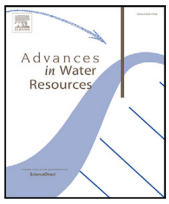
Advances in Water Resources 206 (2025) 105140

Contents lists available at ScienceDirect

Advances in Water Resources

journal homepage: www.elsevier.com/locate/advwatres



Transfer learning of neural surrogates on multifidelity groundwater simulations

A. Chiofalo [a], V. Ciriello [a,*], D.M. Tartakovsky [b]

[a] Department of Civil, Chemical, Environmental and Materials Engineering, University of Bologna, Italy
[b] Department of Energy Science and Engineering, Stanford University, USA

ARTICLE INFO

Dataset link: https://github.com/Model-Reduction-and-UQ-Group/Transfer_Learning_K_reconstruction, https://doi.org/10.5281/zenodo.17087811

Keywords:
Multifidelity data
Convolutional neural network
Reduced-order model
Hydrology
Transfer learning
Inverse problem

ABSTRACT

Computationally inexpensive surrogates of process-based models, such as deep neural networks, enable ensemble-based computations used in risk assessment, data assimilation, etc. However, generation of large datasets required to train a neural network can be as expensive as the ensemble simulations themselves. We ameliorate this challenge by using data from multifidelity (MF) groundwater simulations and transfer learning (TL) to reduce data generation costs while maintaining model accuracy. As a computational example, we train a deep convolutional neural network (CNN) to reconstruct permeability fields from saturation maps derived from a multiphase flow model. Starting with very low- and low-fidelity data generated on increasingly coarse meshes, we pretrain the CNN, followed by output-layer training and fine-tuning using only a limited number of high-fidelity samples. We demonstrate the surrogate's robustness when interpreting low-quality inputs — such as interpolated maps or data affected by noise — which has strong implications for the applicability in practical hydrogeological scenarios. This multilevel MF-TL strategy achieves a favorable trade-off between computational efficiency and predictive accuracy, significantly outperforming high-fidelity-only approaches under the same computational budget.

1. Introduction

Machine learning (ML) techniques are increasingly applied in scientific domains, including fluid dynamics. Among these, neural networks (NNs) are a widely adopted approach in scientific computing (Karniadakis et al., 2021; Shen et al., 2023). Here, we are interested in the role of NNs as surrogate models to speed up ensemble-based computations (EBCs), which are often prohibitively expensive. In this context, surrogates serve as efficient alternatives to high-fidelity simulations by providing approximate solutions at a fraction of the computational cost (e.g., Oladyshkin et al., 2012; Zhou and Tartakovsky, 2020; Focaccia et al., 2021; Libero et al., 2024; Zhao et al., 2024).

Deep learning is routinely used to develop fast surrogates of groundwater flow and transport models. For example, physics-informed NNs incorporate knowledge of the governing equations directly into the training process by minimizing a combined loss that penalizes deviations from both the physical laws and auxiliary conditions (Raissi et al., 2019). This strategy has been used (e.g., He et al., 2020; Tartakovsky et al., 2021) to combine physical residuals with simulation data to tackle single-phase flow and transport processes. Purely data-driven techniques rely only on labeled (experimental or simulated) data to learn a NN mapping between inputs to outputs. Among these, convolutional NNs (CNNs) represent spatial structures of the data; a convolutional encoder–decoder architecture (e.g., Zhu and Zabaras, 2018) was used for single-phase flow predictions and, in combination with autoregressive strategies, to forecast long-term simulation outputs (Mo et al., 2019a,b, 2020).

A persistent challenge in surrogate modeling is the curse of dimensionality: performance tends to degrade as the number of uncertain inputs increases (Barajas-Solano and Tartakovsky, 2016). NNs are attractive because of their ability to handle high-dimensional input–output mappings. However, their effectiveness depends on the availability of large training datasets, which require numerous model evaluations over varying parameter realizations. Therefore, the deployment of NN surrogates is only advantageous if the training phase is less computationally demanding than performing the full set of EBCs. In practice, if we consider the time needed for the training and the hyperparameter

* Corresponding author.
E-mail address: v.ciriello@unibo.it (V. Ciriello).

https://doi.org/10.1016/j.advwatres.2025.105140
Received 17 July 2025; Received in revised form 3 October 2025; Accepted 3 October 2025
Available online 8 October 2025

tuning, these costs are often on par, casting doubt on the overall efficiency of NN-based surrogate models (Song and Tartakovsky, 2022).

A way to expedite the generation of a robust dataset for NN training is by using multifidelity (MF) data. In the literature, numerous studies have explored the use of MF simulations to train data-driven models to enhance both accuracy and computational efficiency (Geneva and Zabaras, 2020; Meng and Karniadakis, 2020; Howard et al., 2024). Recently, Song and Tartakovsky (2022) introduced the idea of using transfer learning (TL) on MF data to train a CNN surrogate of a system of nonlinear parabolic PDEs governing multi-phase flow in a heterogeneous porous medium (Taverniers et al., 2020), where MF data are obtained by solving the PDEs on fine and coarse meshes. Further developments are proposed in Propp and Tartakovsky (2025) that train a CNN surrogate model on a mixture of numerical solutions to both the d-dimensional problem and its (d-1) dimensional approximation, and in Chiofalo et al. (2025) that use two physical models with different levels of complexity and reliability to generate MF data.

In this work, we build upon these advancements to analyze how the performance of NN surrogates can improve, in terms of computational cost and accuracy, by using multiple levels of multifidelity data. We also demonstrate the capability of the NNs pre-tuned on low-fidelity data to interpret low-accurate input data either obtained by interpolation or affected by error noise. That is particularly relevant in hydrogeological applications.

As an application example, we use the same dataset in Song and Tartakovsky (2022), representing two-dimensional multiphase flow in heterogeneous media. In Song and Tartakovsky (2022), the CNN surrogate maps a permeability field to multiple saturation outputs. In this configuration, 1 image to N images, the problem is generally well-posed and deterministic. TL in this setting primarily enhances the richness and resolution of the output, allowing the network to refine physically plausible patterns learned from low-fidelity data using limited high-fidelity data. As such, the transfer dynamics see early layers encoding permeability features remaining more stable during fine-tuning while mainly decoder layers adjust. Differently, we derive a CNN-based surrogate trained on inverted data, where the network infers a permeability field from multiple saturation maps, in a configuration N images to 1 image. Typically, this problem is ill-posed and much more sensitive to uncertainty and noise. Here, TL plays a more critical role in stabilizing the inverse mapping by providing an initial prior learned from approximate data, but may also need to unlearn biased or ambiguous features. As such, the transfer dynamics ask early layers to fuse and interpret complex spatiotemporal inputs and may require more aggressive adaptation. The study of the effect of TL in the case 1 to N (Song and Tartakovsky, 2022) is not expected to be generalizable to the N to 1 due to the fundamentally different nature of the tasks. Therefore, a dedicated analysis is essential to properly assess the benefits and limitations of TL in this case.

Beyond their structural and learning differences, the two different CNN surrogate configurations also serve distinct practical purposes. The direct configuration, where permeability maps are used to predict time-evolving saturation fields, is particularly well-suited for forward uncertainty quantification through Monte Carlo simulations. Once trained, this surrogate can rapidly generate large ensembles of outputs for different permeability realizations, enabling efficient propagation of geological uncertainty through the flow model. On the other hand, the indirect configuration, which infers permeability from observed saturation fields, is better aligned with inverse modeling tasks such as history matching, data assimilation, or even real-time reservoir characterization. In these applications, the goal is to reconstruct the most plausible subsurface properties given partial or time-sequenced flow observations, tasks that are computationally intensive with traditional inversion techniques. The CNN surrogate, in this case, can act as a fast approximation of the inverse map, supporting real-time decision-making or iterative data assimilation workflows. Traditional inversion methods (e.g., MCMC, EnKF) remain more robust and interpretable for inversion tasks, especially in high-stakes applications requiring uncertainty bounds or sparse data assimilation. As such, rather than replacing these classical approaches, CNN surrogates are best used as complementary tools, by guiding prior selection or acting as efficient approximates within hybrid physics-informed ML frameworks (Zhou and Tartakovsky, 2020; Zhou et al., 2022). However, here, we do not aim to develop a general-purpose inverse surrogate, instead, we focus on systematically exploring the effect of TL in this configuration.

In this work, we also employ three levels of multi-fidelity data. This possibility was explored in Song (2022), to study further the practical trade-off between training cost and accuracy in large-scale Monte Carlo simulations. Here, we do so in the N to 1 configuration not only to analyze the benefit from a training cost perspective but also the possible increase in the surrogate's robustness to degraded inputs, typical of real-world applications, where saturation observations used for inversion are often sparse, noisy, or limited in resolution. Specifically, we explore the accuracy and computational saving obtained by training the CNN on two or three levels of MF data generated through simulations of the forward model on fine, coarse, and very coarse meshes. In the following, we refer to these data as high-fidelity (HF), low-fidelity (LF), and very low-fidelity (VLF) data, respectively. We demonstrate that our approach provides an optimal balance between computational speed-up and predictive accuracy. Specifically, CNNs trained on MF data can achieve significantly lower root-mean-square errors compared to those trained solely on HF data while using an equivalent data generation budget. Moreover, by optimizing the TL strategy, we achieve rapid convergence during training and reliable predictions, thereby validating the effectiveness of MF data in solving inverse problems.

The paper is structured as follows. Section 2 describes the CNN architecture and the training process using transfer learning and multiple levels of MF data. Section 3 provides a brief description of the physical forward model used to generate the fine and coarse image dataset. Section 4 introduces the CNN surrogate designed for solving inverse problems. In Section 5, we showcase the accuracy and computational efficiency of the CNN-based surrogate. Finally, Section 6 contains our closing remarks.

2. Materials and methods

This section is devoted to the description of the deep convolutional encoder–decoder surrogate model, and of TL as a method to train the CNN on MF data.

2.1. Deep convolutional encoder–decoder network

The deep convolutional encoder–decoder is a well-suited architecture for producing accurate surrogates in various fluid flow problems. In this study, we select a CNN with an encoder–decoder architecture, which has been successfully applied to single-phase and multi-phase flow problems in the context of uncertainty quantification (Mo et al., 2019a,b; Song and Tartakovsky, 2022). This specific CNN reformulates the surrogate modeling task as an image-to-image regression problem. The idea is to capitalize on the strengths of CNNs in image processing. Indeed, CNNs are particularly effective for processing image-like data, as they explicitly account for data spatial dependency. Additionally, the encoder–decoder architecture is highly compatible with training on MF data generated by solving a PDE-based model on different grids (Song and Tartakovsky, 2022).

