

D IMPLEMENTATION OF DINES

This appendix describes the implementation of DINEs independently of the tasks; the values of the settings that depend on the task are listed in Appendix E.

D.1 LEARNABLE ELEMENTS AND INTERCONNECTION

Elements. How each element is implemented depends on the prior knowledge given about it, that is, on whether the element is assumed to be linear (see Table 7). If a storage element is assumed to be linear, its energy is the quadratic function $H_{\theta,i}(z_i) = \frac{1}{2} 10^{\vartheta_i} z_i^2$ with a learned exponent ϑ_i, so

that the coefficient is positive. For nonlinear storage elements, a neural network $\mathrm{NN}_{\theta,i}$ is added to the quadratic term as $H_{\theta,i}(z_i) = \mathrm{NN}_{\theta,i}(z_i) + \frac{1}{2}10^{\vartheta_i} z_i^2$.

The resistive relation of every resistive element is an explicit map from the flow to the effort, the sum of a neural network and a linear term, $e = \hat{\mathcal{R}}_{\theta,j}(f) = \mathrm{NN}_{\theta,j}(f) - 10^{\vartheta_j} f$ in our sign convention. The last affine layer of every network and every exponent ϑ are initialized to zero, so that at initialization the network outputs vanish and every element is linear with unit coefficient. Each element network has two hidden layers of 200 units with the hyperbolic tangent, with hidden-layer weights and biases initialized uniformly on $[-1/\sqrt{d_{\mathrm{in}}}, 1/\sqrt{d_{\mathrm{in}}}]$, where d_{in} is the layer's input dimension. No sign constraint is imposed on the resistive relation for simplicity.

Interconnection. The matrices (F, E) are parameterized as in Theorem 1. The orthogonal matrix W_0 is drawn once at initialization as a random orthogonal matrix, namely the orthogonal factor of the QR decomposition of a matrix with independent standard normal entries (Mezzadri, 2007), and Ω is initialized to zero. The true (F, E) are never used.

D.2 Learning and prediction

Trajectory network. The trajectory network χ_η takes the time t, normalized by the duration T of the training trajectories, and the index of the trajectory. The index is embedded into a 256-dimensional space. The time is encoded by random Fourier features (Tancik et al., 2020) at three scales, 1, 10, and 100, following the multi-scale construction of Wang et al. (2021). At each scale $s \in \{1, 10, 100\}$, the time is mapped to 128 sine and 128 cosine features, $\sin(\omega_\ell t/T)$ and $\cos(\omega_\ell t/T)$, with frequencies drawn independently from $\mathcal{N}(0, s^2)$ and held fixed during training. These features are concatenated with the index embedding and passed through a subnetwork shared by the three scales, with four hidden layers of 256 units and the hyperbolic tangent. The outputs of the three scales are concatenated and mapped by a final affine layer to the outputs of χ_η. The outputs are the state of every storage element, the flow of every resistive element, and the variable conjugate to the prescribed variable of every source port, whether observed or not. The affine layers are initialized by the Glorot uniform scheme with zero biases. The trainable trajectory embeddings are initialized with independent standard normal entries.

Loss. In Eq. (6), each squared norm is implemented as a mean of squares: over the observed channels for the observation term, and over the n rows of the residual $Ff + Ee$ for the residual term. The observation error is not normalized by the standard deviations of the channels. The weight is $\lambda = 1$ unless stated otherwise. At every update, the observation term is evaluated on 2,000 pairs of a trajectory and one of its sample times, drawn uniformly with replacement from all such pairs, and the residual term on 2,000 pairs of a trajectory and a collocation time drawn uniformly over the training duration.

Regularization of unobserved flows. When the flow of a resistive element is not observed, its scale is a gauge freedom that the observations do not determine. To avoid vanishing, a regularization term that pushes the variance of each such flow over the minibatch toward one is added to the loss (see Appendix E.3 for the elements to which it is applied),

$$\mathcal{L}_{\mathrm{var}} = \frac{0.1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \left(\mathrm{Var}_{\mathcal{B}}[f_{R,j}] - 1 \right)^2, \tag{10}$$

where $\mathcal{J}$ is the set of these elements, $\mathcal{B}$ is the minibatch, and $\mathrm{Var}_{\mathcal{B}}$ is the population variance over the minibatch.

Initial state. Before prediction, the initial state z_0, and the initial rate $\dot{z}_0$, and the algebraic variables a_0 (the flows of the resistive elements and the conjugate variables of the source ports) are estimated from the observations y_0 and the input u_0 at the first time step by minimizing the observation error $\frac{1}{2}\|\hat{y}(z_0, \dot{z}_0, a_0; u_0) - y_0\|^2$ subject to the learned DAE at t_0. When needed, the derivative of the learned DAE is also used as a constraint. When a history of n_h samples of inputs and observations is available, the unknowns at all time steps of the history are estimated by minimizing the sum of the observation errors over the history, subject to the learned DAE and the trapezoidal rule between consecutive samples.

Integration. At prediction, the learned DAE is integrated from the initial state by the three-stage Radau IIA method of order five (Hairer & Wanner, 1996) with the fixed step equal to the sampling interval Δt. At each step, the stage equations are solved simultaneously by Newton's method. The prescribed variables of the source ports, which are given as functions of time, are evaluated analytically at every stage.

D.3 Isolation and composition

Gauge of the boundary variable. When the boundary port of a subsystem is connected to a source during training, the conjugate boundary variable is an output of χ_η during training and, for prediction, is computed from the learned relations. Its scale is not determined by the observations of the subsystem (see Section 3), whereas the composition equates the boundary variables of two subsystems. Before composition, this scale is therefore calibrated, without retraining, by the following rules. The calibration is applied consistently to (F, E), the constitutive relations, and the momenta, preserving the observed trajectories of each subsystem.

Fixing the scale from the interconnection. For junction interconnections, the nonzero connection coefficients are known a priori to have unit magnitude. This prior is used to fix scales and is not imposed on coefficients representing unknown transformer ratios. When the learned interconnection, read in reduced row echelon form, has a row that relates the boundary variable, whose scale is undetermined, to an observed variable, whose physical scale is known, through a junction, that is, with a coefficient of unit magnitude, the scale is fixed by normalizing the absolute value of the learned coefficient to 1. In this case, no additional observation of the boundary variable is needed for the calibration. When a single element is isolated for reuse, its scale is fixed in the same way from the learned interconnection of the subsystem in which it was learned, before the element is taken out. When the row involves several observed variables, the coefficient of the largest magnitude is normalized.

Fixing the scale from measurements. Otherwise, when the scale cannot be fixed from the interconnection and the existing observations, it is calibrated from additional measurements of the boundary variable. With the boundary effort $\hat{e}_\mathrm{b}$ predicted by the model and the measured boundary effort $e_\mathrm{b}^\mathrm{obs}$, the scale α and the intercept β are estimated by

$$(\alpha, \beta) = \arg\min_{a,b} \sum_{j=1}^{N_\mathrm{cal}} \left(a\hat{e}_{\mathrm{b},j} + b - e_{\mathrm{b},j}^\mathrm{obs}\right)^2,$$

where N_cal is the number of measurements, and the calibrated boundary effort is $\tilde{e}_\mathrm{b} = \alpha\hat{e}_\mathrm{b} + \beta$. The intercept is a constant shift of the boundary effort. Since the effort rows $Ee = 0$ of the DAE are preserved by any shift δe of the efforts with $E\,\delta e = 0$, the intercept is absorbed by constant shifts c_i of the efforts of the C-type storage elements, $\tilde{e}_{C,i} = \alpha e_{C,i} + c_i$, which is equivalent to replacing the energy of each C-type storage element by $\tilde{H}_i(q_i) = \alpha H_i(q_i) + c_i q_i$; the shifts are chosen so that $E\,\delta e = 0$ holds for the rescaled matrix E and δe with c_i at the ports of the C-type elements, β at the boundary port, and 0 at the other ports.

