Math Geosci (2022) 54:177–205
https://doi.org/10.1007/s11004-021-09962-w

Automatic Semivariogram Modeling by Convolutional Neural Network

Honggeun Jo[1] · **Michael J. Pyrcz**[1,2,3]

Received: 9 May 2020 / Accepted: 19 June 2021 / Published online: 12 July 2021

Abstract Modeling the semivariogram to characterize spatial continuity requires expert geostatistical knowledge and domain expertise about the spatial phenomenon of interest. Moreover, although practitioners may have experience in semivariograms, their interpretations may vary due to experimental semivariogram noise and ambiguity. In general, modeling semivariograms remains highly subjective. This paper presents a data-driven, deep learning-based automated semivariogram modeling method known as automatic semivariogram modeling with convolution-based deep learning (ASMC) that improves the utilization of available spatial information to reduce the subjectivity of semivariogram modeling. Training models are generated by sequential Gaussian simulation (SGS) and labeled with their associated semivariogram parameters (i.e., maximum correlation length, aspect ratio of major and minor direction ranges, and azimuth of major direction). ASMC consists of two convolutional neural networks (CNNs). The first CNN model maps the sparse spatial samples to the exhaustive SGS-derived spatial models, and the second CNN maps the SGS spatial model to the semivariogram parameters. Both CNNs are trained with realistic spatial training data, and their validity is also checked with validation data withheld from training. Two-dimensional synthetic, but realistic, case studies demonstrate that the first CNN successfully learns the spatial characteristics among spatial data and generates realistic subsurface model estimates. The second CNN learns the spatial context of the estimated subsurface model and successfully predicts the semivariogram parameters with greater than 96% accuracy. The proposed

✉ Honggeun Jo
honggeun.jo@utexas.edu

[1] Hildebrand Department of Petroleum and Geosystems Engineering, The University of Texas, Austin, TX, USA

[2] Bureau of Economic Geology, Jackson School of Geosciences, The University of Texas, Austin, TX, USA

[3] Jackson School of Geosciences, The University of Texas, Austin, TX, USA

machine, deep learning-based workflow improves the utilization and objectivity of spatial data in semivariogram-based spatial continuity modeling. With the optimal design of the experiment for training and tuning of model hyperparameters, this method may be generalized for application for a wide range of spatial modeling projects.

Keywords Automatic semivariogram modeling · Machine learning · Deep learning · Convolutional neural network

1 Introduction

Characterization and modeling of the spatial continuity of the spatially sampled feature of interest are challenging and essential steps in spatial modeling workflows (David 1997; Jensen et al. 1997; Deutsch and Journel 1998; Larrondo et al. 2003; Pyrcz and Deutsch 2014; Liu and Pyrcz 2020; Cho et al. 2020; Salazar and Lake 2020; Chilès and Delfiner 2012). The semivariogram model is a spatial two-point statistic that characterizes feature variability over lag distance and is often applied in spatial modeling for estimation (e.g., kriging) or simulation (e.g., sequential Gaussian simulation). The typical procedure for calculating semivariogram models is to (1) check the stationarity of the available data over the volume of interest, (2) detrend or segment the data to ensure stationarity, (3) compute experimental semivariograms over a variety of directions to identify the directions of maximum and minimum continuity, and (4) fit multiple nested positive definite semivariogram structures to the computed experimental semivariogram.

The semivariogram model is parameterized by maximum correlation lengths (ranges), major and minor continuity direction, and contribution to variance (i.e., the proportion of the sill) for each nested structure. For example, in a typical semivariogram graph, the range is the distance, where the semivariogram reaches the sill (i.e., the upper limit of semivariogram, identical to the variance of data). If the range is in the major direction, it is called the major range, and if the range is along the minor continuity direction, it is called the minor range. Moreover, the directions of major and minor continuity are parameterized by the azimuth in the horizontal plane and dip in the vertical axis. In the case of the existence of more than one semivariogram (e.g., zonal anisotropy), contribution to the variance of each nested semivariogram structure and each semivariogram may have different ranges over different major and minor continuity directions.

The application of this approach, along with an assumption of geometric anisotropy for inference of spatial continuity for off-diagonal directions, guarantees solvable kriging systems of equations and valid, positive kriging variances (Pyrcz and Deutsch 2014). However, semivariogram modeling is generally conducted manually with incomplete, noisy, experimental semivariograms and requires a fair amount of tradecraft and geostatistical knowledge, which may not be readily available, often resulting in low repeatability between practitioners and, for some cases, poor semivariogram model inference.

Least-square and maximum likelihood methods are proposed as an alternative workflow for automated semivariogram modeling. The least-square workflows mirror the common manual semivariogram modeling approach up to the computation of multiple experimental semivariograms and then automates the selection of the semivariogram parameters to minimize the misfit with the experimental semivariogram points. In other words, least-square workflows convert semivariogram modeling problems into the optimization of a cost function based on misfit (i.e., commonly based on L2 norm, mean square error) between experimental semivariogram points and the semivariogram model at the same lags. Several researchers have proposed using various stochastic optimization algorithms, including random perturbation (Larrando et al. 2003), simulated annealing (Emery 2010; Wilde and Deutsch 2012), the Gauss–Newton algorithm (Dasassis and Renard 2013), and the genetic algorithm (Li et al. 2018; Yasojima et al. 2019).

Even though these least-square methods may perform well in some cases, they have several limitations. First, most of the algorithms are vulnerable to convergence issues, especially in the poor choice of initial values and solution boundaries. Moreover, the number of fitting parameters should be small for the practical solution of the optimization. As a result, most of the applications are confined to two-dimensional cases. Second, the least-square workflows assume experimental semivariogram certainty, yet the experimental semivariogram is uncertain due to sensitivity to search template parameters such as lag distance, lag tolerance, azimuth or dip tolerance, and bandwidth. Variations in these semivariogram calculation parameters may result in a significant difference in experimental semivariograms, and least-square workflows may converge with a large irreducible error or demonstrate instability in the assessed semivariogram models. Mitigation through the use of large enough spatial datasets to reduce variability in experimental semivariograms and expert knowledge to identify optimum search template settings may not be possible.

Alternatively, maximum likelihood methods estimate semivariogram parameters (e.g., range, sill, and nugget) by minimizing a negative log-likelihood function without any intermediate steps such as initial value estimate and iteration for optimization (Mardia and Marshall 1984; Pardo-Igúzquiza et al. 2009; Oliver and Webster 2014). Therefore, maximum likelihood methods enable direct estimation of semivariogram model parameters and quantification of uncertainty in the parameters. Pardo-Igúzquiza et al. (2009) demonstrate that the maximum likelihood method can also detect possible linear or quadratic trends in semivariogram modeling in the case of a groundwater dataset. However, the maximum likelihood methods require the variables of interest to be multivariate Gaussian-distributed, a strong assumption that is often not met and is difficult to verify with typical data sparsity (Kerry and Oliver 2007; Zhang et al. 2017). Moreover, ill-posed assumptions with respect to the nugget effect (i.e., the spatial variability observed at distances less than the minimum data spacing) and nonstationary trends may result in unrealistic semivariogram models (Pardo- Igúzquiza et al. 2009; Oliver and Webster 2014).

Artificial intelligence and deep machine learning algorithms are adding value for spatial analysis, specifically geoscience, and geostatistical workflows (Dimitrakopoulos 1993; Dowd and Sarac 1994). These data-driven methods enhance efficiency and may improve estimation accuracy over conventional statistical algorithms.

Dimitrakopoulos (1993) introduced a conceptual framework of artificial intelligence in geostatistics for computing an experimental semivariogram and performing ordinary kriging. Dowd and Sarac (1994) applied neural networks to geostatistical simulation. Bergen et al. (2019) extensively summarized machine learning applications in geoscience.

The convolutional neural network (CNN), which consists of multiple feature maps, has become one of the most popular deep learning algorithms for applications with spatial data (LeCun et al. 1998). By transferring information through kernels between the feature maps, a CNN is well suited for image data to learn spatial configurations (Guo et al. 2016). For this reason, the CNN is the most commonly applied machine learning method in various computer vision applications, including image denoising (Xuejiao et al. 2015; Zhang et al. 2017), image segmentation (Ronneberger et al. 2015), and image inpainting (Yeh et al. 2016). In addition, CNN-based workflows have been applied for reconstructing porous structures in rock (Mosser et al. 2017; Wang et al. 2018), data conditioning in rule-based reservoir models (Jo et al. 2020; Pan et al. 2021; Jo et al. 2021), developing a surrogate reservoir flow model (Park et al. 2021; Maldonado-Cruz and Pyrcz 2021), and developing proxy pore flow simulation in digital rock (Santos et al. 2020, 2021).

In this paper, machine learning-assisted, automatic semivariogram modeling, known as automatic semivariogram modeling with convolution-based deep learning (ASMC), is proposed to maximize the use of spatial data and reduce semivariogram modeling subjectivity by omitting traditional experimental semivariogram computation and modeling. The workflow includes calculation of two CNNs; the first neural network maps sparse data to exhaustive sequential Gaussian simulation (SGS) images, and the second convolutional neural network maps the SGS images to semivariogram parameters (i.e., range, azimuth of major direction, and aspect ratio between major and minor directions). By dividing the workflow into two steps, the proposed method makes the role of the individual CNN model explicit and stabilizes the training process. Moreover, the proposed method enables geoscientists and engineers to build subsurface models to explore and examine a wide range of possible semivariogram models with the least prior assumptions. As such, ASMC is opposed to the previous least-square methods that require solution boundaries to resolve convergence issues and the maximum likelihood methods that required an assumption of multivariate Gaussianity for the sparse data. The proposed machine learning-based workflows provide improved efficiency and simplification; therefore, the experts may spend more effort in interpreting results and designing geological models. The proposed workflow may be applied to the case of exhaustive images (e.g., acoustic impedance or seismic interpretation models), where the second CNN model directly estimates semivariogram model parameters from the exhaustive images.

The remainder of the paper is organized as follows. Section 2 describes the general CNN design and the training process. Section 3 covers the proposed CCN design and related hyperparameters for automatic semivariogram modeling. Sections 4 and 5 include several demonstrations in synthetic data and a real case study (i.e., the North Cowden field) with performance evaluations. Finally, conclusions are given in Sect. 6.

2 Deep Learning

2.1 Architecture of the CNN

LeCun et al. (1989) first proposed the concept of the convolutional neural network, but CNN was not initially considered practical because of the large amount of computation necessary and the instability in training. Implementation of graphics processing unit (GPU)-based programming and various regularization techniques such as dropout (Srivastava et al. 2014) and batch normalization (Ioffe and Szegedy 2015) made the CNN practical for tackling a variety of problems, especially in the field of computer vision applications. Unlike the conventional neural network for which each layer is fully connected to the next layer, the CNN applies a kernel filter to transfer information between the layers. While the conventional neural network is composed of multiple hidden layers (i.e., one-dimensional sets of nodes) between input and output, the CNN has feature maps (i.e., higher than or equal to two-dimensional matrices).

