*ANS M&C 2021 - The International Conference on Mathematics and Computational Methods Applied
to Nuclear Science and Engineering · Raleigh, North Carolina · October 3–7, 2021*

SAND2021-1933C

# USING DEEP NEURAL NETWORKS TO PREDICT MATERIAL TYPES IN CONDITIONAL POINT SAMPLING APPLIED TO MARKOVIAN MIXTURE MODELS

**Warren L. Davis IV[1], Aaron Olson[1], Gabriel Popoola[1],
Dan Bolintineanu[1], Theron Rodgers[1], and Emily Vu[1,2]**
[1]Sandia National Laboratories
1515 Eubank Blvd. SE, Albuquerque, NM 87123

[2]Department of Nuclear Engineering Radiological Sciences
University of Michigan, Ann Arbor, MI 48109, USA

wldavis@sandia.gov, aolson@sandia.gov, gapopoo@sandia.gov

## ABSTRACT

Conditional Point Sampling (CoPS) is a recently developed stochastic media transport algorithm that has demonstrated a high degree of accuracy in 1-D and 3-D calculations for binary mixtures with Markovian mixing statistics. In theory, CoPS has the capacity to be accurate for material structures beyond just those with Markovian statistics. However, realizing this capability will require development of conditional probability functions (CPFs) that are based not on explicit Markovian properties, but instead are based on latent properties extracted from material structures. Here, we describe a first step towards extracting these properties by developing CPFs using deep neural networks (DNNs). This new approach lays the groundwork for enabling accurate transport on many classes of stochastic media. We train DNNs on ternary stochastic media with Markovian mixing statistics and compare their CPF predictions to those made by existing CoPS CPFs derived based on Markovian mixing properties. We find that the DNN CPF predictions usually outperform the existing approximate CPF predictions, but with wider variance. In addition, even when trained on only one material volume realization, the DNN CPFs are shown to make accurate predictions on other realizations that have the same internal mixing behavior. Our results show that it is possible to form a useful CoPS CPF by using a DNN to extract correlation properties from realizations of stochastically mixed media, thus establishing a foundation for creating CPFs for mixtures other than those with Markovian mixing, where deriving an acurate analytical CPF may not be possible.

KEYWORDS: machine learning, Conditional Point Sampling, stochastic media, Monte Carlo transport

## 1. INTRODUCTION

*Conditional Point Sampling* (CoPS) is an algorithm for effecting transport in stochastic media [1–4], simulating particle histories using Woodcock tracking so that material in the domain only needs to be defined at pseudo-collision sites. By only sampling the material at pseudo-collision sites,

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

CoPS does not need to realize or store complex material realizations and, therefore, transport results from whole ensembles of realizations can be approximated cheaply. At each pseudo-collision site, the material present is sampled in order to evaluate the pseudo-collision. Once a point is sampled, it is remembered and used to modify the sampling probabilities of subsequently sampled points. We call the mathematical/computational means for predicting these probabilities the CoPS *conditional probability function* (CPF). To date, several CPFs have been developed for use in CoPS when simulating media with Markovian mixing statistics by leveraging Markovian-mixing pseudo-interface properties [1–4], but no attempt has been made to develop a CPF that can be straightforwardly adapted to other kinds of mixing. Deep neural networks (DNNs) are machine-learning (ML) systems that can be used to model latent relationships present in data. They have been employed successfully for many regression and classification problems. In this paper, we describe the creation of a novel DNN model that serves as a CoPS CPF capable of learning spatial correlation relationships based on realizations of stochastic media. This is a significant first step towards developing CPFs adaptable to yielding high accuracy in various types of stochastic mixing. For reference, we compare the DNN predictions to predictions made by existing CPFs that were derived for use in CoPS when simulating transport in Markovian-mixed media. We specifically work with a ternary mixture, which requires a modest extension to existing pseudo-interface-based CoPS CPF derivations.

## 2. METHODOLOGY

A set of 3-D realizations consisting of three materials mixed according to Markovian mixing statistics are generated for this study using the method described for binary mixtures in [5] and extended to mixtures of an arbitrary number of materials in [6]. As in [6], the geometry is translated to a voxel geometry structure by querying what material exists at points on a regular grid and assigning those material types to voxels.