Fixing the Transformer ratio from measurements. Similarly, if two subsystems are connected through a transformer with an unknown ratio, the ratio is identified after the scales of both subsystems are fixed. The boundary outputs of the two subsystems are evaluated at a few states of the composed system, and the ratio is obtained by closed-form least squares from the relation of the transformer. The ratio is then frozen. This procedure is required for the piston area of Task 4, mentioned in Appendix E.5.

Composed DAE. The composed system is the DAE formed by the frozen DAEs of the subsystems and the connection; the boundary variables are unknowns of this DAE and are solved at every stage of the Radau IIA method together with the states. No data of the composed system are used to retrain the subsystems.

Table 4: Systems of Tasks 1–4.

Task: system	C	I	R	Se	Sf	n	(n_f, n_e)	Observations	Inputs
1: Three-way junction	3	3	–	–	–	6	(2, 4)	6 states	–
2: RC ladder	4	–	7	2	–	13	(8, 5)	10 of 13 currents	2 source voltages
3: Subsystem A	3	3	2	1	–	9	(6, 3)	6 states	boundary force
Subsystem B	1	2	1	2	–	6	(4, 2)	3 states	boundary and external forces
4: Hydraulic	1	1	1	1	1	5	(3, 2)	2 states	source pressure, boundary flow
Mechanical	3	2	2	–	1	8	(5, 3)	5 states	boundary velocity

Boundary variables are recorded separately for training the boundary networks of Coupled Neural ODEs and, for the mechanical subsystem of Task 4, calibrating DINEs (see Table 7).

D.4 Ablations

DINEs (soft). DINEs (soft) replace the exact parameterization of Theorem 1 by independently learned matrices $F, E \in \mathbb{R}^{n\times n}$ and soft penalties. The loss of Eq. (6) is augmented by

$$\mathcal{L}_1 = \left\|EF^\top + FE^\top\right\|_{\mathrm{F}}^2, \qquad \mathcal{L}_2 = \left\|FF^\top + EE^\top - I\right\|_{\mathrm{F}}^2,$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm, and both penalties have unit weight. The second term penalizes deviations from row orthonormality of $[\,F \mid E\,]$; exact row orthonormality implies the rank condition of Eq. (2). Both penalties vanish identically under the parameterization of Theorem 1 (see Appendix C for proof). The entries of F and E are initialized independently from a Gaussian distribution with mean zero and variance $1/(2n)$. No block structure or value of n_f is enforced. The remaining training settings and the initial-state estimation and time-integration procedures are unchanged. The learned matrices are used directly for prediction, without projection onto a Dirac structure. For this ablation, "not converged" in Tables 2 and 8 denotes a numerical failure during initial-state estimation or time integration that prevents completion of all ten test trajectories for a seed. This designation is not based on a threshold on the training loss or prediction error. Means and standard deviations are reported only over seeds that complete all ten trajectories, with the number of excluded seeds stated in the tables.

Misspecified n_f. DINEs with a misspecified n_f are the model with n_f set to a value different from that of the true interconnection and $n_e = n - n_f$; all other settings are unchanged. By Theorem 1, the true Dirac structure is then outside the model family, so that the misspecification leads to a catastrophic failure rather than to a silently wrong model (see Section 3). The values of n_f examined are listed in Appendix E.2.

E Experiments

E.1 Common settings

Systems. Table 4 lists the elements, observations, and inputs of the learned systems of Tasks 1–4. The coefficients of the true systems below are disclosed for reproducibility and are not given to any model.

Sampling and data. All trajectories are sampled with the time step $\Delta t = 0.1$ in the units of the simulation, without noise. Durations are counted in steps, so that a trajectory of K steps has $K + 1$ samples, and the frequencies of the inputs are given in cycles per step. Every training set, of Tasks 1 and 2 and of each subsystem of Tasks 3 and 4, consists of 100 trajectories of 500 steps, and every test set of 10 trajectories of 10,000 steps.

Reference trajectories. The reference trajectories are generated by integrating the true equations with the eighth-order Runge–Kutta method DOP853 (Hairer et al., 1993) with the relative tolerance 10^{-11}, the absolute tolerance 10^{-13}, and the maximum step Δt. The generated data are checked against a second integration with the tolerances 10^{-12} and 10^{-14} and the maximum step $\Delta t/2$.

Training. All models are trained with five seeds, which change the initialization and the sampling of minibatches, with minibatches of 2,000. The optimizer is Adam (Kingma & Ba, 2015) with the learning rate 10^{-3}, $(\beta_1, \beta_2) = (0.9, 0.999)$, and no weight decay, with cosine annealing (Loshchilov & Hutter, 2017) of the learning rate to zero over all updates. The number of updates is 100,000 unless stated otherwise. Training is in single precision. Whenever a loss is normalized by standard deviations, these are the sample standard deviations over the training set. The parameters after the last update are used, without early stopping and without model selection on the test set.

Prediction. Predictions are computed in double precision. The learned DAEs of DINEs, the composed systems, and the receiving system of Task 5 are integrated as described in Appendices D.2 and D.3. The ODE models are integrated by the adaptive Dormand–Prince method with the relative tolerance 10^{-9} and the absolute tolerance 10^{-11}, the maximum step Δt, and outputs at every step.

Evaluation metrics. The metrics of Section 4.1 are defined as follows. Let $\hat{y}(t_k)$ and $y(t_k)$ be the predicted and true observations at the k-th sample and d the number of observed channels. The squared error at sample k and the MSE over an evaluation horizon of K steps ($K = 10{,}000$ unless stated otherwise), including the initial sample, are

$$e_k = \frac{1}{d}\left\|\hat{y}(t_k) - y(t_k)\right\|^2, \qquad \mathrm{MSE} = \frac{1}{K+1}\sum_{k=0}^{K} e_k. \tag{11}$$

The valid prediction time (VPT) (Vlachas et al., 2020) for a threshold ε is the fraction of the horizon before e_k first exceeds ε: with k_* the first index such that $e_{k_*} > \varepsilon$,

$$\mathrm{VPT}(\varepsilon) = \frac{\max(k_* - 1, 0)}{K}, \tag{12}$$

and $\mathrm{VPT}(\varepsilon) = 1$ if the threshold is never exceeded. Both metrics are averaged over the ten test trajectories of each seed.

E.2 Task 1: three-way junction

System. Three masses m_i are attached through springs to one massless junction. With the elongation q_i of spring i, the velocity v_i of mass i, and the velocity v_J of the junction,

$$\dot{q}_i = v_i - v_J, \qquad m_i \dot{v}_i = -e_i(q_i), \qquad 0 = \sum_{i=1}^{3} e_i(q_i), \qquad i = 1, 2, 3,$$

where the last equation is the force balance at the junction. The spring forces are $e_i(q_i) = a_i q_i + b_i q_i^3$ with $(a_i, b_i) = (0.4, 0.05), (0.3, 0.4), (0.25, 0.15)$, and the masses are $(m_1, m_2, m_3) = (1.0, 1.2, 0.8)$; there are no dampers and no inputs. For the reference trajectories, the force balance is differentiated in time, which gives $v_J = \sum_i k_i(q_i) v_i / \sum_i k_i(q_i)$ with the tangent stiffness $k_i(q_i) = a_i + 3 b_i q_i^2$; this v_J is substituted, and the resulting ODE is integrated. In port-based terms, the system has three C-type and three I-type elements, and the junction imposes two constraints among the flows and four among the efforts.

Data. The initial elongations q_1 and q_2 are uniform on $[-1.2, 1.2]$ and the initial velocities uniform on $[-0.6, 0.6]$; q_3 is solved from the force balance, whose solution is unique since e_3 is monotonic. All six states $(q_1, q_2, q_3, v_1, v_2, v_3)$ are observed. The initial states of the test trajectories are drawn from the same distributions as those of the training trajectories.