The main steps of a CNN are convolution, pooling, and upsampling (Guo et al. 2016). Convolution projects information from current feature maps to the next feature maps through the weighted sum of each pixel based on a weighting kernel. Figure 1 shows an example of a convolution operation from 5×5 input to 4×4 output with a kernel size of 2×2. Pooling and upsampling reshape the size of the feature map; pooling shrinks the feature map, and upsampling enlarges the feature map, as shown in Fig. 2. The CNN also may have fully connected layers like the conventional neural network. In this case, a feature map should be flattened to connect to a hidden layer, or a hidden layer should be rearranged to connect to a feature map. Figures 3 and 4 illustrate the CNN models used in this paper. The first CNN maps spatial data to a possible subsurface model, and the second CNN maps the subsurface model to semivariogram model parameters.

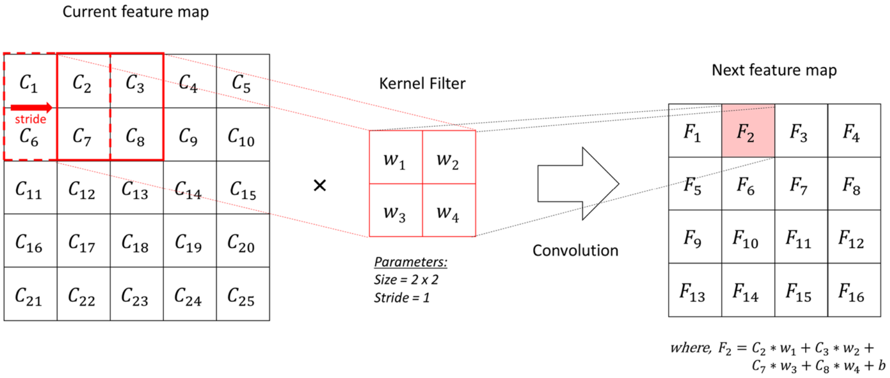
Current feature map
C_1 C_2 C_3 C_4 C_5
stride
C_6 C_7 C_8 C_9 C_{10}
C_{11} C_{12} C_{13} C_{14} C_{15}
C_{16} C_{17} C_{18} C_{19} C_{20}
C_{21} C_{22} C_{23} C_{24} C_{25}
×
Kernel Filter
w_1 w_2
w_3 w_4
Parameters:
Size = 2 x 2
Stride = 1
Convolution
Next feature map
F_1 F_2 F_3 F_4
F_5 F_6 F_7 F_8
F_9 F_{10} F_{11} F_{12}
F_{13} F_{14} F_{15} F_{16}
where, $F_2 = C_2 * w_1 + C_3 * w_2 + C_7 * w_3 + C_8 * w_4 + b$

Fig. 1 Schematic diagram of a convolution operation: The input and output feature maps have 5×5 and 4×4 dimensions, respectively. The kernel filter size is 2×2 with a stride (i.e., kernel shift between convolution operations) of 1

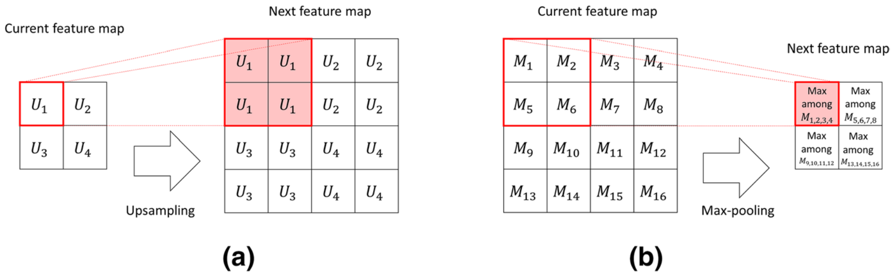

Fig. 2 Schematic diagram of **a** upsampling and **b** max pooling

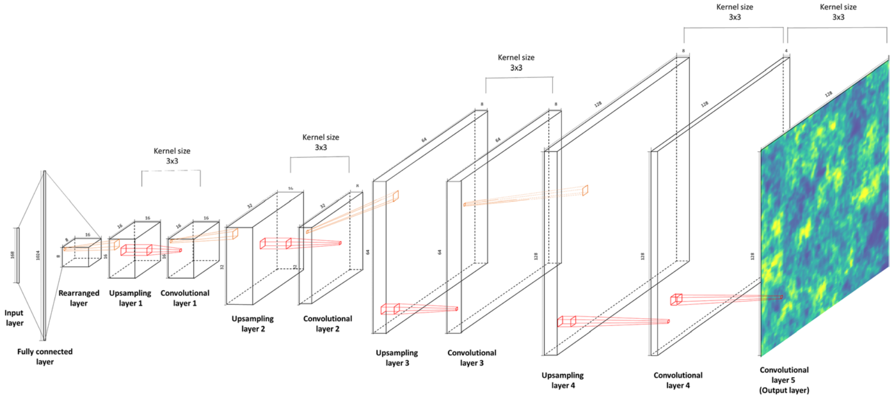

Fig. 3 A CNN model with five convolutional layers, four upsampling layers, and one fully connected later: This CNN model maps spatial data to an exhaustive subsurface model. Red lines describe convolution, whereas orange lines indicate upsampling

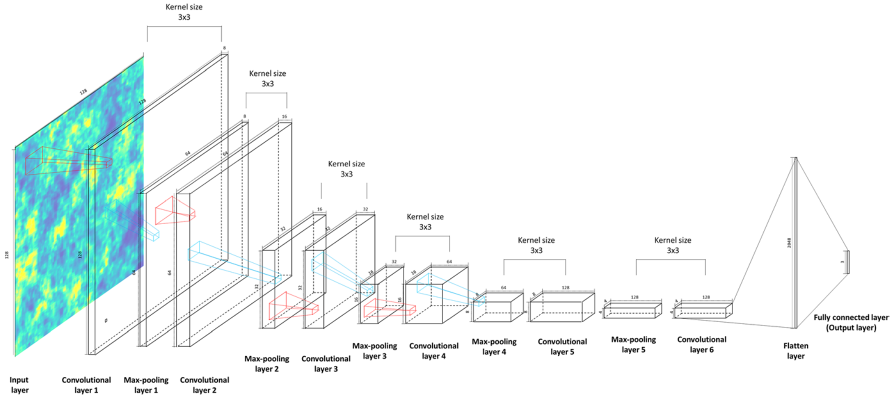

Fig. 4 A CNN model with six convolutional layers, five max-pooling layers, and one fully connected later: This CNN model maps an exhaustive subsurface model to semivariogram parameters. Red lines describe convolution, whereas blue lines indicate max pooling

The CNN in Fig. 3 consists of an input layer, one fully connected layer, five convolutions, four upsampling layers, and an output layer. As the first CNN takes a vector (i.e., spatial conditioning data) and maps it to a subsurface model, the first fully connected layer expands the number of nodes to reshape the feature map. In the next layers, the convolution operations extract the main geometric features, and upsampling operations expand the feature map dimensions so that the size of the output layer is equivalent to the subsurface model size. The kernel filter size is often 3×3, 4×4, or 5×5, but our example shows that 3×3 works better than others. The number of channels in layers controls the CNN model's complexity, which is tuned to the given problem's complexity. The CNN's optimum structure (e.g., the sequence of operations, kernel filter size, and the number of channels) is mainly adjusted in a heuristic manner. The details of the first CNN are:

1. A vector of spatial data (i.e., values for the feature of interest in sample locations) is inputted into the model. In Fig. 3, the length of the input layer is assumed to be 168, but this can be updated depending on the number of given spatial sample data.
2. Next, the input layer is fully connected to the next hidden layer, whose number of elements is 1,024, and the hidden layer is reshaped to 8×8×16 (i.e., width×height×channel). The channel is a depth of an image, and each channel captures a different component of information from the image. For example, a colored image has three channels for red, blue, and green. Each channel contains the intensity of the corresponding color. The feature map (8×8×16) is enlarged to 16×16×16 through upsampling. Upsampling repeats the rows and columns twice, respectively, as shown in Fig. 2a.
3. The first upsampling is followed by a convolution layer through 16 kernel filters, and the size of each kernel is 3×3; therefore, each kernel filter consists of nine weights. The next feature maps encode various geometric characteristics in the current feature map with multiple sets of kernels. The value of the (i,j) pixel of the k th channel in the feature map can be described as $f_k(i,j)$.

$$f_k(i,\, j) = R\left(\sum_{t=1}^{16} \left(U_{1_t} * W_k\right)(i,j) + b_k\right), \tag{1}$$

   where $R(*)$ is the activate function (i.e., ReLu from LeCun et al. 1989), U_{1_t} is the tth channel of upsampling layer 1, $*$ denotes the convolution operation (Guo et al. 2016), W_k is the kth filter of 3×3 in size, and b_k is the bias.
4. The above upsampling and convolution are repeated three more times to result in the feature maps of 32×32×8, 64×64×8, and 128×128×4, respectively.
5. Finally, the last convolution maps 128×128×4 to 128×128×1, a spatial subsurface model. The value of the (i,j) pixel of the output layer is $f(i,j)$ and can be calculated by

$$f(i,\ j) = R\left(\sum_{t=1}^{4} \left(C_{4_t} * W\right)(i,j) + b\right), \tag{2}$$

where $R(*)$ is the sigmoid activate function (LeCun et al. 1989) that is varying from zero to 1, C_{4_t} is the tth channel of convolution layer 4, $*$ denotes the convolution operation, W is the 3×3 filter, and b is the bias.

The second CNN model in Fig. 4 consists of six convolutional layers, five max-pooling layers, and one fully connected layer to map the subsurface model to semivariogram parameters (i.e., the range, aspect ratio, and azimuth of major direction). In the first few layers of the second CNN model, convolution layers convey the main geometric features in feature maps, whereas max pooling downsizes and imposes location invariance in the feature maps. In the last few layers, the feature map is flattened and converged into three nodes for semivariogram parameters. Kernel filter dimensions for convolution layers are determined as 3×3 through hyperparameter tuning, the same as the first CNN model. The second CNN includes the following:

1. A $128 \times 128 \times 1$ subsurface model is inputted into the model.
2. Next, eight kernel filters 3×3 in size convolve with the input image to generate the feature map of $128 \times 128 \times 8$. Like the first CNN model, this process extracts the main features of the input image with different kernel filters. For example, the value of the (i,j) pixel of the kth channel in the feature map is expressed as $f_k(i,j)$.

   $$f_k(i,\ j) = R\left(\left(I * W_k\right)(i,j) + b_k\right), \tag{3}$$

   where $R(*)$ is the ReLu activation function, I is the input image (i.e., a subsurface model), $*$ denotes the convolution operation, W_k is the k th kernel filter, and b_k is the bias.
3. The first convolution layer is followed by a pooling layer that shrinks the feature map along the spatial dimension (i.e., width and height) and results in the feature map of $64 \times 64 \times 8$. Max pooling and average pooling are most commonly implemented (Guo et al. 2016); Max pooling is applied in this research and illustrated in Fig. 2b.
4. The second convolution computation is conducted on max-pooling layer 1. This process converts the feature map of $64 \times 64 \times 8$ into a new feature map of $64 \times 64 \times 16$. The value of the (i,j) pixel of the k th channel in the feature map is expressed as $f_k(i,j)$.

   $$f_k(i,\ j) = R\left(\sum_{t=1}^{8} \left(M_{1_t} * W_k\right)(i,j) + b_k\right), \tag{4}$$

   where $R(*)$ is the activate function (i.e., ReLu), M_{1_t} is the tth channel of max-pooling layer 1, $*$ is the convolution operation, W_k is the kth filter of 3×3 in size, and b_k is the bias parameter.

