I&EC research
Industrial & Engineering Chemistry Research

Article

Cite This: *Ind. Eng. Chem. Res.* 2019, 58, 12312−12322

pubs.acs.org/IECR

Acceleration of the *NVT* Flash Calculation for Multicomponent Mixtures Using Deep Neural Network Models

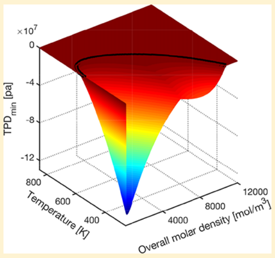
Yiteng Li,[†] Tao Zhang,[†] and Shuyu Sun*[,†,‡]

[†]Physical Science and Engineering Division (PSE), King Abdullah University of Science and Technology (KAUST), Thuwal, 23955-6900, Saudi Arabia

[‡]Institute of Geophysics and Geomatics, China University of Geosciences, Wuhan 430074, People's Republic of China

S *Supporting Information*

ABSTRACT: Phase equilibrium calculation, also known as flash calculation, has been extensively applied in petroleum engineering, not only as a standalone application for a separation process but also as an integral component of compositional reservoir simulation. Previous research devoted numerous efforts to improve the accuracy of phase equilibrium calculations, which place more importance on safety than speed. However, the equation-of-state-based flash calculation consumes an enormous amount of computational time in compositional simulation and thus becomes a bottleneck to the broad application of compositional simulators. Therefore, it is of vital importance to accelerate flash calculation without much compromise in accuracy and reliability, turning it into an active research topic in the past two decades. With the rapid development of computational techniques, machine learning brings another wave of technology innovation. As a subfield of machine learning, the deep neural network becomes a promising computational technique due to its great capacity to deal with complicated nonlinear functions, and it thus attracts increasing attention from academia and industry. In this study, we establish a deep neural network model to approximate the iterative flash calculation at given moles, volume, and temperature, known as the *NVT* flash. A dynamic model designed for *NVT* flash problems is iteratively solved to generate data for training the neural network. In order to test the model's capacity to handle complex fluid mixtures, three real reservoir fluids are investigated, including one Bakken oil and two Eagle Ford oils. Compared to previous studies that follow the conventional flash framework in which stability testing precedes phase splitting calculation, we incorporate stability test and phase split calculation together and accomplish two steps by a single deep learning model. The trained model is able to identify the single vapor, single liquid, and vapor−liquid states under the subcritical region of the hydrocarbon mixtures. A number of examples are presented to show the accuracy and efficiency of the proposed deep neural network. It is found that the trained model makes predictions at most 244 times faster than the iterative *NVT* flash calculation for the given cases and meanwhile preserves high accuracy.



1. INTRODUCTION

Phase equilibrium calculations are not only performed independently, for instance, in separation processes, but also play a significant role in compositional simulation during which complex phase behaviors often take place and accompany considerable mass transfer between hydrocarbon phases. As secondary recovery methods hardly bring further economical production from conventional reservoirs, enhanced oil recovery (EOR) techniques, such as miscible flooding, are increasingly employed to improve oil productivity. The simulation of EOR processes requires accurate prediction of phase behaviors that could be modeled by the commonly used equation of states (EOSs), such as the Peng−Robinson EOS[1] or the Soave−Redlich−Kwong EOS,[2] thus placing great importance on the EOS-based phase equilibrium calculation. However, due to the large nonlinearity of phase equilibria problems, the popular iterative solution method for stability test and phase split calculation at given conditions (usually pressure, temperature, and feed composition) is computationally expensive. It has been reported that up to 70% of computational time is consumed by flash calculation in compositional reservoir simulation. Although good initial approximations for phase splitting calculation can be obtained from the stability test, accuracy is preserved at the cost of efficiency, which is attributed to the repeated stability analysis, the high-dimension formulation, and the iterative algorithm itself. Since speed is one of critical concerns in current compositional simulators, speeding up flash calculations without too much compromise in accuracy and reliability is an active research topic in both academia and industry.

In reality, a hydrocarbon mixture may consist of tens to hundreds of species, which is far more than the number of components in numerical simulations. Thus, in order to accelerate phase equilibrium calculation, what first comes to mind is to lump the fluid mixture into a smaller number of pseudocomponents without losing much accuracy of the EOS model.[3] Clearly, the fewer the number of components is, the more computational time can be saved and meanwhile the more accuracy will be lost. Despite this, the speedup given by

Received: February 2, 2019
Revised: June 15, 2019
Accepted: June 15, 2019
Published: June 15, 2019

ACS Publications

DOI: 10.1021/acs.iecr.9b00527

reducing the number of components is barely satisfactory, therefore stimulating a large batch of researchers to develop acceleration strategies for phase equilibrium calculation over the past two decades. Among them, the reduction method was first proposed by Michelsen[4] in 1986, showing that only three equations need to be solved in the two-phase flash problem under the van der Waals mixing rule. The central idea of the reduced approach is to decrease the number of nonlinear equations and unknowns by taking advantage of the low-rank property of the binary interaction coefficient (BIC) matrix. However, his method, assuming all zero binary interaction coefficients, is severely circumscribed so that it is hardly applicable to real reservoir fluids. Later, Jensen and Fredenslund[5] extended Michelsen's research by introducing one non-hydrocarbon component for practical application. The nonzero BICs between the non-hydrocarbon and hydrocarbons yield five reduced variables, irrespective of the number of components, but still limits the applicability of their approach. A generalized reduction theorem was then presented by Hendriks,[6] which gets rid of the restriction on the BIC matrix. He reformulated phase equilibria problems with the transformed chemical potential, as a function of reduction variables. By introducing the truncated spectral method, Hendriks and Van Bergen[7] simulated two-phase equilibrium in reduced space using Newton−Raphson iterations.