DINEs. The I-type elements are assumed to be linear, with their masses learned, and the springs are nonlinear. The main setting is $(n_f, n_e) = (2, 4)$; in the sweep, $n_f \in \{1, 2, 3, 4, 5\}$ with $n_e = 6 - n_f$, and all other settings, including the initialization rule and the observations, are unchanged. DINEs estimate the initial state from the six observations at the first sample.

Neural ODEs and PoDiNNs. Neural ODEs (Chen et al., 2018) take the observed states as the state x and learn the vector field by a network with two hidden layers of 200 units and the hyperbolic tangent. PoDiNNs (Khosrovian et al., 2025) consist of one network per C-type element for its

Table 5: Resistors of Task 2.

Resistor	ΔV_j	r_j	k_j
R_0	$u_\mathrm{A} - a$	0.8299	0.7991
R_1	$a - v_1$	0.6711	4.6312
R_2	$a - v_2$	0.7407	6.3965
R_3	$b - v_3$	0.6981	6.5022
R_4	$b - v_4$	0.5908	5.1852
R_5	$a - b$	0.7813	2.3884
R_6	$b - u_\mathrm{B}$	0.7018	0.7792

energy, of the same multilayer-perceptron structure as the element networks of DINEs but without an explicit quadratic term, following the original implementation, one learned mass 10^{ϑ_i} per I-type element, and a constant matrix $B_\theta \in \mathbb{R}^{n_I \times n_C}$, initialized to zero, that couples the two:

$$\dot{q} = B_\theta^\top v, \qquad \dot{v} = -M^{-1} B_\theta \, \nabla_q H_\theta(q),$$

where M is the diagonal matrix of the masses, and n_C and n_I are the numbers of C-type and I-type elements. Here, B_θ is the I–C coupling block of the matrix of the input-output representation in Section 2.2; on Task 1, which has no resistive elements and no sources, the coupling parameters of the original implementation reduce to this zero-initialized constant matrix. PoDiNNs (Khosrovian et al., 2025) are given the list of elements, the correspondence between the observed channels and the states of the elements, and the linearity of the I-type elements. Neither the constraint nor the state-dependent matrix $B(q)$ of the reduced ODE is supplied, and the states are not projected onto the true constraints. Both models are trained by one-step prediction: from a training sample x_k, the state after Δt is predicted by one step of the classical fourth-order Runge–Kutta method. The loss is the squared error to x_{k+1}, normalized by the sample standard deviation of each channel over the training set and averaged over the channels and the minibatch. Predictions start from the observed initial state, $x_0 = y_0$, and integrate the learned ODE by the adaptive Dormand–Prince method (Dormand & Prince, 1980).

E.3 Task 2: RC ladder

System. Two voltage sources u_A and u_B, two nodes a and b, four grounded linear capacitors $\mathrm{C}_1, \ldots, \mathrm{C}_4$, and seven nonlinear resistors $\mathrm{R}_0, \ldots, \mathrm{R}_6$ are connected as follows: R_0 connects source A to node a, R_1 and R_2 connect node a to C_1 and C_2, R_5 connects node a to node b, R_3 and R_4 connect node b to C_3 and C_4, and R_6 connects node b to source B. In Figure 2, the source voltages are denoted by $V_{E_1} = u_\mathrm{A}$ and $V_{E_2} = u_\mathrm{B}$. Capacitor i has the charge q_i and the voltage $v_i = \gamma_i q_i$ with $(\gamma_1, \ldots, \gamma_4) = (0.375, 2.1, 0.45, 2.4)$, that is, the capacitances are $1/\gamma_i$. Resistor j has the voltage drop $\Delta V_j = r_j I_j + k_j I_j^3$ in the direction of Table 5. The currents satisfy $I_0 = I_1 + I_2 + I_5$ and $I_5 = I_3 + I_4 + I_6$ at the two nodes, and the charges satisfy $\dot{q} = (I_1, I_2, I_3, I_4)$. The node voltages are algebraic variables determined by Kirchhoff's current laws together with the nonlinear resistor relations. For the reference trajectories, the resulting nonlinear equations are solved by Newton's method at every evaluation. In port-based terms, the system has 13 ports, four C-type elements, seven resistive elements, and two effort sources; its graph has 13 branches and nine nodes, hence eight current-law constraints and five voltage-law constraints.

Inputs and data. Each source voltage is a sum of six cosines and an offset, $u(t) = c + \sum_{\ell=1}^{6} A_\ell \cos(2\pi \nu_\ell \, t/\Delta t + \phi_\ell)$, where the frequencies ν_ℓ are in cycles per step, so that $\nu_\ell/\Delta t$ is the frequency per unit time; the inputs are evaluated analytically at all sample, collocation, and stage times. The two sources are generated as a common mode and a differential mode, $u_\mathrm{A} = u_\mathrm{c} + u_\mathrm{d}$ and $u_\mathrm{B} = u_\mathrm{c} - u_\mathrm{d}$, each of which is a sum of three cosines and an offset. The trajectories are divided into six groups, which combine the upper frequency $\nu_\mathrm{max} \in \{0.03, 0.08\}$ with the level $\kappa \in \{4, 12, 20\}$, with the group labels randomly permuted over the trajectories; for each ν_max, the training groups contain 17, 17, and 16 trajectories for the three increasing levels and the test groups 2, 2, and 1 trajectories. In each group, the frequencies of the common mode are uniform on the lower half $[0.0025, (0.0025 + \nu_\mathrm{max})/2]$ of the band and those of the differential mode on the upper half $[(0.0025 + \nu_\mathrm{max})/2, \nu_\mathrm{max}]$, the phases are uniform on $[0, 2\pi]$, and, before the rescaling below,

the offset of each mode is uniform on $[-0.5\kappa, 0.5\kappa]$. The rescaling gives the two modes the same mean square on a fixed grid. Before the rescaling, the amplitudes are uniform on $[0.25\kappa, 0.65\kappa]$ for the common mode and on $[0.2\kappa, 0.52\kappa]$ for the differential mode. For the amplitude normalization, provisional frequencies $\nu_{m,\ell}^{(0)}$ are first drawn independently and uniformly from the full band $[\nu_{\min}, \nu_{\max}]$ for both modes, where $\nu_{\min} = 0.0025$ and $m \in \{c, d\}$. Using these provisional frequencies and the original phases, the sum S of the mean squares of the two modes, including their offsets, is computed on the grid $t_j = j\Delta t/2$, $j = 0, \ldots, 500$. The frequencies are then mapped to the respective half-bands by

$$\nu_{c,\ell} = \frac{\nu_{\min} + \nu_{c,\ell}^{(0)}}{2}, \qquad \nu_{d,\ell} = \frac{\nu_{\max} + \nu_{d,\ell}^{(0)}}{2},$$

without drawing new frequencies, and the differential-mode phases are shifted by $\pi/2$. Finally, the amplitudes and offset of each mode are multiplied by a single factor specific to that mode, so that its mean square, evaluated with the remapped frequencies and updated phases on the same grid, equals $S/4$. The initial capacitor voltages are obtained from $\tilde{v}_i = a_i^0 \xi_i$ with ξ_i uniform on $[-1.1, 1.1]$ and $a^0 = (0.75, 1.05, 0.9, 1.2)$: within the pairs $(1, 2)$ and $(3, 4)$, the mean of the two voltages is kept and their difference is tripled, which gives $\hat{v}_i$, and then $v_i(0) = s\hat{v}_i$ with $s = \left(\sum_i \tilde{v}_i^2/a_i^0\right)^{1/2}\left(\sum_i \hat{v}_i^2/a_i^0\right)^{-1/2}$, so that the weighted sum of squares $\sum_i v_i(0)^2/a_i^0$ equals that of $\tilde{v}$. This transformation increases the within-pair voltage differences relative to their common components while preserving the specified weighted sum of squares. The initial charges are $q_i(0) = v_i(0)/\gamma_i$. The initial states of the test trajectories are drawn from the same distributions as those of the training trajectories. The observed channels are the currents at ten of the 13 ports: the source currents $I_{\mathrm{A}} = I_0$ and $I_{\mathrm{B}} = -I_6$, the currents of the four capacitors, and the currents of $\mathrm{R}_1, \ldots, \mathrm{R}_4$; all ten are given to all models alike. The current labels in Figure 2 satisfy $I_{E_1} = I_{\mathrm{A}} = I_0$, $I_{E_2} = I_{\mathrm{B}} = -I_6$, and $I_{C_i} = I_{R_i} = I_i = \dot{q}_i$ for $i = 1, \ldots, 4$. The charges, the node voltages, and the currents of R_0, R_5, and R_6 are not observed. That the capacitor currents coincide with the currents of $\mathrm{R}_1, \ldots, \mathrm{R}_4$, and I_{A} and I_{B} with the currents of R_0 and R_6 up to sign, is not given to any model. Each test trajectory has 20 samples of inputs and observations before the prediction as the history.