Two CPFs, each derived for use with Markovian-mixed media, are used: the pseudo-interface-based *CoPS2* CPF [1,3,4] that makes probability predictions based on the point nearest to the new point and the *CoPS3PO* CPF [4,7,8] that makes predictions based on the nearest point on each side of the new point. While, in general, the notation “CoPS2” only denotes use of CoPS with a two-point CPF, we use it in this paper as shorthand to specifically mean CoPS with the two-point pseudo-interface-based CPF. For binary media, CoPS2 has been shown to perform roughly as accurately as the Limited Realization Preserving method for computing reflectance in 1-D and 3-D and more accurate than benchmarked non-CoPS methods for transmittance for 1-D and 3-D [1,3]. We therefore use CoPS2 CPF evaluations as a metric for “good” predictions. The CoPS3PO CPF, for which “PO” stands for pseudo-interface optimum, makes errorless probability assessments in Markovian-mixed media, when the three points of interest are the only relevant points and lie perfectly on a line. Here, CoPS3PO is used to establish the “true” probabilities for 1-D relationships. Previous derivations of these CPFs are generalized to mixing of $N$ materials in a later section.

For machine learning predictions, we use a DNN with three hidden layers and rectified linear hidden activation units, described more fully in a later section. The goal of the modeling algorithms is to take nearby point data as input and predict the distribution of the material types for the reference point.

## 2.1. Derivation of CoPS CPFs for $N$-ary Mixing

The pseudo-interface-based CoPS2 CPF has been derived for use with binary media in 1D [1,4] and 3D [3]. The pseudo-interface-based CoPS3PO CPF has been derived in [4] and used in [8] and [7]. In order to use these CPFs for ternary Markovian mixing in this study, we follow the same method as for binary mixing in [4] to derive these CPFs for use with Markovian mixing involving an arbitrary number of constituent materials.

These media are describable by a Poisson process of material cells in which the material type of each cell is independent of the material type in other cells and determined by the nominal material abundance. Neighboring cells can be the same material type. Cell boundaries are called pseudo-interfaces. The frequency of $k$ pseudo-interfaces existing in a distance $r$ is Poisson-distributed $f(k, I(r)) = \frac{I^k(r)}{k!}e^{-I(r)}, I(r) = \frac{r}{\Lambda_c}$ and driven by the correlation length $\Lambda_c$. For the following derivations, we use the expressions in Eq. (1) to probabilistically assess whether two points distance $r_i$ apart have at least one pseudo-interface between them–and are thus in the same cell and necessarily the same material type–or not.

$$f(k=0, I(r=r_i)) = e^{-\frac{r_i}{\Lambda_c}}; \qquad f(k>0, I(r=r_i)) = 1 - e^{-\frac{r_i}{\Lambda_c}} \tag{1}$$

The probability of two randomly sampled points distance $r_1$ apart from one another being material types $m_1$ and $m_0$ is expressed as

$$\begin{aligned} \pi(m_1, m_0|r_1) = \quad & \delta_{m_1,m_0} \quad p_{m_1} \quad f(k=0, I(r=r_1)) \\ + \quad & \qquad\qquad p_{m_1}p_{m_0} \quad f(k>0, I(r=r_1)), \end{aligned} \tag{2}$$

where $p_{m_i}$ is the material abundance of material $i$ and $\delta_{jk} = \begin{cases} 1, & \text{if } j = k, \\ 0, & \text{if } j \neq k \end{cases}$. The first row in Eq. (2) is the probability that there are no pseudo-interfaces between the two points and that the cell containing them is material type $p_{m_1}$. The second row contributes the probability that there is at least one pseudo-interface between the points and that the points are material type $m_1$ and $m_0$. Eq. (2) holds for $m_1, m_0 \in \{0, \ldots, N-1\}$. We note that $m_1$ and $m_0$ may hold the same value. The same procedure of evaluating the probability of having or not having at least one pseudo-interface between each pair of points and the probability that each point is a certain material type is followed for determining the probability that three randomly sampled colinear points with spacing $\vec{r} = \{r_1, r_2\}$ are material types $m_1$, $m_0$, and $m_2$:

$$\begin{aligned} \pi(m_1, m_0, m_2|\vec{r}) = \quad & \delta_{m_1,m_0}\delta_{m_0,m_2} \quad p_{m_1} \quad f(k=0, I(r=r_1+r_2)) \\ + \quad & \delta_{m_0,m_2} \quad p_{m_1}p_{m_2} \quad f(k>0, I(r=r_1))f(k=0, I(r=r_2)) \\ + \quad & \delta_{m_1,m_0} \quad p_{m_1}p_{m_2} \quad f(k=0, I(r=r_1))f(k>0, I(r=r_2)) \\ + \quad & \qquad p_{m_1}p_{m_0}p_{m_2} \quad f(k>0, I(r=r_1))f(k>0, I(r=r_2)). \end{aligned} \tag{3}$$