Similarly, Pan and Firoozabadi developed reduction models for phase stability testing[8] and phase equilibrium calculation[9] based on the spectral theory of linear algebra. For stability analysis, they found the tangent-plane distance (TPD) function in the reduced space has a smoother surface accompanied by a unique minimum, both of which significantly improve the robustness of their method. On the other hand, by extensively testing the conventional approach and the reduced approach of the two-phase flash calculation, they concluded that the direct solution of the reduced model makes flash calculations more efficient and robust. In addition, Li and Johns[10] developed their reduction model for rapid flash calculation by decomposing the BICs into two parts. The new phase split formulation includes at most six reduced parameters, either degenerating to Michelsen's reduction model with three reduced parameters when all BICs are zero or the five-reduced-parameter model of Jensen and Fredenslund when the nonzero BICs only exist between a single component and the other components. Moreover, their method avoids solving the nonlinear Rachford−Rice equation,[11] a difficulty in phase splitting calculation, and meanwhile retains the accuracy of conventional approaches. Recently, Nichita and Graciaa[12] designed a reduced flash formulation with a new set of independent variables and error equations, leading to simpler expressions of the Jacobian matrix. Compared to the classical reduction methods, their approach exhibits better efficiency and meanwhile preserves the robustness. Instead of decomposing BIC matrices by the conventional spectral analysis approach, Gaganis and Varotsis[13] introduced a new decomposition method to minimize the approximation errors of the energy parameter, which allows the phase equilibrium calculation to be performed at a given accuracy with fewer reduced variables. Even though all the above reduction methods have shown great potential in acceleration of the flash calculation, recent research[14−17] questioned its efficiency against the conventional method. No agreement has been reached at which points the reduced methods become faster than the conventional ones. However, the reduction model seems to be more applicable to mixtures with 6−15 components.

In addition, there are many other approaches to speed up flash calculation within the framework of compositional simulator. Wang and Stenby[18] developed a noniterative flash calculation algorithm on the assumption that thermodynamic equilibrium is instantaneously established at any time of the simulation. With flash calculations decoupled from the compositional flow, the iterative flash procedure can be linearly approximated so that the mole numbers of components in each phase are computed based on the solution of compositional flow between the previous and current time steps. There is no need to solve the Rachford−Rice equation for the phase saturation, thereby improving the efficiency of their method further. However, the reduction of CPU time varies case by case. Given that the repetitive phase stability test is time-consuming, Rasmussen et al.[19] proposed the shadow region method in order to bypass unnecessary stability testing based on the location of any given condition in the phase diagram. It is assumed that the pressure, temperature, or compositions have small changes in a certain grid block between two consecutive time steps. In the two-phase region, the stability test procedure is completely bypassed by initializing flash calculation with the equilibrium solution in the last time step. Furthermore, a one-sided stability test, initialized by the prestored nontrivial stability solution, replaces the full stability test within the shadow region, while outside the shadow region the distance to the critical point is used to determine whether stability analysis is skipped or not. Consequently, the computation time spent on flash calculation is significantly reduced. On the other hand, to accelerate the phase splitting calculation, Voskov and Tchelepi[20−22] presented the compositional space adaptive tabulation (CSAT) method in which a tieline table is constructed using the prestored phase split solutions to parametrize the compositional space. If the investigated composition is on the predetermined tieline or its extension, the phase equilibria solution can be approximated on the basis of the tieline compositions; otherwise, the standard flash procedure is performed and an additional tieline will be added into the lookup table. It is noted that the accuracy and efficiency of the CSAT method heavily depend on the number of stored tielines, as well as the tolerance set to accept the feed composition. As the tieline table becomes larger, we gain more accuracy but lose more efficiency as well. Inspired by Voskov and Tchelepi's research, Belkadi et al.[23] suggested using the tieline distance-based approximation (TDBA) method to approximate flash solutions in the two-phase region. In their approach, the shortest distance from the new feed composition to the tieline in the same grid block is used to determine if the rigorous flash solution is approximated, which minimizes the access and buildup process of the tieline table.

