University of Dundee

GINNs

Hall, Eric J.; Taverniers, Søren; Katsoulakis, Markos A.; Tartakovsky, Daniel M.

GINNs: Graph-Informed Neural Networks for Multiscale Physics

Eric J. Hall[a,1,*], Søren Taverniers[b,1], Markos A. Katsoulakis[c], Daniel M. Tartakovsky[b,*]

[a]*Division of Mathematics, University of Dundee, Dundee, DD1 4HN, UK*
[b]*Department of Energy Resources Engineering, Stanford University, Stanford, CA 94305, USA*
[c]*Department of Mathematics and Statistics, University of Massachusetts Amherst, Amherst, MA 01003, USA*

Abstract

We introduce the concept of a Graph-Informed Neural Network (GINN), a hybrid approach combining deep learning with probabilistic graphical models (PGMs) that acts as a surrogate for physics-based representations of multiscale and multiphysics systems. GINNs address the twin challenges of removing intrinsic computational bottlenecks in physics-based models and generating large data sets for estimating probability distributions of quantities of interest (QoIs) with a high degree of confidence. Both the selection of the complex physics learned by the NN and its supervised learning/prediction are informed by the PGM, which includes the formulation of structured priors for tunable control variables (CVs) to account for their mutual correlations and ensure physically sound CV and QoI distributions. GINNs accelerate the prediction of QoIs essential for simulation-based decision-making where generating sufficient sample data using physics-based models alone is often prohibitively expensive. Using a real-world application grounded in supercapacitor-based energy storage, we describe the construction of GINNs from a Bayesian network-embedded homogenized model for supercapacitor dynamics, and demonstrate their ability to produce kernel density estimates of relevant non-Gaussian, skewed QoIs with tight confidence intervals.

Keywords: Deep learning; Surrogate model; Bayesian network; Probabilistic Graphical Model (PGM); Uncertainty propagation; Electrical Double Layer Capacitor (EDLC);

1. Introduction: decision-making using physics-based models and surrogates

Modeling and simulation of complex nonlinear multiscale and multiphysics systems requires the inclusion and characterization of uncertainties and errors that enter at various stages of the computational workflow. Typically this requires casting the original *deterministic* physics-based model into a *probabilistic* framework where inputs or control variables (CVs) are treated as random variables with probability distributions derived from available experimental data, manufacturing constraints, design criteria, expert judgment, and/or other domain knowledge (e.g., see [1]). Running the physics-based model with CVs sampled according to these distributions yields corresponding realizations of the system response as characterized by quantities of interest (QoIs). Analysis of the uncertainty propagation from the CVs to the QoIs informs decision-making, e.g., it informs engineering decisions aimed at improving the quality and reliability of designed products and helps identify potential risks at early stages in the design and manufacturing process.

Quantitatively assessing uncertainty propagation presents a fundamental challenge due to the computational cost of the underlying physics-based model. Even for a low number of CVs and QoIs, uncertainty quantification (UQ) for, e.g., accelerating the simulation-aided design of multiscale systems and data-centric engineering tasks more generally ([2]), requires a large number of repeated observations of QoIs to achieve a high degree of confidence in such an analysis. The sampling cost is further exacerbated in real-world applications where distributions on QoIs are typically non-Gaussian, skewed, and/or mutually correlated, and therefore need to be characterized by their full probability density function (PDF) rather than through summary statistics such as mean and variance. The computational cost of nonparametric methods to estimate these densities can become prohibitively high when using a fully-featured physics-based model to compute each sample.

One approach to alleviate the computational burden is to derive a cheaper-to-compute *surrogate* for the physics-based model's response enabling much faster generation of output data and thus overcoming computational bottlenecks. Also known as metamodels, emulators, or response surfaces/hypersurfaces, such data-driven surrogate

---

[*]Corresponding authors

Email addresses: `ehall001@dundee.ac.uk` (Eric J. Hall), `tartakovsky@stanford.edu` (Daniel M. Tartakovsky)
[1]Both authors contributed equally to this work.

models are statistical models emulating the system response (e.g., [3, 4]). Their accuracy and fidelity depends on a number of factors including the amount of physics-based model run data available for "training" and how the corresponding inputs are selected in the parameter space; they differ from reduced-order model and model hierarchy surrogates which attempt to capture a simplified or lower-fidelity representation of the physical system. However, all surrogates are unified in their aim: computationally cheaper predictions of the response. The use of surrogates in lieu of physics-based representations paves the way for data-driven UQ including sensitivity studies or model calibration with tight confidence intervals for the estimated metrics.

A plethora of surrogate modeling techniques have been developed for physics-based modeling and simulation including statistical learning of coarse-grained models ([5–7]), radial basis function networks ([8, 9]), space mapping ([9]), kriging ([8–14]), polynomial chaos ([15–17]), and neural networks (NNs) (including early works [18–21] and more recent works that take advantage of modern advances in computing, e.g., [22–25]). Machine learning approaches such as NNs and deep NNs, i.e., NNs that contain multiple hidden layers between their input and output layers, have received significant attention in recent years thanks to, in part, the advent of off-the-shelf software like TensorFlow [26] and PyTorch [27], which automate the computation of the training loss function gradient via back-propagation [28] (a special case of reverse mode automatic differentiation) and tremendously simplify NN design and implementation.

We establish a framework for constructing domain-aware surrogate models, built via deep learning, to support simulation-based decision-making in complex multiscale systems. A fundamental difference between simulation-based decision-making and other scenarios where deep learning is typically used is that in the former setting the user drives the data generation process. With this insight, we deploy a deep learning approach that incorporates well-known strategies from stochastic and predictive modeling in the following way. First, we embed a probabilistic graphical model (PGM) into the physics-based representation to encode complex dependencies among model variables that arise from domain-specific information and to enable the generation of physically sound distributions. Second, from the embedded PGM we identify computational bottlenecks intrinsic to the underlying physics-based model and replace them with a NN. The graph, i.e., PGM, informs (i) the selection of the complex physics that the NN learns, (ii) the supervised learning of the NN, and (iii) the predictions made by the trained NN. These last two features are facilitated through the use of structured priors on CVs that serve as inputs to the NN, as highlighted in Fig. 3, and that differ from the typical use of independent input layer nodes. We refer to the resulting hybrid PGM/NN surrogate as a "Graph-Informed Neural Network" (GINN) as the supervised learning and predictions are guided by the PGM. The PGM-embedded physics-based model yields a domain-aware surrogate and also lends interpretation to the GINN's predictions.

Related to but different from our GINN approach, two main paradigms have emerged with respect to the use of deep NNs for building surrogates of physics-based models described by partial differential equations (PDEs): physics-informed NNs (PINNs) [25, 29–32] and "data-free" physics-constrained NNs [24, 33–35]. Both approaches drive supervised learning by enforcing physical constraints. While PINNs include both the governing PDE and its initial/boundary conditions in the training loss function, physics-constrained NNs enforce the initial/boundary conditions through a bespoke NN architecture while encoding the PDE in the training loss. In contrast, GINNs use simulation data from a domain-aware model without modifying the training loss function. This facilitates their deployment in complex problems that involve a system of PDEs or differential equations and additional constraints of various types. Overfitting in a GINN is controlled using standard non-intrusive and readily available techniques, such as testing the NN on unseen data (i.e., data independent of the training samples) and using ℓ_1 (lasso regression) or ℓ_2 (ridge regression) regularization ([36]).

Further, while GINNs are informed by a graph (e.g., a PGM), they are not simply graph NNs (e.g., [37, 38]). A typical application of graph NNs is to use a NN for node classification tasks, i.e., deciding how to label nodes of a given graph from available labeled data on the remaining nodes. In our approach, the PGM is used to build a domain-aware physics-based model, thus synthesizing stochastic and multiscale modeling. Then computational bottlenecks in the PGM are identified and replaced by a NN whose supervised learning and prediction are further informed by the PGM, e.g., through structured priors on CVs.

Our methods are general and can be applied to a wide range of complex models with intrinsic computational bottlenecks. However, here we showcase the GINN approach via a real-world application of interest in energy storage. The optimal and robust design of electrical double layer (EDL) supercapacitors for use in long-term energy storage devices critically relies on the multiscale modeling of novel nanoporous metamaterials. The presence of nonlinear multiscale physics in this complex system translates into nontrivial correlations both across and within problem scales and necessitates the use of structured probabilistic models (like PGMs) to describe the dependencies among the model variables in order to maintain physically sound distributions. Using a homogenized model of an EDL supercapacitor ([39]) as a computational testbed, we derive a domain-aware physics-based model following

the Bayesian Network (BN) PDE framework developed in [16]. The BN PDE model[2] for supercapacitor dynamics, formulated in Section 2, is then used to train a GINN surrogate in Section 3. The GINN replaces the expensive computation of intermediate variables by learned features in its hidden layers. Hence, the GINN replaces random PDE mappings from CVs to QoIs with a NN surrogate model that is domain-aware and physics-informed. The GINN surrogate can then be leveraged to make sufficiently many predictions to quantify uncertainties with a high degree of statistical confidence, as described in Section 4 for nonparametric kernel density estimation of QoIs. Finally, Section 5 is reserved for conclusions and follow-up work.

2. Domain-aware physics-based models: the BN PDE

2.1. Motivation for the use of structured probabilistic models

The rigorous homogenization in [39] enables the derivation of macroscopic quantities from microscale counterparts with clearly defined limits of applicability, in contrast to relying on phenomenological relations. While it may be possible to minimize the computational burden with an appropriately chosen simulation technique, such as a multilevel Monte Carlo method as in [40], the homogenization and solution of the corresponding closure equations are an *integral feature* of this multiscale physics-based model and thus the associated computational bottleneck is intrinsic. Moreover, the complicated dependencies among the components, some of which are viewed as CVs for the QoIs, demand specialized tools, such as PGMs, to recast the physics-based model into a probabilistic framework.

Since their introduction, PGMs have proven to be a fundamental mathematical concept for modeling uncertainty in artificial intelligence [41, 42] and machine learning [43]. BNs are a class of PGMs that can be represented by a directed acyclic graph with nodes representing random variables and edges representing conditional dependencies. The directed nature of BNs makes describing dependencies intuitive and is therefore well-suited to physics-based modeling. Such structured probabilistic models are necessary in the context of complex systems as, e.g., independent selection of CVs will often lead to non-physical predictions. In [16], a stochastic modeling framework is presented for embedding BNs into physics-based models. The resulting BN PDEs are random PDEs that incorporate a BN, thereby encoding correlations into stochastic models and providing a platform for uncertainty propagation. More specifically, BN PDEs use the hierarchical structure of BNs to bring together both statistical and multiscale mathematical modeling in a systematic way by informing the physics-based model with domain knowledge including available data, which are typically sparse or incomplete, along with expert opinion, engineering design constraints, and dependencies between CVs. For the problem of interest, a BN encoding the supercapacitor dynamics is shown in Fig. 1.

2.2. Formal definition of a BN

The BN PDE framework centers around constructing a structured probabilistic model for the joint probability density function (PDF) of model variables that captures correlations and constraints among variables in a systematic way. Formally, a BN is defined as a probability model, i.e., probability distribution,

$$P(\boldsymbol{Z} \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} P(Z_i \mid \mathrm{Pa}_{Z_i}, \theta_{Z_i \mid \mathrm{Pa}_{Z_i}}), \tag{1}$$

for random variables $\boldsymbol{Z} = \{Z_1, \ldots, Z_n\}$ where Pa_{Z_i} is the set of parents of each Z_i and $\boldsymbol{\theta} = \{\theta_{Z_i \mid \mathrm{Pa}_{Z_i}}\}_{i=1,\ldots,n}$ are statistical model hyperparameters for each conditional probability distribution (CPD) $P(Z_i \mid \mathrm{Pa}_{Z_i})$. Here, we assume without loss of generality that the hyperparameters are independent (if not, they can instead be treated as variables). BNs enable the modeling of large and complex joint distributions containing correlations and the identification of conditionally independent variables significantly reduces the overall dimensionality of (1) thus making parameter inferences from data feasible.