Suppressing the expression of $r_1$ and $r_2$ for brevity, we use Bayes' Theorem to derive the probability of sampling a new point of material type $m_0$ relative to either one previously sampled point distance $r_1$ away or one point on each side of the new point distances $r_1$ and $r_2$ away:

$$\pi(m_1, m_0|m_1, \kappa) = \frac{\pi(m_1, \kappa|m_1, m_0)\pi(m_1, m_0)}{\pi(m_1, \kappa)} = \frac{\pi(m_1, m_0)}{\sum_{m=0}^{N-1}\pi(m_1, m)}, \tag{4}$$

$$\pi(m_1, m_0, m_2|m_1, \kappa, m_2) = \frac{\pi(m_1, \kappa, m_2|m_1, m_0, m_2)\pi(m_1, m_0, m_2)}{\pi(m_1, \kappa, m_2)} = \frac{\pi(m_1, m_0, m_2)}{\sum_{m=0}^{N-1} \pi(m_1, m, m_2)}, \tag{5}$$

where $\kappa$ represents an unknown material type and $m$ represents a material type index. The pseudo-interface-based CoPS2 CPF returns probabilities according to either material abundance or Eq. (4), using the nearest point to the new point, based on whether any points have been sampled in the domain or not. Due to the Markovian property of the media, this CPF is errorless in such media if points are only sampled colinearly and in forward procession–otherwise its evaluations are approximate. The pseudo-interface-based CoPS3PO CPF returns probabilities according to either material abundance, Eq. (4), or Eq. (5) based on whether no points have been sampled, whether previously sampled points are colinear and all on one side of the new point, or whether previously sampled points are colinear and at least one point exists on each side of the new point. The CoPS3PO CPF is errorless in this type of media for colinear evaluations. We thus use the CoPS3PO CPF to establish truth probability values for colinear relationships.

### 2.2. Data

The datasets used are all ternary Markovian mixture volumes with 300 voxels in each dimension (27,000,000 voxels total). Five different realizations were created using the following parameters: correlation length $\Lambda_C = 20.8$ voxels and material abundances $p_0 = 0.06, p_1 = 0.65$, and $p_2 = 0.29$.

The material type of the original voxel, along with the distance and material types of the auxiliary voxels, form the features for an individual data observation. For each material structure sample, one million voxels were selected randomly. For each voxel, 2 auxiliary voxels were selected at a random integer distance along a single axis and on opposite sides of the original voxel within varying ranges. The voxels were selected such that auxiliary voxels always lie inside the Markovian mixture volume. Therefore, voxels near the edge of the volume were excluded from the random voxel selection. Two examples are shown in Table 1.

**Table 1: Examples of features used in the material probability prediction task. The reference material ($m_0$) is listed, followed by the material in the negative X direction ($m_1$), the distance between $m_0$ and $m_1$ ($r_1$), the material in the positive X direction ($m_2$), and the distance between $m_0$ and $m_2$ ($r_2$), as well.**

| **Example** | $m_0$ | $m_1$ | $r_1$ | $m_2$ | $r_2$ |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 7 | 0 | 5 |
| 2 | 2 | 2 | 12 | 1 | 13 |

However, representing the material types as categorical values would be problematic, as our research is actually attempting to solve a regression problem. We want to predict the probability of each material type instead of giving a single material prediction. The output material types would not be interpretable if constrained to a single categorical variable. For instance, in the original form, a material prediction of ".99" could be interpreted as the DNN being very confident that

**Table 2: Examples of the features in Table 1 after one-hot encoding. The representation of the positive material and negative material for the examples have both been changed from one column to three columns, represented as a binary variable across material types.**

| **Example** | $m_{0,0}$ | $m_{0,1}$ | $m_{0,2}$ | $m_{1,0}$ | $m_{1,1}$ | $m_{1,2}$ | $r_1$ | $m_{2,0}$ | $m_{2,1}$ | $m_{2,2}$ | $r_2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 7 | 1 | 0 | 0 | 5 |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 | 12 | 0 | 1 | 0 | 13 |

the material type is type 1, or split between deciding whether it should predict material type 0 or material type 2. To remove this confusion, both the input and output variables are modified with *one-hot encoding* [9]. This gives a separate binary feature for each class; in this instance, each material type. Table 2 shows an example of the data from Table 1 after the encoding process. One-hot encoding expands the complexity of the inputs and outputs to the DNN. Instead of 4 inputs, the DNN will have 8; instead of one output, the network has 3. The nodes in the hidden layers must be increased to compensate for this increased complexity. However, this increase in complexity helps to avoid ambiguity of the DNN response during learning and during post-processing.