Recently, Yan et al.[24] accelerated phase behavior calculations by interpolating initial approximations from the compositional space constructed, prior to the flash calculation, for reservoir fluids with both fixed and dramatically varying composition. For those hydrocarbon mixtures with significant compositional change during simulation, an additional pseudokey component is introduced, together with pressure, to control the phase behavior prediction. Although their approach exhibits an enhanced performance under Rachford−Rice preconditioning, the resolution of the compositional spaces constructed offline play a critical role in this kind of methods as expected. In addition, Wu et al.[25] utilized the sparse grid surrogate model to approximate flash results during the two-phase compositional simulation. Under a similar interpolation philosophy, a remarkable speedup of compositional simulation was achieved, but several intrinsic defects restrict the applicability of their method. Subsequently, Wu and Chen[26] extended their previous work by approximating the equilibrium solution directly from the sparse grid instead of the surrogate model. The introduction of the "layer" concept and array structure makes the sparse grid construction efficient and inexpensive. However, two issues, including the violation of saturation boundedness and inaccurate detection of the single-phase region, need to be resolved for reliable application of sparse grid methods in phase equilibrium calculation.

Apart from the aforementioned accelerating strategies, other alternatives, such as negative flash calculation[27,28] or linear transformation of the Rachford−Rice equation,[29] can be used to improve the efficiency of flash calculation as well. All the above speedup approaches are only the tip of the iceberg, and few of them can avoid iteratively solving phase equilibria problems. In addition to the iterative nature of flash algorithms, the increasing demand for high-resolution reservoir models, usually with millions of grid blocks, also sharply increases the computational costs of phase equilibrium calculations during compositional simulation. Thus, it is imperative to perform both phase stability test and phase split calculation in a fast, noniterative, and robust way. With the rapid development of computing techniques, machine learning has become a powerful tool to deal with those sophisticated and time-consuming problems. Since 2012, due to the breakthrough of AlexNet,[30] deep learning, a subset of machine learning, made revolutionary progress in image recognition, language processing, artificial intelligence, and so forth. The great potential of machine learning has drawn close attention from the petroleum industry, and remarkable efforts have been made in many aspects, such as reservoir characterization,[31] development planning optimization,[32] and permeability estimation.[33] Furthermore, a number of studies have been devoted to speeding up both phase stability test and phase split calculation using the machine learning approach. Gaganis and Varotsis[34] designed a stability discriminating function, which yields the same sign and zero points as the conventional tangent plane distance (TPD) approach, to eliminate the time-consuming iterations in stability analysis. Under the framework of support vector machine (SVM),[35−37] phase stability test is treated as a binary classification problem and has an explicit expression on the basis of pressure, temperature, and feed composition. For phase split calculation, they developed regression models using the artificial neural network (ANN) to predict the conventional equilibrium coefficients[38] and later the prevailing reduced variables,[39] in order to approximate the closed-form solution of the original flash formulation and the reduced one, respectively. Recently, Gaganis[40] presented a new classification technique using two simplified discriminating functions to achieve rapid stability determination without much compromise in accuracy to depict the phase boundary.

Instead of using SVM, Kashinath et al.[41] utilized two relevance vector machine (RVM) classifiers for stability analysis. The supercritical classifier identifies the supercritical state at the given condition, while the subcritical classifier predicts the number of stable phases. Similar to Gaganis and Varotsis's approach, they estimated the equilibrium coefficients for the phase split problem using the regression model established on the basis of ANNs. Unlike the aforementioned works, Wang et al.[42] built machine learning models based on ANNs for both stability test and phase split calculation. The ANN model for stability testing predicts the saturation pressure, which is used for stability determination in comparison with the system pressure. The other ANN model provides initial mole fractions and equilibrium coefficients for phase split calculations. In addition, Li et al.[43] approximately solved the vapor−liquid equilibria problem using a deep learning model. Different from the aforementioned works, they directly estimated the equilibrium compositions rather than the equilibrium coefficients that work as the intermediate variables. A variety of effects are investigated, such as the number of hidden layers, the number of nodes per layer, and the type of activation function, on the performance of the deep neural network, which helps them optimize their model to improve efficiency and accuracy. However, as a shortcoming of their work, they only discussed the cases where both vapor and liquid phases are present.

In this study, instead of using the conventional *NPT* formulation, the phase equilibrium problem is formulated at constant moles, volume, and temperature (the so-called *NVT* flash). A dynamic model is designed for this isothermal−isochoric phase equilibria problem and iteratively solved by Newton−Raphson methods with line search. To validate the capacity of our dynamic model, as well as the deep learning model, for handling complex fluid mixtures, three realistic reservoir fluids are investigated: one from the Bakken shale formation includes five components and the other two from the Eagle Ford shale formation consist of 8 and 14 components, respectively. With the training data provided by the iterative flash calculation, we develop a deep neural network to predict phase compositions at different overall concentrations and temperatures with fixed feed composition, enabling it to deal with depletion processes. To overcome the overfitting issue inherent in neural networks, the dropout technique is applied in combination with a regularized loss function. In addition, batch normalization significantly reduces the time spent on training the network model, thus enabling us to construct a more complex multilevel structure for better approximation. The network configuration is designed based on our previous research.[43] Different from the preceding studies, one key effort of this study is to incorporate stability test and phase split calculation together so that both steps in the conventional flash framework can be accomplished by a single deep neural network. An independent variable is introduced in the training process to determine the number of stable phases. Meanwhile, the trained model identifies the fluid state (e.g., single vapor, single liquid, or vapor−liquid) under the subcritical region. We present a number of numerical examples to validate the accuracy and efficiency of the proposed deep learning model.