The convolutional encoder–decoder architecture consists of initial layers that process the input dimensions. Subsequent layers encode the input into the latent state, a low-dimensional representation of the original inputs. This encoded latent state is then decoded to match the output dimensions. The encoder–decoder framework serves as a dimensionality reduction technique, helping to alleviate the "curse of dimensionality", which is a key factor in the failure of many surrogate methods when training on high-dimensional inputs with limited data (Mo et al., 2019b).

The CNN layers are arranged in dense blocks, where all layers within each block are directly connected to maximize information flow. To maintain the feed-forward nature, each layer receives additional inputs from all preceding layers and passes its feature maps to all subsequent layers. A dense block has two design parameters: the number of internal layers and the number of output features of each layer. Each layer consists of three consecutive operations: a batch normalization (Ioffe and Szegedy, 2015), a rectified linear unit, and a convolution (Goodfellow, 2016). Transition layers, known as encoding layers in the encoder, and decoding layers in the decoder, are inserted between each set of dense blocks. These layers modify the feature size using convolutional and transposed convolutional operations during the encoding and decoding processes and also reduce the number of feature maps used during concatenation in the dense blocks (Mo et al., 2019a).

Given this architecture, to train the CNN surrogate model, we need to perform repeated solves of the (coupled, nonlinear) PDEs which governs the selected physical process:

$$\mathcal{N}(\mathbf{u};\boldsymbol{\Theta}) = g(\mathbf{x},\ t;\boldsymbol{\Theta}), \qquad (\mathbf{x},\ t) \in D \times (0,\ T]. \tag{1}$$

Eq. (1) describes the spatio-temporal evolution of a state variable $\mathbf{u}(\mathbf{x}, t)$ in the computational domain D over simulation time horizon $(0,\ T]$. To account for the variability of the inputs $\boldsymbol{\Theta}(\mathbf{x})$, that parameterize the differential operator $\mathcal{N}$, the source function g, and auxiliary functions in the initial and boundary conditions, a large number of solves is required.

In case an inverse CNN is trained, the discretized solution $\mathbf{u}(\mathbf{x}_i,\ t_k)$ of PDEs at N_{ts} time steps $\{t_k\}_{k=1}^{N_{\text{ts}}}$ (from the corresponding forward model) are used in input, while the values of $\boldsymbol{\Theta}(\mathbf{x}_i)$ in N_{el} elements $\{\mathbf{x}_i\}_{i=1}^{N_{\text{el}}}$ of a numerical grid as output. To ensure robust training of the CNN and to effectively capture the underlying patterns in the data, it is essential to utilize a sufficiently large training dataset, comprising N_{train} samples.

Once the dataset is generated, the CNN training consists of finding a set of N_{w} weights, $\mathbf{w} = (w_1, \dots, w_{N_{\text{w}}})^{\top}$, that minimizes the loss function

$$\mathcal{L}(\mathbf{w}) = \sum_{m=1}^{N_{\text{train}}} \|\boldsymbol{\Theta}_{\boldsymbol{m}}(\mathbf{x}) - \hat{\boldsymbol{\Theta}}_{\boldsymbol{m}}(\mathbf{x},\mathbf{w})\|_1 + \lambda\|\mathbf{w}\|_2^2. \tag{2}$$

The first term in Eq. (2) represents the L_1-norm discrepancy between the reference map of parameter $\boldsymbol{\Theta}(\mathbf{x})$ and its CNN representation, $\hat{\boldsymbol{\Theta}}(\mathbf{x},\mathbf{w})$. The second term in (2) involves L_2 squared regularization, which penalizes large weights to prevent overfitting and improving generalization. The amount of penalty applied is determined by the regularization parameter λ.

2.2. Transfer learning on multiple levels of multifidelity data

In physical sciences, observational data are seldom sufficient, and typically data are generated by solving a governing equation, as described in Section 2.1. Specifically, to create enough training data for building DNN surrogate models, it may be necessary to perform several thousand HF simulations. As a result, the cost of generating training data raises doubts about the practicality of using DNNs as surrogate models in real-world applications (Song and Tartakovsky, 2022). To address this issue, we propose a framework that employs TL and MF simulations to reduce the cost of data generation for the training of an inverse CNN surrogate for the reconstruction of permeability fields (see Section 3).

TL involves using knowledge gained from one task to help learn another related task (Hosna et al., 2022). By utilizing the knowledge stored in the NN trained for the initial task, it is possible to significantly reduce the number of training samples required for the second task. Following this idea, the CNN is first trained using a large amount of LF input data. Subsequently, a relatively small amount of HF input data is used to fine-tune the network. Here, HF and LF data refer to the values of the state variable $\mathbf{u}(\mathbf{x}_i,\ t_k)$ obtained solving (1) on a fine ($N_{\text{el}} = N_{\text{el}}^{\text{HF}}$) and coarse ($N_{\text{el}} = N_{\text{el}}^{\text{LF}}$) mesh, respectively.

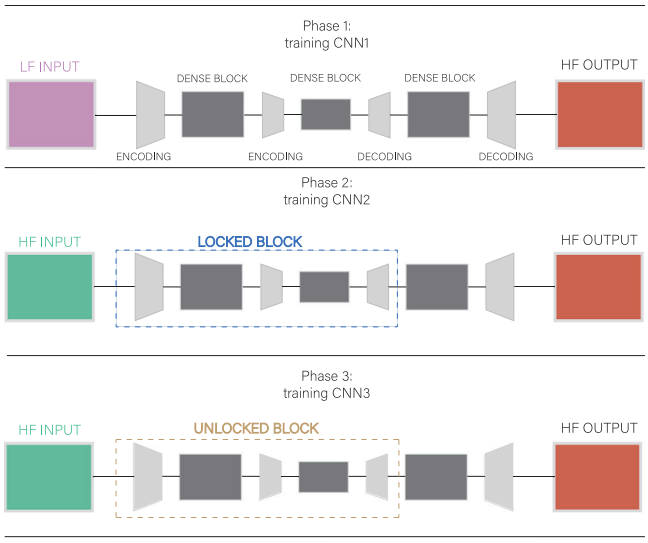


Fig. 1. Workflow of CNN training on two levels of MF data via TL. During Phase 1, LF input data are used and the weights ($\mathbf{w}_{\text{LF}}$) are updated in each layer. In Phase 2, HF input data are used to update the weights in the final dense block and decoding layer ($\mathbf{w}_{\text{HF}}$) while the others are frozen. In Phase 3, all weights are unlocked and updated during training on the same HF data.

In detail, as shown in Fig. 1, the TL applied to two levels of MF data consists of three phases involving the training of CNNs denoted in the following as CNN_1, CNN_2, and CNN_3. In Eq. (2), let N_{w} represent the number of weights in the CNN preliminary trained on a small number of HF data. In the first phase, we perform the training of CNN_1 on LF data, and the weights $\mathbf{w}_{\text{LF}} = (w_1, w_2, \dots, w_{N_{\text{LF}}})^{\top}$ with $\mathbf{w}_{\text{LF}} < N_{\text{w}}$ are all updated. In the second phase, we train CNN_2 on the HF data by minimizing the loss function (2) over the weights $\mathbf{w}_{\text{HF}} = (w_{N_{\text{LF}+1}}, w_{N_{\text{LF}+2}}, \dots, w_{N_{\text{w}}})^{\top}$ of the last dense block and decoding layer, while keeping the remaining weights $\mathbf{w}_{\text{LF}}$ of the first layers of the architecture fixed at the values got in the first phase. In the third phase, we train CNN_3 on HF data by allowing all weights $\mathbf{w}$ of the CNN_2 to be updated during the minimization.

This idea can be extended at three levels of MF data as depicted in Fig. 2, with five training phases in this case and where VLF data refer to the values of the state variable $\mathbf{u}(\mathbf{x}_i,\ t_k)$ obtained solving (1) on a very coarse ($N_{\text{el}} = N_{\text{el}}^{\text{VLF}}$) mesh.

2.3. Conceptual difference in direct and inverse CNN configurations from a transfer learning perspective

When training a CNN where input–output images are 1 to N, the network performs an expansion operation, which requires learning a rich latent representation from limited input and then decoding it into multiple structured outputs, a task generally accomplished by using encoder–decoder architectures. This setup increases the complexity of the output layer, which must handle multiple channels and preserve spatial coherence across all N outputs. Moreover, in this case, the network is asked to generate information that may not be explicitly contained in the single input, making it prone to underfitting or generating overly smooth predictions. Conversely, when training with N input images to produce a single output image, i.e., N to 1, the network performs a compression task, which requires extracting and integrating spatial features from multiple input channels into a unified representation. This is the more typical use of CNNs. This setup places more demand on the early layers to successfully capture correlations across input channels. In this case, the network is more prone to overfitting if

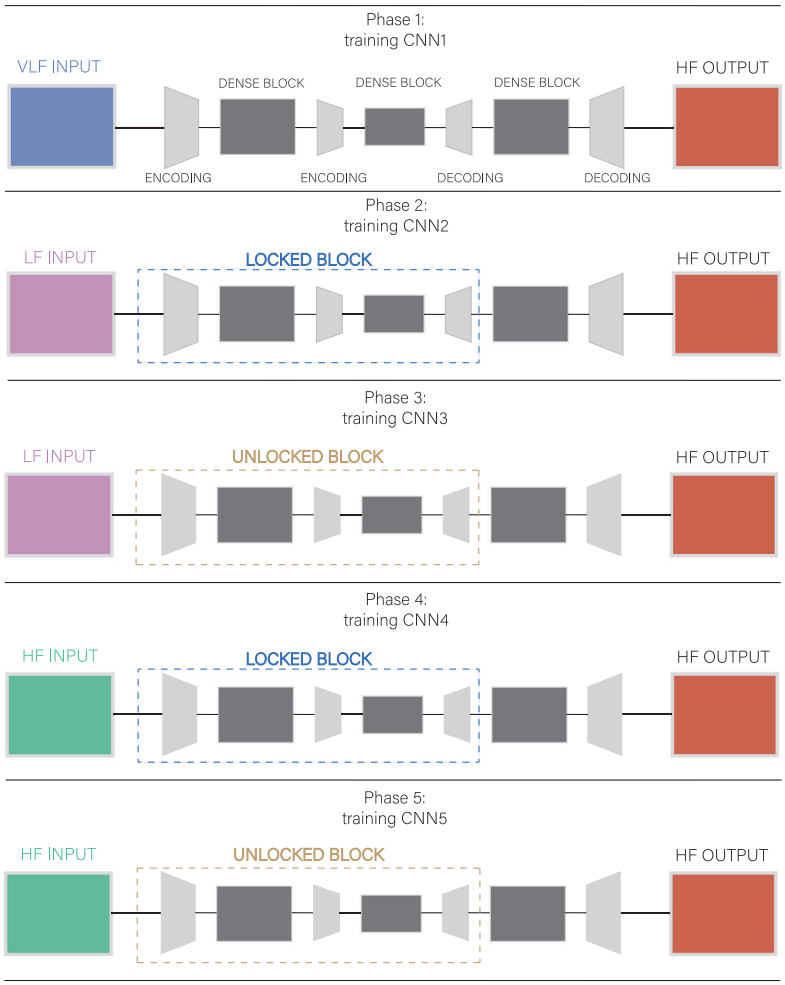


Fig. 2. Workflow of CNN training on three levels of MF data via TL. Phases 1–3 replicate the three phases used in the case of two levels of data for VLF and LF data. Phases 4–5 are analog to Phases 2–3 in the case of two levels of data.

the number of input channels is large and training data is limited. In addition, in terms of sensitivity, this case is higher since noise across multiple input channels can significantly affect the feature aggregation process, leading to degraded or unstable predictions. Indeed, from a calibration perspective, the loss function is applied to a single target and mostly relies on the accurate multi-channel input processing step. As such, considering that our analysis is focused on the case of limited HF input data, to analyze the effect of TL on MF data for the CNN operating N to 1, is not only conceptually different but more significant under this perspective.