DINEs. The capacitors are assumed to be linear with unknown capacitances, and the resistors are nonlinear. DINEs use $\lambda = 10$, the regularization term $\mathcal{L}_{\mathrm{var}}$ (as defined in Appendix D.2) for the flows of the three unobserved resistors R_0, R_5, and R_6, and 500,000 updates. The initial state is estimated from the history of 20 samples, starting from zero charges. The model is then advanced by one step from the last sample of the history, and the following 10,000 steps are evaluated, including the first predicted sample.

CT-SUBNET. CT-SUBNET (Beintema et al., 2023) consists of an encoder ψ_θ that maps the inputs and outputs of the $n_h = 20$ past samples to the state $z \in \mathbb{R}^{n_z}$, a vector field $\dot{z} = \varphi_\theta(z, u)$, and an observation $\hat{y} = h_\theta(z, u)$. The observation map takes the input as well, unlike the original form $h_\theta(z)$, because the observed currents depend directly on the source voltages; otherwise the currents would have to be reproduced from the latent state alone. Each of the three maps is the sum of a network with two hidden layers of 64 units and the hyperbolic tangent and an affine map from its input to its output, with zero initial biases. The inputs and outputs are standardized by their means and sample standard deviations over the training set, and the output of h_θ is transformed back to physical units. The output of the network of the vector field is divided by a time scale of $50\Delta t$. At every update, 2,000 windows are drawn from the training set, each consisting of the n_h past samples for the encoder and the following 50 samples. Over the 50 samples, the model is integrated by the classical fourth-order Runge–Kutta method with one step per Δt and the analytic inputs at the stage times. The mean squared error of the 50 predicted outputs, normalized by the sample standard deviation of each output channel, is minimized with respect to ψ_θ, φ_θ, and h_θ jointly. For prediction, the encoder maps the history to the state at the first predicted sample once, and the model is integrated with the known inputs by the adaptive Dormand–Prince method; neither the encoder nor the state is updated by observations afterwards. The state dimension is $n_z \in \{4, 8, 12, 16, 20\}$, and the model is trained for 500,000 updates.

Table 6: Subsystems of Tasks 3 and 4.

Subsystem	Inertias	Springs (a, b)	Dampers (a, b)	Boundary (prescribed $\to$ recorded)	Observed
Task 3, A	$1, 1.2, 1.4$	$(0.4, 0.1)$ $(0.3, 0.1)$ $(0.2, 0.1)$	$(0.07, 0.45)$ $(0.09, 0.55)$	$F_\mathrm{b} \to v$	$3\,q, 3\,v$
Task 3, B	$0.9, 1.1$	$(0.35, 0.15)$	$(0.08, 0.5)$	$F_\mathrm{b} \to v$	$1\,q, 2\,v$
Task 4, hydraulic	0.9	$(0.8, 0)$ (reservoir)	fluid resistor	$Q_\mathrm{b} \to P$	$1\,q, 1\,Q$
Task 4, mechanical	$1, 1.4$	$(0.4, 0.1)$ $(0.3, 0.1)$ $(0.2, 0.1)$	$(0.07, 0.45)$ $(0.09, 0.55)$	$v \to F_\mathrm{b}$	$3\,q, 2\,v$

E.4 Task 3: rigid coupling

System. Each subsystem is a chain of masses, springs, and dampers with one boundary port. Springs have $e = aq + bq^3$, and dampers have $e = -R(f)$ with $R(f) = af + bf^3$, in our sign convention; the equations of motion below are written with the resistive forces R, which are the negatives of the efforts. Subsystem A is a chain from a wall, a spring with a damper, mass m_1, a spring, mass m_2, a spring with a damper, and mass m_3, to which the boundary force u_A is applied. With the absolute positions x_1, x_2, x_3 of the masses, the elongations of the three branches are $(q_1, q_2, q_3) = (x_1, x_2 - x_1, x_3 - x_2)$, which are observed together with the velocities v_1, v_2, v_3, and the equations of motion are

$$\dot{q}_1 = v_1, \qquad \dot{q}_2 = v_2 - v_1, \qquad \dot{q}_3 = v_3 - v_2,$$
$$m_1\dot{v}_1 = -T_1 + T_2, \qquad m_2\dot{v}_2 = -T_2 + T_3, \qquad m_3\dot{v}_3 = -T_3 + u_\mathrm{A},$$

where T_i is the total force of the spring and the damper of branch i, $T_i = e_i(q_i) + R_i(\dot{q}_i)$ for $i = 1, 3$ and $T_2 = e_2(q_2)$. Subsystem B consists of mass $m_{\mathrm{B}1}$, to which the boundary force $u_{\mathrm{B}0}$ is applied, a spring with a damper, and mass $m_{\mathrm{B}2}$, to which the external force $u_{\mathrm{B}1}$ is applied. With $q_\mathrm{B} = x_{\mathrm{B}2} - x_{\mathrm{B}1}$ and $T_\mathrm{B} = e_\mathrm{B}(q_\mathrm{B}) + R_\mathrm{B}(v_{\mathrm{B}2} - v_{\mathrm{B}1})$,

$$\dot{q}_\mathrm{B} = v_{\mathrm{B}2} - v_{\mathrm{B}1}, \qquad m_{\mathrm{B}1}\dot{v}_{\mathrm{B}1} = T_\mathrm{B} - u_{\mathrm{B}0}, \qquad m_{\mathrm{B}2}\dot{v}_{\mathrm{B}2} = -T_\mathrm{B} - u_{\mathrm{B}1}.$$

The recorded boundary velocities are v_3 of A and $v_{\mathrm{B}1}$ of B; that they coincide with observed mass velocities is not given to any model. Table 6 lists the elements of the subsystems.

Boundary port. During the training of a subsystem, its boundary port is a source: the prescribed boundary variable, a force, is a known input, and the conjugate variable, a velocity, is recorded but excluded from the losses of the dynamics training of every model. The recorded boundary variables of the training data are used for the training of the boundary network of Coupled Neural ODEs (and, in Task 4, for the calibration of DINEs). No observation of the boundary variables is given during the prediction of the composed systems. In the composed system, the boundary velocities are equated, the reaction force μ acts with $+\mu$ on A and $-\mu$ on B and is an unknown, and $u_{\mathrm{B}1}$ is the only known input; no data of the composed system are used for training. The composed systems are integrated as DAEs with the boundary variables as unknowns, and the initial state is estimated from the nine observations at the first sample.

Data. The initial positions of A are uniform on $[-1, 1]$ and its velocities on $[-0.6, 0.6]$; the initial elongation of B is uniform on $[-0.8, 0.8]$ and its velocities on $[-0.5, 0.5]$. The boundary force of A is a sum of three cosines with frequencies uniform on $[0.005, 0.02]$, amplitudes uniform on $[0.4, 0.8]$, and phases uniform on $[0, 2\pi]$. Each force of B is a sum of four cosines with frequencies drawn from the multiples of the reciprocal of the training duration within $[0.006, 0.018]$, amplitudes uniform on $[0.08, 0.24]$, and phases uniform on $[0, 2\pi]$. The reference trajectories of the composed system are generated by substituting the coupling variables, computed analytically, into the differential equations and integrating the resulting ODE in the original states of the subsystems with the method and tolerances of Appendix E.1. Here, m_A and m_B are the two coupled masses, F_A and F_B are the forces acting on them other than the reaction force, and the reaction force $\mu = (m_\mathrm{A}F_\mathrm{B} -$

$m_{\mathrm{B}}F_{\mathrm{A}})/(m_{\mathrm{A}} + m_{\mathrm{B}})$ keeps the boundary velocities equal; the initial boundary velocities are set equal.