The remainder of this paper is organized as follows. In section 2, we formulate the dynamic model for the *NVT* flash problem in the bulk phase and compare our simulation results with the published ones. In section 3, the deep learning algorithm is elaborated in detail with some techniques to enhance its performance. We discuss the prediction accuracy of the proposed deep neural network and compare its efficiency with the iterative flash calculation in section 4. At the end, we make our conclusions in section 5.

2. *NVT* FLASH CALCULATIONS

Compared to the classical *NPT* flash calculation, the *NVT* flash exhibits some advantages and has attracted a lot of attention from researchers in recent years. Without inverting the equation of state, the *NVT* formulation has a unique solution and thus eliminates the root-selection procedure that usually takes place in *NPT* flash problems. Additionally, volumes of pure substances under saturation pressure cannot be uniquely determined using the *NPT* formulation, since all states (two phases, single vapor phase, or single liquid phase) of a pure substance share the same pressure on the phase boundary.[44] Similar behavior has been observed in multicomponent mixtures with three or four phases as well.[45] Despite the fact that the *NPT* flash is the most commonly used flash technique in compositional simulators, numerous efforts have been made to improve the performance of the *NVT* flash calculation and also extend its applications. Jindrová and Mikyška[44] developed a fast and robust *NVT* flash algorithm for two-phase equilibria problems by directly minimizing the total Helmholtz free energy. Subsequently, they extended their research to multiphase systems and investigated the equilibrium problem involving aqueous phase through the cubic-plus-association EOS.[45] Similarly, Nichita[46] formulated the *NVT* flash problem as unconstrained minimization of the Helmholtz free energy as well. Unlike Jindrová and Mikyška's method, volume is treated as a dependent variable on mole numbers, leading to a "*PT*-like" iteration framework for the *NVT* flash calculation. To model the dynamic process from any nonequilibrium state to the equilibrium state, Kou et al.[47] established evolution equations for mole numbers and volume, which were solved by a well-designed energy-stable numerical algorithm. In addition to the aforementioned bulk phase flash problems, the confined phase behaviors at constant moles, volume, and temperature have been studied by taking into account adsorption,[48] capillary pressure,[49,50] or confinement effect resulting from the interaction between fluid molecules and pore walls.[51,52] In this study, we restrict our discussion to the bulk phase equilibrium of two-phase systems in which the fluid mixture is described by the Peng–Robinson EOS.

2.1. Dynamic Model for Bulk Phase Equilibria.

Let us assume a fluid mixture of M components stays in the vapor–liquid equilibrium at the specified moles, volume, and temperature. Thus, the total Helmholtz free energy of this system can be expressed as

$$F = f(\mathbf{n}^{\mathrm{G}})V^{\mathrm{G}} + f(\mathbf{n}^{\mathrm{L}})V^{\mathrm{L}} \tag{1}$$

where $\mathbf{n}^{\alpha} = [n_1^{\alpha}, ..., n_M^{\alpha}]^{\mathrm{T}}$ and V^{α} denote the molar density and volume of phase α (α = G, L), respectively. The Helmholtz free energy density $f(\mathbf{n})$ has the following form:

$$f(\mathbf{n}) = RT\sum_{i=1}^{M} n_i(\ln n_i - 1) - nRT\ln(1 - bn) + \frac{a(T)n}{2\sqrt{2}b}\ln\left(\frac{1 + (1 - \sqrt{2})bn}{1 + (1 + \sqrt{2})bn}\right) \tag{2}$$

In eq 2, $n = \sum_j n_j$ is the overall molar density, R is the ideal gas constant, and T is the temperature. The Peng–Robinson parameters a and b are functions of the molar composition. Considering the total mole numbers $\mathbf{N}^{\mathrm{t}} = [N_1^{\mathrm{t}}, ..., N_M^{\mathrm{t}}]^{\mathrm{T}}$ and total volume V^{t} are fixed, the following mole and volume constraints have to be satisfied whenever both vapor and liquid phases are present in the system:

$$N_i^{\mathrm{G}} + N_i^{\mathrm{L}} = N_i^{\mathrm{t}} \qquad i = 1, ..., M \tag{3}$$

$$V^{\mathrm{G}} + V^{\mathrm{L}} = V^{\mathrm{t}} \tag{4}$$

where N_i^{α} represents the mole number of component i in phase α.