### 2.3. Deep Neural Network Architecture

We created a network with 8 input nodes; 2 vectors, each of length 3, containing one-hot encodings of the material types, and two scalars representing the respective distances of the materials. Between the input and ouput layers, we utilized a default of 3 dense hidden layers with 12 nodes, 9 nodes, and 6 nodes, respectively. The initial hidden layer expands the number of nodes by 50% as a default setting to allow the network to express and take advantage of variable dependencies, then reduces the number of nodes for subsequent layers in a uniform fashion. The final output layer is a vector-length 3 one-hot encoding of the reference material type. The networks use rectified linear unit (ReLU) activations [10,11] in all layers except the final layer, which uses a linear activation. The loss is calculated using mean squared error. The optimizer is Nadam, the standard Adam optimizer with Nesterov momentum [12]. The models were implemented using Keras [13], with a Tensorflow [14] backend. All optimizer, activation, etc. parameters were utilized with the default values.

We experimented with changes to the DNN depth and nodes, and we found that network performance was mostly robust to such changes. Different configurations changed the average number of training epochs and training time, but generally did not change the accuracy due to the adaptive nature of the Nadam optimizer. Our configuration reasonably balances accuracy and training time.

### 2.4. Machine Learning Training

Each network runs with early-stopping, based on the validation loss. When the validation stops decreasing, the network stops and rolls back to the network's state at the best loss. The validation loss can be quite dynamic, so the network allows up to 20 epochs after the best validation loss before it stops. The DNNs trained in the range from 15 to 73 epochs before stopping due to lack of validation error decrease. The batch size for each network is set to 32.

The output values for the DNN do not have the same semantics as the output of the non-DNN CoPS CPFs examined here. The linear output activations do not force the material probability predictions to sum to 1, nor do they constrain the probability predictions between 0 and 1. There are architectures that can arbitrarily constrain the output values; however, we hypothesized that these constraints could be learned just by modeling the data.

In the end, our DNN probability predictions mostly adhered to the expected unit sum relationship. However, the constraint was not perfect, and several times there were either negative probability predictions or individual material probability predictions that exceeded 100%. To resolve those discrepancies, we applied a normalizing process to the DNN predictions. For all observations with negative material probability predictions, we first add the absolute value of the minimum material probability to the material probability distribution of that observation. We then use the preprocessing.normalize function from Scikit-Learn [15] to scale the adjusted probability prediction vectors to the unit norm, which ensures that the probabilities are in [0,1] and that they all sum to one. These DNN predictions can now be treated like material probabilities.

### 2.5. Experiments

We began our experiments by examining scenarios where the material to be predicted lies in the middle of two auxiliary points along one dimension. For each Markovian mixture sample, one million voxels were selected randomly. For each voxel, 2 auxiliary voxels were selected at a random integer distance, in voxels, along a single axis and on opposite sides of the original voxel. The distance for each auxiliary voxel was selected independently and from within a uniform distribution in the range of 1 and 20 voxels away.

Because the materials have different distributions, we were careful not to build a model that was rewarded merely for picking the most abundant material type. From the million voxels, we subsampled, randomly and without replacement, a number of observations, such that each class was evenly represented in the realization. This process yielded 30,000 observations of each material type in the training set (90,000 total training observations) and 10,000 observations of each material type in the test set (30,000 test observations). The result is a partitioning of 75% of the data for training, and 25% for testing. Of the training data, we used 20% for validation. The validation set is chosen from within the training set randomly and without replacement, agnostic of material types.

## 3. Results

The performance of the methodologies is evaluated against the observations in the test set. We evaluate and compare the accuracy of the predictions made by the CoPS2, CoPS3PO, and DNN CPFs.