Test trajectories. The test trajectories of the composed systems are generated by the following rule. Each candidate starts from independently selected subsystem training initial states, adjusted to satisfy the coupling constraints. The input frequencies and amplitudes are taken from the selected training trajectories, with independently redrawn uniform phases. A candidate is accepted if, at all samples of the 10,000 steps of its reference trajectory, every state, known input, recorded boundary variable, and resistive flow stays within the range of the corresponding channel in the training data of the subsystems, and the residual of the coupling constraints, $\max_k |v_3(t_k) - v_{\mathrm{B1}}(t_k)|$ at the samples, stays below 2×10^{-7} (an acceptance criterion after generation, not a tolerance of the integrator). The first ten accepted candidates, out of at most 500, form the test set, which is common to both methods and all seeds.

DINEs. The I-type elements are assumed to be linear with unknown coefficients, and the springs and dampers are nonlinear. The scale of the recorded boundary velocity of each subsystem is fixed from the learned interconnection as described in Appendix D.3, without additional observations of the boundary variable.

Coupled Neural ODEs. Subsystem dynamics. For each subsystem, a Neural ODE of Appendix E.2 takes the observed states as its state x and the prescribed boundary variable, together with the other known inputs, as its input u, $\dot{x} = f_\theta(x, u)$, and is trained by the same one-step loss, with the known inputs evaluated analytically at the stage times.

Boundary network. Composition also requires the boundary variable conjugate to the prescribed boundary input, so a separate network $h_\phi(x)$ with two hidden layers of 64 units, the hyperbolic tangent, and one scalar output predicts it from the current state of the subsystem only: the boundary velocity on Task 3. After the training of f_θ, its parameters are frozen and h_ϕ is trained separately on the recorded states x_j^{obs} and the recorded boundary variables b_j^{obs} of the training data of the subsystem, not on states generated by the learned vector field. At every update, a minibatch $\mathcal{B}$ of 2,000 pairs is drawn with replacement, and

$$\mathcal{L}_{\mathrm{boundary}} = \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \left(\frac{h_\phi(x_j^{\mathrm{obs}}) - b_j^{\mathrm{obs}}}{s_b} \right)^2$$

is minimized, where s_b is the sample standard deviation of the recorded boundary variable over the training set, with the optimizer, the schedule, and the number of updates of Appendix E.1. The normalization is applied to the loss only, so the network predicts the boundary variable in its recorded physical units.

Composition and prediction. Both vector fields and both boundary networks are frozen before composition. During prediction, each boundary network is evaluated on the current predicted state of its subsystem, and its output enters the same connection as for DINEs: on Task 3, the outputs of the two boundary networks are equated as the boundary velocities, and the reaction force is an unknown input of the vector fields. The composed system is integrated, and its initial state is estimated, using the procedures of Appendices D.2 and D.3 and the settings stated above; neither the vector fields nor the boundary networks are retrained on data of the composed system.

E.5 Task 4: hydraulic–mechanical coupling

System. Both subsystems are chains with one boundary port as in Task 3 (see Appendix E.4), the mechanical subsystem with masses, springs, and dampers, and the hydraulic subsystem with a fluid inertia, a reservoir, and a fluid resistor; the springs and dampers have the same forms as in Task 3, with the coefficients of Table 6. The hydraulic subsystem consists of a pressure source u_P, a fluid resistor, one fluid inertia, and a reservoir, with the states (q_h, Q_1) and the known inputs u_P and the boundary flow Q_{b}:

$$P = 0.8\, q_h, \qquad \dot{q}_h = Q_1 - Q_{\mathrm{b}}, \qquad 0.9\, \dot{Q}_1 = u_P - P - R(Q_1),$$

where $R(Q) = 0.16\, Q \sqrt{Q^2 + (0.18/0.16)^2}$; the boundary pressure P is recorded. The mechanical subsystem is a chain from the boundary with the prescribed velocity v_{b}, a spring, mass m_1, a spring

with a damper, mass m_2, and a spring with a damper, to a wall, with the states $(q_1, q_2, q_3, v_1, v_2)$ and the input v_b only:

$$\dot{q}_1 = v_\mathrm{b} - v_1, \qquad \dot{q}_2 = v_1 - v_2, \qquad \dot{q}_3 = v_2,$$
$$m_1\dot{v}_1 = e_1 - e_2 - R_1(v_1 - v_2), \qquad m_2\dot{v}_2 = e_2 - e_3 + R_1(v_1 - v_2) - R_2(v_2),$$

where e_i and R_i are the spring forces and the damper forces of Table 6; the boundary force $F_\mathrm{b} = -e_1(q_1)$ is recorded. The coupling is $Q_\mathrm{b} = A_\mathrm{p} v_\mathrm{b}$ and $A_\mathrm{p}P + F_\mathrm{b} = 0$ with the unknown piston area A_p, whose true value 1.3 is not given.

Boundary port. The protocol of Task 3 applies: the boundary flow Q_b is prescribed and the pressure P is recorded for the hydraulic subsystem, and the velocity v_b is prescribed and the force F_b is recorded for the mechanical subsystem (see Table 6). In the composed system, only the pressure source is a known input, and Q_b and v_b are unknowns. The composed system is integrated as a DAE with the boundary variables as unknowns, and the initial state is estimated from the seven observations at the first sample.

Data. The initial states of the hydraulic subsystem are uniform on $[-0.7, 0.7]$ for q_h and on $[-0.45, 0.45]$ for Q_1; those of the mechanical subsystem are uniform on $[-0.7, 0.7]$ for q_1 and q_2, with $q_3 = -q_1 - q_2$, and on $[-0.45, 0.45]$ for v_1 and v_2. Every input is a sum of five cosines: the frequency band is divided into five equal sub-bands and one frequency is drawn uniformly from each, the phases are uniform on $[-\pi, \pi]$, and the amplitudes are $A_\ell = s\, b_\ell / \sqrt{\sum_\ell b_\ell^2/2}$ with relative amplitudes b_ℓ uniform on $[0.7, 1.3]$ and a nominal root-mean-square level s. For the pressure source of the hydraulic subsystem, $s \in \{0.8, 1.8, 2.6\}$, and for its boundary flow, $s \in \{0.12, 0.35, 0.65\}$; the band is $[0.002, 0.010]$ or $[0.002, 0.035]$. For each of these two inputs, the three levels and the two bands form six groups, and the group labels are shuffled separately for the two inputs, so that neither the levels nor the bands are constrained to match within a trajectory. For the boundary velocity of the mechanical subsystem, $s \in \{0.20, 0.56, 1.00\}$ with the band $[0.005, 0.020]$. The reference trajectories of the composed system are generated as in Task 3. The coupling variables follow from differentiating the force balance $A_\mathrm{p}P + F_\mathrm{b} = 0$ with $P = 0.8\, q_h$ and $F_\mathrm{b} = -e_1(q_1)$: $v_\mathrm{b} = (0.8A_\mathrm{p}Q_1 + e_1'(q_1)v_1)/(0.8A_\mathrm{p}^2 + e_1'(q_1))$ and $Q_\mathrm{b} = A_\mathrm{p} v_\mathrm{b}$, which are substituted into the ODE of the seven states; the initial state satisfies the force balance. The five trajectories of 100 steps of the composed system, provided for the identification of the area and for the scales of the initialization, start from states consistent with the force balance and within the training ranges.

Test trajectories. The test trajectories of the composed system are generated by the rule of Task 3 (see Appendix E.4), where the residual of the coupling constraints is $\max_k |A_\mathrm{p}P(t_k) + F_\mathrm{b}(t_k)|$.