In any two-phase system, we can arbitrarily select the mole composition and volume of one phase as primary variables and compute the counterparts of the other phase by the aforementioned mole and volume constraints. Here the mole composition and volume of the vapor phase, denoted by $\mathbf{N}^{\mathrm{G}}$ and V^{G}, are chosen as the primary variables so that the Helmholtz free energy is reduced as $F(\mathbf{N}^{\mathrm{G}},V^{\mathrm{G}})$. The partial derivatives of the Helmholtz free energy in eq 1 with respect to N_i^{G} and V^{G} yield

$$\frac{\partial F(\mathbf{N}^{\mathrm{G}}, V^{\mathrm{G}})}{\partial N_i^{\mathrm{G}}} = \mu_i(\mathbf{n}^{\mathrm{G}}) - \mu_i(\mathbf{n}^{\mathrm{L}}) = \mu_i^{\mathrm{G}} - \mu_i^{\mathrm{L}} \tag{5}$$

$$\frac{\partial F(\mathbf{N}^{\mathrm{G}}, V^{\mathrm{G}})}{\partial V^{\mathrm{G}}} = p(\mathbf{n}^{\mathrm{L}}) - p(\mathbf{n}^{\mathrm{G}}) = p^{\mathrm{L}} - p^{\mathrm{G}} \tag{6}$$

where μ_i^{α} and p^{α} are the chemical potential of component i and pressure in phase α, respectively. By applying the chain rule, the time derivative of the total Helmholtz free energy $F(\mathbf{N}^{\mathrm{G}},V^{\mathrm{G}})$ can be derived as follows:

$$\frac{\partial F}{\partial t} = \sum_{i=1}^{M}\frac{\partial F}{\partial N_i^{\mathrm{G}}}\frac{\partial N_i^{\mathrm{G}}}{\partial t} + \frac{\partial F}{\partial V^{\mathrm{G}}}\frac{\partial V^{\mathrm{G}}}{\partial t} = \sum_{i=1}^{M}(\mu_i^{\mathrm{G}} - \mu_i^{\mathrm{L}})\frac{\partial N_i^{\mathrm{G}}}{\partial t} + (p^{\mathrm{L}} - p^{\mathrm{G}})\frac{\partial V^{\mathrm{G}}}{\partial t} \tag{7}$$

The mole and volume evolution equations are formulated based on Onsager's reciprocal principle, shown as below, to characterize the dynamic bulk phase equilibrium process.

$$\frac{\partial N_i^{\mathrm{G}}}{\partial t} = \sum_{j=1}^{M}\psi_{i,j}(\mu_j^{\mathrm{G}} - \mu_j^{\mathrm{L}}) + \psi_{i,M+1}(p^{\mathrm{L}} - p^{\mathrm{G}}) \tag{8}$$

$$\frac{\partial V^{\mathrm{G}}}{\partial t} = \sum_{j=1}^{M}\psi_{M+1,j}(\mu_j^{\mathrm{G}} - \mu_j^{\mathrm{L}}) + \psi_{M+1,M+1}(p^{\mathrm{L}} - p^{\mathrm{G}}) \tag{9}$$

According to the second law of thermodynamics, the total Helmholtz free energy in a closed system is supposed to dissipate with time, implying the Onsager coefficient matrix $\mathbf{\Psi} = (\psi_{i,j})_{i,j=1}^{M+1}$ shall be negative definite. Following a similar strategy in ref 50, we compute the Onsager coefficient matrix based on the equilibrium conditions. A revised modified Cholesky factorization is applied to ensure its negative definiteness; see ref 53 for more details. It is worth mentioning that this well-designed coefficient matrix helps to achieve the persistent dissipation of the Helmholtz free energy, but also simultaneously solve all evolution equations as a whole system in an iteration approach. When the Helmholtz free energy is minimized, the investigated system reaches the equilibrium state, including the chemical equilibrium ($\mu_i^{\mathrm{G}} = \mu_i^{\mathrm{L}}$) and the pressure equilibrium ($p^{\mathrm{G}} = p^{\mathrm{L}}$).

2.2. Numerical Experiments.

To perform the *NVT* flash calculation, each phase split calculation is initialized by the stability test in this study. Detailed implementation procedures