2.3. Choice of tunable control variables and identification of computational bottlenecks

We begin by recalling the physical parameters of interest, and their roles, for the dynamics of an EDL supercapacitor described in [39]; for reproducibility and to provide a self-contained work, the model equations are detailed in Appendix A. At the macro or continuum scale, the effective ion diffusion coefficients D_+^{eff} and D_-^{eff} (scalar quantities in (A.7) arising from (A.1)) are used to compute the effective electrolyte conductivity κ^{eff} in (A.9)

---

[2]We refer to our derived model as a BN PDE since the principal variables involved in the computational bottlenecks are governed by random PDEs.
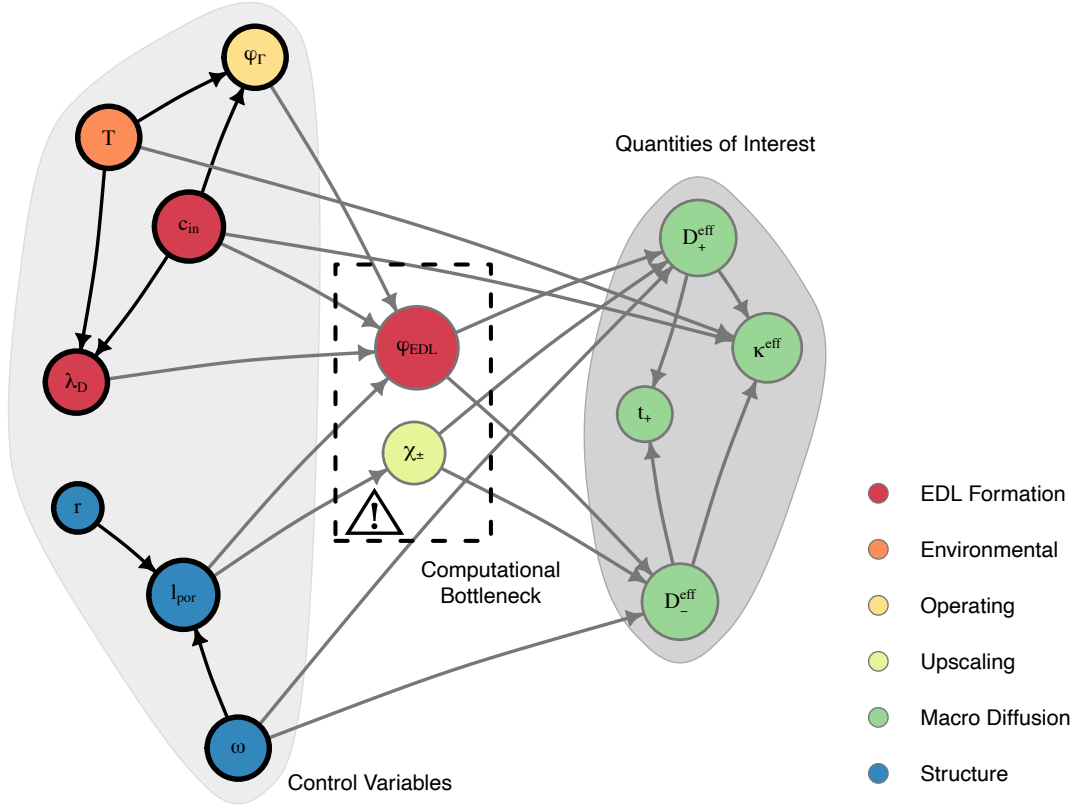
Figure 1: A Bayesian Network (BN), a particular type of PGM, describing supercapacitor dynamics in a nanoporous material encodes conditional relationships for key variables (see Appendix A and Fig. A.8). The BN enables the systematic and intuitive inclusion of domain knowledge into the stochastic multiscale model and ensures the resulting BN PDE makes physically sound predictions. PGMs guide the supervised learning of GINNs to overcome intrinsic computational bottlenecks in the physics-based model (cf. see Fig. 3 and Fig. 4).

and transference number t_+ in (A.10). Ultimately, these four QoIs are used to inform the state of EDL cells in (A.11). The macroscale QoIs depend on microscale parameters, including the solid radius r and pore throat size l_{por} of the nanoporous structure (that are also related to the material porosity ω, see Fig. A.8), via a deterministic homogenization (upscaling) with closure $\chi_\pm$ in (A.3). Additionally, the QoIs depend on the temperature T, initial ion concentration c_{in}, fluid-solid interface potential φ_Γ in (A.6), Debye length λ_D in (A.5), and EDL potential φ_{EDL} in (A.4).

Based on the aforementioned parameters (i.e., **??** A.1–A.6 and (A.8) to (A.10)), we select thirteen variables,

$$\boldsymbol{Z} := \{\boldsymbol{X}, \boldsymbol{Y}\} = \{X_{\varphi_\Gamma}, X_{c_{\mathrm{in}}}, X_T, X_\omega, X_{l_{\mathrm{por}}}, X_r, X_{\varphi_{\mathrm{EDL}}}, X_{\lambda_D}, X_{\boldsymbol{\chi}_\pm}, Y_{D_+^{\mathrm{eff}}}, Y_{D_-^{\mathrm{eff}}}, Y_{\kappa^{\mathrm{eff}}}, Y_{t_+}\}, \tag{2}$$

where for simplicity of notation we will use labels instead of indices as in (1) when no confusion arises. The variables

$$\boldsymbol{Y} := \{Y_{D_+^{\mathrm{eff}}}, Y_{D_-^{\mathrm{eff}}}, Y_{\kappa^{\mathrm{eff}}}, Y_{t_+}\}, \tag{3}$$

represent QoIs that correspond to macroscopic diffusion quantities that parametrize models characterizing the behavior of EDLC cells (cf. Appendix A.3). The variables

$$\boldsymbol{X} := \{X_{\varphi_\Gamma}, X_{c_{\mathrm{in}}}, X_T, X_\omega, X_{l_{\mathrm{por}}}, X_r, X_{\lambda_D}, X_{\boldsymbol{\chi}_\pm}, X_{\varphi_{\mathrm{EDL}}}\}, \tag{4}$$

associated, respectively, with electrode surface (fluid-solid interface) potential, initial ion concentration, temperature, porosity, (half) pore throat size, solid radius, Debye length, (upscaling) closure variables, and EDL potential, correspond to both independent and dependent inputs as well as solutions to physical model equations. In particular, the variables

$$\boldsymbol{X}_b := \{X_{\boldsymbol{\chi}_\pm}, X_{\varphi_{\mathrm{EDL}}}\} \subset \boldsymbol{X}, \tag{5}$$

4

represent solutions to random PDEs, i.e., the PDEs in (A.3) and (A.4) with random coefficients and/or boundary conditions. These variables correspond to computationally intensive portions of the physics-based model and therefore constitute a computational *bottleneck* for UQ. We investigate the remaining seven variables in (4) as tunable CVs,

$$\boldsymbol{X}_c := \{X_{\varphi_\Gamma}, X_{c_{\text{in}}}, X_T, X_\omega, X_{l_{\text{por}}}, X_r, X_{\lambda_D}\} \subset \boldsymbol{X}\,, \tag{6}$$

related to the engineering design process and experimental conditions.

We cast the (originally deterministic) homogenized problem into a probabilistic framework by modeling the CVs $\boldsymbol{X}_c$ as random variables, see e.g. approach followed for a similar problem in [40]. The type and support of the distributions placed on $\boldsymbol{X}_c$ need to reflect a combination of expert opinion, available data, physical and design constraints, and other domain knowledge, in order to ensure the generation of physically meaningful distributions on the QoIs $\boldsymbol{Y}$. While equally valid alternative choices can be made, we select the CVs X_T, $X_{c_{\text{in}}}$, X_r, and X_ω to be independent and assume the prior distributions on them to be uniform on an interval of $\pm 35\%$ (for X_T and $X_{c_{\text{in}}}$) or $\pm 25\%$ (for X_r and X_ω) around a physically relevant baseline value (see Table 1). That is, each of these variables is uniform,

$$X_i \mid \theta_i \sim \text{Uniform}([\theta_i^{\min}, \theta_i^{\max}])\,, \tag{7}$$

where the hyperparameters $\theta_i = \{\theta_i^{\min}, \theta_i^{\max}\}$ represent the minimum and maximum values that are endpoints of the support intervals.

Table 1: Independent CVs distributed according to (7) over the physically relevant ranges, i.e., the hyperparameters $\theta^{\min}$ and $\theta^{\max}$, that are selected using expert knowledge and available experimental data (cf. dependent inputs (8) and in Fig. 5).

Variable label	$\theta^{\min}$	$\theta^{\max}$	Mean/Baseline	Variation	Units
T	208	432	320	$\pm 35\%$	K
c_{in}	0.52	1.08	0.80	$\pm 35\%$	mol/l
r	1.05	1.75	1.40	$\pm 25\%$	nm
ω	0.5025	0.8375	0.6700	$\pm 25\%$	-

The distributions of the CVs X_{λ_D}, X_{φ_Γ}, and $X_{l_{\text{por}}}$ are then determined by their relation to these independent inputs, captured by (A.5), (A.6), and (A.8), and the uniform distributions (7). It follows that the conditional distributions on these variables,

$$P(X_{\lambda_D} \mid X_T, X_{c_{\text{in}}}, \theta_T, \theta_{c_{\text{in}}})\,, \tag{8a}$$

$$P(X_{\varphi_\Gamma} \mid X_T, X_{c_{\text{in}}}, \theta_T, \theta_{c_{\text{in}}})\,, \tag{8b}$$

$$P(X_{l_{\text{por}}} \mid X_\omega, X_r, \theta_\omega, \theta_r)\,, \tag{8c}$$

are nontrivial. For example, X_{φ_Γ} in (8b) depends on both X_T and $X_{c_{\text{in}}}$ as the transcendental equation for φ_Γ,

$$
\begin{aligned}
\varphi_\Gamma &= \frac{V}{2} - \varphi_{\text{ecm}} - \frac{\sigma}{C_{\text{H}}}\,, \\
\sigma &= \sqrt{4\mathcal{E}RTz^2 c_{\text{in}}} \sqrt{\cosh\left(\frac{\mathrm{e}\varphi_\Gamma}{k_{\text{B}}T}\right) - \cosh\left(\frac{\mathrm{e}\varphi_{\min}}{k_{\text{B}}T}\right)}\,,
\end{aligned}
\tag{9}
$$

depends on both T and c_{in} (we refer to (A.6) in Appendix A for a detailed discussion of (9)). Fig. 2 displays slices (averages) from the empirical conditional PDF corresponding to (8b) based on $M = 10^7$ observations.

These conditional dependencies do not necessarily represent causal relationships. For example, while ω is an emergent property of r and l_{por}, we treat ω and r as independent CVs and which forces us to make l_{por} a dependent input (Fig. A.8). Inspired by the recent work in [44], this choice allows us to explore numerically a broad range of porosities that was guided by, but not limited to, values in the literature based on prior experiments. In more general terms, such an approach allows parameters for which data are missing over certain ranges to be systematically incorporated and combined with real data through the priors. This being said, investigating the additional inclusion of causal reasoning and its implications for causal inference [45] are a natural extension of this line of research.