5. The above max pooling and convolution are carried out four more times to result in the feature maps of $32\times32\times32$, $16\times16\times64$, $8\times8\times128$, and $4\times4\times128$, respectively. While downsizing the image and increasing the number of channels, this CNN model learns the main features of the input image.
6. The last feature map is flattened to a hidden layer of 2,048 elements, and this hidden layer is fully connected to a final layer with three nodes. Each node indicates the range, aspect ratio, and azimuth of the major direction.

The detailed structures of CNN models are described in Tables 1 and 2.

2.2 Training

Each step of convolution in the CNN has trainable model parameters such as filter weights and biases, and fully connected layers also have connection weights and node biases. These trainable model parameters, θ, are represented as

$$\theta^1 = \left\{W_1^1, W_2^1,\ \ldots,\ W_6^1,\ B_1^1, B_2^1, \ldots,\ B_6^1\right\}, \tag{5}$$

$$\theta^2 = \left\{W_1^2, W_2^2,\ \ldots,\ W_7^2,\ B_1^2, B_2^2, \ldots,\ B_7^2\right\}, \tag{6}$$

Table 1 Architecture of CNN #1, which maps sparse data to an exhaustive subsurface model

Layer	Type	Dimension (width × height × channel)	Kernel size	Stride	Activate function	Trainable parameters
Input	Sparse data	# of spatial data	–	–	–	–
1	Fully connected	1,024	–	–	ReLU	W_1^1, B_1^1
2	Reshape	(8 × 8 × 16)	–	–	–	–
3	Upsampling	(16 × 16 × 16)	–	–	–	–
4	Convolution	(16 × 16 × 16)	3 × 3	1	ReLU	W_2^1, B_2^1
5	Batch normalization	(16 × 16 × 16)	–	–	–	–
6	Upsampling	(32 × 32 × 16)	–	–	–	–
7	Convolution	(32 × 32 × 16)	3 × 3	1	ReLU	W_3^1, B_3^1
8	Batch normalization	(32 × 32 × 16)	–	–	–	–
9	Upsampling	(64 × 64 × 16)	–	–	–	–
10	Convolution	(64 × 64 × 8)	3 × 3	1	ReLU	W_4^1, B_4^1
11	Batch normalization	(64 × 64 × 8)	–	–	–	–
12	Upsampling	(128 × 128 × 8)	–	–	–	–
13	Convolution	(128 × 128 × 4)	3 × 3	1	ReLU	W_5^1, B_5^1
14	Batch normalization	(128 × 128 × 4)	–	–	–	-
15	Convolution	(128 × 128 × 1)	3 × 3	1	Sigmoid	W_6^1, B_6^1
Output	Subsurface model	(128 × 128 × 1)	–	–	–	–

Table 2 Architecture of CNN #2, which maps an exhaustive subsurface model to semivariogram parameters (i.e., maximum range, azimuth of major direction, and aspect ratio between major and minor range)

Layer	Type	Dimension (width × height × channel)	Kernel size	Stride	Activate function	Trainable parameters
Input	Subsurface model	(128 × 128 × 1)	–	–	–	–
1	Convolution	(128 × 128 × 8)	3 × 3	1	ReLU	W_1^2, B_1^2
2	Max pooling	(64 × 64 × 8)	–	–	–	–
3	Convolution	(64 × 64 × 16)	3 × 3	1	ReLU	W_2^2, B_2^2
4	Max pooling	(32 × 32 × 16)	–	–	–	–
5	Convolution	(32 × 32 × 32)	3 × 3	1	ReLU	W_3^2, B_3^2
6	Max pooling	(16 × 16 × 32)	–	–	–	–
7	Convolution	(16 × 16 × 64)	3 × 3	1	ReLU	W_4^2, B_4^2
8	Max pooling	(8 × 8 × 64)	–	–	–	–
9	Convolution	(8 × 8 × 128)	3 × 3	1	ReLU	W_5^2, B_5^2
10	Max pooling	(4 × 4 × 128)	–	–	–	–
11	Convolution	(4 × 4 × 8)	3 × 3	1	ReLU	W_6^2, B_6^2
12	Dropout	(4 × 4 × 8)	–	–	–	–
13	Flatten	2,048	–	–		–
14	Fully connected	3	–	–	Sigmoid	W_7^2, B_7^2
Output	Semivariogram parameters	3	–	–	–	–

where θ^1 and θ^2 are trainable parameter sets of the first and second CNN models, respectively. W_i $(i = 1, 2, \ldots, n)$ are the weights of the convolutional kernel filter or fully connected layer in the ith step (e.g., in Fig. 3, the dimension of the kernel filter in the first convolution layer W_2 is $3 \times 3 \times 16$), $B_i(i = 1, 2, \ldots, n)$ is the bias of the ith step (e.g., in Fig. 3, the dimension of B_2 is 16), and n denotes the number of steps that are either a convolution layer or a fully connected layer (e.g., the n is 6 in Figs. 3 and 7 in Fig. 4) in the CNN model.

The training process is conducted by minimizing the loss function $L_1(\theta^1)$ for the first CNN model and $L_2\left(\theta^2\right)$ for the second CNN model. In this paper, the mean square error is used as loss functions

$$L_1\left(\theta^1\right) = \frac{1}{N^1} \sum_{i=1}^{N^1} \left(y_i^1 - \widetilde{y^1}_i\right)^2, \tag{7}$$

$$L_2\left(\theta^2\right) = \frac{1}{N^2} \sum_{i=1}^{N^2} \left(y_i^2 - \widetilde{y^2}_i\right)^2. \tag{8}$$

where N^k is the number of training data of the kth CNN model, y_i^k is the true value, and $y^k{}_i$ is the estimate from the kth CNN model. These loss functions are minimized using the rectified adaptive moment (Adam) optimization (Liu et al. 2019). The learning rates for both CNN models are set as 0.0001 from the hyperparameter tuning process.

3 Automated Semivariogram Modeling with a Convolutional Neural Network

ASMC consists of two steps, each with separate CNN models. The first CNN model maps the sparsely sampled spatial data to an estimate of the exhaustive subsurface model, and the second CNN model maps the estimated exhaustive spatial model to the estimated semivariogram parameters. For brevity, this paper will refer to the first CNN model as CNN #1 and the second CNN as CNN #2. Figure 5 presents the entire workflow of ASMC, and Fig. 6 describes the detailed training process of two CNN models.

First, the spatial feature of interest (e.g., porosity) is pooled from available samples under the assumption of univariate distribution and semivariogram stationarity. For this demonstration, the spatial model extent is 1 km × 1 km, and the spatial data are plotted in a two-dimensional grid composed of 128 × 128 grid cells. Therefore, the spatial data can be visualized in a location map, as shown in step 1 of Fig. 5. Next, multiple semivariogram models with variable parameters are sampled through the design of experiments (in Table 3). Then, SGS realizations are calculated with all semivariogram models from the design. Each SGS realization is labeled with its semivariogram parameters. Third, both CNN #1 and #2 models are trained with these SGS realizations. Figure 6 illustrates the approach for training CNN models. There are two groups of SGS realizations: training and validating groups. Spatial data are extracted from the SGS

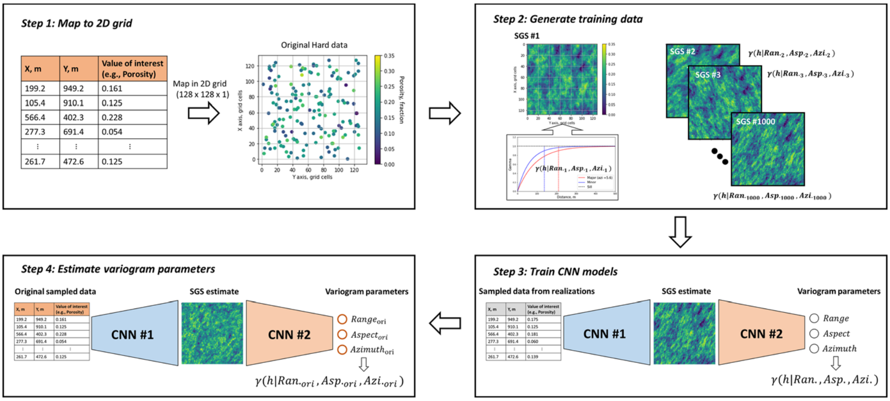

Fig. 5 Schematic graph of the automated semivariogram modeling with CNN: The first step is to transform a given sparse sample data into a two-dimensional grid. Then, a variety of semivariogram parameters are selected from random sampling, and SGS realizations are generated with the semivariogram parameters. Each realization is labeled by its semivariogram model. CNN #1 and CNN #2 are trained with the SGS realizations and their label. Finally, original spatial data is input to CNN #1 to estimate the SGS image, and the SGS image is input to CNN #2 to estimate semivariogram parameters

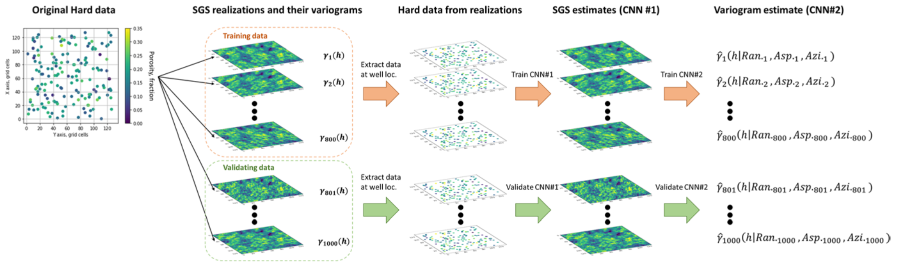
Original Hard data
X axis, grid cells
Y axis, grid cells
Porosity, fraction
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
20
40
60
80
100
120
SGS realizations and their variograms
Training data
$\gamma_1(h)$
$\gamma_2(h)$
$\gamma_{800}(h)$
Validating data
$\gamma_{801}(h)$
$\gamma_{1000}(h)$
Extract data at well loc.
Extract data at well loc.
Hard data from realizations
Train CNN#1
Validate CNN#1
SGS estimates (CNN #1)
Train CNN#2
Validate CNN#2
Variogram estimate (CNN#2)
$\hat{\gamma}_1(h|Ran._1, Asp._1, Azi._1)$
$\hat{\gamma}_2(h|Ran._2, Asp._2, Azi._2)$
$\hat{\gamma}_{800}(h|Ran._{800}, Asp._{800}, Azi._{800})$
$\hat{\gamma}_{801}(h|Ran._{801}, Asp._{801}, Azi._{801})$
$\hat{\gamma}_{1000}(h|Ran._{1000}, Asp._{1000}, Azi._{1000})$

Fig. 6 Schematic diagram of training and validating CNN #1 and CNN #2: First, SGS realizations are generated with different semivariogram parameters and divided into training and validating groups. The data at sample locations are extracted from the realizations. CNN #1 is trained to map spatial data to a subsurface model, and CNN #2 is trained to map the subsurface model to semivariogram parameters (i.e., range, aspect ratio of major/minor range, azimuth of major direction). After training CNN #1 and CNN#2 with training data, they are validated with the data in the validating group

Table 3 Design of experiments to generate multiple semivariogram models with different designs of parameters: for example, if range is 100 m and aspect ratio is 2, the range in the minor direction becomes 50 m. An azimuth of 0 degree indicates the major range is along the north–south direction, whereas an azimuth of 90 degrees is an east–west direction

Semivariogram parameters	Range	Unit
Range of major direction	80–240 (or 10–30)	Meter (or grid cells)
Aspect ratio of major/minor	1–3	Ratio
Azimuth of major direction	0–180	Degree

realizations at sample locations, and CNN #1 maps the spatial data to estimate their exhaustive subsurface model. CNN #2 maps the exhaustive subsurface models to estimates of semivariogram parameters. Training data is only used for training, whereas validation data are withheld from model parameter training to apply in the model validation step to check model performance with unseen inputs and to avoid significant overfitting issues. After training and validating CNN #1 and CNN #2, the two-part ASMC model efficiently maps the original spatial data to estimates of the underlying semivariogram model.