**Table 3: Accuracy of the predictions on each realization (Real #) by simply choosing the material with the highest probability.**

| Method | Real 0 | Real 1 | Real 2 | Real 3 | Real 4 | Average |
|---|---|---|---|---|---|---|
| Truth | 77.8 | 79.2 | 84.1 | 82.4 | 75.0 | 79.7 |
| CoPS2 | 80.4 | 81.9 | 86.6 | 84.5 | 77.9 | 82.3 |
| DNN | **83.9** | **86.5** | **89.6** | **87.1** | **81.9** | **85.8** |

### 3.1. Accuracy based on highest probability

The first metric is the accuracy of the prediction based on the highest probability. For each observation, we find the material that receives the highest probability score. The predicted material is compared to the actual reference material from the dataset. If it is the same, the model is deemed correct for that observation. We divide the number of correct predictions by the total number of elements in the test set. As Table 3 shows, the best performer using this method is the DNN CPF in all cases. The DNN CPF outperforms the CoPS2 CPF by a wider margin than the DNN CPF outperforms the Truth CPF. The differences in the individual CPF performances are consistent across all realizations.

While this metric has some value, additional metrics are needed to assess the usefulness of the CPF prediction for CoPS. In CoPS, new points must be sampled such that there is some probability of sampling the less likely material. In many material types, including those with Markovian mixing, always chosing the material type with the higher probability would yield all points in the domain being sampled as the same material as the first sampled point.

### 3.2. Accuracy based on random probabilistic selection

Another way of measuring accuracy is by randomly selecting a material type based on the material probability prediction distributions. In this way, a random number is generated and binned based on the predicted probabilities for an observation. Table 4 shows that the DNN predictions perform best for all realizations, although by less of a margin when the metric of accuracy is based on the highest probability. On average, the difference is around 3 percentage points.

This metric not only measures the DNN CPF's ability to assess which material is more likely, but also measures the DNN CPF's ability to predict the probability of sampling a new material. It is therefore more useful than our first metric and is expected to have strong correlation with the accuracy of CoPS when using the CPF of interest. A third metric, introduced in the next section, is believed to the be the most useful of those proposed here, since it compares predicted probabilities to the true probabilities, but can only be used when the true probabilities are known. Therefore, the random probabilistic selection metric is important to provide a measure of accuracy when true probabilities are not known, such as for predictions involving multi-D point relationships.

**Table 4: Accuracy of the predictions by randomly choosing a material based on the per-observation probability distributions.**

| Method | Real 0 | Real 1 | Real 2 | Real 3 | Real 4 | Average |
|---|---|---|---|---|---|---|
| Truth | 73.2 | 75.4 | 80.1 | 77.8 | 70.5 | 75.4 |
| CoPS2 | 69.1 | 70.2 | 73.0 | 71.7 | 67.3 | 70.3 |
| DNN | **76.4** | **78.7** | **84.9** | **80.7** | **72.3** | **78.6** |

### 3.3. Accuracy based on divergence from the true probabilities

The most useful model performance metric is the measurement of how closely the probability distributions match the true material distributions, as determined by the CoPS3PO CPF. Kullback-Liebler divergence (KL-divergence) is a common measurement of the divergence for probability distributions [16]. However, because the KL-divergence is asymmetric, the value will vary depending on which distribution is chosen as the reference distribution. In addition, the divergence is sometimes undefined for non-continuous distributions. We use, instead, a modification of the KL-divergence, called the Jensen-Shannon divergence (JS-divergence) [17], which incorporates the KL-divergence and corrects the problems of undefined values and asymmetry [18]. As with KL-divergence, the JS-divergence for two probability distributions decreases as the distributions become more similar, with identical distributions having a divergence of zero.

In Table 5, we compare the average JS-divergences of both the CoPS2 and the DNN CPFs to the CoPS3PO true probability distributions. The results show that the material probability distributions predicted by the DNN CPF are, on average, comparable to the CoPS3PO truth values. This gives us confidence that the probability distributions predicted by the network in 3-D would be representative of the true material properties. Although the JS-Divergence of the DNN CPFs were slightly better on average than the CoPS2 CPF, the JS-divergences for DNN CPFs varied greatly between realizations. In the best case, Realization 2, the divergence was comparatively very low. However, for Realizations 0 and 3, the variances were much greater. Such variance can be undesirable in applications where consistency of CPFs is more important than overall accuracy. This variance may be ameliorated by training on many different realizations. In contrast, the CoPS2 CPF had a consistent divergence across all realizations.