When applying TL to CNN surrogates for multiphase flow, starting with training on LF data and progressively fine-tuning on HF data while gradually unfreezing network layers, the benefits manifest differently depending on the modeling direction. If the permeability field is the input and multiple saturation maps are the output, the CNN initially learns general spatial-to-temporal patterns of saturation behavior, capturing broad physical relationships such as flow directionality, heterogeneity effects, or large-scale saturation trends from the LF simulations, and then, the fine-tuning with HF data is useful to refine these patterns and reflect more realistic features, such as sharper fronts or small-scale nonlinear dynamics, with early layers preserving general permeability features while deeper layers adapt to the increased physical complexity. This allows the network to improve output accuracy without retraining from scratch, even when HF data is limited. In the opposite case, TL plays a critical role in stabilizing the inversion process, which is inherently ill-posed. The LF phase helps the network learn a rough inverse mapping, extracting coarse permeability structure from saturation data, while HF fine-tuning sharpens the reconstruction of fine-scale heterogeneities and improves sensitivity to subtle spatial variations. Progressive unfreezing is especially helpful here, as it prevents overfitting early in the process and allows a smooth transition toward more accurate inversion.

3. Application

3.1. High-fidelity forward model

Dealing with multi-phase flow in porous media is known to be extremely challenging. The computational cost of solving the governing PDEs, which are characterized by a high level of nonlinearity and stiffness, is generally not negligible and can hinder uncertainty quantification and detailed analyses requiring numerous simulations in the parameter space.

Here, we are examining the horizontal flow of two incompressible and immiscible fluids, each with viscosity represented by μ_1 and μ_2, through a heterogeneous, incompressible, and isotropic porous medium D. Propagation of the saturation maps $\mathrm{S}_l(\mathbf{x}, t)$ of the lth phase ($l = 1$, 2) is described by the mass conservation equation:

$$\phi \frac{\partial S_l}{\partial t} + \nabla \cdot \mathbf{v}_l + q_l = 0, \qquad \mathbf{x} \equiv (x_1, x_2)^\top \in D, \quad t \in [0, T], \tag{3}$$

where the porosity is assumed to be constant at 0.25. The phase saturation, denoted as $S_l(\boldsymbol{x}, t)$, is subject to the constraint $S_1 + S_2 = 1$.

The source/sink term, q_l, is assumed to be zero. The time domain t ranges from zero to a specified terminal time T. The macroscopic velocity, denoted as $\mathbf{v}_l(\mathbf{x}, t)$, is described by the generalized Darcy's law (4) for the lth phase

$$\mathbf{v}_l = -\mathbf{k}(\mathbf{x})\frac{k_{rl}}{\mu_l}\nabla P_l, \tag{4}$$

where $\mathbf{k}(\mathbf{x})$ is the intrinsic permeability tensor (since the medium is isotropic, $\mathbf{k}(\mathbf{x})$ becomes a scalar and will be denoted by k), k_{rl} is the relative permeability of the lth phase that depends on the corresponding saturation by the Brooks-Corey constitutive model, $k_{rl} = k_{rl}(S_l)$ (Corey, 1954). According to Taverniers et al. (2020), the capillary forces are neglected, assuming pressure within the two phases to be equal, $P_1 = P_2 \equiv P(\mathbf{x}, t)$.

We solve numerically this multi-phase flow scenario in a two-dimensional domain D of dimension 150 m × 150 m with impermeable boundaries at the top (Γ_t) and bottom (Γ_b). Along the left (Γ_l) and right (Γ_r) boundaries, the Dirichlet boundary conditions are imposed

$$\frac{\partial P}{\partial x_2} = 0, \qquad \mathbf{x} \in \Gamma_b \cup \Gamma_t, \tag{5}$$

with P = 10.2 MPa and S_1 = 1.0 for $\mathbf{x} \in \Gamma_l$, and P = 10.1 MPa for $\mathbf{x} \in \Gamma_r$. The initial conditions are: $P(\mathbf{x}, 0)$ = 10.1 MPa; $S_1(\mathbf{x}, 0) = 0$, with $\mathbf{x} \in D$ (see Fig. 3). While all input parameters are assumed to be constant and known, the intrinsic permeability $k(\mathbf{x})$ is modeled as a second-order stationary random field, where $Y(\mathbf{x}) = \ln k$ is multivariate Gaussian with mean $\langle Y \rangle = 0$, variance $\sigma_Y^2 = 2.0$, and an exponential two-point covariance $C_Y(\mathbf{x}, \mathbf{y}) = \sigma_Y^2 \exp(-|\mathbf{x}, \mathbf{y}|/\lambda_Y)$ with a correlation length of λ_Y = 19 m.

This value plays a crucial role in defining the separation between fidelity levels. The spatial domain of 150 m × 150 m is discretized on a 128 × 128 grid for HF simulations, resulting in a cell size of approximately 1.17 m, so that each correlation length spans about 16 grid cells, allowing fine-scale heterogeneity to be accurately resolved. To generate LF and VLF data, we first upscale the conductivity field to 64 × 64 and 32 × 32 grids, respectively. The corresponding saturation maps are then downscaled back to 128 × 128 resolution, so that all inputs to the CNN surrogate share the same dimensionality while retaining the degradation in spatial detail due to coarsening (see Section 3.2). This structured reduction in resolution, relative to the fixed λ_Y, defines meaningful fidelity levels. In the 64 × 64 and 32 × 32 cases, the correlation length spans approximately 8 and 4 cells, respectively, leading to a loss of small-scale variability. This setup supports the proposed TL strategy, where initial training on coarse data enables the surrogate to capture broad spatial patterns, and subsequent fine-tuning with HF data refines local heterogeneity. The choice of the selected correlation length thus ensures a sufficient fidelity gap to make TL both effective and significant.

Eqs. (3) and (4) are discretized using a finite volume scheme in space and an implicit Euler scheme in time. We use the Newton–Raphson method to obtain iteratively the solution at each time step, and the modified Appleyard saturation update dumping (Appleyard et al., 1981) to improve the convergence. To guarantee the convergence of both flow (pressure) and transport (saturation) solutions, three convergence criteria are defined: the normalized residual norm, the maximum saturation update, and the maximum relative pressure update with tolerances of $\epsilon_1 = 10^{-6}, \epsilon_2 = 10^{-2}$, and $\epsilon_3 = 10^{-3}$ respectively (Taverniers et al., 2020).

3.2. Multifidelity data acquisition

In this Section, we describe the data acquisition procedure for generating HF, LF, and VLF data for CNN training according to the TL framework. In Fig. 4, the flowchart connecting fine, coarse, and downscaled snapshots is shown, while the data acquisition procedure is summarized in Fig. 5.

Mesh coarsening is performed using a uniform coarsening strategy, where every second node in each spatial direction is removed, effectively doubling the element size. This approach reduces the total number of elements by a factor of four in 2D. We apply coarsening globally and we verify that the solution retains acceptable accuracy while significantly reducing computational time.

3.2.1. High-fidelity data

HF training data are generated by solving (3)–(4) on a grid 128 × 128, obtaining N_{ts} = 16 temporal snapshots of the saturation $S_1(\mathbf{x}, t)$. To capture the randomness in permeability fields, according to the "rule of thumb" requirement, the numerical mesh should have at least four elements of the length of the spatial discretizations, Δx, per correlation length, λ_Y, i.e., $4\Delta x \leq \lambda_Y$. In our case, with Δx = 1.17 m and the correlation length λ_Y = 19 m, this requirement is satisfied.

3.2.2. Low-fidelity data

LF data are obtained by solving (3)–(4) on a "coarse" mesh of size 64 × 64. Specifically, the mesh coarsening is achieved by upscaling the random permeability generated at the finer scale (128 × 128). With λ_Y = 19 m, a grid of size 64 × 64, and Δx = 2.34 m, we ensure that $4\Delta x \leq \lambda_Y$.

Among different upscaling strategies, we have chosen the one proposed by Durlofsky (2005). We denote the fine-scale permeability tensor as $\mathbf{k}_{fine}$. We consider one realization of this random field and obtain the corresponding realization of its coarse-scale counterpart, $\mathbf{k}_{coarse}$, whose diagonal elements are computed using distance-weighted harmonic mean in the direction of flow, and the distance-weighted arithmetic mean in the direction perpendicular to the flow, while the off-diagonal components are zero. The resulting tensor $\mathbf{k}_{coarse}$ is still diagonal but anisotropic (see Algorithm 1). The selected method yields directionally consistent effective properties, by generating anisotropic diagonal coarse-scale permeability tensors from isotropic fine-scale fields. While more sophisticated global upscaling approaches are available, the applied local upscaling strategy offers an effective, computationally efficient alternative that preserves the relevant flow features (Taverniers et al., 2020).