DINEs. The reservoir is assumed to be linear with an unknown coefficient, and the fluid resistor is nonlinear; the I-type elements, springs, and dampers are treated as in Task 3. The scale of the boundary pressure of the hydraulic subsystem is fixed from the learned interconnection as in Task 3, whereas the boundary force of the mechanical subsystem is calibrated from measurements as described in Appendix D.3, with an intercept in addition. For the mechanical subsystem, $N_\mathrm{cal} = 4$ observations of the boundary force, preselected from the training data of the subsystem by a fixed random draw, are used. The area is identified from four fixed states of the five trajectories of the composed system by the closed-form least squares of Appendix D.3, $\hat{A}_\mathrm{p} = -\sum_j \hat{P}_j \hat{F}_{\mathrm{b},j} / \sum_j \hat{P}_j^2$, where $\hat{P}_j$ and $\hat{F}_{\mathrm{b},j}$ are the calibrated boundary outputs of the two subsystems at the j-th state. The hydraulic boundary output $\hat{P}$ is evaluated with the boundary flow set to zero. The observations of the composed system do not determine the initial rates. In the initial-state estimation, the observation error is therefore normalized by the sample standard deviations σ_j of the observed channels over the five trajectories, and a penalty $\frac{10^{-6}}{2}\sum_j (\dot{\hat{y}}_{0,j}/\rho_j)^2$ on the initial rates of the observed channels is added, where ρ_j are the sample standard deviations of the increments divided by Δt over the same trajectories.

Coupled Neural ODEs. Coupled Neural ODEs are as in Task 3 (see Appendix E.4), except that the boundary network predicts the boundary pressure of the hydraulic subsystem, whose boundary input is the flow, and the boundary force of the mechanical subsystem, whose boundary input is the velocity, and that, in the composed system, the boundary flow and velocity are unknown inputs, and

the boundary networks supply the pressure and the force of the coupling constraints of the piston. If the connection has a transformer with an unknown ratio, the ratio is identified from the same states of the composed system as for DINEs by the same closed-form least squares, with the outputs of the frozen boundary networks, and is then frozen. The same normalized initial-state objective and rate penalty are used for Coupled Neural ODEs.

E.6 Task 5: reuse of the learned spring

Isolation. The donor is DINEs trained on subsystem A of Task 3, and all five training seeds are used. The energy network of the second spring, including its quadratic term, is isolated and frozen as the constitutive relation $\hat{e}(q) = \partial_q \hat{H}(q)$. The scale of the force is fixed from the learned interconnection as described in Appendix D.3, since the coefficients of the junctions have unit magnitude and the subsystem contains no unknown transformer ratio. With the learned flow of the source port written in the observed velocities of the masses as $f_s = \sum_i \omega_i v_i$, the coefficient of the largest magnitude is selected for each seed, and the scale is

$$i^\star = \arg\max_i |\omega_i|, \qquad \alpha = 1/|\omega_{i^\star}|,$$

so that the inertia connected to the boundary is identified from the learned interconnection rather than prescribed by its index. The reused constitutive relation is $g(q) = \alpha \hat{e}(q)$; the network and its displacement axis are unchanged, and no offset of the force is corrected. Nothing else is transferred from the donor, neither (F, E), χ_η, the other springs, the dampers, nor the masses, and neither additional measurements of the boundary variable or of the spring force, nor the true constitutive relation of the spring, nor the response of the receiving system is used to determine the scale.

Receiving system. The receiving system is known: a wall is connected to a main mass M by a linear spring k_h with a damper c_h in parallel, the input $u(t)$ acts on M, and an auxiliary mass m_a is connected to M by the reused spring with a damper c_a in parallel. With the positions x and x_a measured from the wall and the relative displacement $r = x_\mathrm{a} - x$ of the reused spring,

$$\begin{aligned} M\ddot{x} &= -k_\mathrm{h} x - c_\mathrm{h}\dot{x} + g(r) + c_\mathrm{a}\dot{r} + u(t), \\ m_\mathrm{a}\ddot{x}_\mathrm{a} &= -g(r) - c_\mathrm{a}\dot{r}, \end{aligned}$$

where $g(r)$ is the derivative of the scaled energy of the spring with respect to r. The observed channels are $y = (x, x_\mathrm{a}, v, v_\mathrm{a})$ with $v = \dot{x}$ and $v_\mathrm{a} = \dot{x}_\mathrm{a}$. All coefficients are fixed, $M = 2$, $m_\mathrm{a} = 0.3$, $k_\mathrm{h} = 2$, $c_\mathrm{h} = 0.04$, and $c_\mathrm{a} = 0.02$, and all initial positions and velocities are zero. The prediction is terminated if the argument of the reused constitutive relation, including its evaluations within the Newton iterations, leaves the range of the displacements of the second spring in the training data of the donor; the argument is never clamped to this range.

Data. The reference trajectories use the true relation of the second spring of subsystem A of Task 3, $e_\star(q) = 0.3q + 0.1q^3$, in place of g in the equations of the receiving system, and are generated and checked as in Appendix E.1. Each input is a sum of five cosines, $u(t) = \sum_{\ell=1}^{5} A_\ell \cos(2\pi\nu_\ell\, t/\Delta t + \phi_\ell)$, with one frequency drawn uniformly from each of the bands $[0.005, 0.010]$, $[0.010, 0.015]$, $[0.015, 0.020]$, $[0.020, 0.025]$, and $[0.025, 0.030]$, phases uniform on $[0, 2\pi]$, and amplitudes $A_\ell = s\, b_\ell / \sqrt{\sum_\ell b_\ell^2/2}$ with relative amplitudes b_ℓ uniform on $[0.7, 1.3]$ and a nominal root-mean-square level $s \in \{0.05, 0.15\}$. At each level, candidates are generated in order, at most 500 per level, and the first five whose reference responses keep the deformation of the reused spring within the interval between the first and the 99th percentiles of the displacements of the second spring in the training data of the donor are retained; no candidate was rejected. This selection uses the reference system, not the prediction errors, and the same ten inputs are used for all five seeds of the donor. Each test trajectory has 10,000 steps, as in the other tasks.

F Additional results and discussions

Prior knowledge given to the models. Table 7 summarizes the prior knowledge and assumptions given to the models. The available prior knowledge is not identical across the models, but the comparison was performed under the same data and evaluation procedures.

Table 7: Prior knowledge given to the models. The last row is used for scoring only.

	DINEs	Neural ODEs	PoDiNNs	CT-SUBNET	Coupled Neural ODEs
Training observations and known inputs	yes	yes	yes	yes	yes
Types and numbers of elements	yes	no	C-type, I-type	no	dimensions only
Correspondence of observations to elements	yes	as states	yes	as outputs	states, boundary port
Linearity of elements (coefficients unknown)	yes	no	I-type elements	no	no
(n_f, n_e)	yes	no	no	no	no
True wiring and coefficients	no	no	no	no	no
Unobserved true states	no	–	–	no	–
Boundary records of the subsystem training data	calibration (Task 4, mechanical only)	–	–	–	boundary-network training
Connection of the boundary ports at composition	yes	–	–	–	yes
States of the composed system (Task 4)	area, initialization scales	–	–	–	area, initialization scales
Displacement–force measurements of the true spring (Task 5)	no	–	–	–	–
Test observations after the first sample (after the history on Task 2)	scoring	scoring	scoring	scoring	scoring

Table 8: VPT of Tasks 1–5 for further thresholds ε.