### 3.4. Extension to 3-D reference points

In 3-D, we choose auxiliary voxels along all 3 axes, instead of just one axis, resulting in 6 total auxiliary voxels surrounding the reference voxel. We refer to this orientation of auxiliary voxels as a “star shape.” Unlike with the 1-D case, there is no provably true 3-D comparison currently available. We can still use the metric of accuracy based on random probabilistic selection, as we did earlier, to compare the accuracy of the DNN CPF to that of the CoPS3PO CPF in material prediction.

**Table 5: Jensen-Shannon divergences across material realizations. Lower numbers are better, indicating less of a divergence from the truth values.**

| Method | Real 0 | Real 1 | Real 2 | Real 3 | Real 4 | Average |
|---|---|---|---|---|---|---|
| CoPS2 | **.147** | **.146** | .154 | .151 | **.147** | .149 |
| DNN | .169 | .153 | **.112** | **.129** | .166 | **.146** |

As this data has more input values (24 instead of 8), the architecture of the DNN was modified accordingly. Between the input and output layers, the network utilizes 3 hidden layers of 36, 25, and 14 nodes, respectively. All other network parameters were the same as for the 1-D cases.

In Table 6, we compare the accuracies based on random selection with the predicted probability distributions from the 3-D prediction models. As with the experiments in 2-D, the DNN CPF is more accurate for all realizations than the COPS2 CPF.

**Table 6: Accuracy percentages of the predictions by randomly choosing a material based on the per-observation probability distributions for 3-D reference voxels. Higher numbers are better.**

| Method | Real 0 | Real 1 | Real 2 | Real 3 | Real 4 | Average |
|---|---|---|---|---|---|---|
| CoPS2 | 84.8 | 83.6 | 90.2 | 86.1 | 83.2 | 85.6 |
| DNN | **87.8** | **87.3** | **91.1** | **87.4** | **86.0** | **87.9** |

### 3.5. Inter-realization model inferencing

One problem encountered in machine learning is that models trained on one dataset may not work on different datasets. This can happen even when dataset-specific information is not explicitly included in the model (e.g., when the data from one set is drawn from a different distribution than the other). Although the realizations were given identical material parameters, the realizations include stochastic variance, which can affect the actual data distributions. To investigate the question of whether models trained on one realization would be accurate for other realizations sampled with the same parameters, we performed experiments using a model trained on a 3-D realization and then projected across the other realizations.

Table 7 shows that the best performance is usually attained by the intra-realization models, where the models are trained and tested on independent datasets extracted from the same realizations. However, on average, the differences are minor, never reaching one percentage point. In a couple

**Table 7: Comparison of intra-realization accuracy versus inter-realization accuracy. Higher numbers are better.**

| Method | Real 0 | Real 1 | Real 2 | Real 3 | Real 4 |
|---|---|---|---|---|---|
| Intra-Realization Inferencing | **87.8** | **87.3** | **91.1** | **87.4** | **86.0** |
| Inter-Realization Minimum | 86.6 | 86.5 | 90.4 | 85.9 | 84.3 |
| Inter-Realization Maximum | 87.4 | **87.3** | 90.7 | **87.4** | 85.7 |
| Inter-Realization Average | 87.1 | 86.9 | 90.6 | 86.9 | 85.2 |

of cases, inter-realization models performed as well as intra-realization models. This gives us confidence that models trained on one realization (or, ideally, a set of realizations) will inference accurately on realizations that have not been seen, yet are drawn from the same material property distributions.

## 4. CONCLUSIONS

A DNN workflow was developed to serve as the CPF in stochastic media transport using CoPS. For 1-D relationships drawn from 3-D geometry realizations, the DNN CPF was shown to yield a higher predictive accuracy than existing CoPS CPFs according to highest probability and random probabilistic metrics for sample points drawn from all five realizations. When using the JS-divergence approach, the DNN CPF is shown to outperform the CoPS2 CPF for making probability predictions on 1-D relationships drawn from 3 of 5 3-D datasets. The Jensen-Shannon DNN accuracies vary more across the realizations than the CoPS2 predictions, but yield higher average accuracy. While an analytic evaluation of true probabilities is not currently known for 3-D relationships, DNN probability predictions on a 3-D star shape compared favorably with the CoPS2 CPF, outperforming the CoPS2 CPF when drawing samples from each of five 3-D realizations. An inter-realization model inferencing study was performed for 3-D predictions in which the DNN CPF is used to make predictions on a different realization than it was trained on. The inter-realization minimum accuracy in the random probabilistic selection metric was greater than the CoPS2 accuracy on four of the five realizations, which strongly suggests that the DNN can be trained on one or more realizations and then make accurate predictions on other realizations that have the same internal mixing behavior. The DNNs were able to capture the internal material properties through data, with no *a priori* material science knowledge. The results demonstrate that the DNN can make accurate, predictive CPF probability evaluations in 1-D and 3-D spatial relationships.