After upscaling the permeability field, we solve Eqs. (3)–(4) to obtain 16 temporal snapshots of the saturation maps $S_1(\mathbf{x}, t)$, each with dimensions of 64 × 64, denoted as S_{coarse}. To match the dimensions of HF data, we then apply a downscaling technique to the 64 × 64 saturation maps by taking the Kronecker product between these LF data matrices and a 2 × 2 matrix of ones, thus allowing us to obtain S_{fine}^{ds} (see Algorithm 2). The assumption underlying this approach is that the spatial covariance is approximately separable, i.e., it can be modeled as a Kronecker product of lower-dimensional covariances. This assumption is reasonable when the heterogeneity in the permeability field is mildly anisotropic or exhibits layered regular patterns, which is often the case after permeability upscaling, where small-scale variability is averaged out. Moreover, the Kronecker approach is advantageous here due to its computational efficiency.

3.2.3. Very low-fidelity data

VLF data are obtained by solving (3)–(4) on a very coarse mesh of size 32 × 32. As for LF data, the mesh coarsening is achieved by upscaling the random permeability generated at the finer scale (128 × 128). With λ_Y = 19 m, a grid of size 32 × 32, and Δx = 4.69 m, we ensure that $4\Delta x \leq \lambda_Y$. To match the dimensions of HF data, downscaling is applied to the 32 × 32 saturation maps. As in the previous case, Algorithms 1–2 summarize the procedure, but this time the upscaling factor is $n = N/4$ and the 32 × 32 saturation maps are downscaled by taking the Kronecker product between these VLF data matrices and a 4 × 4 matrix of ones.

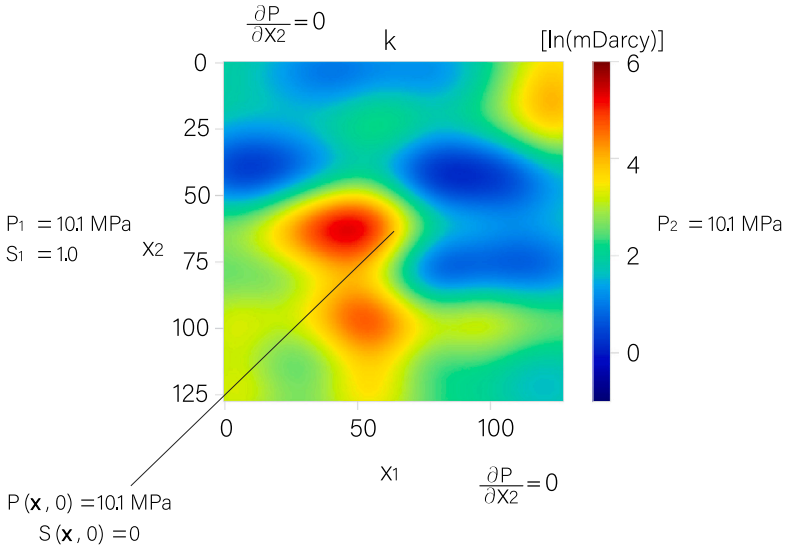


Fig. 3. Domain setup for the two-phase flow problem with one realization of the log permeability field.

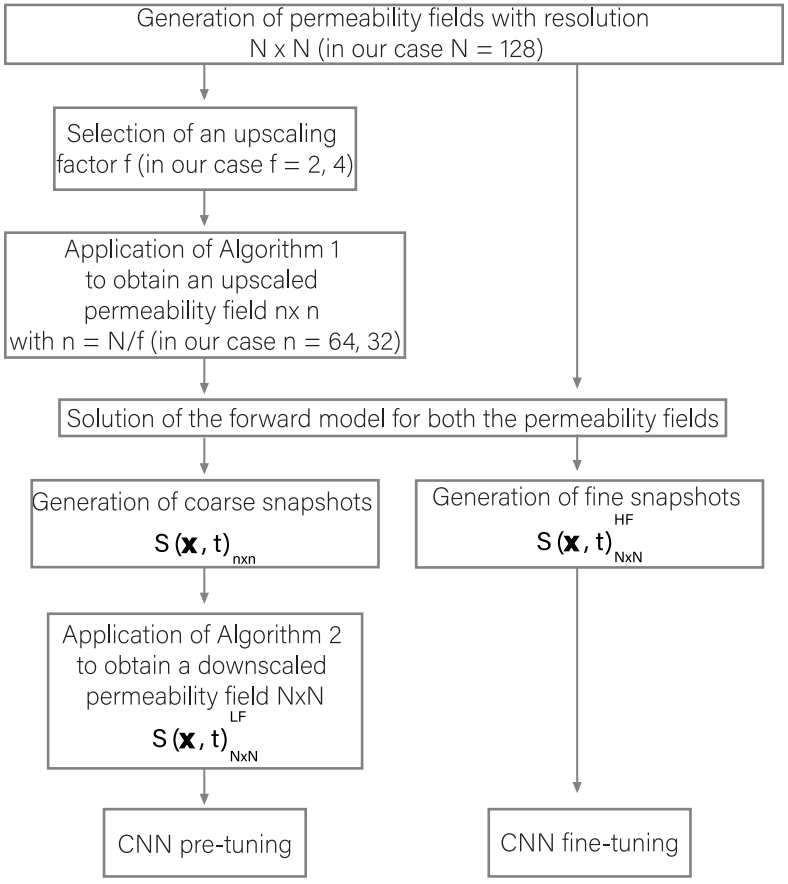


Fig. 4. Flowchart connecting fine, coarse, and downscaled snapshots through Algorithms 1 and 2.

4. Inverse CNN surrogate and hyperparameter performance

The training of the inverse CNN surrogate is performed to learn the mapping from temporal snapshots of saturation maps back to the permeability field using MF data. With the CNN architecture represented in Fig. 6, we patch each phase of training according to the TL framework (Figs. 1–2).

The design of the network configuration follows Song and Tartakovsky (2022). The model block description and the input and output dimensions of each model block are reported in Table 1. We adopt the same CNN architecture used in the 1 to N forward problem in Song and Tartakovsky (2022) for the N to 1 inverse configuration analyzed here, in order to create a controlled comparison framework that isolates the effects of data fidelity and transfer learning. While this architecture may not be optimal for the inverse task, where early fusion and robust feature aggregation across multiple input channels are typically more effective, it supports our goal of maintaining consistency. Architectures

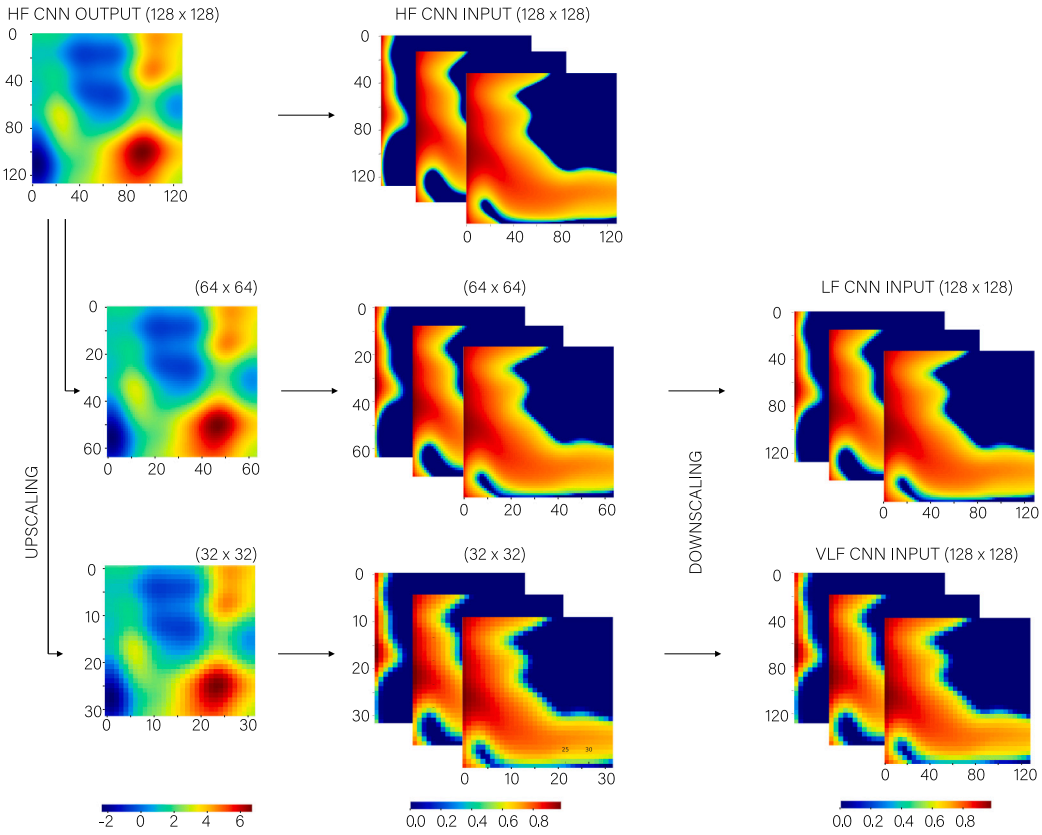


Fig. 5. Generation of HF, LF, and VLF data for CNN training.

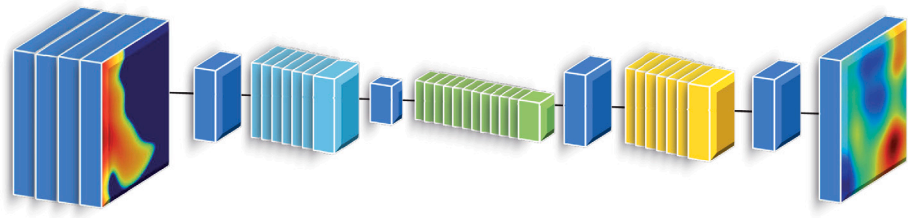


Fig. 6. CNN architecture. Details are reported in Table 1.

better suited for N to 1 compression tasks often use sequential convolutional layers, residual connections, and downsampling to integrate spatial and temporal features efficiently, avoiding upsampling and transposed convolutions that are unnecessary or potentially destabilizing for single-output predictions. Since our objective is not to optimize inversion performance but to explore the behavior of transfer learning in a multi-fidelity setting, we deliberately reuse the same architecture to control for architectural variables. This ensures that observed differences in performance can be attributed to the learning strategy and data fidelity, rather than to differences in model design.

In support of this decision, we performed exploratory tests by modifying the number of layers within the dense blocks and found that the configuration used in Song and Tartakovsky (2022) offers good empirical performance and training stability, even in the inverse setting. Consequently, our efforts focused on tuning training-related hyperparameters (e.g., learning rate, weight decay, factor, and minimum learning rate). A more extensive sensitivity analysis on architectural parameters could be valuable future work, particularly for adapting the surrogate model to different inverse problems.