Finally, we divide the CVs into subgroups associated with environmental conditions $\{X_T\}$, operating conditions $\{X_{\varphi_\Gamma}\}$ (as (A.6) depends on the externally applied voltage), structural constraints $\{X_{l_{\text{por}}}, X_\omega, X_r\}$, EDL formation $\{X_{\varphi_{\text{EDL}}}, X_{\lambda_D}, X_{c_{\text{in}}}\}$, and upscaling/homogenization $\{X_{\boldsymbol{\chi}_\pm}\}$. Along with the macroscopic diffusion QoIs $\boldsymbol{Y}$, these groupings attach additional layers of significance to the underlying probabilistic model that aid in interpretation
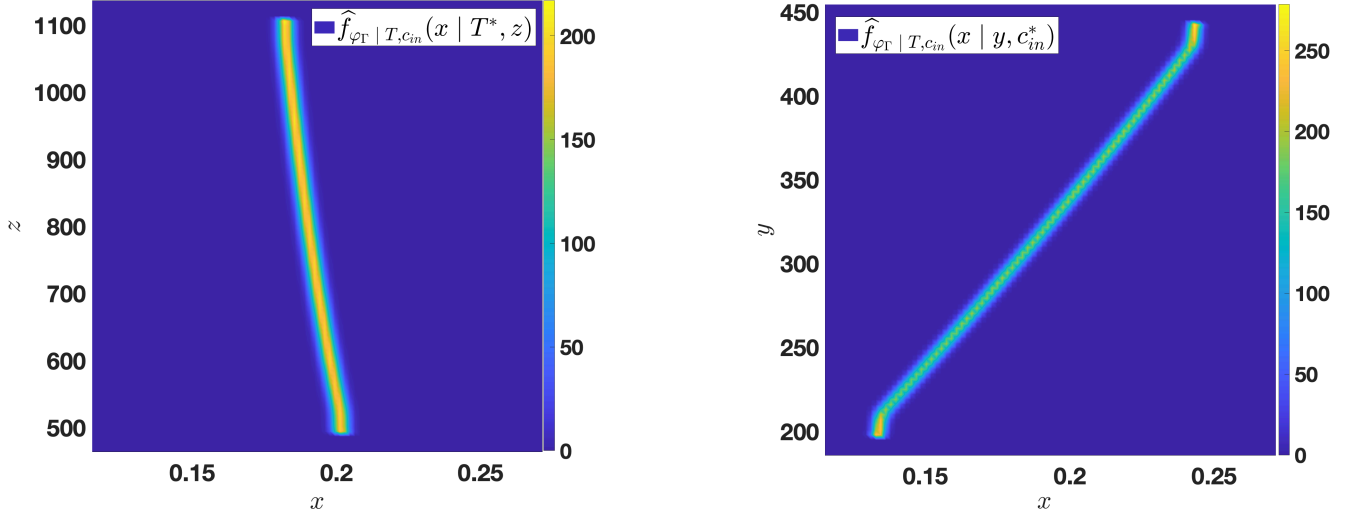
Figure 2: Estimated conditional PDFs, based on $M = 10^7$ samples using kernel density estimation techniques (see (10)), for the dependent CV X_{φ_Γ} in (8b) that is constrained by the nonlinear transcendental equation (9) describing its relation to other CVs. From left to right, we show a view of two-dimensional slices of the conditional PDFs, in this case a function of three variables, along T^*, the mean value of X_T, and along c_{in}^*, the mean value of $X_{c_{\text{in}}}$. The conditional densities (8a) and (8c) can be estimated similarly.

and are not necessarily unique. The BN Fig. 1 encodes conditional relationships both between problem scales, via the rigorous pore-to-Darcy scale homogenization (e.g., between macroscopic variables, closure variables, and microscopic structural features), and within single problem scales, such as the geometry and topology of the pore structure.

2.4. BN PDE for supercapacitor dynamics

The joint PDF on all model variables in (2) represents the underlying probabilistic model for our application of interest. Using the distributions on the CVs (7) and (8), which we refer to as structured priors, we decompose the probabilistic model according to (1) to arrive at the BN for supercapacitor dynamics, visualized in Fig. 1. This allows us to formally propagate uncertainty from $\boldsymbol{X}_c$ via $\boldsymbol{X}_b$ to $\boldsymbol{Y}$ following the relationships in Fig. 1. For our application of interest this involves solving a chain of transcendental and algebraic equations, to obtain the dependent CV values, and BN PDEs, associated with the computational bottleneck $\boldsymbol{X}_b$ in our physics-based model. A visual representation of these steps (to model and propagate uncertainty using the BN and BN PDE) is included in the flowchart for the global GINN algorithm Fig. 4. We shall observe, in Fig. 6 in Section 4, that the marginal densities of QoIs $\boldsymbol{Y}$ are skewed and non-Gaussian. Moreover, continuous densities are required for downstream computations related to the EDLC cell state model in (A.11). Therefore density estimation is the appropriate tool for a corresponding UQ analysis. However, as this requires a large volume of simulations of the physics-based model, in the next section we develop an appropriate surrogate model to accomplish this task.

3. GINN surrogates for complex systems

Predicting macroscopic QoIs with confidence requires repeated solves of the physics-based model. On the one hand, mathematical homogenization is a central feature of the multiscale model that enables the rigorous propagation of uncertainty using the framework in Section 2. On the other hand, this upscaling is associated with computational bottlenecks that limit our ability to generate sufficiently many realizations of the physics-based model and therefore to carry out a subsequent UQ analysis. Arguably, upscaled models represent the "best-case scenario" as direct simulation of the microscale physics is even more computationally demanding in most applications. To address this challenge, we formulate a domain-aware GINN surrogate model suitable for complex systems.

A GINN is essentially a hybrid PGM/NN model that replaces nodes in the PGM associated with computationally expensive solves, e.g., corresponding to the homogenization procedure and closure problem, with a deep NN. In the application of interest, rather than solving the computational bottleneck $\boldsymbol{X}_b$ (5) directly using the physics-based model, the GINN learns these intermediate variables as hidden layer features (see Fig. 3) which, at least conceptually, mirror the boxed nodes of the BN highlighted in Fig. 1. While replacing nodes in the BN with learned features removes the possibility of making interventions on them, importantly the CVs $\boldsymbol{X}_c$ serve as inputs to the GINN
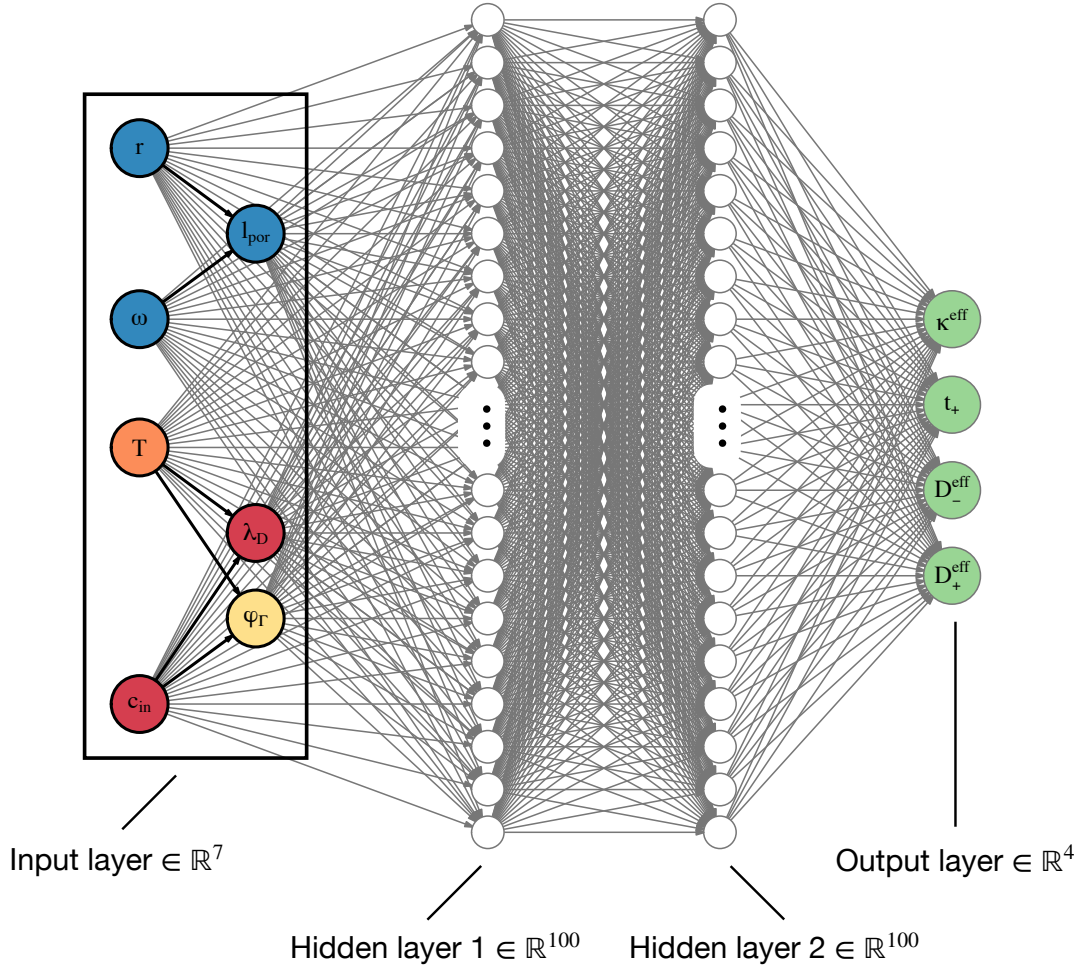
Figure 3: The GINN surrogate for the multiscale model of supercapacitor dynamics. Trained using data simulated with the BN PDE (see Section 2 and Fig. 4), the GINN takes domain-aware graphs as inputs (i.e., structured priors on CVs in Fig. 1) and predicts output QoIs that bypass computational bottlenecks (highlighted in Fig. 1) using a deep NN.

rather than being learned features and therefore can still be tuned. Notably, as expressed through the structured priors of the graphical model in Fig. 1, some of these CVs are mutually correlated, while typically the input layer nodes of NNs are independent. Failure to account for these dependencies would lead to nonphysical distributions for the CVs and hence also for the QoIs.

In lieu of replacing $X_{\varphi_{\mathrm{EDL}}}$ and $X_{\chi_{\pm}}$ in Fig. 1 with a single NN surrogate, our PGM-based representation also accommodates the use of separate PINNs for each of these computational bottlenecks. Likewise, other deep learning approaches, e.g., [24, 46, 47], might also be integrated if appropriate for problems where the stochastic dimension is high. While recent work on benchmark problems showed that PINNs can be successfully incorporated into a multifidelity framework where they are trained on both high- and low-fidelity data ("multifidelity PINNs" in [32]), it remains an open question whether replacing multiple systems of PDEs with PINNs in our application of interest yields a tractable multifidelity approach. By retaining the domain knowledge and correlations between CVs in the input layer of the NN, GINNs decouple optimizing the training loss function from encoding relevant physics into the NN and therefore can be easily cast into a multifidelity framework such as the one proposed by [48].