Note that the random seed that controls the stochastic realizations of SGS (Pyrcz and Deutsch 2014) is identical in all SGS realizations, resulting in greater consistency among realizations and allowing for ASMC to train rapidly with a relatively small number of training cases. On the other hand, a diverse random seed integrates inessential variability in the training data, degrading the predictive performance of ASMC. The primary purpose of ASMC is to predict one best solution subsurface model and semivariogram parameter rather than multiple possible subsurface models and the associated semivariogram parameters. After predicting semivariogram parameters, we can span the uncertainty in the subsurface models through SGS. Therefore, generating SGS realizations with a fixed random seed does not limit the capability of ASMC but makes the training process more efficient. The effect of fixing and varying random seeds in ASMC is investigated in Sect. 4.8.

Table 4 Required computation time in each task of ASMC: training data consists of 1,000 SGS models from GSLIB, and both CNN models are trained over 500 epochs with an Intel E5-1650 3.60 GHz CPU and NVIDIA Quadro M6000 24 GB GPU

Task	Computation time, s	Average computation time, s
Generating training data	1,000–1,100	1,050
Training CNN #1	100–130	115
Training CNN #2	150–170	160

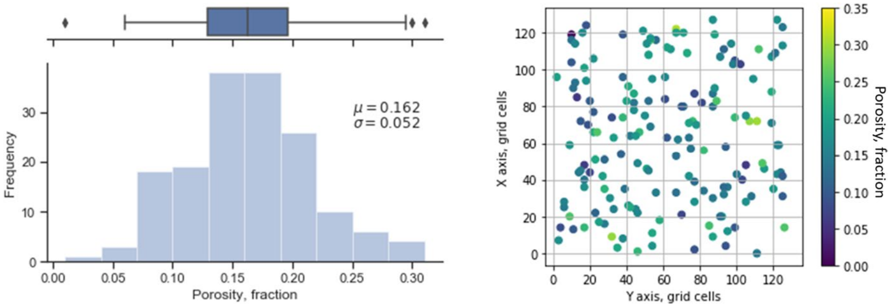

Fig. 7 The histogram and box plot of the given porosity data (left) and location map of the given data (right) from 163 samples

Moreover, in the SGS realizations, the values at sample locations are replaced by SGS outcomes, referenced by surrounding simulated values. This results in locally consistent realizations with the data but not overly constrained by the available spatial data. The CNN models should be customized to a given spatial data. In other words, the entire workflow of generating possible SGS realizations and training CNN models should be repeated for each spatial dataset. Nevertheless, the total computational time is practical on a typical workstation (see Table 4). The average computation time, including generating SGS realizations and training two CNNs, is around 20 min on a desktop computer with an Intel E5-1650 3.60 GHz CPU environment.

4 Results

Synthetic porosity data is calculated in a two-dimensional space to demonstrate the ASMC workflow. The extent of the subsurface model is 1 km × 1 km, and there are porosity measures available at 163 samples. The porosity samples have a mean of 16.2% and a standard deviation of 5.2%, as visualized in Fig. 7 (left). The distribution shows a positive skew but no outliers (based on the Tukey method). The porosity data is transferred to the two-dimensional grid of 128 × 128 in size. Figure 7 (right) presents the location map of 163 samples with their porosity values.

The sample locations are widely spaced over the entire subsurface model, and there are no apparent trends, as the spatial data is assumed to be stationary. Otherwise, the detrending process is necessary before the application of the ASMC workflow (Pyrcz and Deutsch 2014). Moreover, in the severe non-Gaussian distribution in the hard data, normal score transformation (Pyrcz and Deutsch 2014) can be used to alleviate possible artifacts before ASMC.

A total of 800 SGS realizations are calculated as a training set. A further 200 SGS realizations were retained as a validating set to detect any possible overfitting issues and evaluate the prediction performance of ASMC. Figure 8 shows all SGS realizations: each realization depends on the original porosity data, distribution, and semivariogram model parameters from the design of experiments (Table 3). For example, the first realization integrates a semivariogram model with a major range of 23.3 grid cells, a minor range of 15.3 grid cells, and an azimuth of 153 degrees (northern-west to southern-east).

4.1 Subsurface Model Estimate from Spatial Data

The performance of CNN #1 that maps spatial data to a subsurface model estimate is evaluated with the validation set of 200 SGS realizations (see Fig. 9). The first column presents the SGS realizations and their univariate distributions. From the realizations, the porosity values at sample locations are extracted and mapped in the second column. New simulated values replace the original values at sample locations. The porosity values at data locations are input to CNN #1 to estimate the subsurface models, as shown in the third column. For example, the first realization shows the dominant continuity from northwest to southeast. Even though it is challenging to identify this from visual inspection of the spatial data alone, CNN #1 learns this well from the sparse sample data. The subsurface model estimate of the first realization is not perfectly fit to the original SGS realization. However, it shows a similar spatial continuity to the original one as the dominant continuity in the original subsurface model with a major continuity direction from northwest to southeast. The results of the second and third realizations demonstrate consistency in the performance of CNN #1. In other words, CNN #1 successfully learns the main spatial characteristics of the porosity data at sample locations and estimates subsurface models estimating a similar spatial continuity type.

4.2 Estimating Semivariogram Parameters from the Subsurface Model

Figure 10 demonstrates the semivariogram model prediction performance of the ASMC workflow for the validation data withheld from model training. The SGS realizations that are conditioned to the original porosity are in the first column. The second column of Fig. 10 presents the experimental semivariogram from the porosity values at sample locations and their respective semivariogram models in major and minor directions. Recall the experimental semivariogram is defined as

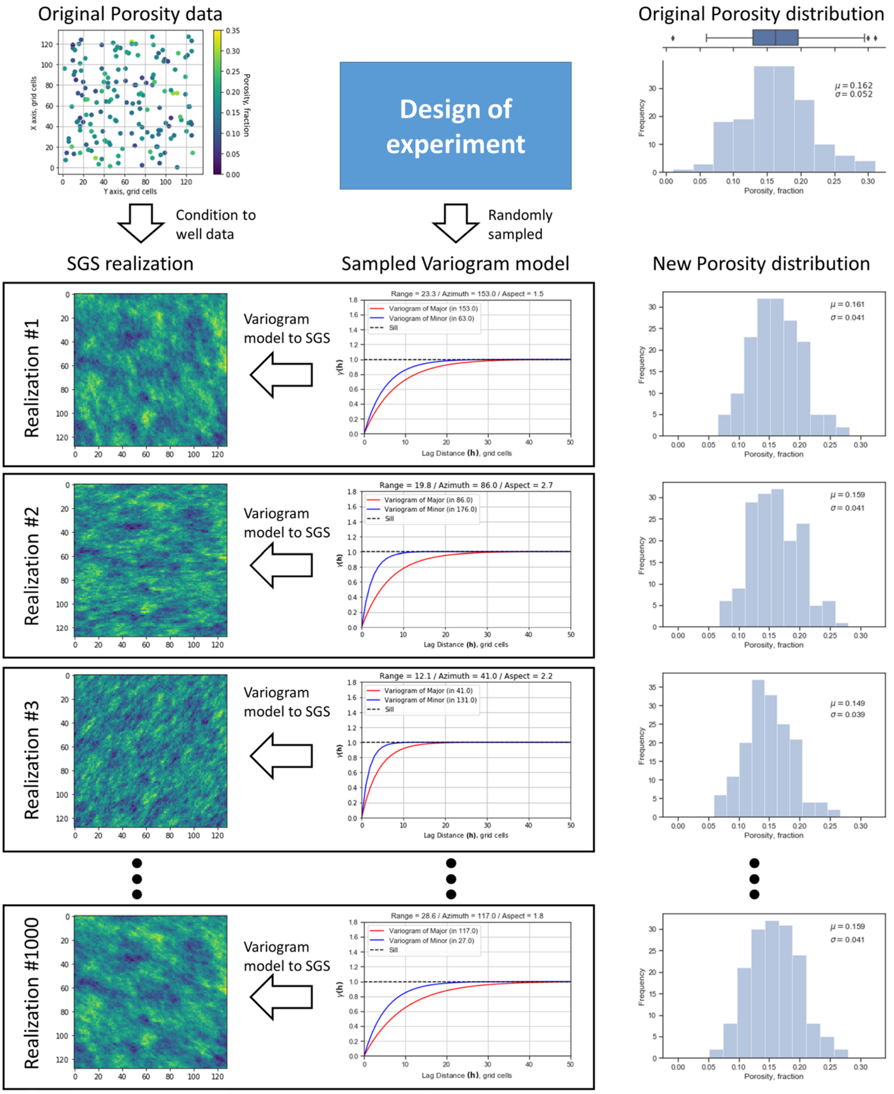

Fig. 8 Overview of the SGS realizations as training/validating data for ASMC: In the first row, the original porosity map and its distribution are given. From the second row, the first column is the SGS realization that is conditioned to the original porosity data and semivariogram model in the second column. The semivariogram model parameters are randomly sampled from Table 3. The third column shows the distribution of the new porosity values at well locations

$$\gamma(\boldsymbol{h}) = \frac{1}{2N(\boldsymbol{h})} \sum_{i=1}^{N(\boldsymbol{h})} \left(x_i - y_i\right)^2, \tag{9}$$

where $\boldsymbol{h}$ denotes a lag vector (i.e., offset between spatial locations), x_i is the variable of interest (e.g., porosity) at the start of the pair i, and y_i is the corresponding end