The model implementation and training is done using `Pytorch` and other source packages. The computations are carried out utilizing 32 cores Intel Ice Lake @2.6 GHz CPU 512 GB RAM, 4 NVIDIA Ampere A100 GPU 64 GB vRAM. (Although 32 cores on a single node are available, only 1 core per node is used. All 4 GPUs are used).

As mentioned, CNN's performance is influenced by key hyperparameters such as Learning Rate (LR) and Weight Decay (WD), managed by the Adam optimizer. The Adam optimizer is commonly used for optimizing deep learning models and adjusts the LR and WD during training to improve convergence and prevent overfitting (Kingma and Ba, 2014). Additionally, a factor (F) and the minimum learning rate (mLR) are controlled by the ReduceLROnPlateau scheduler, which adjusts the learning rate during training based on the validation loss, enhancing the efficiency and convergence of the training process for CNNs (Li and Arora, 2019). While CNN training includes numerous additional hyperparameters, we adhere to their default settings in `Pytorch` (Paszke et al., 2019).

The investigation involves iterating through variations of LR, WD, F, and mLR, in this specific order. For each hyperparameter, we fix the best-performing value before moving on to the next, using 100 HF data points and training for 200 epochs. The selected values are those that minimize the RMSE on the HF test data (see Fig. 7). This process is a form of empirical tuning, similar to a coarse grid search over a

Table 1
Dense encoder–decoder CNN architecture. Each layer of the dense blocks, as well as the encoding and decoding transition layers, consists of these consecutive operations: batch normalization (BN), followed by a rectified linear unit (relu), convolution (conv), and transposed convolution (convTranspose) for the encoding and decoding phases, respectively.

Layer	Activation functions	Input	Output size
Input: Saturation map $\hat{S}$		$n_{ts} \times 128 \times 128$	
Transition layer	conv (k5, s2, p2)	$n_1 \times 128 \times 128$	$n_2 \times 64 \times 64$
Dense Block (7 layers)	BN, relu, conv (k3, s1, p1)	$n_2 \times 64 \times 64$	$n_3 \times 64 \times 64$
Transition layer (encoding)	BN, relu, conv(k1, s1);		
	BN, relu, conv(k3, s2, p1)	$n_3 \times 64 \times 64$	$n_4 \times 32 \times 32$
Dense Block (12 layers)	BN, relu, conv (k3, s1, p1)	$n_4 \times 32 \times 32$	$n_5 \times 32 \times 32$
Transition layer (decoding)	BN, relu, conv(k1, s1);		
	BN, relu, convTranspose(k4, s2, p1)	$n_5 \times 32 \times 32$	$n_6 \times 64 \times 64$
Dense Block (7 layers)	BN, relu, conv (k3, s1, p1)	$n_6 \times 64 \times 64$	$n_7 \times 64 \times 64$
Transition layer	BN, relu, conv(k1, s1);		
	convTranspose (k4, s2, p1)	$n_7 \times 64 \times 64$	$1 \times 128 \times 128$
Output: Permeability field k		$1 \times 128 \times 128$	

Note: n_{ts} denote the number of input time steps of saturation maps, $n_{ts} = 16$. The number of channels, n_1, n_2, n_3, n_4, n_5, n_6 and n_7 are the following: 64, 344, 172, 652, 326, 606, 303, respectively. The values of kernel size, k, stride, s, and zero-padding, p, are denoted by the number at the right; for example for $k5, s2, p2 : k = 5, s = 2$, and $p = 2$. The convolution kernels are the same for all convolutional layers within the dense blocks. The first and second convolution kernels correspond to the first and second (transposed) convolutions of the encoding–decoding layers, respectively.

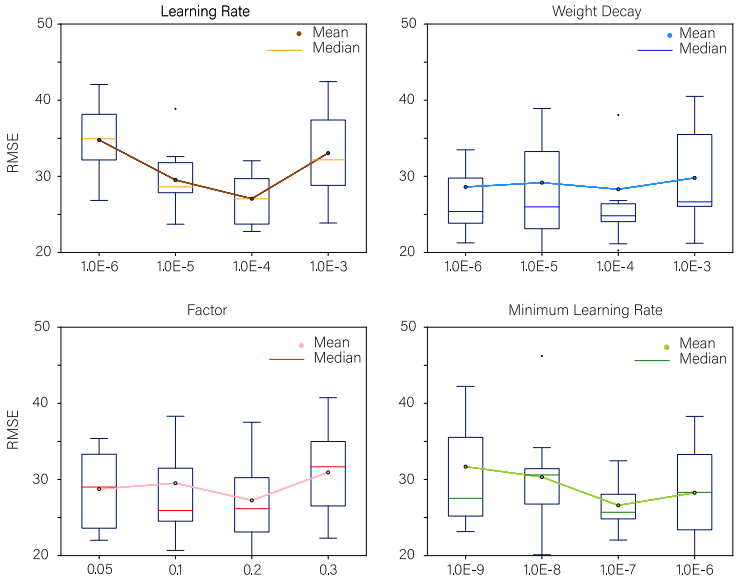


Fig. 7. Hyperparameter performance measured in terms of root mean square error (RMSE) on the test data. Each box plot displays the corresponding median (colored line) and mean (point) values calculated from 10 training sessions.

predefined subset of values. These optimal values then serve as the starting point for transfer learning with MF data. For the two-level MF transfer learning setup, organized into three sequential training phases, we adopt a progressive training strategy. Starting from the best LR identified in the HF-only tuning phase, we progressively reduce it in later phases to allow for finer weight updates as the model became more specialized. A similar rationale guides the choice of epochs, with 200 epochs in the initial phase and reduced durations, 180 and 160, in the following phases. This choice is supported by the analysis of the loss function at each training phase, which shows that RMSE tends to plateau more rapidly in the later training stages (see Section 5). In the three-level MF case, we maintain a constant number of epochs of 200 across all phases while still applying a decaying LR schedule, as shown in Table 2. This choice balances consistency with the need for finer adjustments as the fidelity level increases. Overall, hyperparameter tuning and scheduling are designed to control learning dynamics across fidelities, enabling a fair and effective evaluation of transfer learning performance.

Table 2
Learning rate and epoch values at each training phase in the case of either two or three levels of MF data.

TL strategy	Phase	Learning rate	Epochs
MF - 2 levels	1	$1.0 \cdot 10^{-4}$	200
MF - 2 levels	2	$0.5 \cdot 10^{-4}$	180
MF - 2 levels	3	$0.5 \cdot 10^{-5}$	160
MF - 3 levels	1–3	$1.0 \cdot 10^{-4}$	200
MF - 3 levels	4–5	$0.5 \cdot 10^{-4}$	200

5. Results and discussion

First, we compare the accuracy of the CNNs trained on only HF and LF data, hereinafter CNN^{HF} and CNN^{LF}, respectively, for different training data generation budgets. We observe in Fig. 8, that for a data generation budget of less than 12 h, CNN^{LF} outperforms CNN^{HF}, in terms of RMSE. This is because limited budgets do not allow for the generation of sufficient HF data for training. As the budget increases, the error associated with CNN^{LF} also increases, reaching a relatively

Algorithm 1 Upscaling algorithm pseudocode.

Require: $\mathbf{k}_{fine} \in \mathbb{R}^{N\times N}$
Ensure: $N \mod 2 \equiv 0$

function UPSCALE($\mathbf{k}_{fine}$, N, direction)
  $n \leftarrow N/2$
  for $i \leftarrow 1, \dots, n$ **do**
    $l \leftarrow (i-1) * 2 + 1$

    for $j \leftarrow 1, \dots, n$ **do**
      $m \leftarrow (j-1) * 2 + 1$

      if direction is X **then**
        $k_{coarse}(i,j) = \mathcal{AA}(\mathcal{HA}(k_{fine}(l,m), k_{fine}(l,m+1)), \mathcal{HA}(k_{fine}(l+1,m), k_{fine}(l+1,m+1)))$

      else if direction is Y **then**
        $k_{coarse}(i,j) = \mathcal{AA}(\mathcal{HA}(k_{fine}(l,m), k_{fine}(l+1,m)), \mathcal{HA}(k_{fine}(l,m+1), k_{fine}(l+1,m+1)))$

      else
        $k_{coarse}(i,j) = \mathcal{AA}(k_{fine}(l,m), k_{fine}(l+1,m), k_{fine}(l,m+1), k_{fine}(l+1,m+1))$
  return $\mathbf{k}_{coarse} \in \mathbb{R}^{n\times n}$

Algorithm 2 Downscaling algorithm pseudocode.

Require: $S_{coarse} \in \mathbb{R}^{n\times n}$

function DOWNSCALE(S_{coarse})
  $S^{ds}_{fine} \leftarrow S_{coarse} \otimes \mathbf{1}_{2\times 2}$
return $S^{ds}_{fine} \in \mathbb{R}^{N\times N}$ with $N = 2n$

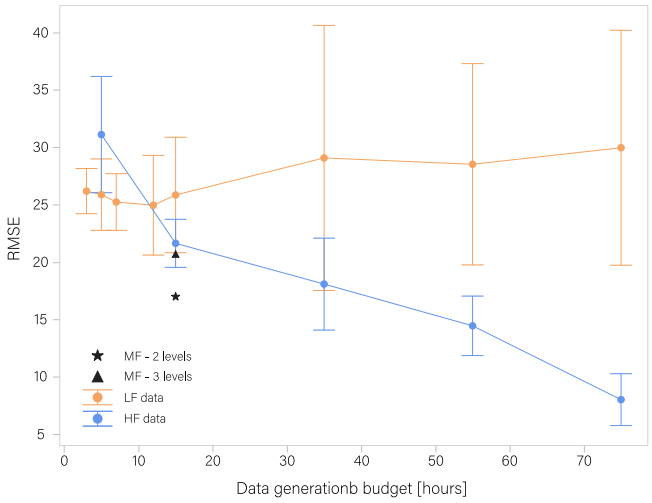


Fig. 8. RMSE on HF test data for alternative CNN training strategies as a function of the budget allocated for training data generation. The CNNs are trained on HF (blue circles), LF (orange circles), two levels (the black star), or three levels (the black triangle) of MF data. Each point in the Figure is the average value of the RMSE over 20 training iterations, while the variability among the iterations is described through the standard deviation.

Table 3
Numbers of HF, LF, two levels and three levels of MF data generated with a budget of 15 h and a computer Intel(R) Core(TM) i7-9700 CPU (3 GHz), 32 GB RAM.