A modeling choice was made to have the GINN predict all four QoIs simultaneously (compare the "flattened" output layer in Fig. 3 with the nontrivial dependencies in Fig. 1) rather than through a separate post-processing step, however the original structure could also be retained. Anecdotally, the discrepancy between the learned samples of $Y_{\kappa^{\mathrm{eff}}}$ and Y_{t_+} and their counterparts computed by post-processing appears typically to be of $\mathcal{O}(10^{-3})$, which is small compared to other errors and uncertainties considered.

To train the GINN, we obtain simulation data $(\boldsymbol{X}_c, \boldsymbol{Y})$ comprised of input-output (io) pairs of CVs $\boldsymbol{X}_c$ and the resultant QoIs $\boldsymbol{Y}$ using the framework in Section 2. The CVs for our training data are sampled according to the structured prior distributions in (7) and (8) (see also BN in Fig. 1). These io pairs are then used to train the

GINN, depicted in Fig. 3, using supervised learning. Specifically, the workflow to construct the GINN consists of the following steps which correspond to the numbered boxes in the flowchart for the global GINN algorithm Fig. 4.

1. **Generating data (BN PDE):** We draw $N_{\text{sam}} = 10^4$ input samples from the structured priors on CVs $\boldsymbol{X}_c$ in (6) (see Fig. 1) and produce N_{sam} corresponding samples of the QoIs $\boldsymbol{Y}$ in (3) using the BN PDE model. These simulations are performed using a co-simulation framework of COMSOL Multiphysics® and MATLAB®. This procedure results in N_{sam} io pairs $(\boldsymbol{X}_c, \boldsymbol{Y})$ of simulated data, which we then divide into training and test sets of size $N_{\text{train}} = 0.8 N_{\text{sam}}$ and $N_{\text{test}} = 0.2 N_{\text{sam}}$, respectively.

2. **Training:** We train the GINN on the training data set through supervised learning. First, we normalize the inputs and outputs to lie on the interval $[-1, 1]$ (for the independent CVs) or $[0, 1]$ (for the dependent CVs and the outputs) as this increases the rate of decay of the training loss function (mean squared error) as a function of the number of Epochs (one Epoch corresponds to seeing all the training data once). Next, we define a fully connected NN using Google's TensorFlow 2 software ([26]) consisting of:

   (i) an input layer comprised of the 7 CVs,
   (ii) two hidden layers each comprised of 100 neurons, and
   (iii) an output layer comprised of the 4 QoIs,

   where all neurons are activated using the ReLU (Rectified Linear Unit) activation function. The structure of the GINN specified above is visualized in Fig. 3. Finally, we train the GINN on the N_{train} io pairs using 50 Epochs, which was sufficient to yield a training loss Err_{train} below a prespecified tolerance TOL of $\mathcal{O}(10^{-4})$.

3. **Testing:** This step deals with improving the generalization capability of the model, i.e., avoiding overfitting, by validating that it makes suitable predictions on unseen test data. This is done by computing the test loss function (also mean squared error) using the N_{test} test io pairs. If the test loss Err_{test} is comparable to the training loss, then we proceed to use the trained GINN to predict new io sample pairs.

4. **Predicting:** We generate $N_{\text{sam}}^{\text{pred}}$ inputs for the GINN by sampling from the structured priors, i.e., the same distributions as those used to generate the training and test samples. Using the trained GINN, we obtain $N_{\text{sam}}^{\text{pred}}$ output samples, each consisting of (normalized) values for the four output quantities, which are then de-normalized to obtain the final predicted QoIs.

Simulation of $N_{\text{sam}} = 10^4$ io pairs with the BN PDE model using the co-simulation framework (described in Appendix A) on an Ubuntu system with 8 cores (16 hyperthreads) running at 2.60 GHz and having 64 GB of RAM, given a typical time of 20 seconds per run, takes 3326.4 minutes. Learning the optimal parameter values of the GINN using $N_{\text{train}} = 8 \times 10^3$ training io pairs and $N_{\text{test}} = 2 \times 10^3$ test io pairs, and generating $N_{\text{sam}}^{\text{pred}} = 10^7$ new io pairs using the trained GINN takes about 5 minutes on a 16-inch MacBook Pro (MBP) with 8 cores running at a little under 4 GHz and having 64 GB of RAM. Accounting for the 50% faster clock speed of the MBP compared to the Ubuntu workstation, predicting 10^7 sample pairs with the GINN takes 2222.6 minutes on the MBP (i.e., including the time needed to generate the training/test data and to learn the GINN's parameters). Generation of the same amount of data would take 2.2176×10^6 minutes on the MBP. We conclude that the *cost of generating data with the GINN is almost three orders of magnitude lower than that of generating the data with direct simulation methods*, basically amounting to the ratio between the required number of training/test sample pairs and the number of predicted sample pairs.

Given new values of the CVs sampled according to the structured prior distributions, the learned GINN predicts corresponding samples of the QoIs much faster than would be possible with direct simulation using the physics-based model. This enables the generation of io data sets that are orders of magnitude larger, which drives uncertainty propagation in the next section.

4. GINN accelerated uncertainty propagation for supercapacitor dynamics

By replacing the intrinsic computational bottlenecks of the homogenized model for supercapacitor dynamics by a NN, while retaining domain knowledge and correlations between input nodes, GINNs accelerate the prediction of relevant QoIs and thereby enable data-driven UQ and data-centric engineering approaches to simulation-based decision-making, e.g., design of novel metamaterials. A challenge in replacing a well-understood physics-based model with a black box surrogate lies in interpreting and explaining surrogate model predictions. Although we replace the co-simulation framework with a more computationally advantageous surrogate, the complementary BN PDE serves as an anchor for interpreting surrogate predictions. The structured priors on CVs, which encode domain
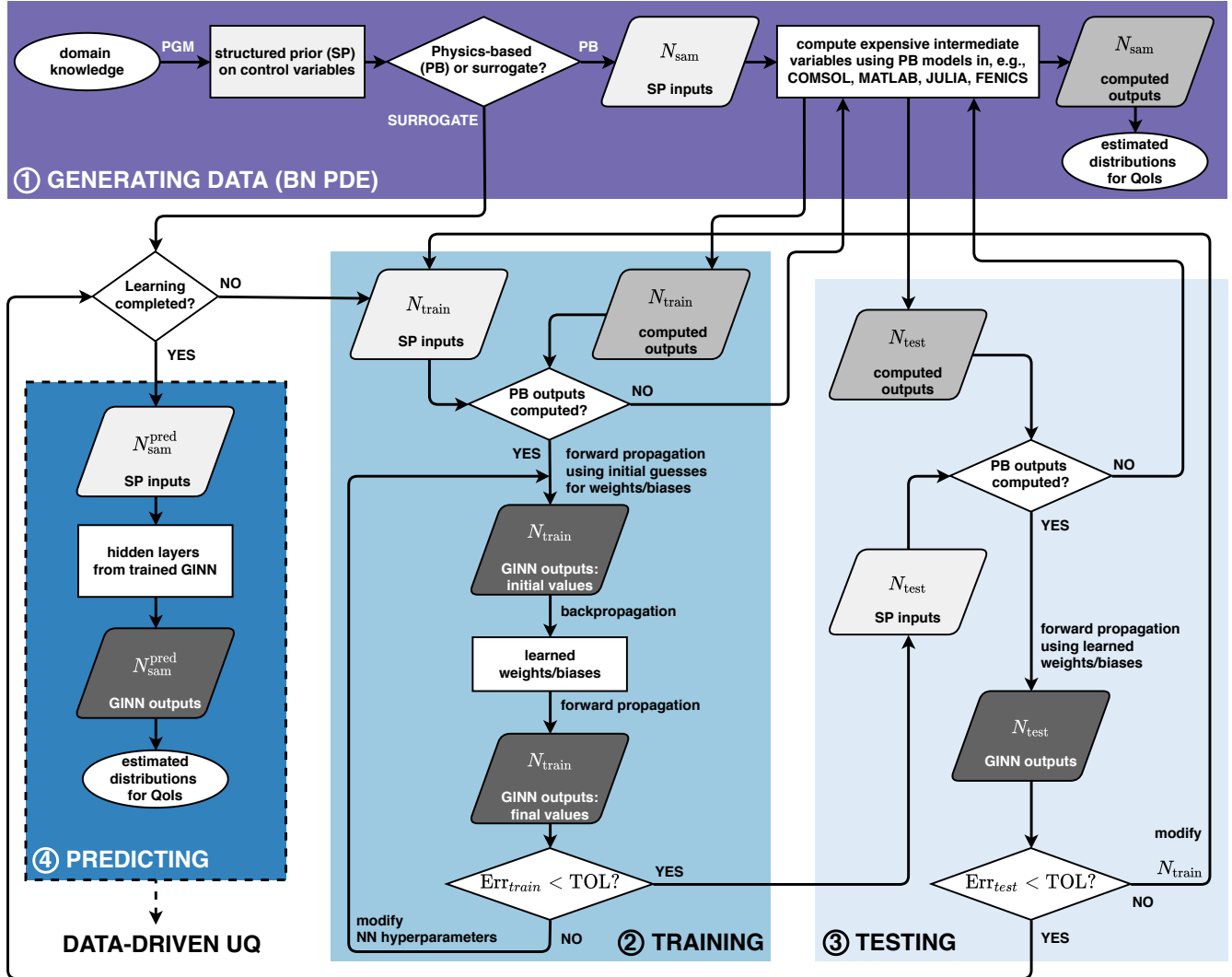
Figure 4: An overview of the global algorithm for the GINN surrogate. (1) Generation of input-output samples using the BN PDE is expensive because of computationally intensive solves for intermediate variables using physics-based model simulation methods. (2) Training a GINN on a small amount of physics-based model data computed with the BN PDE removes this computational bottleneck by replacing the intermediate variables with learned features (hidden layers). (3) Testing the learned GINN on unseen data avoids overfitting and reduces its generalization error. (4) The resulting trained GINN accelerates the prediction of output data at a much reduced computational cost, thus enabling a rigorous data-driven UQ analysis.

knowledge and constraints, ensure that the inputs to the GINN and hence the resulting output QoIs are physically sound.

Given the possibility of fast generation of sample data using the GINN, we can now estimate marginal and joint densities for QoIs with appropriate confidence intervals. Density estimates form the basic elements for understanding uncertainty propagation between and within scales and are used as building blocks for other UQ tasks such as sensitivity analysis. As the QoIs considered here inform a wide array of phenomena including the three-equation state model for EDLC cells (A.11), it is crucial to retain the continuous nature of these variables for downstream computations. Also, as we will demonstrate below, the QoIs for the application of interest are non-Gaussian and skewed. Therefore, density estimates, as opposed to summary statistics, are crucial to provide a complete picture of macroscopic QoIs and their complex interactions.