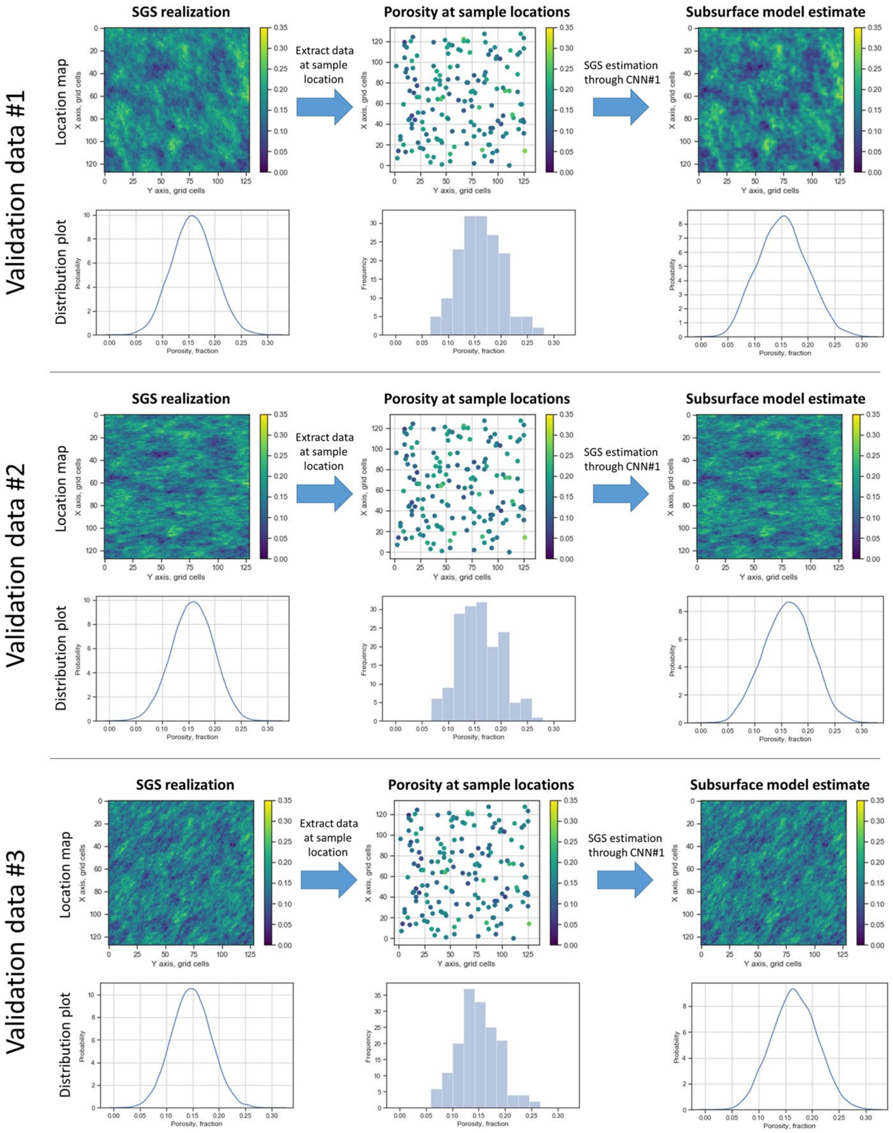

Fig. 9 Demonstration of CNN #1: SGS realization and its probability density function (in the first column), porosity at sample locations and its histogram (in the second column), and subsurface model estimate through CNN #1 and the distribution of the estimated porosity (in the third column). Even though the spatial subsurface model estimate has a tendency to be slightly blurred and its distribution is not entirely identical to the original SGS, CNN #1 successfully understands the main spatial characteristics of the porosity data at sample locations and reconstructs the subsurface model estimate by showing the similar continuity trend

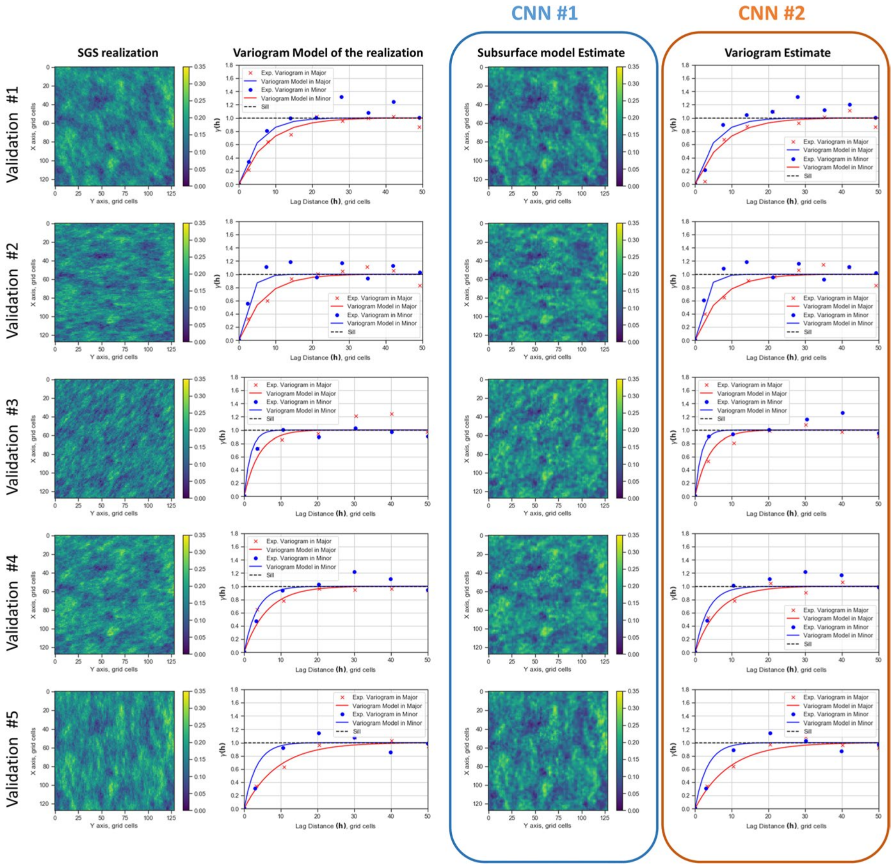

Fig. 10 Demonstration of ASMC: The first column is SGS realization from the validating group, and the second column shows both the experimental semivariogram from porosity at sample locations and the true semivariogram model that is utilized in generating the SGS realization. The third column is the subsurface model estimates through CNN #1, and the fourth column shows both the experimental semivariogram from estimated porosity at sample locations and the semivariogram model that is computed from CNN #2. ASMC successfully estimates the original semivariogram models

value. $N(\boldsymbol{h})$ stands for the number of sample pairs available for lag $\boldsymbol{h}$. The number of lags should be carefully selected based on the minimum data spacing and the extent of the study area.

Moreover, the semivariogram model is usually modeled by one or a combination of these three known, positive definite semivariogram models (Deutsch and Journel 1998; Pyrcz and Deutsch 2014)

$$\gamma_{model}(\boldsymbol{h}|C_0,a_0)=\begin{cases} min\left(C_0, C_0\left[\frac{3}{2}\frac{\boldsymbol{h}}{a_0}-\frac{1}{2}\left(\frac{\boldsymbol{h}}{a_0}\right)^3\right]\right), & Spherical \quad (10a)\\ C_0\left[1-\exp\left(-\frac{3\boldsymbol{h}^2}{a_0^2}\right)\right], & Gaussian \quad (10b)\\ C_0\left[1-\exp\left(-\frac{3\boldsymbol{h}}{a_0}\right)\right]. & Exponential \quad (10c)\end{cases}$$

where C_0 is the structure contribution, and a_0 denotes the range. As the values are standardized to have a unity variance, C_0 is always equal to 1 in this demonstration (we assume a single structure). Please note that as Gaussian and exponential semivariograms never reach the sill (i.e., their sills are asymptotic), their ranges are defined where the semivariogram reaches around 95% of the sill. Moreover, the semivariogram is assumed to follow the exponential type, and only the 2D case is considered, so the estimated semivariogram parameters include a_0 values in major and minor directions, and the azimuth of major direction.

In the second column of Fig. 10, the experimental semivariograms are shown with the truth semivariogram models that are used to generate training data and ASMC workflow estimates. The third column of Fig. 10 presents the estimated subsurface models through CNN #1 that have a similar continuity trend as the original SGS realizations. The last column of Fig. 10 is a plot of the experimental semivariograms calculated from the porosity values in subsurface model estimates and the semivariogram models computed from CNN #2. The experimental semivariograms (i.e., dots in the semivariogram plots) of subsurface model estimates are quite similar to those of the original SGS realizations. This similarity indicates that CNN #1 is accurate in estimating the underlying continuity models in SGS realizations. Moreover, the semivariogram model estimates from CNN #2 are also like the calculated experimental semivariograms as well as the original truth semivariogram models. Thus, the ASMC workflow successfully interprets spatial continuity from the porosity data at the sample locations and predicts reasonable semivariogram model parameters.

4.3 Model Validation of ASMC

In training CNN models, loss functions should be inspected over training epochs to measure the model accuracy and stability and detect any possible issues (e.g., overfitting). Figure 11 shows the loss functions (i.e., mean square error) of CNN #1 and CNN #2 over 500 epochs. The loss function of CNN #1 decreases rapidly until 100 epochs and then stabilizes. Both loss curves for training and validation data decrease simultaneously, indicating no issue with overfitting. On the other hand, the loss function of CNN #2 improves up to 300 epochs and then stabilizes. The loss function of training data slightly deviates from validating data after 200 epochs, but this is assessed to be insignificant. The best CNN #1 and #2 parameters with the least loss values are automatically saved during the training process.

Figure 12 compares the estimated semivariogram model parameter values to the true values. The blue cross indicates the estimation performance of training data, whereas the red dot is for validating data. The R-squared values in the range

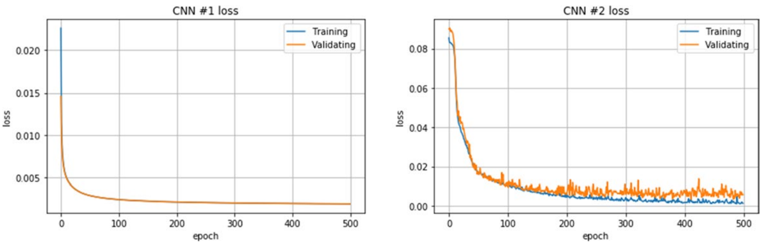

Fig. 11 Loss curve of CNN #1 (left), loss curve of CNN #2 (middle), and accuracy of CNN #2 (right) over the training epoch: Both training and validating loss functions of CNN #1 and CNN #2 decrease effectively. No significant overfitting is observed

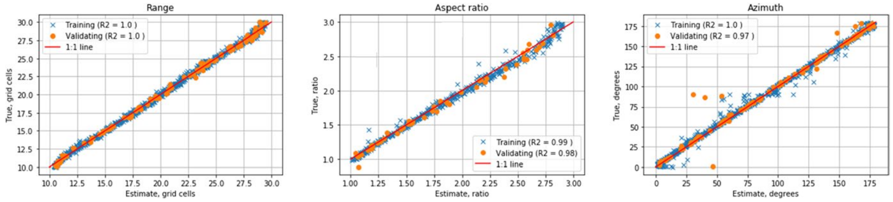

Fig. 12 True values vs. estimates of CNN #2: the range in the major direction (left), the aspect ratio between the major range and minor range (middle), and the azimuth of the major direction (right). Both training and validating data show over 0.97 explained variance scores

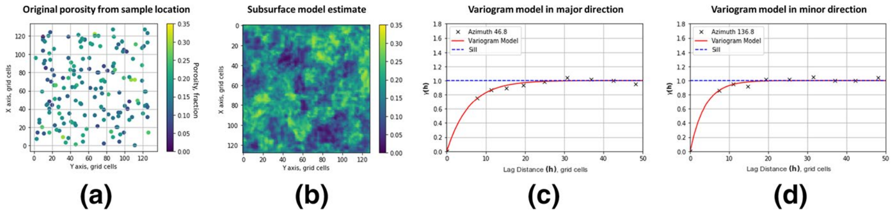

Fig. 13 ASMC demonstration for original porosity data: **a** Original spatial data, **b** subsurface model estimate from the original spatial data, **c** experimental semivariogram and estimated semivariogram model in major direction, and **d** experimental semivariogram and estimated semivariogram model in the minor direction. The estimated range is 24 (grid cells), and the estimated azimuth of the major direction is 46.8 degrees. The estimated aspect ratio is 2.18. Both semivariogram models are well fitted to the experimental semivariogram

prediction, aspect ratio prediction, and azimuth prediction show over 0.97 for training and validation datasets. The proposed prediction model performs reliably.