The ultimate goal is to make accurate CPF predictions for arbitrary geometry types and arbitrary point arrangement shapes to yield accurate stochastic media transport in stochastic media of various mixing types. Future work includes similar accuracy comparisons to those performed in this paper for other types of mixing such as Voronoi mixing, similar accuracy comparisons for shapes other than the colinear and star shapes studied here, and accuracy studies of final transport quantities compared to benchmarks when using the DNN CPF in CoPS.

## ACKNOWLEDGEMENTS

Supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. This material is based upon work supported by a Department of Energy Nuclear Energy University Programs Graduate Fellowship. The authors thank Dr. Jacquilyn Weeks for writing consultation in the preparation of this document.

## REFERENCES

[1] E. H. Vu and A. J. Olson. "Conditional Point Sampling: A Novel Monte Carlo Method for Radiation Transport in Stochastic Media." *Trans Am Nucl Soc*, **volume 120**, pp. 471–474 (2019).

[2] E. H. Vu and A. J. Olson. "An Extension of Conditional Point Sampling to Quantify Uncertainty Due to Material Mixing Randomness." In *M&C2019*. American Nuclear Society, Portland, OR (2019).

[3] A. J. Olson and E. H. Vu. "An Extension of Conditional Point Sampling to Multi-Dimensional Transport." In *M&C2019*. American Nuclear Society, Portland, OR (2019).

[4] E. Vu and A. Olson. "Conditional Point Sampling: A Monte Carlo method for radiation transport in stochastic media." *J Quant Spectrosc and Rad Transfer* (2021, in process).

[5] C. Larmier, F. Hugot, F. Malvagi, A. Mazzolo, and A. Zoia. "Benchmark Solutions for Transport in d-Dimensional Markov Binary Mixtures." *J Quant Spectrosc and Rad Transfer* (2017).

[6] A. J. Olson, S. D. Pautz, D. S. Bolintineanu, and E. H. Vu. "Theory and Generation Methods for N-ary Stochastic Mixtures with Markovian Mixing Statistics." In *M&C 2021*. American Nuclear Society, Raleigh, NC (2021).

[7] E. Vu and A. Olson. "Amnesia radius versions of Conditional Point Sampling for radiation transport in 1D stochastic media." In *M&C 2021*. American Nuclear Society, Raleigh, NC (2021).

[8] E. H. Vu and A. J. Olson. "Recent memory versions of Conditional Point Sampling for radiation transport in 1d stochastic media." *Trans Am Nucl Soc*, **volume 123**, pp. 663–666 (2020).

[9] D. Harris. *Digital Design and Computer Architecture*. Elsevier/Morgan Kaufmann, Amsterdam Boston (2012).

[10] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung. "Digital Selection and Analogue Amplification Coexist in a Cortex-Inspired Silicon Circuit." *Nature*, **volume 405**(6789), pp. 947–951 (2000). URL https://doi.org/10.1038/35016072.

[11] X. Glorot, A. Bordes, and Y. Bengio. "Deep Sparse Rectifier Neural Networks." In G. Gordon, D. Dunson, and M. Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 315–323. PMLR, Fort Lauderdale, FL, USA (2011).

[12] T. Dozat. "Incorporating Nesterov Momentum into Adam." (2016).

[13] F. Chollet et al. "Keras." (2015). URL https://github.com/fchollet/keras.

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

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, **volume 12**, pp. 2825–2830 (2011).

[16] S. Kullback and R. A. Leibler. "On Information and Sufficiency." *Ann Math Statist*, **volume 22**(1), pp. 79–86 (1951). URL https://doi.org/10.1214/aoms/1177729694.

[17] C. Manning. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Mass (1999).

[18] L. Lee. "Measures of Distributional Similarity." In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, p. 25–32. Association for Computational Linguistics, USA (1999). URL https://doi.org/10.3115/1034678.1034693.