Kernel density estimation is a nonparametric statistical procedure for estimating probability density functions from samples of a given (univariate) random variable or (multivariate) random vector (see, e.g., [49]). For d-dimensional identically distributed random vectors $\mathbf{Z}^{(1)}, \ldots, \mathbf{Z}^{(m)}$, a kernel density estimator (KDE) for the unknown d-variate probability density f is given by

$$\widehat{f}(\boldsymbol{\zeta}; \mathbf{B}) = \frac{1}{M|\mathbf{B}|^{1/2}} \sum_{m=1}^{M} K\left(\mathbf{B}^{-1/2}(\boldsymbol{\zeta} - \mathbf{Z}^{(m)})\right),$$

where $K : \mathbb{R}^d \to \mathbb{R}$ is a smooth multivariate kernel function and $\mathbf{B}$ is a $d \times d$ symmetric and positive definite bandwidth matrix. As selection of the kernel shape does not play an important role, we select the widely used Gaussian kernel K_G,

$$K_G(\boldsymbol{\zeta}) := \frac{\exp(-\boldsymbol{\zeta}^\top \boldsymbol{\zeta}/2)}{\nu_d}, \quad \nu_d := \int e^{-\boldsymbol{\zeta}^\top \boldsymbol{\zeta}/2} d\boldsymbol{\zeta} = (2\pi)^{d/2}.$$

The choice of the bandwidth matrix $\mathbf{B}$ however is crucial to the performance of $\widehat{f}$, and we choose $\mathbf{B}$ to be diagonal with $B_{ij} = \delta_{ij} b_i^2$, for $i, j = 1, \ldots, d$, with bandwidths $b_i > 0$. With these selections of kernel and bandwidth matrix, and defining $\mathbf{b} = (b_1, \ldots, b_d)^\top$, Section 4 becomes

$$\widehat{f}_Z(\boldsymbol{\zeta}; \mathbf{b}) = \frac{(2\pi)^{-d/2}}{M \prod_{j=1}^{d} b_j} \sum_{m=1}^{M} \prod_{j=1}^{d} \exp\left[-\frac{\left(\zeta_j - Z_j^{(m)}\right)^2}{2b_j^2}\right]. \tag{10}$$

To automate the computation of bandwidths, we utilize the Improved Sheather–Jones method, a direct plug-in bandwidth selector from [50].

As for any estimated quantity, confidence regions can, and should, be given for KDEs. The asymptotic normality of the pointwise error enables one to define a $1 - \alpha$ confidence interval pointwise for $\widehat{f}_Z(\boldsymbol{\zeta}; \mathbf{b})$,

$$C_{1-\alpha}(\boldsymbol{\zeta}) = \left[\widehat{f}_Z(\boldsymbol{\zeta}; \mathbf{b}) - z_{\alpha/2}\sqrt{\frac{\mu_{K,d}\widehat{f}_Z(\boldsymbol{\zeta}; \mathbf{b})}{M \prod_{j=1}^{d} b_j}}, \widehat{f}_Z(\boldsymbol{\zeta}; \mathbf{b}) + z_{\alpha/2}\sqrt{\frac{\mu_{K,d}\widehat{f}_Z(\boldsymbol{\zeta}; \mathbf{b})}{M \prod_{j=1}^{d} b_j}}\right], \tag{11}$$

where $z_{\alpha/2}$ is defined through $\Phi(z_{\alpha/2}) = 1 - \frac{\alpha}{2}$ with Φ the cumulative distribution function for the standard normal (see, e.g., [51]) . The parameter $\mu_{K,d}$ is a constant that depends on the kernel K and dimension d; e.g., for a Gaussian kernel K_G and $d = 1$, $\mu_{K_G,1} = (2\sqrt{\pi})^{-1}$ (and for $d = 2$, $\mu_{K_G,2} = (4\pi)^{-1}$). $C_{1-\alpha}(\boldsymbol{\zeta})$ is an easy to compute random interval such that

$$P\left(\mathbb{E}[\widehat{f}_Z(\boldsymbol{\zeta}; \mathbf{b})] \in C_{1-\alpha}(\boldsymbol{\zeta})\right) \geq 1 - \alpha,$$

that is, the confidence interval holds pointwise for $\boldsymbol{\zeta}$.

KDEs for the marginal distributions, (10) with $d = 1$, associated with the CVs $X_{l_{\text{por}}}$, X_{λ_D}, and X_{φ_Γ} are displayed in Fig. 5 together with appropriate confidence intervals (11), based on both $M = 8 \times 10^3$ and $M = 10^7$ observations. The reliability of the KDEs, i.e., absence of spurious features, and the tightness of the confidence intervals very much depend on the amount of available data. As sampling the structured priors for the CVs is relatively inexpensive, the density estimates in Fig. 5 can be obtained with arbitrarily high confidence by making additional observations a priori. In contrast, QoIs, as child nodes of the highlighted bottleneck in Fig. 1, are expensive to sample using simulation of the physics-based model, such as direct computation of the BN PDE nodes $\boldsymbol{X}_b$. Instead, we can use our GINN surrogate model to construct KDEs for the marginal and joint densities of the QoIs with a high degree of statistical confidence.
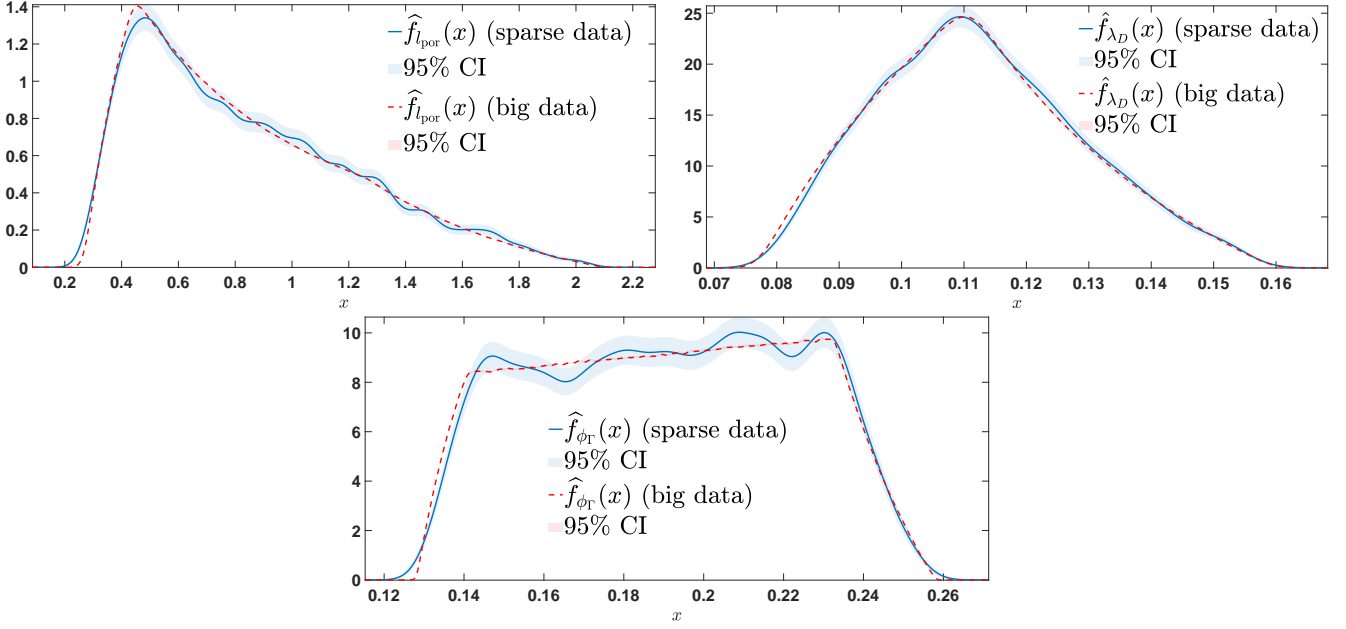
Figure 5: The reliability of KDEs depends on the amount of data available, as illustrated for the marginal PDFs on dependent CVs in (8) by the spurious features with loose confidence intervals resulting from $M = 8 \times 10^3$ (solid/blue) observations compared to the smooth densities with tight confidence intervals resulting from $M = 10^7$ (dashed/red). A large volume of observations is computationally feasible as the the structured priors on CVs, (7) and (8) and Table 1, precede the computational bottleneck in the BN Fig. 1.

Density estimates (10), together with appropriate confidence intervals, provide a great deal of insight into the propagation of uncertainties that are lacking in summary statistics. This is especially true here because of the skewed and non-Gaussian nature of the QoIs Figs. 6 and 7. In Fig. 6, we plot a KDE for each QoI based on $M = 8 \times 10^3$ samples simulated using the BN PDE with the co-simulation forward model framework and on $M = 10^7$ samples using the GINN. For each KDE, optimal kernel bandwidths are again chosen using the Improved Sheather–Jones method. The structured priors over broad ranges in Table 1 produce a wide landscape of effective macroscopic dynamics observed in Figs. 6 and 7. A qualitative comparison of the densities in Fig. 6 reveals that the GINN is making faithful predictions that do not include spurious features observed with cost-limited data obtained from the physics-based model simulation. Moreover, we can report the KDE with significantly increased confidence. That is, the confidence interval for the KDE based on the large number of samples enabled by the GINN is vanishingly small. In Fig. 7, we again observe the highly skewed and non-Gaussian nature of the joint densities for all combinations of QoIs based on $M = 10^7$ samples from the GINN. Likewise (although it is not displayed in the plot), we report that the confidence intervals for each corresponding density are vanishingly small. As the surrogate model enables us to generate a large amount of io sample pairs, we do not require bootstrapping techniques to construct these KDEs or their corresponding confidence intervals.

5. Conclusions and future work

We developed a domain-aware surrogate model for simulation-based decision-making in complex multiscale and multiphysics systems that leverages well-known tools from deep learning and stochastic/predictive modeling. By embedding a probabilistic graphical model, specifically, a Bayesian network (BN), into a deterministic homogenized (i.e., physics-based) model for effective ion diffusion in an electrical double layer (EDL) supercapacitor, we distilled a Graph-Informed Neural Network (GINN) surrogate that incorporates domain knowledge via structured priors on tunable, possibly correlated, control variables (CVs) and bypasses computational bottlenecks in the physics-based model by replacing them with a NN. The complementary BN partial differential equation (BN PDE) model provided a context for interpreting the GINN's predictions.

Our analysis yielded the following major conclusions.

1. Both the correlated CVs and output quantities of interest (QoIs) have non-Gaussian and skewed probability distributions, necessitating their characterization using probability density functions (PDFs) rather than via summary statistics.
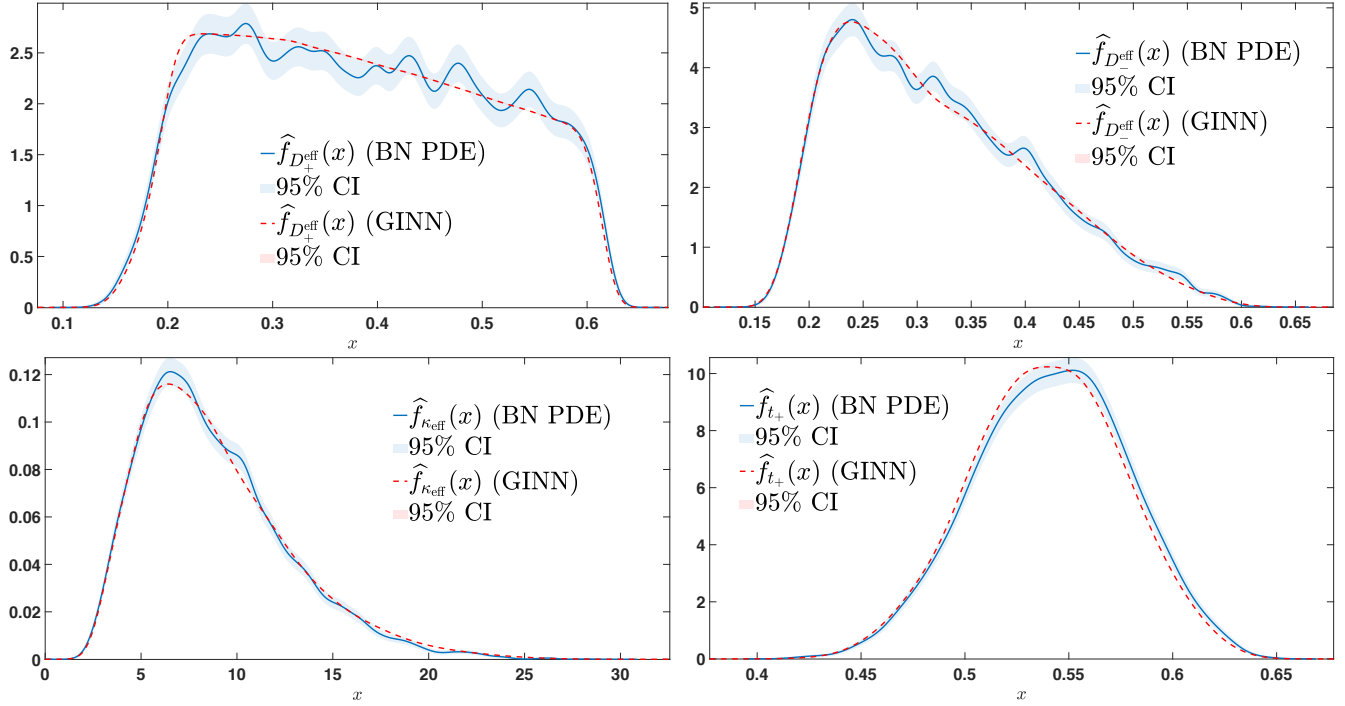
Figure 6: The estimated marginal densities for the QoI variables (3), each based on $M = 8 \times 10^3$ samples computed with the BN PDE (solid/blue) or $M = 10^7$ samples computed with the GINN (dashed/red), are non-Gaussian and skewed. The higher number of samples that can be generated with the GINN surrogate enables tighter estimates that omit spurious features, thereby enabling data-driven UQ analysis.