4.4 Testing ASMC with the Original Data

The best CNN models from the training process are selected by their minimum loss values. The original porosity spatial data is input to ASMC to estimate the associated semivariogram model parameters. Figure 13 includes the original porosity data, and the estimated subsurface model is computed through CNN #1. The maximum continuity is from northeast to southwest. CNN #2 predicts semivariogram model parameters of a 24-grid-cell major range, 11-grid-cell minor range, and the azimuth of the major direction at 46.8 degrees. Experimental semivariograms and estimated semivariogram models are shown in the last two graphs in Fig. 13. The machine learning-based predictions of semivariogram models in major and minor directions closely approximate the experimental semivariograms from the original spatial dataset.

4.5 Comparison with the Least-square Method

The performance of ASMC is compared with the least-square method in the same porosity dataset. The loss function of the least-square method is defined as

$$\mathrm{MSE}\left(a_0, r,\ \theta\right) = \frac{1}{N_{\mathrm{lag}}} \sum_{i=1}^{N_{\mathrm{lag}}} \left(\gamma_{\mathrm{model}}\left(\boldsymbol{h}_i | a_0, r,\ \theta\right) - \gamma\left(\boldsymbol{h}_i | \theta\right)\right)^2, \tag{11}$$

where a_0, r, and θ are semivariogram parameters (i.e., the range, aspect ratio between major and minor range, and azimuth, respectively), γ_{model} is the semivariogram model with the given parameters, and γ is the experimental semivariogram along the azimuth of θ. Moreover, N_{lag} is the number of lags in the experimental semivariogram.

With a fixed aspect ratio, the loss is calculated for the change of the range and azimuth, as shown in Fig. 14a. We implement Nelder–Mead optimization (Nelder and Mead 1965) to find the solutions for the least-square method. Figure 14b shows the contour map of the loss with estimated semivariogram parameters from the ASMC and least-square methods, as well as global minimum location.

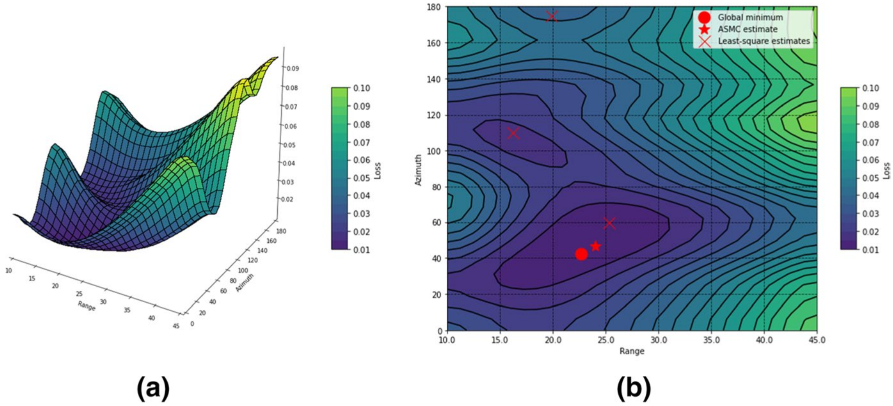

Fig. 14 Response surface of loss (i.e., mean square error) between the experimental semivariogram and the corresponding semivariogram model: **a** three-dimensional view of the loss change with respect to the range and azimuth and **b** contour map of the loss with solutions from ASMC and least-square method. The least-square method falls into multiple local minima with different initial guesses

Due to the non-convex loss function, the least-square solutions converge on multiple local minima from different initial guesses, and these estimates are significantly different from each other. For example, one solution estimates the range of 16 grid cells with an azimuth of 110 degrees, whereas another solution indicates the range of 25 grid cells with an azimuth of 60 degrees. Considering that calculating exhaustive loss response is computationally expensive and unavailable, it is extremely challenging to tell which solution is closer to the global minimum in the typical least-square process. Unlike the least-square solution, ASMC gives a stable solution close enough to the global minimum solution, as shown in Fig. 14b.

4.6 Uncertainty Assessment of Semivariogram Parameters

The uncertainty of ASMC estimates is investigated following the workflow in Fig. 15. With the given degree of measurement error in hard data, multiple possible realizations of the hard dataset are generated through the Monte Carlo simulation. Each realization passes through the two CNNs in ASMC and computes estimated semivariogram parameters. The probability distribution can be computed with reliable uncertainty quantification from the multiple estimates for the range, aspect ratio, and azimuth. As two trained CNNs only require negligible computational time to render estimates, the proposed workflow can assess uncertainties in semivariogram parameters without significant additional computational cost.

Figure 16 presents the uncertainty distribution of semivariogram parameters with different degrees of measurement error in hard data. Each column of plots stands for semivariogram parameters, and each row indicates the significance of measurement errors. The blue lines describe the probability density of estimates in each plot, and the red dashed lines show ASMC estimates with no importing measurement error. For example, the range varies from 22 to 27 grid cells with a 1% measurement error, but it can go from 15 to 35 grid cells with a 5% measurement error. In the same way, the uncertainty ranges of the aspect ratio and azimuth can be quantified.

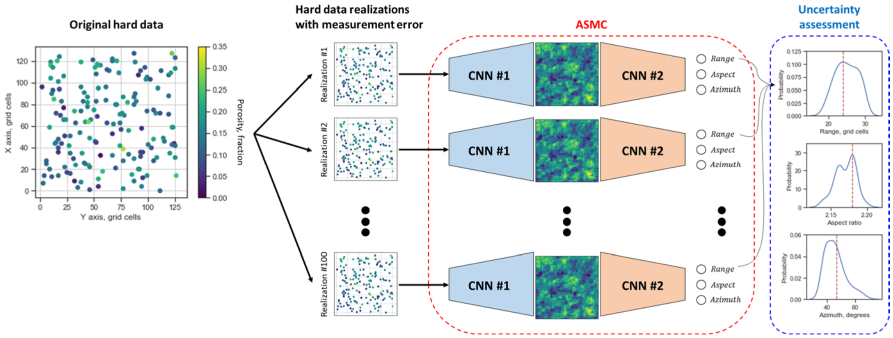

Fig. 15 Schematic diagram for uncertainty assessment of semivariogram parameters

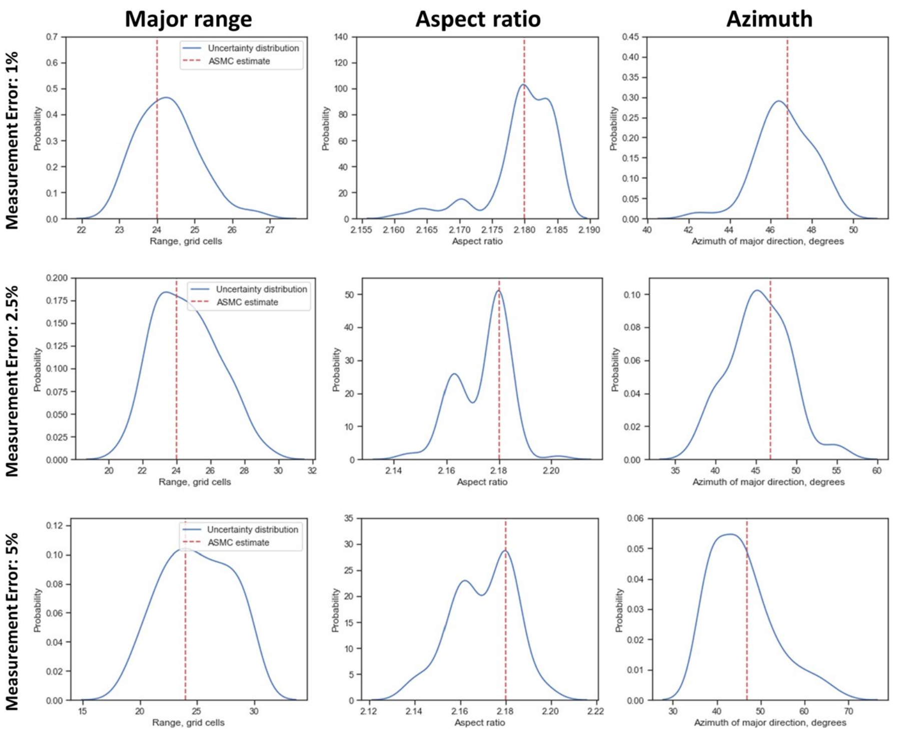

Fig. 16 Uncertainty distribution of semivariogram parameters with different measurement errors: each row indicates a different degree of measurement errors in hard data. Each column presents uncertainty distributions (i.e., solid blue lines) of semivariogram parameters and ASMC estimates (i.e., red dashed lines)

4.7 Sensitivity Analysis of CNN Hyperparameters

The sensitivity analysis for the CNN hyperparameters of the ASMC workflow is shown in Figs. 17 and 18. There are two key hyperparameters in CNNs: (i) kernel size and (ii) the number of channels. With the same training dataset, we assess the performance of ASMC with different complexity of CNNs. Figure 17 shows the losses of ASMC with the different numbers of channels in CNNs, and the base case indicates the CNN structure that we describe in Tables 1 and 2. Moreover, the *y*-axis in Fig. 17 is normalized by the loss of validating data in the base case (e.g., 100% means the same loss for validation data in the base case). Blue and orange bars indicate relative losses of training and validating data, respectively. Using a quarter or a half of the channels of the base case, the losses of training and validating data do not decrease effectively, indicating underfitting. On the other hand, when increasing the number of channels twice or four times, training data losses keep decreasing, whereas those of validating data increase, indicating overfitting. Therefore, we select the structures of CNN #1 and #2 with the most optimum predictive performance of both training and validating data.