Amount and quality of data	Generation cost
214 HF	15 h
1011 LF	15 h
4655 VLF	15 h
754 MF - 2 levels	15 h
2069 MF - 3 levels	15 h

constant value due to training the network on an increasing amount of less accurate data that does not provide additional value for training purposes. In contrast, the RMSE associated with CNN^{HF} continues to decrease as expected.

Then, to assess the benefit of TL, by setting the data generation budget to 15 h, we demonstrate that training the CNN on two levels of MF data results in a significantly lower RMSE compared to the one associated with CNN^{HF} and CNN^{LF}. Also, the CNN trained on three levels of MF data has a RMSE value that deviates from that of CNN^{LF} and remains lower than that of CNN^{HF} on average. The multifidelity cases represented in Fig. 8 are obtained with an optimal ratio of HF-LF and HF-LF-VLF data, respectively, that minimizes the RMSE, as we discuss below.

Focusing on the computational cost, in Table 3 we compare the number of HF, LF, and VLF data generated with a budget of 15 h. The cost of a single simulation is the computational time for the numerical solution of the forward model using a `Matlab`-based multi-phase flow simulator (Song and Tartakovsky, 2022) on a computer with an Intel(R) Core(TM) i7-9700 CPU (3 GHz), 32 GB RAM. We observe that HF data generation is about five times more expensive than LF data and about 22 times more expensive than VLF data. The same Table also reports the number of data for the optimal cases of two and three levels of MF data.

In the case of two levels of MF data, we examine how different ratios of HF and LF training data, for a data generation budget of 15 h, influence the accuracy of the CNN (Fig. 9). We observe that HF-LF ratios of 1:10-1:5 provide significantly higher accuracy than using only HF training data. We observe that the mean across 20 train-test attempts is slightly lower for the ratio of 1:10 compared to 1:5, which has a significantly thinner error bar.

The reuse of an architecture originally designed for the 1 to N forward problem in the N to 1 inverse case may help explain why the optimal HF-LF data ratio shifts to 1:10. Since the architecture includes a relatively heavy decoder component with transposed convolutions suited for output expansion, it may be overparameterized for the inverse task, where only a single permeability map is predicted. In this setting, using a large amount of HF data could lead to overfitting or inefficient learning, whereas relying more heavily on LF data acts as a natural regularizer. The smaller amount of HF data then serves to anchor the network without overwhelming it, allowing better generalization. Thus, while not the only factor, the architectural mismatch may contribute to why a lower HF-LF ratio yields the best performance in the N to 1 case.

Based on this evidence, in the case of three levels of MF data, we examine three distinct scenarios in which the ratio between VLF-LF and LF-HF data is given by a factor f whose value is either 10, 5, or 2. The number of data for each scenario is detailed in Table 4. The best accuracy in terms of RMSE occurs for $f = 5$, which corresponds to about 70 HF data in the MF mix. Overall, our results align with the ratio 1:5, as recommended in a theoretical multilevel Monte Carlo study (Taverniers et al., 2020), and with empirical findings from Song and Tartakovsky (2022).

The CNN training is performed using a 32 cores Intel Ice Lake @ 2.6 GHz CPU 512 GB RAM, 4 NVIDIA Amper A100 GPU 64 vRAM (although 32 cores on a single node are available, only 1 core per node is used. All 4 GPUs are used) (Turisini et al., 2024). In Table 5 the training cost using HF, LF, and two or three levels of MF data, for the budget of 15 h, are reported. The training cost using only LF data is slightly higher than when using HF data, as the training dataset size

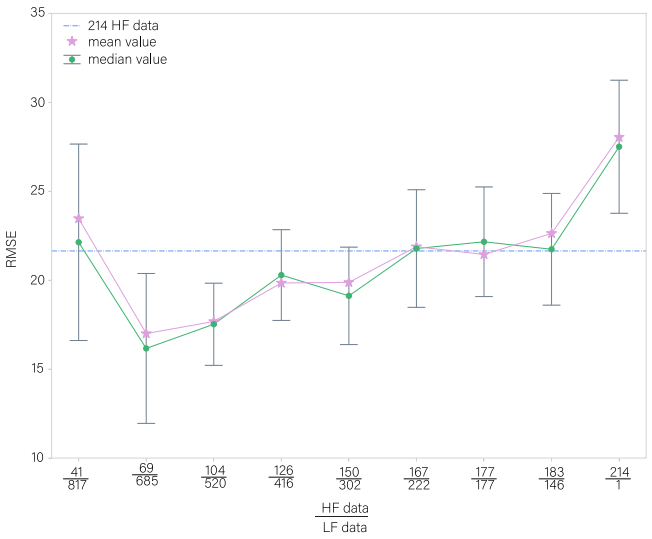


Fig. 9. Influence on the CNN accuracy of different ratios of HF-LF data used for training in the two-level MF data scenario, generated with a budget of 15 h. For each ratio, the CNN training and test are repeated 20 times. While mean and median RMSE values are represented by star and point symbols, the error bars indicate the standard deviation. The blue dot-dash line represents the mean value of the CNN trained using only HF data.

Table 4
Influence on the CNN accuracy of different ratios f of HF-LF and LF-VLF data used for training in the three-level MF data scenario, generated with a budget of 15 h. For each scenario, the CNN training and test are repeated 20 times.

	$f = 10$	$f = 5$	$f = 2$
HF	28	67	133
LF	277	334	266
VLF	2774	1668	532
RMSE	22.44	20.62	21.98

Table 5
CNN training costs for a generation budget of 15 h, performed with a 32 cores Intel Ice Lake @ 2.6 GHz CPU 512 GB RAM, 4 NVIDIA Amper A100 GPU 64 vRAM.

Data type	Training cost	No. of phases	No. of epochs
HF	219 s	1	[200]
LF	907 s	1	[200]
MF-2levels	769 s	3	[200, 180, 160]
MF-3levels	2062 s	5	[200, 200, 200, 200, 200]

increases from 214 to 1011 (see Table 3). In the case of MF data, the dataset size is smaller (754) but the training process involves three phases instead of one. That becomes more evident in the case of three levels of MF data, where the training phases increase to five.

To understand how TL and the use of multiple phases enhance and optimize the training process of CNN, Fig. 10 illustrates, on a semilog scale, the performance, measured in terms of RMSE, of each training phase as a function of the number of epochs. This analysis refers to the best CNN model trained on two levels of MF data, for the data generation budget of 15 h.

Within the framework of the TL strategy, in the first phase, the dataset consists solely of LF data in input. Subsequently, in the second phase, only the last layer's weights are unfrozen and updated using HF input data, while the rest of the network remains frozen. We observe that during this phase, the loss curve decreases only marginally. This limited improvement can be attributed to the fact that unfreezing only the final layer limits the model's capacity to adapt to the HF data, especially in this context in which earlier layers are responsible for extracting meaningful features from the saturation inputs.

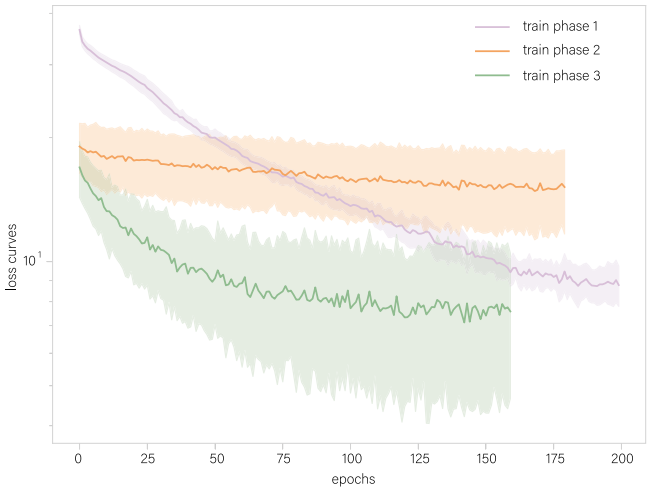


Fig. 10. Loss functions expressed in RMSE to track the quality of the learning process of the CNN trained on two levels of MF data for a generation budget of 15 h and a ratio of 1:10 between HF and LF data. The training process is repeated 20 times. The continuous lines represent the mean value, while the surrounding bands indicate the ± one standard deviation.

The third phase benefits from the knowledge acquired in the previous stages, and by unlocking all weights across all layers and utilizing HF data, the accuracy improves, though with an increase in variability, as the model fine-tunes itself to the sparse HF data. Notably, as we move from the first to the third phase, the plateau is reached more quickly, reducing the total number of epochs required. The achievement of these plateaus validates the choices made during the optimization phase, where the number of epochs for the three training phases was set to 200, 180, and 160, respectively (Table 2).

In retaining the same architecture originally designed for the 1 to N forward problem, our results further highlight the strength of the TL approach. The initial training on LF input data serves as a strong regularizer, allowing the model to learn low-complexity feature representations. The final phase, with full unfreezing and HF input, enables the model to refine those representations using richer data in a more controlled and stable manner. In this way, TL helps to mitigate the limitations of an architecture that may be overparameterized for the N to 1 task, where beginning directly with extensive HF training could otherwise lead to overfitting or inefficient learning.

Finally, we compare the predictions provided by the best CNN surrogates obtained with two and three levels of MF training data. In Fig. 11, we show two test examples of permeability fields and the correspondent approximations provided by the two surrogates. In the case of two levels of MF data, high accuracy is achieved as quantified by the corresponding scatter plots, where the regression lines are closely aligned with the bisector and R^2 values are 0.93 and 0.95, respectively. Furthermore, the probability density functions (PDFs) of the differences between exact and predicted values show a sharp peak at zero and relatively low variance, indicating high consistency. In the case of three levels of MF data, the prediction accuracy slightly decreases. However, the regression lines remain close to the bisector and the R^2 values above 0.83. The PDFs still exhibit a peak around zero, although the variance is higher.