Task	Model	$\varepsilon = 10^{-6}$	10^{-5}	10^{-4}	10^{-3}
1	Neural ODEs	0.008 ± 0.001	0.023 ± 0.004	0.062 ± 0.015	0.160 ± 0.046
1	PoDiNNs	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.001 ± 0.000
1	DINEs (soft)†	0.270 ± 0.160	0.548 ± 0.102	0.898 ± 0.022	1.000 ± 0.000
1	DINEs (ours)	0.204 ± 0.028	0.525 ± 0.031	0.860 ± 0.043	0.988 ± 0.011
2	CT-SUBNET ($n_z = 4$)	0.000 ± 0.000	0.000 ± 0.000	0.039 ± 0.011	0.681 ± 0.036
2	CT-SUBNET ($n_z = 8$)	0.000 ± 0.000	0.001 ± 0.000	0.096 ± 0.084	0.773 ± 0.037
2	CT-SUBNET ($n_z = 12$)	0.000 ± 0.000	0.000 ± 0.000	0.157 ± 0.023	0.759 ± 0.048
2	CT-SUBNET ($n_z = 16$)	0.000 ± 0.000	0.001 ± 0.001	0.205 ± 0.055	0.777 ± 0.033
2	CT-SUBNET ($n_z = 20$)	0.000 ± 0.000	0.002 ± 0.001	0.241 ± 0.018	0.800 ± 0.048
2	DINEs (soft)	(not converged)			
2	DINEs (ours)	0.000 ± 0.000	0.001 ± 0.000	0.711 ± 0.002	0.904 ± 0.001

Task	Model	$\varepsilon = 10^{-8}$	10^{-7}	10^{-6}	10^{-5}
3	Coupled Neural ODEs	0.000 ± 0.000	0.000 ± 0.000	0.002 ± 0.001	0.020 ± 0.005
3	DINEs (ours)	0.003 ± 0.001	0.032 ± 0.018	0.970 ± 0.068	1.000 ± 0.000
4	Coupled Neural ODEs	0.000 ± 0.000	0.001 ± 0.000	0.004 ± 0.001	0.268 ± 0.156
4	DINEs (ours)	0.002 ± 0.002	0.121 ± 0.228	0.831 ± 0.321	1.000 ± 0.000
5	DINEs (ours)	0.001 ± 0.001	0.189 ± 0.164	0.968 ± 0.050	1.000 ± 0.000

† 3 of 5 seeds did not converge; averaged over the remaining 2.

VPT for further thresholds. Table 8 reports the VPT for further thresholds ε. The superiority of DINEs over the other models is consistent.

Readout of the learned interconnection. We show examples of the learned W_f and W_e on Tasks 1 and 2 and read the interconnection from them. For each matrix, linearly independent columns are chosen as the pivot columns and moved to the front, and the matrix is brought to reduced row echelon form, in which the pivot columns form the identity matrix. Since neither system contains transformers, the nonzero coefficients should be ± 1. The coefficients of the elements whose port variables are unobserved, the resistive elements in this case, have gauge freedoms, as mentioned in Appendix D.2. These gauge freedoms are eliminated by normalizing the coefficients to ± 1. In principle, a very small coefficient could be canceled in scale by a very large flow, effort, or constitutive relation of the element (that is, the resistance) and accidentally removed; this does not happen thanks to the regularization term $\mathcal{L}_{\mathrm{var}}$ (as defined in Appendix D.2). Since the learned coefficients are continuous values, those with absolute values of at most 10^{-2} are treated as zero.

Task 1: three-way junction. The ports are ordered as $(\mathrm{C}_1, \mathrm{C}_2, \mathrm{C}_3, \mathrm{I}_1, \mathrm{I}_2, \mathrm{I}_3)$, the learned flow constraint matrix and its reduced row echelon form are

$$W_f \simeq \begin{array}{c} \begin{matrix} \mathrm{C}_1 & \mathrm{C}_2 & \mathrm{C}_3 & \mathrm{I}_1 & \mathrm{I}_2 & \mathrm{I}_3 \end{matrix} \\ \begin{bmatrix} 0.572 & -0.355 & -0.217 & 0.572 & -0.355 & -0.217 \\ -0.080 & -0.455 & 0.535 & -0.080 & -0.455 & 0.535 \end{bmatrix} \end{array},$$

$$\mathrm{rref}(W_f) \simeq \begin{array}{c} \begin{matrix} \mathrm{C}_1 & \mathrm{C}_2 & \mathrm{C}_3 & \mathrm{I}_1 & \mathrm{I}_2 & \mathrm{I}_3 \end{matrix} \\ \begin{bmatrix} 1.000 & 0.000 & -1.000 & 1.000 & 0.000 & -1.000 \\ 0.000 & 1.000 & -1.000 & 0.000 & 1.000 & -1.000 \end{bmatrix} \end{array}.$$

Substituting $f_{\mathrm{C}_i} = -\dot{q}_i$ and $f_{\mathrm{I}_i} = v_i$ into the rows gives

$$\dot{q}_i = v_i - v_J \qquad (i = 1, 2, 3), \qquad v_J := v_1 - \dot{q}_1.$$

This implies that each spring i with the velocity $\dot{q}_i$ is connected between the same junction with the velocity v_J and its own mass with the velocity v_i.

The ports are ordered as $(\mathrm{C}_3, \mathrm{I}_1, \mathrm{I}_2, \mathrm{I}_3, \mathrm{C}_1, \mathrm{C}_2)$ so that four independent columns come first, the learned effort constraint matrix and its reduced row echelon form are

$$W_e \simeq \begin{array}{c} \begin{matrix} \mathrm{C}_3 & \mathrm{I}_1 & \mathrm{I}_2 & \mathrm{I}_3 & \mathrm{C}_1 & \mathrm{C}_2 \end{matrix} \\ \begin{bmatrix} 0.456 & -0.431 & 0.282 & -0.522 & 0.365 & -0.348 \\ -0.036 & -0.371 & -0.153 & 0.359 & 0.694 & 0.476 \\ 0.336 & 0.496 & 0.674 & 0.374 & 0.214 & 0.035 \\ -0.587 & -0.312 & 0.330 & 0.354 & 0.079 & -0.563 \end{bmatrix} \end{array},$$

$$\mathrm{rref}(W_e) \simeq \begin{array}{c} \begin{matrix} \mathrm{C}_3 & \mathrm{I}_1 & \mathrm{I}_2 & \mathrm{I}_3 & \mathrm{C}_1 & \mathrm{C}_2 \end{matrix} \\ \begin{bmatrix} 1.000 & 0.000 & 0.000 & 0.000 & 1.000 & 1.000 \\ 0.000 & 1.000 & 0.000 & 0.000 & -1.000 & 0.000 \\ 0.000 & 0.000 & 1.000 & 0.000 & 0.000 & -1.000 \\ 0.000 & 0.000 & 0.000 & 1.000 & 1.000 & 1.000 \end{bmatrix} \end{array}.$$

The rows give

$$e_{\mathrm{C}_1} + e_{\mathrm{C}_2} + e_{\mathrm{C}_3} = 0, \qquad e_{\mathrm{I}_i} = e_{\mathrm{C}_i} \quad (i = 1, 2, 3).$$

This implies the force balance at the junction and the action and reaction forces between the springs C_i and their masses I_i.