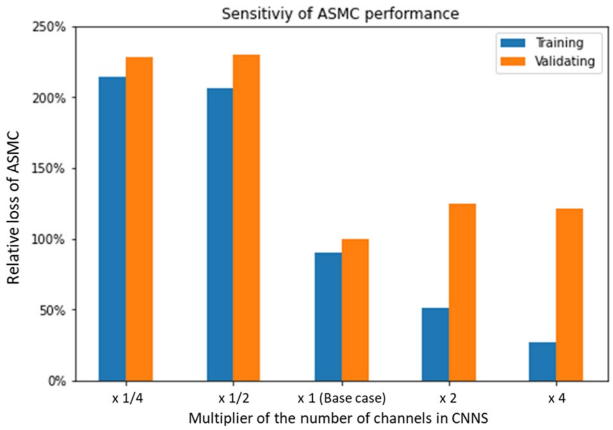

Fig. 17 Sensitivity of ASMC performance with respect to the number of channels in CNNs: the *y*-axis is normalized by the MSE of the validation dataset for the base case that is described in Tables 1 and 2. For example, "×2" indicates that the number of channels is increased by 2, and "×1/2" denotes that the number of channels is decreased by half

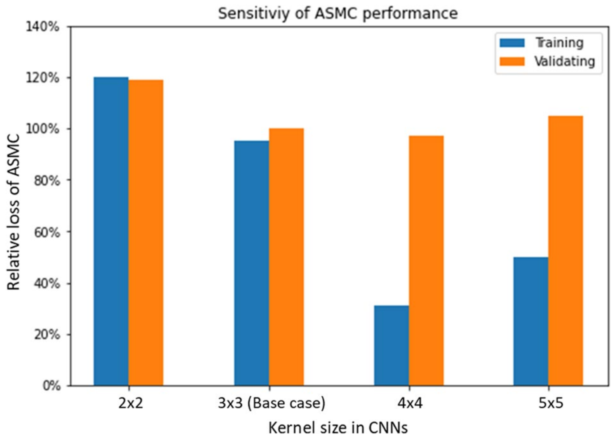

Fig. 18 Sensitivity of ASMC performance with respect to the kernel size in CNNs: the *y*-axis is normalized by the MSE of the validation dataset for the base case that is described in Tables 1 and 2. For example, "2×2" indicates that the kernel size is 2 by 2 in width and height

Figure 18 shows the changes in ASMC's performance concerning the kernel size. The base case uses 3×3, as described in Tables 1 and 2. For a 2×2 kernel size, losses of training and validating data are not as small as the base case. Moreover, for 4×4 and 5×5 kernel sizes, even though the losses of validating data do not change much, the discrepancy between training and validating losses becomes significant (i.e., overfitting). Therefore, we select 3×3 as an optimum kernel size in ASMC. The optimum number of channels and kernel size should be adequately investigated for the given tasks through the hyperparameter tunning process.

4.8 Effect of Random Seeds in SGS Realizations

The effect of fixing random seeds in generating SGS realizations is investigated in Fig. 19. Even though the training losses of both cases are similar in comparison, the validation loss of ASMC that uses varying random seeds in generating SGS realization becomes significantly larger than that of ASMC with a fixed random seed. In other words, such overfitting occurs in conditions with varying random seeds. As such, varying random seeds with unnecessary variability in SGS realizations is shown to be less efficient than using a fixed random seed.

5 Demonstration for the North Cowden field

The ASMC is also demonstrated in a real case study. The North Cowden field is in west Texas, where 62 wells have been drilled. Pyrcz et al. (2006) studied comprehensive geostatistical analysis (i.e., stationary check, declustering, trend modeling and debiasing, object-based modeling, semivariogram, and uncertainty quantifications) in the field. Figure 20a and b show well locations with porosity histogram in units of fractions averaged over the vertical thickness of the reservoir. Note that the observed porosity follows a bimodal distribution, indicating two distinct rock facies (e.g., dolomite and siltstone) and the possible horizontal trends. As such, Pyrcz et al. (2006) applied detrending to the porosity and computed residual porosity. Figure 20c and d shows a spatial scatter map and histogram of the residual porosity. The residual porosity satisfies the stationarity and follows a Gaussian distribution.

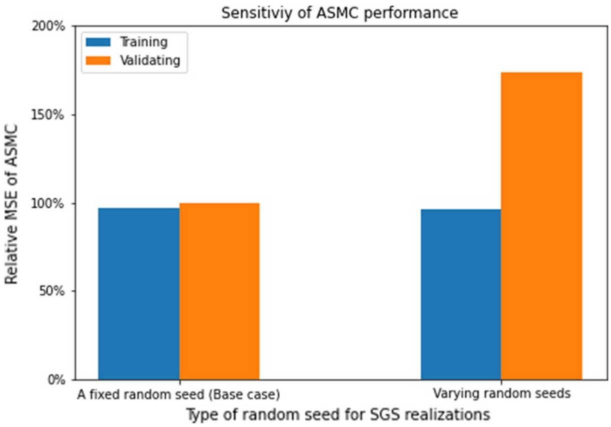

Fig. 19 Performance of ASMC with different types of random seeds in generating SGS realizations: the *y*-axis is normalized by the MSE of the validation dataset for the base case that uses a fixed random seed in generating SGS realizations

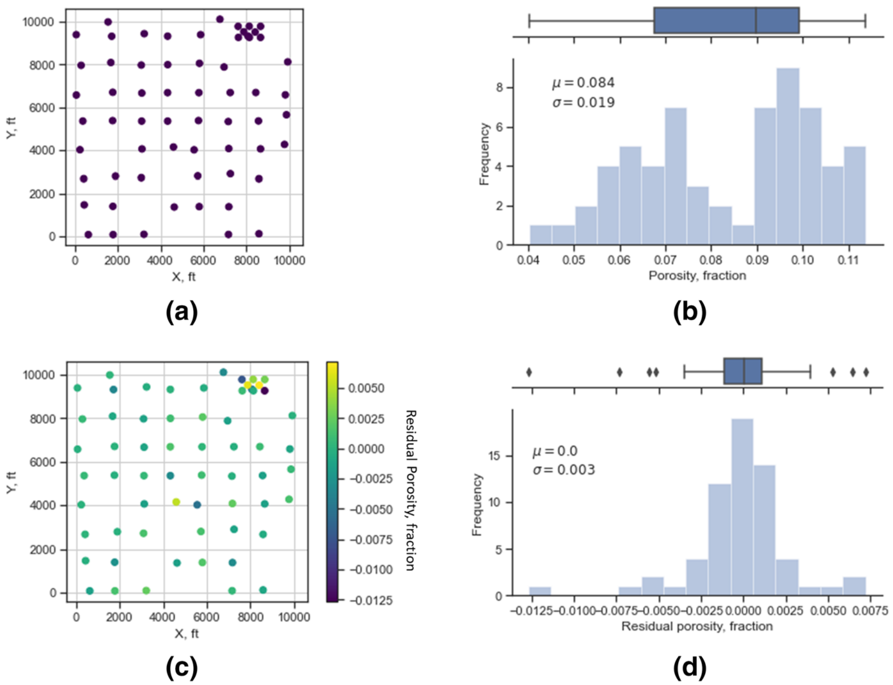

Fig. 20 The spatial distribution of all 62 wells in the North Cowden field and the histogram of the porosity: **a** the location of wells, **b** histogram of porosity at well locations, **c** spatial distribution of residual porosity, and **d** histogram of residual porosity

5.1 ASMC Estimation for Semivariogram Parameters

The extent of the North Cowden field is 10,200 ft × 10,200 ft, and the residual porosity data is transferred to the grid of 128 × 128 cells. The average residual porosity is close to zero, which is typical after an unbiased detrending process, and its standard deviation is 0.3%, as shown in Fig. 20d. Under the same scheme as the synthetic data demonstration, 1,000 SGS realizations are generated with different semivariogram parameters from the design of experiments in Table 5. Moreover, 800 of them are used for training, while the rest are used to validate the two CNNs in ASMC. Note that these two CNNs have the same structure as those in the synthetic data demonstration.

Table 5 Design of experiments to generate multiple semivariogram models with different designs of parameters in the North Cowden field demonstration

Semivariogram parameters	Range	Unit
Range of major direction	2,000–6,000	ft
Aspect ratio of major/minor	1–3	Ratio
Azimuth of major direction	0–180	Degree

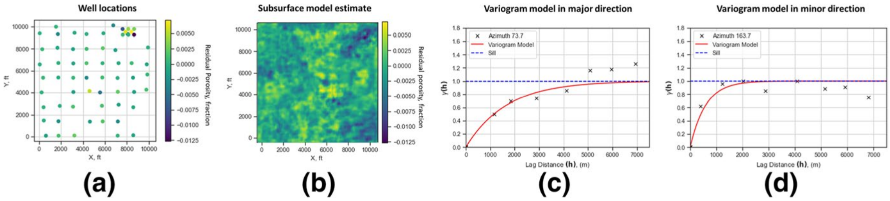

Fig. 21 ASMC demonstration for the North Cowden field data: **a** original residual porosity, **b** subsurface model estimate for residual porosity, **c** experimental semivariogram and estimated semivariogram model in the major direction, and **d** experimental semivariogram and estimated semivariogram model in the minor direction. The estimated range is 4,960 ft, and the estimated azimuth of the major direction is 73.7 degrees. The estimated aspect ratio is 2.81

After training the CNNs of ASMC, the original residual porosity is fed to ASMC to estimate its semivariogram model parameters. Figure 21 shows the original residual porosity, the subsurface model estimate for the residual porosity, and the estimated semivariogram model. CNN #1 maps original residual porosity to subsurface model estimate, whose major continuity direction lies northeast to southwest, as shown in Fig. 21b. Moreover, CNN #2 predicts its semivariogram model parameters of 4,960-ft major range, 1,810-ft minor range, and the azimuth of major direction at 73.7 degrees. Figure 21c and d shows that the experimental variograms in major and minor directions are well fitted to the estimated semivariogram models, which indicates that ASMC successfully predicts the semivariogram of residual porosity in the North Cowden field.

5.2 Uncertainty Assessment of Semivariogram Parameters

The uncertainty analysis scheme is applied to the North Cowden field data. Figure 22 shows the uncertainty of semivariogram parameters with different degrees of measurement error. The range in the major direction varies between 4,930 ft and 4,980 ft with a 1% measurement error, whereas the major range's uncertainty can move from 4,600 ft to 5,100 ft with a 5% measurement error. The larger the measurement errors, the wider the uncertainty ranges of semivariogram model parameters, same as the uncertainty assessment of the synthetic porosity data.