Based on these results, we can note that, for an equal data generation budget, the computational advantage gained from generating VLF data with the forward model, thereby increasing the training dataset size, does not contribute to improving the prediction accuracy of the inverse CNN surrogate. These results suggest that, if VLF data dominate the training set, the model may struggle to adjust its internal representations effectively, particularly when only a limited amount of HF data is available to guide learning. This imbalance could lead to less

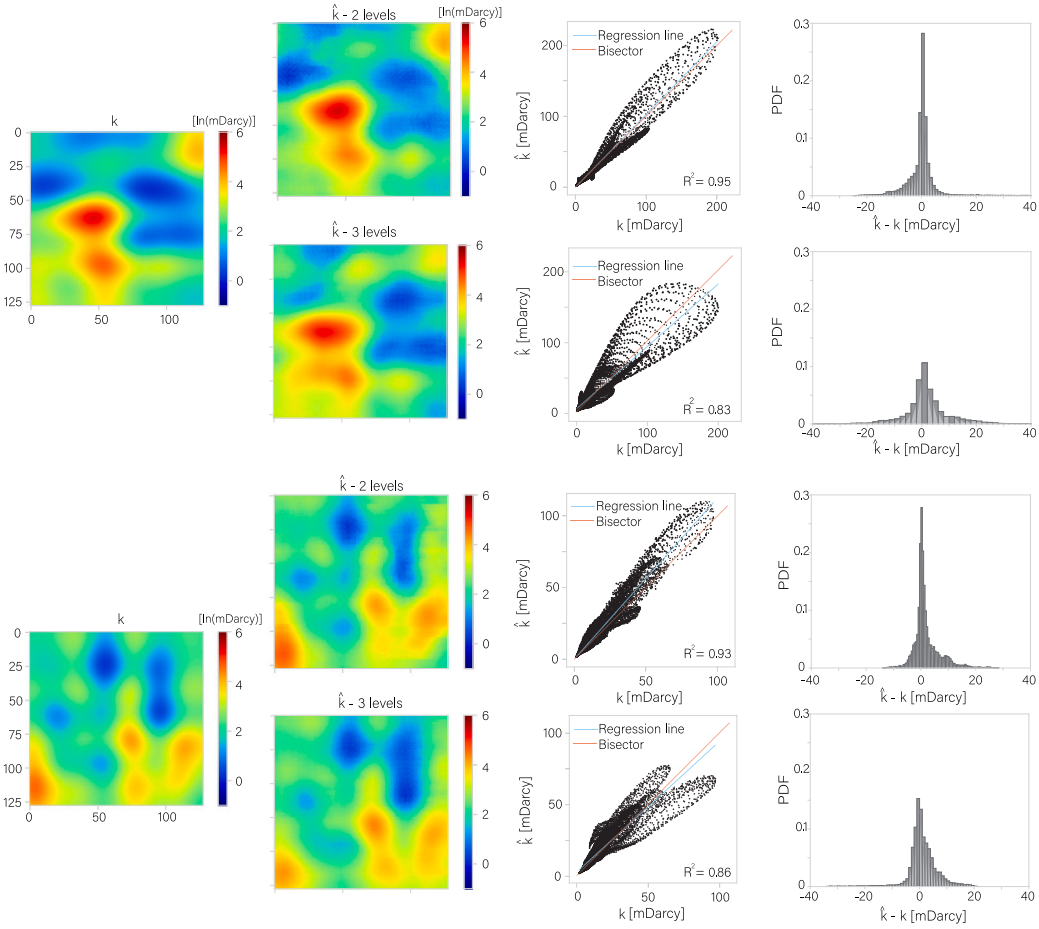


Fig. 11. Comparison between the predictions provided by the two best CNN surrogates trained on two and three levels of MF data against two test examples. For each test, the real and predicted permeability fields, k and $\hat{k}$, are represented with the correspondent scatter plots and the PDFs of the differences between real and predicted values.

informative gradients and, ultimately, suboptimal feature extraction, especially in early layers. Despite this, both surrogates maintain a key advantage in terms of computational efficiency. They require negligible computational time to predict a permeability map, approximately 0.3 s. The computations are carried out utilizing an Intel(R) Core(TM) i7-9700 CPU (3 GHz) processor, with 32 GB RAM.

We also compare the performance of the two CNN surrogates when provided with low-quality inputs. For the first test case shown in Fig. 11, instead of supplying the CNNs with HF saturation maps, we generate interpolated maps using ordinary Kriging from 8×8 data points sampled on a regular grid centered within the domain. The results, reported in the first two rows of Fig. 12, reveal that the CNN trained on three levels of MF data outperforms the one trained on two levels, with a decrease in the R^2 value of about 8% against 43% with respect to the correspondent cases reported in Fig. 11. This suggests that pre-training the surrogate with VLF data, despite offering limited physical accuracy, can increase its robustness to input degradation. Through the TL framework, early exposure to coarse or incomplete data encourages the network to develop low-level features and priors that are more tolerant to sparsity and interpolation artifacts, making the model more stable. This behavior is particularly advantageous in practical scenarios where data availability is limited or unevenly distributed in space.

We further evaluate the surrogates using noisy input maps, created by adding uniform white noise in the range of ± 0.1 to the saturation fields. As shown in the third and fourth rows of Fig. 12, the CNN trained on two levels of MF data performs better in this case, as the added noise does not significantly impair its ability to interpret the input saturation maps. The decrease in the R^2 value is about 2%, for two levels, against 15%, for three levels, with respect to the correspondent cases reported in Fig. 11. Here, the presence of VLF data during training does not appear to confer an advantage, likely because the VLF fields used in training are smooth and physically coherent, whereas the added noise in this test scenario introduces unstructured and localized perturbations. The surrogate trained without the VLF phase may have retained stronger reliance on HF-derived features, allowing it to better filter or ignore high-frequency noise. This highlights a trade-off. While VLF pre-training improves robustness to input sparsity and structural incompleteness, it may limit sensitivity to subtle variations when those variations deviate from the statistical patterns seen during training. Nonetheless, in both cases, the MF surrogates demonstrate resilience to common data imperfections, reaffirming their suitability for real-world applications where measurements are often sparse, and error-prone.

While our results demonstrate the effectiveness of TL in enhancing CNN surrogates for multi-fidelity inversion, it is important to recognize its limitations. The success of TL depends on the choice of fidelity levels and the relative availability of HF and LF data. Inappropriate scaling or overly coarse inputs could limit the transferability of learned features. By itself, TL does not provide systematic uncertainty quantification, which is critical in geoscience and engineering applications. One direction of future research is to combine TL with inversion-specific architectures, adaptive layer-wise strategies, or physics-informed approaches, and to integrate these surrogates with uncertainty quantification frameworks. Such developments would broaden the applicability of TL-enhanced CNNs, enabling fast, reliable, and robust inversion that explicitly accounts for prediction uncertainty.

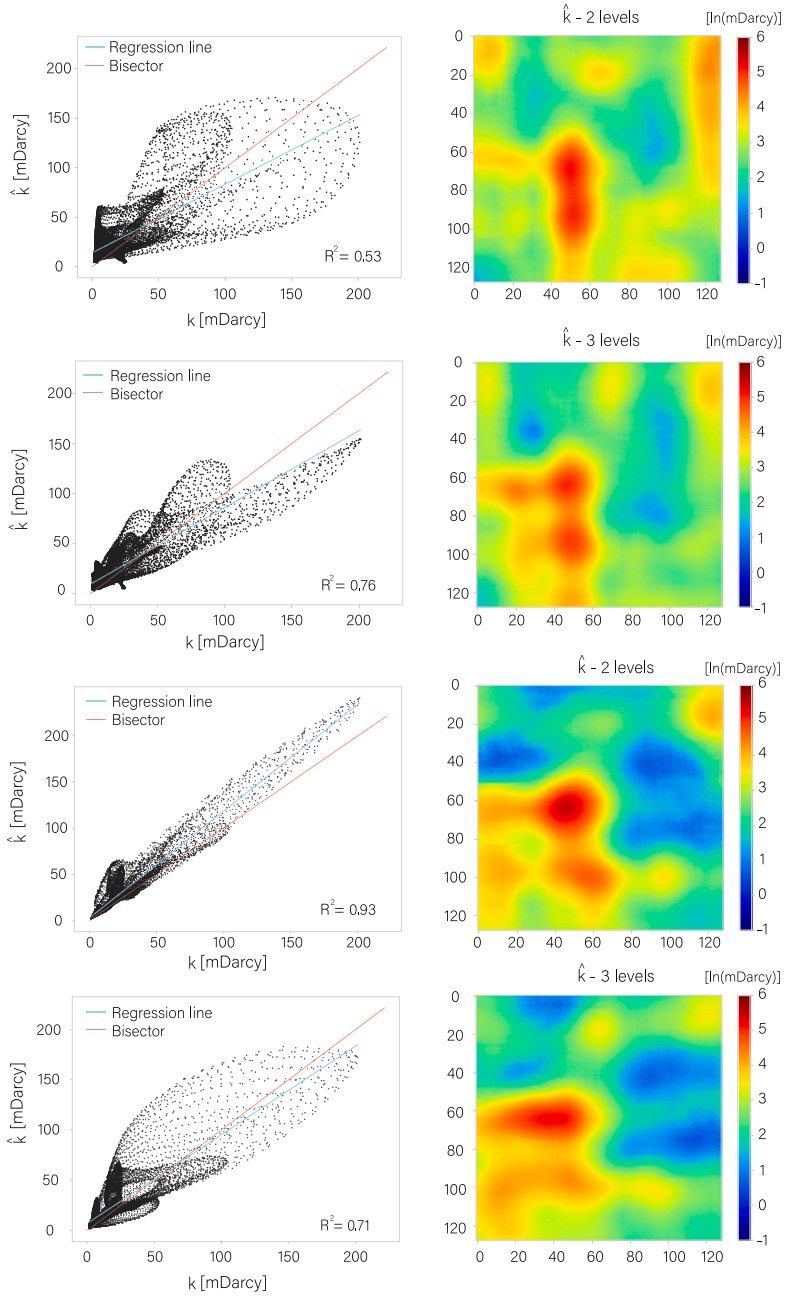


Fig. 12. Comparison between the predictions provided by the two best CNN surrogates trained on two and three levels of MF data against the first test example in Fig. 11. Here, the best models are tested with input data that are generated using ordinary Kriging from 8 × 8 data points sampled on a regular grid centered within the domain (first and second rows) or perturbed by adding uniform white noise in the range of ±0.1 (third and fourth row). For each case, the predicted permeability field, $\hat{k}$, is represented with the corresponding scatter plot between real and predicted values.

We illustrated our strategy on a 2D inversion problem with fixed boundary conditions. More complex settings are left for future research. In 3D applications, the input saturation maps and output permeability fields would increase in dimensionality, requiring larger memory and potentially deeper NN architectures to capture the added spatial complexity. Yet, the core principles of TL—progressive training from coarse to fine data, and phased refinement using high-fidelity inputs—remain unchanged. Similarly, variable boundary conditions can be incorporated by either augmenting the input channels with boundary information or by encoding boundary scenarios as additional features. In either case, training on lower-fidelity or coarsened 3D datasets allows the model to capture global flow patterns, while its refinement with high-fidelity simulations ensures accurate reconstruction of fine-scale heterogeneity under varying boundary scenarios. This approach makes TL-enhanced CNN surrogates a promising tool for realistic 3D multiphase flow inversion and other large-scale geoscience and engineering problems.