Task 2: RC ladder. The flows of three of the seven resistive ports are unobserved. The ports are ordered as $(\mathrm{C}_4, \mathrm{R}_1, \ldots, \mathrm{R}_7 \mid \mathrm{Se}_1, \mathrm{Se}_2, \mathrm{C}_1, \mathrm{C}_2, \mathrm{C}_3)$, where the resistors are numbered by their ports in the learned model and the vertical line separates the pivot columns, the learned flow constraint matrix is

$$W_f \simeq \left[\begin{array}{ccccccccccccc} \mathrm{C}_4 & \mathrm{R}_1 & \mathrm{R}_2 & \mathrm{R}_3 & \mathrm{R}_4 & \mathrm{R}_5 & \mathrm{R}_6 & \mathrm{R}_7 & \mathrm{Se}_1 & \mathrm{Se}_2 & \mathrm{C}_1 & \mathrm{C}_2 & \mathrm{C}_3 \\ \hline .586 & -.173 & .446 & -.264 & .054 & -.044 & .195 & .284 & -.173 & .189 & -.120 & .195 & .340 \\ .061 & .322 & .049 & -.293 & -.308 & .238 & .477 & .117 & .006 & -.497 & -.249 & -.267 & -.180 \\ -.169 & -.271 & .032 & .041 & -.368 & .198 & -.363 & .216 & -.216 & .041 & -.150 & -.563 & .392 \\ -.209 & -.201 & .119 & .114 & .438 & -.189 & -.092 & .335 & -.488 & -.352 & -.291 & .036 & -.305 \\ -.304 & .163 & .167 & -.103 & .316 & .630 & -.032 & -.305 & -.159 & -.059 & -.142 & .276 & .355 \\ .263 & -.194 & -.565 & .360 & .266 & .282 & .261 & .292 & .122 & -.206 & .079 & -.010 & .280 \\ -.200 & .137 & -.079 & -.367 & -.034 & -.044 & -.381 & .543 & .431 & -.197 & .011 & .338 & .138 \\ .056 & -.204 & -.452 & -.426 & .153 & -.106 & -.019 & -.230 & .105 & .234 & -.640 & -.066 & -.036 \end{array}\right],$$

where R_1, R_2, and R_7 are unobserved. Its reduced row echelon form is $\mathrm{rref}(W_f) = [\, I_8 \;\; B_f \,]$ with

$$B_f \simeq \begin{array}{c} \begin{matrix} \mathrm{Se}_1 & \mathrm{Se}_2 & \mathrm{C}_1 & \mathrm{C}_2 & \mathrm{C}_3 \end{matrix} \\ \begin{bmatrix} 1.000 & 1.000 & 0.998 & 0.996 & 0.994 \\ 1.426 & 0.008 & 1.425 & 1.426 & 0.004 \\ -0.708 & -0.001 & 0.002 & 0.003 & 0.000 \\ 0.001 & 0.002 & 0.995 & 0.006 & 0.006 \\ -0.002 & -0.002 & -0.004 & 1.000 & -0.005 \\ -0.003 & -0.004 & -0.001 & -0.004 & 0.992 \\ -1.003 & -1.003 & -1.003 & -0.997 & -1.001 \\ -0.004 & -0.701 & -0.003 & -0.004 & -0.001 \end{bmatrix} \end{array}.$$

Omitting the small coefficients and scaling the remaining coefficients as ± 1 according to their signs gives

$$\widehat{B}_f = \begin{array}{c} \begin{matrix} \mathrm{Se}_1 & \mathrm{Se}_2 & \mathrm{C}_1 & \mathrm{C}_2 & \mathrm{C}_3 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 \end{bmatrix} \end{array}.$$

Each row represents a Kirchhoff current law at a node of the circuit. With the currents $I_{\mathrm{E}_1} = f_{\mathrm{Se}_1}$, $I_{\mathrm{E}_2} = f_{\mathrm{Se}_2}$, and $I_{\mathrm{C}_i} = -f_{\mathrm{C}_i}$, the rows give

$$\begin{gathered} f_{\mathrm{R}_1} = -I_{\mathrm{E}_1} + I_{\mathrm{C}_1} + I_{\mathrm{C}_2}, \qquad f_{\mathrm{R}_2} = I_{\mathrm{E}_1}, \qquad f_{\mathrm{R}_7} = I_{\mathrm{E}_2}, \\ f_{\mathrm{R}_3} = I_{\mathrm{C}_1}, \qquad f_{\mathrm{R}_4} = I_{\mathrm{C}_2}, \qquad f_{\mathrm{R}_5} = I_{\mathrm{C}_3}, \qquad f_{\mathrm{R}_6} = I_{\mathrm{C}_4}, \\ I_{\mathrm{E}_1} + I_{\mathrm{E}_2} = I_{\mathrm{C}_1} + I_{\mathrm{C}_2} + I_{\mathrm{C}_3} + I_{\mathrm{C}_4}. \end{gathered}$$

The ports are ordered as $(\mathrm{Se}_1, \mathrm{Se}_2, \mathrm{C}_1, \mathrm{C}_2, \mathrm{C}_3 \mid \mathrm{C}_4, \mathrm{R}_1, \ldots, \mathrm{R}_7)$, the learned effort constraint matrix is

$$W_e \simeq \left[\begin{array}{ccccc|ccccccc} \mathrm{Se}_1 & \mathrm{Se}_2 & \mathrm{C}_1 & \mathrm{C}_2 & \mathrm{C}_3 & \mathrm{C}_4 & \mathrm{R}_1 & \mathrm{R}_2 & \mathrm{R}_3 & \mathrm{R}_4 & \mathrm{R}_5 & \mathrm{R}_6 & \mathrm{R}_7 \\ \hline -.031 & -.415 & -.085 & -.146 & .485 & .194 & .374 & -.021 & .083 & .147 & -.484 & -.192 & -.292 \\ -.182 & .290 & -.306 & .080 & -.204 & .320 & .580 & -.128 & .305 & -.082 & .203 & -.323 & .202 \\ .099 & -.429 & -.157 & .334 & -.179 & .333 & -.389 & .069 & .156 & -.336 & .178 & -.335 & -.300 \\ .179 & .103 & -.462 & .295 & .230 & -.343 & -.020 & .127 & .456 & -.296 & -.226 & .343 & .072 \\ -.611 & -.021 & .201 & .389 & .135 & -.091 & .029 & -.434 & -.203 & -.389 & -.135 & .091 & -.015 \end{array}\right],$$

and its reduced row echelon form is $\mathrm{rref}(W_e) = [\, I_5 \;\; B_e \,]$ with

$$B_e \simeq \begin{array}{c} \begin{matrix} \mathrm{C}_4 & \mathrm{R}_1 & \mathrm{R}_2 & \mathrm{R}_3 & \mathrm{R}_4 & \mathrm{R}_5 & \mathrm{R}_6 & \mathrm{R}_7 \end{matrix} \\ \begin{bmatrix} -1.000 & -1.426 & 0.708 & -0.001 & 0.002 & 0.003 & 1.003 & 0.004 \\ -1.000 & -0.008 & 0.001 & -0.002 & 0.002 & 0.004 & 1.003 & 0.701 \\ -0.998 & -1.425 & -0.002 & -0.995 & 0.004 & 0.001 & 1.003 & 0.003 \\ -0.996 & -1.426 & -0.003 & -0.006 & -1.000 & 0.004 & 0.997 & 0.004 \\ -0.994 & -0.004 & 0.000 & -0.006 & 0.005 & -0.992 & 1.001 & 0.001 \end{bmatrix} \end{array}.$$

Omitting the small coefficients and scaling the remaining coefficients as ± 1 according to their signs gives

$$\widehat{B}_e = \begin{array}{c} \begin{matrix} \mathrm{C}_4 & \mathrm{R}_1 & \mathrm{R}_2 & \mathrm{R}_3 & \mathrm{R}_4 & \mathrm{R}_5 & \mathrm{R}_6 & \mathrm{R}_7 \end{matrix} \\ \begin{bmatrix} -1 & -1 & 1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ -1 & -1 & 0 & -1 & 0 & 0 & 1 & 0 \\ -1 & -1 & 0 & 0 & -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \end{bmatrix} \end{array}.$$

Each row represents a Kirchhoff voltage law around an independent loop of the circuit. We can confirm $\widehat{B}_e = -\widehat{B}_f^{\top}$. The rows give two common voltages

$$\begin{aligned} V_L &:= e_{\mathrm{Se}_1} + e_{\mathrm{R}_2} = e_{\mathrm{C}_1} - e_{\mathrm{R}_3} = e_{\mathrm{C}_2} - e_{\mathrm{R}_4}, \\ V_R &:= e_{\mathrm{Se}_2} + e_{\mathrm{R}_7} = e_{\mathrm{C}_3} - e_{\mathrm{R}_5} = e_{\mathrm{C}_4} - e_{\mathrm{R}_6}, \end{aligned}$$

and $e_{\mathrm{R}_1} = V_L - V_R$.

From these relations, we can reconstruct the circuit diagram of the RC ladder, including the three resistors whose roles in the interconnection were unknown.