6 Conclusion

This paper introduces a deep learning-based automated semivariogram modeling with convolutional neural networks to improve the efficiency of modeling semivariograms with an automated, data-driven approach. Unlike the least-square automated semivariogram modeling methods, ASMC does not require customizing the cost function to fit the semivariogram model to an experimental semivariogram through the optimization process. Instead, the ASMC workflow generates multiple possible semivariogram parameters and trains two CNNs to learn the mapping from spatial data to an exhaustive spatial model and from an exhaustive spatial model to semivariogram model parameters. The simplified ASMC

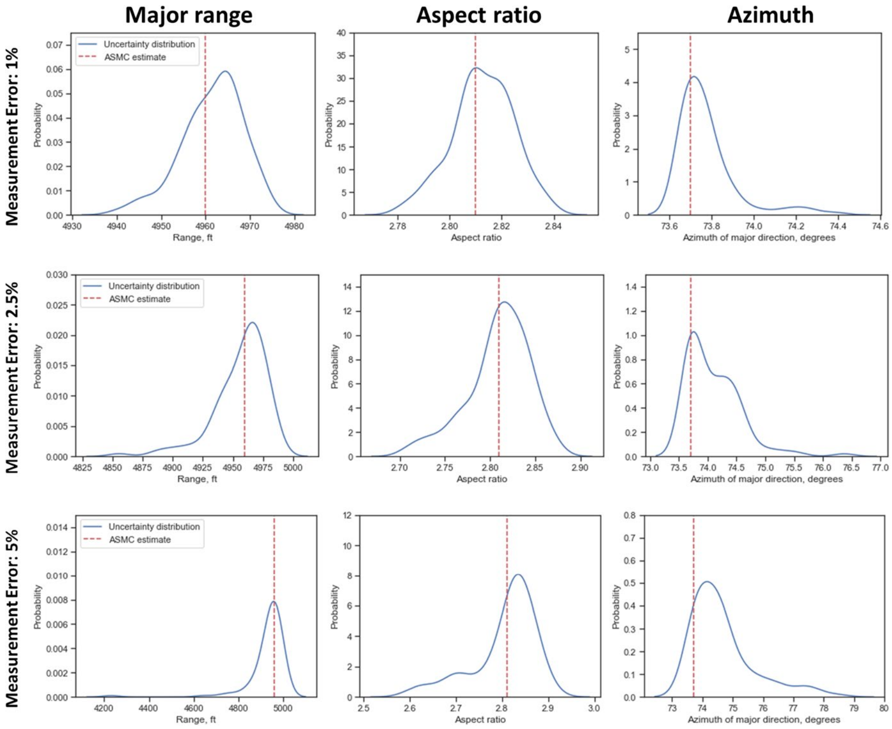

Fig. 22 Uncertainty distribution of semivariogram parameters for the North Cowden field: each row indicates a different degree of measurement error in hard data. Each column presents uncertainty distributions (i.e., solid blue lines) of semivariogram parameters and ASMC estimates (i.e., red dashed lines)

workflow avoids experimental semivariogram calculation and the associated subjectivities of defining semivariogram search parameters. Moreover, the uncertainty of semivariogram model parameters can be measured without adding considerable computation time in ASMC. Therefore, ASMC can efficiently estimate semivariogram model parameters from spatial data and swiftly quantify the associated uncertainties. The flexibility of ASMC allows this workflow to expand to various subsurface features of interest, such as mineral content in mining, permeability in reservoir engineering, or any spatial features of interest, including plant and animal species population densities and precipitation rates.

One limitation of this study is that demonstrations only have been verified in non-nugget, single-structure semivariogram cases. To include the nugget effect and nested structures in semivariograms, it is required to modify the last layer in the second CNN to add more semivariogram model parameters. Moreover, to expand the application of ASME to three-dimensional spatial data, the vertical range and dip in the vertical axis should be added to the semivariogram parameters. Moreover, three-dimensional kernel filters should be applied in CNNs to capture spatial continuity in horizontal and vertical directions. The possible cost of expanding ASMC to more complicated semivariogram cases includes additional training data and consequent computational time increment as the number of estimates and associated trainable parameters in CNNs increases.

Acknowledgements The authors thank the two reviewers for their critical feedback and their help to improve the quality of the manuscript. The authors also appreciate the support of the DIRECT consortium for subsurface data analytics and machine learning at The University of Texas at Austin. The corresponding author is also thankful to the Hildebrand Department of Petroleum and Geosystems Engineering at The University of Texas at Austin, Texas, USA.

References

Bergen KJ, Johnson PA, Maarten V, Beroza GC (2019) Machine learning for data-driven discovery in solid Earth geoscience. Science 363(6433).

Chilès J, Delfiner P (2012) Geostatistics: modeling spatial uncertainty, 2nd edn. Wiley, New York

Cho Y, Cao Y, Zagayevskiy Y, Wong T, Munoz Y (2020) Kriging-based monitoring of reservoir gas saturation distribution using time-lapse multicomponent borehole gravity measurements: Case study, Hastings Field. J Pet Sci Eng 190:107054

Dasassis N, Renard D (2013) Automatic semivariogram modeling by iterative least squares: univariate and multivariate cases. Math Geosci 45:453–470. https://doi.org/10.1007/s11004-012-9434-1

David M (1977) Geostatistical ore reserve estimation. Elsevier, Amsterdam

Deutsch CV, Journel A (1998) GSLIB geostatistical software library and user's guide. Oxford Press, New York

Dimitrakopoulos R (1993) Artificially intelligent geostatistics: a framework accommodating qualitative knowledge-information. Math Geol 25(3):261–279

Dowd PA, Sarac C (1994) A neural network approach to geostatistical simulation. Math Geol 26(4):491–503

Emery X (2010) Iterative algorithms for fitting a linear model of coregionalization. Comput Geosci 36(9):1150–1160

Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, LewMS, (2016) Deep learning for visual understanding: a review. Neurocomputing 187:27–48. https://doi.org/10.1016/j.neucom.2015.09.116

Jensen J, Lake LW, Corbett PW, Goggin D (1997) Statistics for petroleum engineers and geoscientists. Gulf Professional Publishing, New Jersey

Jo H, Santos JE, Pyrcz MJ (2020) Conditioning well data to rule-based lobe model by machine learning with a generative adversarial network. Energy Explor Exploit 38(6):2558–2578

Jo H, Pan W, Santos JE, Jung H, Pyrcz MJ (2021) Machine learning assisted history matching for a deepwater lobe system. J Pet Sci Eng, 109086.

Journel AG, Huijbregts CHJ (1978) Mining geostatistics. Academic Press, London

Kerry R, Oliver MA (2007) Comparing sampling needs for variograms of soil properties computed by the method of moments and residual maximum likelihood. Geoderma 104(4):383–396. https://doi.org/10.1016/j.geoderma.2007.04.019

Larrondo PF, Neufeld CT, Deutsch CV (2003) VARFIT: A program for semi-automatic semivariogram modeling. In: Deutsch CV (ed) CCG Annual report 5. University of Alberta, Edmonton, p 17

LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1:541–551

Li Z, Zhang X, Clarke KC, Liu G, Zhu R (2018) An automatic variogram modeling method with high reliability fitness and estimates. Comput Geosci 120:1150–1160. https://doi.org/10.1016/j.cageo.2018.07.011

Liu W, Pyrcz MJ (2020) A spatial correlation-based anomaly detection method for subsurface modeling. Math Geosci. https://doi.org/10.1007/s11004-020-09892-z

Loffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Paper presented at the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015

Maldonado-Cruz E, Pyrcz MJ (2021) Tuning machine learning dropout for subsurface uncertainty model accuracy. J Pet Sci Eng, 108975.

Mardia KV, Marshall RJ (1984) Maximum likelihood estimation of models for residual covariance in spatial regression. Biometrika 71(1). https://doi.org/10.2307/2336405

Mosser L, Dubrule O, Blunt MJ (2017) Reconstruction of three-dimensional porous media using generative adversarial neural networks. Phys Rev E 96(4):043309. https://doi.org/10.1103/PhysRevE.96.043309

Nelder JA, Mead R (1965) A simplex method for function minimization. Comput J 7(4):308–313. https://doi.org/10.1093/comjnl/7.4.308

Oliver MA, Webster R (2014) A tutorial guide to geostatistics: computing and modelling variograms and kriging. CARENA 13:56–69. https://doi.org/10.1016/j.catena.2013.09.006

Pan W, Torres-Verdín C, Pyrcz MJ (2021) Stochastic Pix2pix: a new machine learning method for geophysical and well conditioning of rule-based channel reservoir models. Nat Resour Res 30(2):1319–1345

Pardo-Igúzquiza E, Mardia KV, Chica-Olmo M (2009) MLMATERN: A computer program for maximum likelihood inference with the spatial Matérn covariance model. Comput Geosci 35:1139–1150. https://doi.org/10.1016/j.cageo.2008.09.009

Park J, Datta-Gupta A, Singh A, Sankaran S (2021) Hybrid physics and data-driven modeling for unconventional field development and its application to US onshore basin. J Pet Sci Eng, 109008.

Pyrcz MJ, Deutsch CV (2014) Geostatistical reservoir modeling. Oxford University Press, New York

Pyrcz MJ, Gringarten E, Frykman P, Deutsch CV (2006) Representative input parameters for geostatistical simulation. In: Coburn TC, Yarus JM, Chambers RL (eds) Stochastic modeling and geostatistics: Principles, methods, and case studies, volume II: AAPG Computer Applications in Geology 5, pp 123–137. https://doi.org/10.1306/1063811CA53230

Pyrcz MJ, Jo H, Kupenko A, Liu W, Gigliotti AE, Salomaki T, Santos JE (2021) GeostatsPy. https://github.com/GeostatsGuy/GeostatsPy

Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. arXiv preprint https://arxiv.org/abs/1505.04597. Accessed 6 May 2020

Salazar JJ, Lake LW (2020) The physical meaning of the koval factor. Math Geosci 52:1017–1033. https://doi.org/10.1007/s11004-020-09883-0

Santos JE, Xu D, Jo H, Landry CJ, Prodanović M, Pyrcz MJ (2020) PoreFlow-Net: A 3D convolutional neural network to predict fluid flow through porous media. Adv Water Resour 138:103539. https://doi.org/10.1016/j.advwatres.2020.103539

Santos JE, Yin Y, Jo H, Pan W, Kang Q, Viswanathan HS, Prodanović M, Pyrcz MJ, Lubbers N (2021) Computationally efficient multiscale neural networks applied to fluid flow in complex 3D porous media. Transp Porous Media. https://doi.org/10.1007/s11242-021-01617-y

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958

Wang Y, Arns CH, Rahman SS, Arns J (2018) Porous structure reconstruction using convolutional neural networks. Math Geosci 50:781–799. https://doi.org/10.1007/s11004-018-9743-0

Wilde BJ, Deutsch CV (2012) Automatic variogram modeling from censored variogram volumes. In: Deutsch CV (ed) CCG Annual report 14. University of Alberta, Edmonton, p 127

XuejiaoW, Qiuyan T, LianghaoW, Dongxiao L, Ming Z (2015) Deep convolutional architecture for natural image denoising. In: Paper presented at the 2015 international conference on wireless communications & signal processing (WCSP), Nanjing, China

Yasojima C, Protzio K, Meiguins B, Neto N, Morais J (2019) A new methodology for automatic cluster-based kriging using k-nearest neighbor and genetic algorithms. Information 10(11):357. https://doi.org/10.3390/info10110357

Yeh RA, Chen C, Lim TY, Schwing AG, Hasegawa-Johnson M, Do MN (2016) Semantic image inpainting with deep generative models. Paper presented at the 30th IEEE conference on computer vision and pattern recognition, Honolulu, Hawaii, USA, 21–26 July 2017. Institute of Electrical and Electronics Engineers Inc, Los Alamitos, pp 6882–6890

Zhang K, Zuo W, Chen Y, Meng D, Zhang L (2017) Beyond a Gaussian Denoiser: residual learning of deep CNN for image denoising. IEEE Trans Image Process 26:3142–3155. https://doi.org/10.1109/TIP.2017.2662206