2. GINNs are able to leverage a "sparse data investment" to yield "big data returns": training on a small set of expensive-to-compute physics-based input-output (io) data, GINNs can cheaply generate a large set of output data to predict QoIs with confidence. This yields several orders of magnitude in computational cost savings compared to using physics-based models alone.

3. Because GINNs can generate io data fast, they can be deployed to estimate the marginal and joint distributions of mutually correlated QoIs with tight confidence intervals.

In a follow-up study, we plan to use the surrogate-based KDEs in Fig. 6 and Fig. 7 to perform data-driven UQ tasks including differential mutual information-based sensitivity studies according to the framework in [16], as opposed to variance-based approaches (e.g., [15]), given the non-Gaussian nature of the QoIs.

The weights and biases of the GINN were determined based on a single run of the entire algorithm including generation of training data with the BN PDE, fitting the GINN with the training set and evaluating its test (generalization) error on an independent test data set. Instead, we could perform, say, N_{real} realizations of the overall procedure, each with a different (but same size) training set with inputs stemming from another set of CV realizations. Then a different set of GINN parameters would be learned, and testing the resulting network would yield a different value for the test loss for the same set of test data. This way, we could build up a PDF for the test loss, which would facilitate comparison of test set predictivity for different complexities of the GINN (e.g., using a different number of hidden layers or neurons per hidden layer), to find the GINN configuration that has the smallest N_{real}-averaged test loss. A similar reasoning was followed in [5] in the context of a PDE based surrogate for the two-dimensional nearest-neighbor kinetic Ising model.
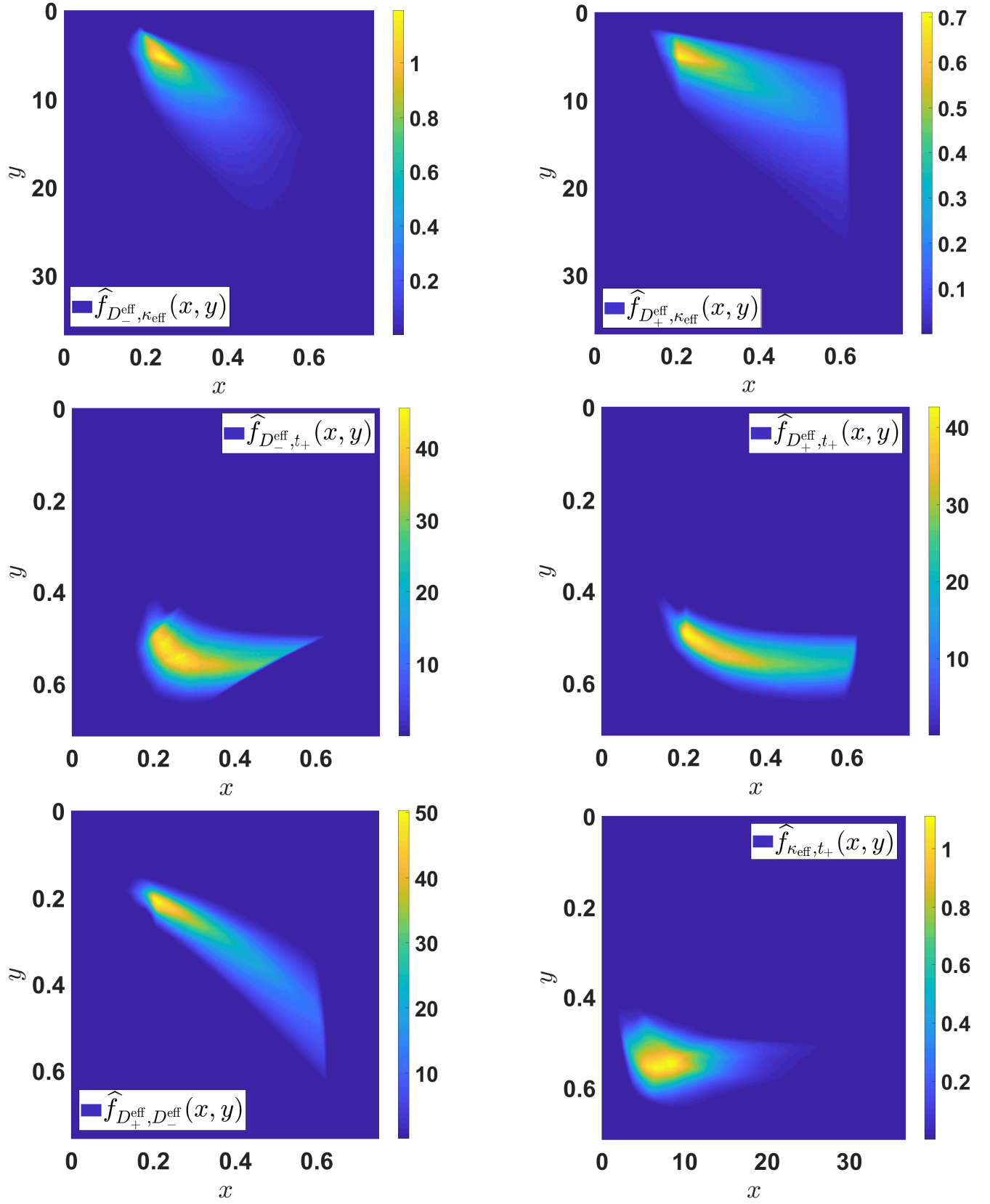
6. Acknowledgments

Figure 7: Estimated joint densities for all combinations of the QoI variables (3), based on $M = 10^7$ samples of each respective output computed with the GINN, exhibit complex relationships that are non-Gaussian and skewed. The generation of faithful multivariate densities for uncertainty propagation requires surrogate modeling to produce a sufficient number of samples for the application of interest.

13

Appendix A. Physics-based model for supercapacitor dynamics

EDL capacitors or EDLCs, as opposed to electrochemical pseudocapacitors, are supercapacitors that rely on the large capacitance of the EDL formed around the surface of their electrodes, typically comprised of a carbon-based hierarchical nanoporous material, for their energy storage capabilities [52]. State models for EDLC cells governing the EDL behavior can be parametrized by effective constants obtained through rigorous mathematical homogenization [39] and related to the diffusion of ions in the cell's electrolyte. Rather than relying on phenomenological relations between microscopic parameters and their macroscopic counterparts, such homogenization or upscaling techniques [53, 54] comprise an *ab initio* approach for deriving macroscopic descriptors which allows them to establish the limits of applicability of the resulting macroscale model.

We consider a supercapacitor with electrodes consisting of a hierarchical nanoporous material $\mathcal{V}$ with characteristic length L that is composed of a pore space $\mathcal{P}$ and impermeable (typically carbon) structure $\mathcal{S}$, i.e., $\mathcal{V} = \mathcal{P} \cup \mathcal{S}$ [39]. The pore space $\mathcal{P}$ is filled with an electrically neutral electrolyte with concentrations of positive ions or *cations* (e.g., tetraethylamonium ions) and negative counter ions or *anions* that evolve in both space and time. The characteristic length scale for this evolution, i.e., a typical pore diameter, is denoted by ℓ and satisfies $\epsilon := \ell/L \ll 1$. The interaction of the electrolyte ions with static charges at the fluid-solid interface Γ gives rise to an electrical double layer (EDL). A representative unit cell $\mathcal{U}$ consisting of pore space $\hat{\mathcal{P}}_{\mathcal{U}}$, structure $\hat{\mathcal{S}}_{\mathcal{U}}$, and fluid-solid interface $\hat{\Gamma}_{\mathcal{U}}$ is illustrated in Fig. A.8 adapted from [39]. We are interested in understanding macroscopic material properties that provide a homogenized (continuum-scale) description of the ion diffusion in the electrolyte, which was derived from a microscale formulation based on the Nernst–Planck equation in [39]. The latter describes the motion of the electrolyte ions under the influence of a concentration gradient and the electric field associated with the gradient of the electric potential in the EDL. The relevant QoIs are then the effective diffusion coefficients of the anions and cations in the electrolyte, and two derived quantities, the electrolyte conductivity and transference number.

Appendix A.1. Effective ion diffusion coefficients

The effective (continuum-scale) diffusion coefficients of the cations and anions are second-order semipositive-definite tensors given by

$$\mathbf{D}_{\pm}^{\text{eff}} := \frac{\mathcal{D}\omega}{G_{\pm}} \int_{\hat{\mathcal{P}}_{\mathcal{U}}} e^{\mp z\hat{\varphi}_{\text{EDL}}}(\mathbf{I} + \nabla_{\mathbf{y}}\boldsymbol{\chi}_{\pm}^{\top})\mathrm{d}\mathbf{y}\,, \quad G_{\pm} := \int_{\hat{\mathcal{P}}_{\mathcal{U}}} e^{\mp\hat{\varphi}_{\text{EDL}}}\mathrm{d}\mathbf{y}\,. \tag{A.1}$$

Here $\mathcal{D}$ $[\text{L}^2/\text{T}]^3$ is the molecular diffusion coefficient of both ion species in the electrolyte;

$$\omega := \|\mathcal{P}\|/\|\mathcal{V}\| \tag{A.2}$$

is the material porosity; z [-] is the ion charge (valence);[4] $\mathbf{I}$ is the identity matrix; and $\boldsymbol{\chi}_{\pm}(\mathbf{y})$ are $\mathcal{U}$-periodic vector functions arising from the homogenization closure equations, i.e., that solve the boundary-value problems

$$\nabla_{\mathbf{y}}\big(e^{\mp z\hat{\varphi}_{\text{EDL}}}(\mathbf{I} + \nabla_{\mathbf{y}}\boldsymbol{\chi}_{\pm}^{\top})\big) = \mathbf{0}\,, \quad \mathbf{y} \in \hat{\mathcal{P}}_{\mathcal{U}}\,; \tag{A.3a}$$

$$\mathbf{n}(\mathbf{I} + \nabla_{\mathbf{y}}\boldsymbol{\chi}_{\pm}^{\top}) = \mathbf{0}\,, \quad \mathbf{y} \in \hat{\Gamma}_{\mathcal{U}}\,; \tag{A.3b}$$

$$\int_{\hat{\mathcal{P}}_{\mathcal{U}}} \boldsymbol{\chi}_{\pm}\mathrm{d}\mathbf{y} = \mathbf{0}\,. \tag{A.3c}$$

In (A.3), $\hat{\varphi}_{\text{EDL}}$ is a non-dimensional formulation of the EDL potential φ_{EDL} [V], that is,

$$\hat{\varphi}_{\text{EDL}} = \frac{F\varphi_{\text{EDL}}}{RT}\,,$$

where $F = 96485$ C/mol is the Faraday constant, R [J/(mol K)] is the gas constant, and T [K] is the temperature. Under the assumption that the spatial variability of $\hat{\varphi}_{\text{EDL}}$ is confined to the nanoscale, it is found by solving

$$\hat{\nabla}^2\hat{\varphi}_{\text{EDL}} = \frac{\ell^2\hat{c}_{\text{b}}}{\epsilon^2\lambda_D^2}\sinh(z\hat{\varphi}_{\text{EDL}})\,, \quad \hat{\mathbf{y}} \in \hat{\mathcal{P}}_{\mathcal{U}}\,; \tag{A.4a}$$

---

[3]We use L for dimensionless units of length and l for liters.