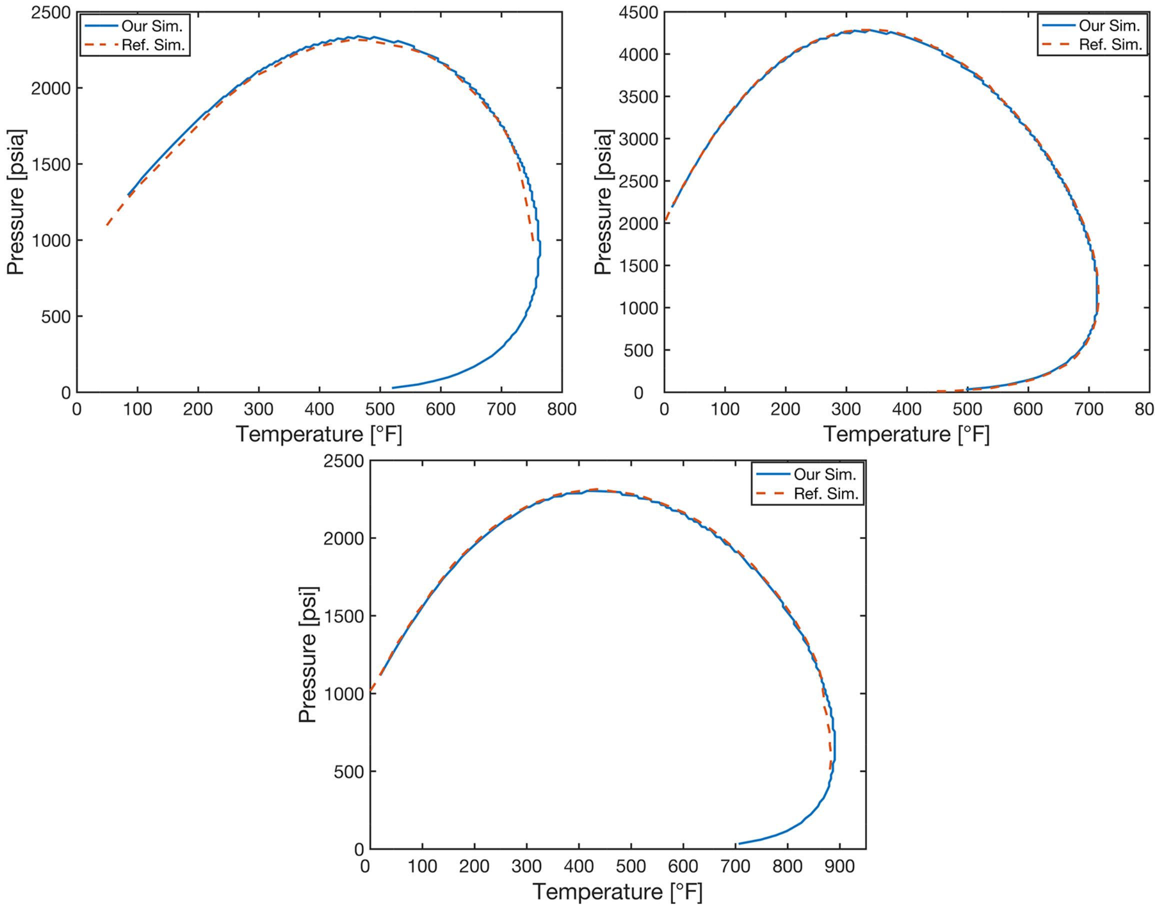


Figure 1. Bulk phase envelopes for the Bakken oil (top left), EagleFord1 oil (top right), and EagleFord2 oil (bottom).

of the bulk phase stability testing can refer to either the work of Mikyška and Firoozabadi[54] or the work of Kou and Sun[49] with negligible capillary pressure. To construct an energy-stable time-marching scheme, which preserves the dissipation of the Helmholtz free energy, the convex splitting technique is employed to decompose the Helmholtz free energy density and chemical potential. In addition, we use a semi-implicit scheme[55] to discretize the mole and volume evolution equations. Specifically, the convex parts of the chemical potential and Helmholtz free energy density are treated implicitly while the concave parts are treated explicitly. The resultant nonlinear system is solved by the Newton−Raphson method with line search. We terminate the Newton iteration if either the maximal number of iterations (500) or the Euclidean norm of the directional vector less than 10^{-8} is satisfied.

In order to validate the accuracy of the proposed dynamic model, three reservoir fluid samples are tested: the Bakken oil consists of five components and the two Eagle Ford oils have 8 and 14 components. For brevity, the Eagle Ford oils are named EagleFord1 for the eight-component sample and EagleFord2 for the other one with 14 components. The molar composition and compositional properties for the Bakken, EagleFord1, and EagleFord2 oils are included in the Supporting Information (see Tables S1−S3). Binary interaction coefficients for these three samples can be found in refs 56, 57, and 58, respectively. To achieve a smooth phase envelope, we uniformly discretize the computational domain of the overall molar density and temperature into a 201 × 201 grid. For the Bakken oil, the investigated molar density varies from 10 to 10 000 mol/m^3 and the temperature ranges from 300 to 850 K. Moreover, the overall concentrations of EagleFord1 and EagleFord2 samples are [10, 12 000] and [10, 10 000] mol/m^3, while both of them have the same temperature interval $T \in [260, 850]$ K. Figure 1 shows the computed phase envelopes for the Bakken oil (top left), the EagleFord1 oil (top right), and the EagleFord2 oil (bottom). It can be seen that the equilibrium solutions of the dynamic model agree with the published results in refs 56−58 very well. In the following, we use these equilibrium results of *NVT* flash calculation, especially the mole fractions of vapor and liquid phases, to train our deep learning model.