6. Conclusion

We demonstrated the effectiveness of TL with MF data for training CNN surrogates. In our application, these surrogates are designed to efficiently solve an inverse problem — permeability reconstruction in a multiphase flow scenario — by learning a mapping from a time series of saturation fields to a high-resolution permeability map. The proposed TL strategy leverages LF and VLF data in the early training stages and refines the model using HF data in later phases. This phased approach not only accelerates convergence and reduces the computational cost

but also enhances predictive accuracy, especially when using limited HF data.

Our results show that CNNs trained on two levels of MF data outperform those trained solely on HF data, offering a favorable balance between accuracy and data generation cost. When extended to a three-level MF setup, further computational savings are achieved by incorporating VLF data. While this leads to a modest reduction in accuracy, the trade-off remains acceptable in most practical contexts.

Crucially, by retaining the same network architecture initially designed for a 1 to N forward problem in Song and Tartakovsky (2022), we demonstrate that TL also plays a compensatory role. The low-fidelity training phase helps regularize a possible overparameterized model, and the final refinement phase enables adaptation to richer HF input while preserving stable representations.

Furthermore, we assess the surrogates under degraded input conditions. When provided with sparse, interpolated input maps or noisy data, MF-trained surrogates demonstrate enhanced robustness. In particular, the three-level MF model shows superior performance with interpolated inputs, highlighting the benefits of pretraining with VLF data in increasing tolerance to coarseness. Conversely, the two-level MF model is more resilient to additive noise, suggesting complementary strengths depending on the nature of the input imperfection.

Overall, our findings underscore the broader value of TL with MF data in developing fast, reliable, and generalizable surrogates. Such models are especially well-suited for real-world applications where high-quality data are limited, expensive, or noisy, making TL-enhanced CNN surrogates a powerful tool for inverse modeling in computational science and engineering.

CRediT authorship contribution statement

A. Chiofalo: Writing – original draft, Methodology, Funding acquisition, Data curation, Conceptualization. **V. Ciriello:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization. **D.M. Tartakovsky:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

AC and VC acknowledge ISCRA for awarding the project TRANSIT "Transfer learning techniques applied to contaminant source identification in groundwater systems" access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CINECA (Italy). AC received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 777822. DMT was supported in part by the Air Force Office of Scientific Research, United States under grant FA9550-21-1-0381, by the National Science Foundation, United States under award 2100927, by the Office of Advanced Scientific Computing Research (ASCR) within the Department of Energy Office of Science, United States under award number DE-SC0023163, and by the Strategic Environmental Research and Development Program (SERDP), United States of the Department of Defense under award RC22-3278.

Data availability

The code used in this study is openly available on GitHub at https://github.com/Model-Reduction-and-UQ-Group/Transfer_Learning_K_reconstruction. git. The input data are available on Zenodo at https://doi.org/10.5281/zenodo.17087811.

References

Appleyard, J., Cheshire, I., Pollard, R., 1981. Special techniques for fully implicit simulators. In: Proc. European Symp. on Enhanced Oil Recovery. Bournemouth, UK, pp. 395–408.

Barajas-Solano, D.A., Tartakovsky, D.M., 2016. Stochastic collocation methods for nonlinear parabolic equations with random coefficients. SIAM/ASA J. Uncert. Quant. 4 (1), 475–494. http://dx.doi.org/10.1137/130930108.

Chiofalo, A., Careddu, L., Ciriello, V., Tartakovsky, D., 2025. AI-enabled cardiovascular models trained on multifidelity simulations data. J. Mach. Learn. Model. Comput. http://dx.doi.org/10.1615/jmachlearnmodelcomput.2025058368.

Corey, A.T., 1954. The interrelation between gas and oil relative permeabilities. Prod. Mon. 38–41.

Durlofsky, L., 2005. Upscaling and gridding of fine scale geological models for flow simulation. In: 8th International Forum on Reservoir Simulation. Vol. 2024, Iles Borromees, Stresa, Italy, pp. pp. 1–59.

Focaccia, S., Panini, G., Pedrazzoli, P., Ciriello, V., 2021. A meta-modeling approach for hydrological forecasting under uncertainty: Application to groundwater nitrate response to climate change. J. Hydrol. 603, 127173. http://dx.doi.org/10.1016/j.jhydrol.2021.127173.

Geneva, N., Zabaras, N., 2020. Multi-fidelity generative deep learning turbulent flows. Found. Data Sci. 2 (4), 391–428. http://dx.doi.org/10.3934/fods.2020019.

Goodfellow, I., 2016. In: Courville, A., Bengio, Y. (Eds.), Deep Learning. In: Adaptive Computation and Machine Learning, The MIT Press, Cambridge, Massachusetts, Includes bibliographical references and index.

He, Q., Barajas-Solano, D., Tartakovsky, G., Tartakovsky, A.M., 2020. Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport. Adv. Water Resour. 141, 103610. http://dx.doi.org/10.1016/j.advwatres.2020.103610.

Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., Azim, M.A., 2022. Transfer learning: a friendly introduction. J. Big Data 9 (1), http://dx.doi.org/10.1186/s40537-022-00652-w.

Howard, A., Fu, Y., Stinis, P., 2024. A multifidelity approach to continual learning for physical systems. Mach. Learn. Sci. Technol. 5 (2), 025042. http://dx.doi.org/10.1088/2632-2153/ad45b2.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. http://dx.doi.org/10.48550/ARXIV.1502.03167.

Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L., 2021. Physics-informed machine learning. Nat. Rev. Phys. 3 (6), 422–440. http://dx.doi.org/10.1038/s42254-021-00314-5.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. http://dx.doi.org/10.48550/ARXIV.1412.6980.

Li, Z., Arora, S., 2019. An exponential learning rate schedule for deep learning. http://dx.doi.org/10.48550/ARXIV.1910.07454.

Libero, G., Tartakovsky, D., Ciriello, V., 2024. Polynomial chaos enhanced by dynamic mode decomposition for order-reduction of dynamic models. Adv. Water Resour. 186, 104677. http://dx.doi.org/10.1016/j.advwatres.2024.104677.

Meng, X., Karniadakis, G.E., 2020. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. J. Comput. Phys. 401, 109020. http://dx.doi.org/10.1016/j.jcp.2019.109020.

Mo, S., Zabaras, N., Shi, X., Wu, J., 2019a. Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification. Water Resour. Res. 55 (5), 3856–3881. http://dx.doi.org/10.1029/2018wr024638.

Mo, S., Zabaras, N., Shi, X., Wu, J., 2020. Integration of adversarial autoencoders with residual dense convolutional networks for estimation of non-Gaussian hydraulic conductivities. Water Resour. Res. 56 (2), http://dx.doi.org/10.1029/2019wr026082.

Mo, S., Zhu, Y., Zabaras, N., Shi, X., Wu, J., 2019b. Deep convolutional encoder-decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media. Water Resour. Res. 55 (1), 703–728. http://dx.doi.org/10.1029/2018wr023528.

Oladyshkin, S., de Barros, F., Nowak, W., 2012. Global sensitivity analysis: A flexible and efficient framework with an example from stochastic hydrogeology. Adv. Water Resour. 37, 10–22. http://dx.doi.org/10.1016/j.advwatres.2011.11.001.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: An imperative style, high-performance deep learning library. http://dx.doi.org/10.48550/ARXIV.1912.01703.

Propp, A., Tartakovsky, D.M., 2025. Transfer learning on multi-dimensional data: a novel approach to neural network-based surrogate modeling. J. Mach. Learn. Model. Comput. 6 (2), 13–27. http://dx.doi.org/10.1615/jmachlearnmodelcomput.2024057138.

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

Shen, C., Appling, A.P., Gentine, P., Bandai, T., Gupta, H., Tartakovsky, A., Baity-Jesi, M., Fenicia, F., Kifer, D., Li, L., Liu, X., Ren, W., Zheng, Y., Harman, C.J., Clark, M., Farthing, M., Feng, D., Kumar, P., Aboelyazeed, D., Rahmani, F., Song, Y., Beck, H.E., Bindas, T., Dwivedi, D., Fang, K., Höge, M., Rackauckas, C., Mohanty, B., Roy, T., Xu, C., Lawson, K., 2023. Differentiable modelling to unify machine learning and physical models for geosciences. Nat. Rev. Earth Environ. 4 (8), 552–567. http://dx.doi.org/10.1038/s43017-023-00450-9.

Song, D.H., 2022. Functional Neural Network Surrogates Trained on Scarce Data. PhD Dissertation, Stanford University.

Song, D.H., Tartakovsky, D.M., 2022. Transfer learning on multifidelity data. J. Mach. Learn. Model. Comput. 3 (1), 31–47. http://dx.doi.org/10.1615/jmachlearnmodelcomput.2021038925.

Tartakovsky, A., Barajas-Solano, D., He, Q., 2021. Physics-informed machine learning with conditional Karhunen-Loève expansions. J. Comput. Phys. 426, 109904. http://dx.doi.org/10.1016/j.jcp.2020.109904.

Taverniers, S., Bosma, S.B.M., Tartakovsky, D.M., 2020. Accelerated multilevel Monte Carlo with Kernel-based smoothing and latinized stratification. Water Resour. Res. 56 (9), http://dx.doi.org/10.1029/2019wr026984.

Turisini, M., Cestari, M., Amati, G., 2024. LEONARDO: A pan-European pre-exascale supercomputer for HPC and AI applications. J. Large Scale Res. Facil. JLSRF 9 (1), http://dx.doi.org/10.17815/jlsrf-8-186.

Zhao, M., Wang, Y., Gerritsma, M., Hajibeygi, H., 2024. A physics-constraint neural network for CO2 storage in deep saline aquifers during injection and post-injection periods. Adv. Water Resour. 193, http://dx.doi.org/10.1016/j.advwatres.2024.104837.

Zhou, Z., Tartakovsky, D.M., 2020. Markov chain Monte Carlo with neural network surrogates: application to contaminant source identification. Stoch. Environ. Res. Risk Assess. 35 (3), 639–651. http://dx.doi.org/10.1007/s00477-020-01888-9.

Zhou, Z., Zabaras, N., Tartakovsky, D.M., 2022. Deep learning for simultaneous inference of hydraulic and transport properties. Water Resour. Res. 58 (10), e2021WR031438. http://dx.doi.org/10.1029/2021wr031438.

Zhu, Y., Zabaras, N., 2018. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. J. Comput. Phys. 366, 415–447. http://dx.doi.org/10.1016/j.jcp.2018.04.018.