[4]For simplicity of presentation, we assume ion charge symmetry ($z_+ = -z_- = z$) and equality of dissociation constants ($\nu_+ = \nu_- = \nu$), i.e., the electrolyte salt is completely dissociated into cations and anions which have equal but opposite charges. The analysis can be easily extended to multicomponent and/or asymmetric electrolytes.

$$\hat{\varphi}_{\text{EDL}} = \hat{\varphi}_{\Gamma}, \quad \hat{\mathbf{y}} \in \hat{\Gamma}_{\mathcal{U}}, \tag{A.4b}$$

where ℓ [L] is the characteristic pore size; $\hat{c}_{\text{b}} := c_{\text{b}}/c_{\text{in}}$ [-] with c_{b} [mol/l] a characteristic ion concentration in the system (e.g., the initial or average concentration) and c_{in} [mol/l] the initial ion concentration; the Dirichlet boundary condition

$$\hat{\varphi}_{\text{EDL}} = \hat{\varphi}_{\Gamma} = \frac{F\varphi_{\Gamma}}{RT}$$

arises from assuming that the surface Γ carries a constant electric (zeta) potential φ_{Γ} which is satisfied if the solid matrix is highly conductive as in the case of carbon aerogels ([55]); and λ_D [L] is the Debye length, a characteristic length of the EDL which in nanoporous materials is of the same order as the characteristic pore size ℓ. The Debye length is given by

$$\lambda_D = \sqrt{\frac{RT\mathcal{E}}{2F^2 z^2 \nu c_{\text{in}}}}, \tag{A.5}$$

where $\mathcal{E}$ [F/m] is the absolute permittivity of the solvent and ν [-] is the dissociation constant. We set $c_{\text{b}} = c_{\text{in}}$ (i.e., $\hat{c}_b = 1$) in our numerical experiments, and compute φ_{Γ} by solving the transcendental equation

$$\varphi_{\Gamma} = \frac{V}{2} - \varphi_{\text{ecm}} - \frac{\sigma}{C_{\text{H}}}, \tag{A.6a}$$

where V is the external voltage, φ_{ecm} is the electrocapillary maximum, C_{H} is the Helmholtz capacitance, and σ is the surface charge density given by

$$\sigma = \sqrt{4\mathcal{E}RTI}\sqrt{\cosh\left(\frac{\mathrm{e}\varphi_{\Gamma}}{k_{\text{B}}T}\right) - \cosh\left(\frac{\mathrm{e}\varphi_{\min}}{k_{\text{B}}T}\right)},$$

$$I = z^2 C, \quad \varphi_{\min} = \min_{\mathbf{y}\in\hat{\mathcal{P}}_{\mathcal{U}}} \varphi_{\text{EDL}}(\mathbf{y}). \tag{A.6b}$$

In (A.6b), e [C] is the elementary charge, k_{B} [J/K] is the Boltzmann constant, I is the ionic strength, C is the macroscopic ion concentration, and $\varphi_{\min}$ is the midplane potential computed by solving (A.4). Since the latter requires knowledge of the boundary potential φ_{Γ} which we are solving (A.6) for, this would require an iterative procedure. Instead, we assume a value of 0.01 for $\varphi_{\min}$. Additionally, for simplicity we set $C = c_{\text{in}}$, i.e., equal to the initial ion concentration. Using an initial guess of 0.3 for φ_{Γ}, performing a constrained ($\varphi_{\Gamma} \leq 0.5$) nonlinear optimization in MATLAB® resulted in physically reasonable solutions for φ_{Γ} over the ranges of T and c_{in} considered. The values of the various constants in (A.6) are given in Table A.2.

Table A.2: Values for the constant parameters in (A.6). For simplicity we take these parameters as fixed values (cf. (8b) where these could also be incorporated as hyperparameters).

Variable	Meaning	Value	Units
V	external voltage	3	V
C_{H}	Helmholtz capacitance	0.45	F/m^2
$\mathcal{E}$	solvent's dielectric constant	6.9×10^{-11}	-
φ_{ecm}	electrocapillary maximum	0.1	V
$\varphi_{\min}$	midplane potential	0.01	V

As opposed to the typical treatment that expresses effective diffusion through phenomenological relations, the rigorous derivation of the effective diffusion tensors in (A.1) using homogenization theory enables one to express them in terms of pore-scale geometry and processes including the EDL potential. In particular, we observe that only the diagonal elements of $\mathbf{D}_{\pm}^{\text{eff}}$ are non-zero since off-diagonal elements of $\nabla_{\mathbf{y}}\boldsymbol{\chi}_{\pm}^{\top}$ are zeros in (A.3). Together with the fact that we consider a homogeneous isotropic nanoporous material (see unit cell in Fig. A.8), the diffusion coefficients in (A.1) become scalars,

$$D_{+}^{\text{eff}} \quad \text{and} \quad D_{-}^{\text{eff}}. \tag{A.7}$$

The assumption of isotropy is reflected in the fact that the pore throat size is identical in both directions (Fig. A.8) and can be expressed in terms of the solid radius r and porosity ω via

$$l_{\text{por}} = -r + 0.5\sqrt{4r^2 + 4r^2 \cdot \left[\frac{\pi}{4\cdot(1-\omega)} - 1\right]}, \tag{A.8}$$

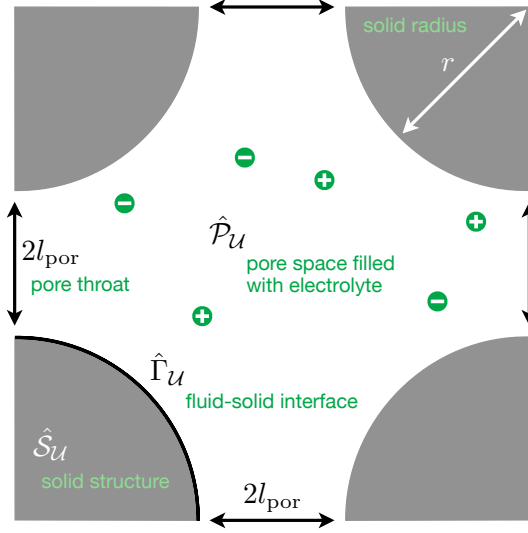where l_{por} refers to the *half* pore throat size.[5]

Figure A.8: Unit cell $\mathcal{U}$ for a homogeneous isotropic nanoporous material containing circular impermeable structures $\hat{\mathcal{S}}_{\mathcal{U}}$ of radius r separated by pore space $\hat{\mathcal{P}}_{\mathcal{U}}$ with pores of throat size $2l_{\mathrm{por}}$ in both the transverse and longitudinal directions. The fluid-solid interface $\hat{\Gamma}_{\mathcal{U}}$, involved in the formation of the EDL, is a key component in the investigation of EDLC cells. Data-driven UQ for simulation-aided design of nanoporous metamaterials is crucial to data-centric approaches for engineering advanced supercapacitors for long-term energy storage applications.

Appendix A.2. Effective electrolyte conductivity and transference number

The effective diffusion coefficients in (A.1) are used to compute other key QoIs including the effective electrolyte conductivity κ^{eff} [mS/cm],

$$\kappa^{\mathrm{eff}} := \nu z^2 \frac{F^2 c_{\mathrm{in}}}{RT}(D_+^{\mathrm{eff}} + D_-^{\mathrm{eff}}), \tag{A.9}$$

and the transference number t_+ [-] (fraction of the current carried by the cations),

$$t_+ := \frac{D_+^{\mathrm{eff}}}{D_+^{\mathrm{eff}} + D_-^{\mathrm{eff}}}. \tag{A.10}$$

Appendix A.3. Three-equation model governing the EDL

The effective ion diffusion parameters D_+^{eff}, D_-^{eff}, κ^{eff}, and t_+ serve as coefficients in a three-equation model ([39, 56, 57]) for macroscopic state variables that characterize the behavior of EDLC cells: electrolyte ionic concentration $C(x,t)$, electrolyte potential $\Phi(x,t)$, and electric potential of the solid phase $\Phi_s(x,t)$, satisfying,

$$\mathcal{C}_{\mathrm{EDL}} \frac{\partial(\Phi_s - \Phi)}{\partial t} = \frac{\partial}{\partial x}\left(\sigma_s \frac{\partial \Phi_s}{\partial x}\right), \tag{A.11a}$$

$$\frac{\partial}{\partial x}\left(\sigma_s \frac{\partial \Phi_s}{\partial x} + \kappa^{\mathrm{eff}} \frac{\partial \Phi}{\partial x} + \kappa^{\mathrm{eff}} RT \frac{2t_+ - 1}{zF} \frac{\partial \ln C}{\partial x}\right) = 0, \tag{A.11b}$$

$$\omega \frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\left(\frac{2D_+^{\mathrm{eff}} D_-^{\mathrm{eff}}}{D_+^{\mathrm{eff}} + D_-^{\mathrm{eff}}} \frac{\partial C}{\partial x} - \alpha \frac{\partial(\Phi_s - \Phi)}{\partial t}\right), \tag{A.11c}$$

for $x \in [0, L]$ and $t > \infty$, subject to suitable boundary conditions where σ_s is the electric conductivity of the solid phase, $\mathcal{C}_{\mathrm{EDL}}$ is the EDL capacitance, and $\alpha = \alpha(x)$ is a piecewise defined function of t_+ that varies according to electrode thickness and separator thickness. In the remainder of this work, we will use their normalized counterparts $D_+^{\mathrm{eff}}/\mathcal{D}$ and $D_-^{\mathrm{eff}}/\mathcal{D}$, but retain the notation D_+^{eff} and D_-^{eff} to refer to these normalized effective diffusion coefficients.

---

[5]We take the *half* instead of the *full* pore throat size as an input parameter for reasons of convenience.

(A.1) to (A.8) and (A.11) constitute two stages of a complex model for supercapacitor dynamics in EDLC cells that includes nonlinear multiscale and multiphysics interactions. For constructing our BN PDE model, we focus only on the first stage covered in Appendix A.1 and Appendix A.2. A similar reasoning can be followed for the EDLC state model detailed in the current section.