3. DEEP LEARNING ALGORITHM

Artificial neural networks are computational models that are able to unearth the underlying correlations and features of input data and process them in a manner analogous to the biological nervous system. Such a characteristic enables the ANN model to have great problem-solving capability. Various applications have been developed on the basis of ANNs, which are performed in either supervised or unsupervised approach, such as machine learning or pattern recognition. Unlike the unsupervised ANN model, the supervised machine learning tries to seek a function that can be used to predict the values of desired variables from input data under a given accuracy. In this study, instead of using shallow neural networks, a deep neural network, which consists of multiple nonlinear transformation layers, is applied to replace the iterative *NVT* flash algorithm in order to accelerate phase equilibrium calculations. Figure 2 displays the schematic diagram of the proposed deep

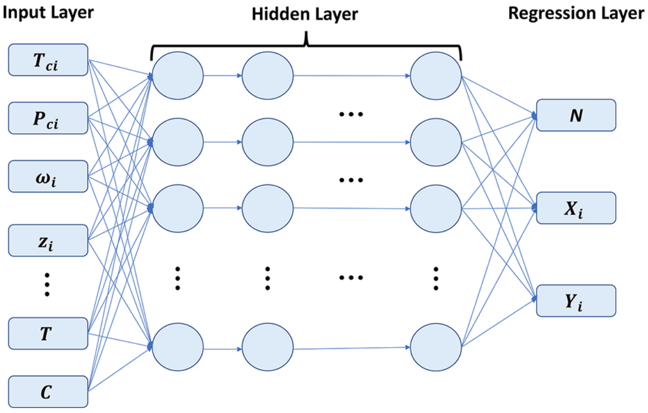


Figure 2. Deep neural network to approximate *NVT* flash calculation.

neural network, which processes the input data, including compositional properties (e.g., critical pressure P_c, critical temperature T_c, acentric factor ω), overall mole fractions $\boldsymbol{z} = [z_1, ..., z_M]^T$, overall molar concentration C, and temperature T in a sequential approach. Each activation layer transforms the output of the preceding layer and yields results as input to the following one. This hierarchical structure enables the deep neural network to extract useful features and patterns layer by layer and then learn complicated correlations behind the enormous data volume. Benefiting from the powerful capacity of the deep learning model, we accomplish both stability test and phase split calculation by a single network model so that the number of phases is determined free of additional stability test. This differs from the recent research on the acceleration of flash calculation by machine learning models, in which the stability test is preserved to determine if the investigated fluid is stable or not as the prevailing flash framework does. As a result, the output of the regression layer in the proposed neural network model includes three parts: the number of phases (N), the mole fraction of vapor components X_i, and the mole fraction of liquid components Y_i.

The performance of a neural network model can be evaluated based on the loss, which is defined as the difference between the model output and the ground truth. Here we use the mean square error as the loss function, evaluated in each iteration, to describe the deviations of the model prediction from the solution of iterative flash calculation. However, overfitting, as the inherent issue of neural networks, gravely affects the performance of the deep learning algorithm, which manifests that the trained model exhibits excellent performance based on the training data while it yields terrible predictions for its further application. This inconsistent behavior is attributed to the numerous parameters of the model waiting for optimization and insufficient training data, thus making the model overparametrized. In order to protect the deep neural network model from the overfitting issue, additional constraints are imposed on the parameters to reduce the freedom of the model. Generally, if the model is overfitted, the norm of the weight parameters will become very large. To avoid the overfitting problem, an additional constraint is added on the norm of weight parameters to penalize the large weights. In practice, we introduce a regularization term, related to the norm of the weights, in the loss function, which yields

$$L = \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{o} - \hat{\mathbf{o}}\|^2 + \lambda \, \|\mathbf{W}\|_2^{\,2} \tag{10}$$

where N is the number of the training data, $\mathbf{o}$ is the model output, $\hat{\mathbf{o}}$ is the observation value, $\mathbf{W}$ is the weight parameters, and λ is the regularization coefficient for weight penalization. In addition to the regularized loss function, we also use the dropout technique to resolve the overfitting problem, which reduces the freedom of the deep neural network model by abandoning certain nodes and connections during the training stage. In particular, with a specified threshold probability p ($0 < p < 1$), each node of a layer is evaluated independently and retained if its probability is less than the threshold value; otherwise, the node will be removed. The connections belonging to those disregarded nodes are removed as well, therefore yielding a reduced neural network for training. After this, the full model is recovered by inserting the removed nodes and connections back and enters the next training cycle.

In order to design a robust and efficient neural network, it is crucial to properly initialize the weight parameters since they have significant influence on the convergence rate and performance of the deep learning algorithm. If the initial weights are underestimated, the variance of the input data will drop to a very small value through some intermediate transformations, which reduces the model complexity and damages its performance. On the other hand, the overestimated weight parameters increase the variance of the input rapidly, making the gradient either vanishing or exploding and eventually failing to train the model. In this study, the weight parameters are initialized following the Gaussian distribution. To control the variance during the training process, we initialize the weights with a variance δ and make the output variance always the same as the input variance on the same activation layer. For example, in Figure 3, the variance of y is expected to expand as follows:

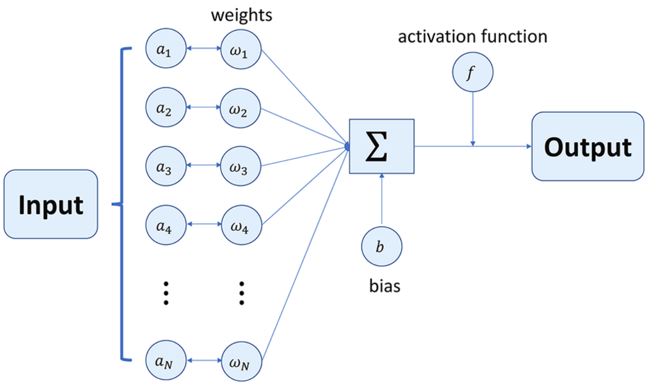


Figure 3. Flowchart of the learning process in each node.