References

[1] R. C. Smith, Uncertainty Quantification: Theory, Implementation, and Applications, Vol. 12, SIAM, Philadelphia, PA, 2013.

[2] F. D.-H. Lau, N. M. Adams, M. A. Girolami, L. J. Butler, M. Z. Elshafie, The role of statistics in data-centric engineering, Stat. Prob. Lett. 136 (2018) 58–62. `doi:10.1016/j.spl.2018.02.035`.

[3] M. Eldred, A. Giunta, S. Collis, Second-order corrections for surrogate-based optimization with model hierarchies, in: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2004. `doi:10.2514/6.2004-4457`.

[4] M. Frangos, Y. Marzouk, K. Willcox, B. van Bloemen Waanders, Surrogate and Reduced-Order Modeling: A Comparison of Approaches for Large-Scale Statistical Inverse Problems, John Wiley & Sons, Ltd, 2010, Ch. 7, pp. 123–149. `doi:10.1002/9780470685853.ch7`.

[5] S. Taverniers, T. S. Haut, K. Barros, F. J. Alexander, T. Lookman, Physics-based statistical learning approach to mesoscopic model selection, Phys. Rev. E 92 (2015) 053301. `doi:10.1103/PhysRevE.92.053301`.

[6] B. C. Daniels, I. Nemenman, Automated adaptive inference of phenomenological dynamical models, Nature Comm. 6 (1) (2015) 8133. `doi:10.1038/ncomms9133`.

[7] V. Harmandaris, E. Kalligiannaki, M. Katsoulakis, P. Plecháč, Path-space variational inference for non-equilibrium coarse-grained systems, J. Comput. Phys. 314 (C) (2016) 355–383. `doi:10.1016/j.jcp.2016.03.021`.

[8] O. Sen, S. L. Davis, G. B. Jacobs, H. S. Udaykumar, Evaluation of convergence behavior of metamodeling techniques for bridging scales in multi-scale multimaterial simulation, J. Comput. Phys. 294 (2015) 585–604.

[9] O. Sen, N. J. Gaul, K. K. Choi, G. B. Jacobs, H. S. Udaykumar, Evaluation of multifidelity surrogate modeling techniques to construct closure laws for drag in shock-particle interactions, J. Comput. Phys. 371 (2018) 434–451.

[10] O. Sen, N. J. Gaul, S. Davis, K. K. Choi, G. B. Jacobs, H. S. Udaykumar, Evaluation of kriging based surrogate models constructed from mesoscale computations of shock interaction with particles, J. Comput. Phys. 336 (2017) 235–260.

[11] O. Sen, N. J. Gaul, K. K. Choi, G. B. Jacobs, H. S. Udaykumar, Role of pseudo-turbulent stresses in shocked particle clouds and construction of surrogate models for closure, Shock Waves 28 (2018) 579–597.

[12] S. Mak, C.-L. Sung, X. Wang, S.-T. Yeh, Y.-H. Chang, V. R. Joseph, V. Yang, C. F. J. Wu, An efficient surrogate model for emulation and physics extraction of large eddy simulations, J. Am. Stat. Assoc. 113 (524) (2018) 1443–1456. `doi:10.1080/01621459.2017.1409123`.

[13] O. Sen, S. Taverniers, P. Das, G. Jacobs, H. S. Udaykumar, Machine-learning based multi-scale model for shock-particle interactions, Bull. Am. Phys. Soc. 64 (8) (2019).

[14] A. Nassar, N. K. Rai, O. Sen, H. S. Udaykumar, Modeling mesoscale energy localization in shocked HMX, part I: machine-learned surrogate models for the effects of loading and void sizes, Shock Waves 29 (2019) 537–558.

[15] K. Um, X. Zhang, M. Katsoulakis, P. Plecháč, D. M. Tartakovsky, Global sensitivity analysis of multiscale properties of porous materials, J. Appl. Phys. 123 (7) (2018) 075103. `doi:10.1063/1.5009691`.

[16] K. Um, E. J. Hall, M. A. Katsoulakis, D. M. Tartakovsky, Causality and Bayesian Network PDEs for multiscale representations of porous media, J. Comput. Phys. 394 (2019) 658–678. `doi:10.1016/j.jcp.2019.06.007`.

[17] E. Torre, S. Marelli, P. Embrechts, B. Sudret, A general framework for data-driven uncertainty quantification under complex input dependencies using vine copulas, Prob. Engrg. Mech. 55 (2019) 1–16. `doi:10.1016/j.probengmech.2018.08.001`.

[18] H. Lee, I. S. Kang, Neural algorithm for solving differential equations, J. Comput. Phys. 91 (1) (1990) 110 – 131. `doi:https://doi.org/10.1016/0021-9991(90)90007-N`.

[19] D. C. Psichogios, L. H. Ungar, A hybrid neural network-first principles approach to process modeling, AIChE J. 38 (10) (1992) 1499–1511. `doi:10.1002/aic.690381003`.

[20] I. E. Lagaris, A. Likas, D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Trans. Neural Net. 9 (5) (1998) 987–1000. `doi:10.1109/72.712178`.

[21] I. E. Lagaris, A. C. Likas, D. G. Papageorgiou, Neural-network methods for boundary value problems with irregular boundaries, IEEE Trans. Neural Net. 11 (5) (2000) 1041–1049.

[22] G. Balokas, S. Czichon, R. Rolfes, Neural network assisted multiscale analysis for the elastic properties prediction of 3D braided composites under uncertainty, Composite Struct. 183 (2018) 550–562. `doi:10.1016/j.compstruct.2017.06.037`.

[23] R. K. Tripathy, I. Bilionis, Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification, J. Comput. Phys. 375 (2018) 565–588. `doi:10.1016/j.jcp.2018.08.036`.

[24] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, J. Comput. Phys. 394 (2019) 56–81. `doi:https://doi.org/10.1016/j.jcp.2019.05.024`.

[25] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707. `doi:10.1016/j.jcp.2018.10.045`.

[26] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).

[27] A. Paszke, et al., PyTorch: An imperative style, high-performance deep learning library33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada, 2019. Software available from pytorch.org.

[28] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, `http://www.deeplearningbook.org`.

[29] M. Raissi, Z. Wang, M. S. Triantafyllou, G. E. Karniadakis, Deep learning of vortex-induced vibrations, J. Fluid Mech. 861 (2019) 119–137.

[30] D. Zhang, L. Lu, L. Guo, G. E. Karniadakis, Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems, J. Comput. Phys. 397 (2019) 108850. `doi:10.1016/j.jcp.2019.07.048`.

[31] Y. Yang, P. Perdikaris, Adversarial uncertainty quantification in physics-informed neural networks, J. Comput. Phys. 394 (2019) 136–152. `doi:https://doi.org/10.1016/j.jcp.2019.05.027`.

[32] X. Meng, G. E. Karniadakis, A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems, J. Comput. Phys. 401 (2020) 109020.

[33] J. Sirignano, K. Spiliopoulos, DGM: A deep learning algorithm for solving partial differential equations, J. Comput. Phys. 375 (2018) 1339–1364.

[34] J. Berg, K. Nyström, A unified deep artificial neural network approach to partial differential equations in complex geometries, Neurocomputing 317 (2018) 28–41.

[35] L. Sun, H. Gao, S. Pan, J.-X. Wang, Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data, Comput. Meth. Appl. Mech. Engrg 361 (2020) 112732. `doi:https://doi.org/10.1016/j.cma.2019.112732`.

[36] T. Hastie, R. Tibshirani, M. Wainwright, Statistical Learning with Sparsity: The Lasso and Generalizations, Taylor & Francis, 2015.

[37] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE transactions on neural networks 20 (1) (2009) 61–80.

[38] J. Zhou, et al., Graph neural networks: A review of methods and applications, arXiv preprint arXiv:1812.08434 (2019).

[39] X. Zhang, D. M. Tartakovsky, Effective ion diffusion in charged nanoporous materials, J. Electrochem. Soc. 164 (4) (2017) E53–E61. `doi:10.1149/2.0491704jes`.

[40] S. Taverniers, D. M. Tartakovsky, Estimation of distributions via multilevel Monte Carlo with stratified sampling, J. Comp. Phys.Article in press (2020).

[41] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Elsevier, San Francisco, CA, 2014. `doi:10.1016/C2009-0-27609-4`.

[42] J. Pearl, Causality, CUP, Cambridge, 2009. `doi:10.1017/CBO9780511803161`.

[43] D. Koller, N. Friedman, Probabilistic Graphical Models, Adaptive Computation and Machine Learning, MIT Press, Cambridge, MA, 2009.

[44] Z. Li, et al., Tuning the interlayer spacing of graphene laminate films for efficient pore utilization towards compact capacitive energy storage, Nature Energy 5 (2020) 160–168.

[45] J. Pearl, M. Glymour, N. P. Jewell, Causal inference in statistics, John Wiley & Sons, Ltd., Chichester, 2016, a primer.

[46] Y. Zhu, N. Zabaras, Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification, J. Comput. Phys. 366 (2018) 415 – 447. `doi:https://doi.org/10.1016/j.jcp.2018.04.018`.

[47] S. Karumuri, R. Tripathy, I. Bilionis, J. Panchal, Simulator-free solution of high-dimensional stochastic elliptic partial differential equations using deep neural networks, J. Comput. Phys. 404 (2020) 109120. `doi:https://doi.org/10.1016/j.jcp.2019.109120`.

[48] M. Motamed, A multi-fidelity neural network surrogate sampling method for uncertainty quantification, arXiv preprint (2019). `arXiv:1909.01859`.

[49] L. A. Wasserman, All of nonparametric statistics: with 52 illustrations, Springer, 2006.

[50] Z. I. Botev, J. F. Grotowski, D. P. Kroese, Kernel density estimation via diffusion, Ann. Statist. 38 (5) (2010) 2916–2957. `doi:10.1214/10-AOS799`.

[51] Y.-C. Chen, A tutorial on kernel density estimation and recent advances, Biostatistics & Epidemiology 1 (1) (2017) 161–187. `doi:10.1080/24709360.2017.1396742`.

[52] F. Béguin, E. Frąckowiak, G. Q. Max Lu, Supercapacitors: Materials, Systems, and Applications, 1st Edition, Wiley-VCH, 2013.

[53] X. Zhang, K. Urita, I. Moriguchi, D. M. Tartakovsky, Design of nanoporous materials with optimal sorption capacity, J. Appl. Phys. 117 (24) (2015) 244304. `doi:10.1063/1.4923057`.

[54] B. Ling, A. M. Tartakovsky, I. Battiato, Dispersion controlled by permeable surfaces: surface properties and scaling, Journal of Fluid Mechanics 801 (2016) 13–42. `doi:10.1017/jfm.2016.431`.

[55] T.-Y. Ying, K.-L. Yang, S. Yiacoumi, C. Tsouris, Electrosorption of ions from aqueous solutions by nanostructured carbon aerogel, J. Colloid Interface Sci. 250 (1) (2002) 18–27. `doi:https://doi.org/10.1006/jcis.2002.8314`.

[56] J. Newman, K. E. Thomas-Alyea, Electrochemical Systems, John Wiley & Sons, 2012.

[57] M. W. Verbrugge, P. Liu, Microstructural analysis and mathematical modeling of electric double-layer super-capacitors, J. Electrochem. Soc. 152 (5) (2005) D79. `doi:10.1149/1.1878052`.