$$\begin{aligned} \mathrm{var}(y) &= \mathrm{var}(w_1 \times a_1 + w_2 \times a_2 + ... + w_n \times a_n + b) \\ &= \mathrm{var}(w_1) \times \mathrm{var}(a_1) + \mathrm{var}(w_2) \times \mathrm{var}(a_2) + ... \\ &\quad + \mathrm{var}(w_n) \times \mathrm{var}(a_n) \\ &= n \times \mathrm{var}(w_i) \times \mathrm{var}(a_i) \end{aligned} \tag{11}$$

with the identical distribution assumption for all w_i and a_i. Since we want $\mathrm{var}(y)$ equal to $\mathrm{var}(a_i)$, the following equality has to be satisfied:

$$n \times \mathrm{var}(w_i) = 1 \tag{12}$$

Therefore, the weights of each layer are initialized following the Gaussian distribution with the variance of $1/n$, where n is the number of weights in that layer. This initializer is known as the Xavier initializer.

It has to be mentioned that training a deep neural network is time-consuming due to massive parameters waiting for optimization over all the hidden layers. In addition, some undesirable properties of the multilevel model, as well as the input data of different scale at each activation layer, slow down the training process. Among others, the varying distribution of the activations plays a critical role in the slow convergence of training deep neural networks, since each layer has to adjust itself to a new distribution in each training step. To speed up the learning process, batch normalization is employed to adjust and scale the outcomes of the preceding layer, which can be regarded as a preprocessing step for each activation layer. Consequently, a significant improvement of the training speed is achieved, which enables training of more complicated models in a reasonable amount of time.

In this study, we employ a deep neural network model to approximate the iterative *NVT* flash calculation using TFlearn on a MacBook Pro laptop with an Intel Core i7 processor. A simplified flowchart of the working process in each node of the proposed model is presented in Figure 3. Let $\mathbf{a}_i$ and $\mathbf{y}_i$ denote the input and output of the ith activation layer, respectively. The output of the ith layer is given by

$$\mathbf{y}_i = f_i(\mathbf{W}_i \times \mathbf{a}_i + \mathbf{b}_i) \qquad (13)$$

where $\mathbf{W}_i$ is the weight, $\mathbf{b}_i$ is the bias, and f_i is the activation function of the ith layer. The neural network in Figure 2, for instance, with three activation layers, can be expressed as

$$\mathbf{o} = f_3(\mathbf{W}_3 \times f_2(\mathbf{W}_2 \times f_1(\mathbf{W}_1 \times \mathbf{x}_1 + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3) \qquad (14)$$

As shown in eq 14, the activation function is the source of the nonlinearity in neural networks, which helps the deep learning algorithm approximate complex functions and gives it great problem-solving capability. There are a variety of activation functions, including Rectified linear unit (ReLU), Parameteric rectified linear unit (PReLU), TanH, Sigmoid, Softplus, Softsign, Leaky rectified linear unit (Leaky ReLU), Exponential linear unit (ELU), and Scaled exponential linear unit (SELU). Among them, the most commonly used activation functions are ReLU

$$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases} \qquad (15)$$

and Sigmoid

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \qquad (16)$$

4. RESULTS AND DISCUSSION

The training data for the deep learning model are provided by the *NVT* flash calculation. Three real reservoir fluids are investigated, including the five-component Bakken oil, eight-component EagleFord1 oil, and 14-component EagleFord2 oil. The compositional parameters for each reservoir fluid are presented in the Supporting Information. Following the theory and practice described in section 3, a deep neural network is established with five activation layers, each of which contains 100 nodes, and a total of 4000 iterations. "ReLU" is chosen as the activation function. It is worth mentioning that the performance of this network configuration has been validated in ref 43. To investigate the effect of data size on the performance of the deep neural network model, we calculate equilibrium results of the *NVT* flash for the EagleFord1 oil on the same computational domain with 51 × 51, 71 × 71, 101 × 101, 151 × 151, 201 × 201, and 301 × 301 uniform grids. In addition, the results of the Bakken oil and EagleFord2 oil are computed on the specified concentration and temperature intervals, which are uniformly divided into 301 × 301 grids. All eight data sets are used to train the proposed neural network, and the efficiency and accuracy of the trained model are tested. One key effort of this study is to investigate the possibility in achievement of both stability test and phase split calculation by a single neural network model, which is different from the conventional two-step framework that all the preceding research follows based on machine learning models.

4.1. Deep Learning Model Training.

As shown in Figure 2, the compositional properties of fluid components, overall molar concentration, and temperature are used as the input, and the proposed deep neural network predicts mole fractions of components in both vapor and liquid phases. The key parameters of the model are the weights of each activation layer, which control the model prediction under the given input data. At the beginning, those weights are initialized randomly, implying that the model initially yields useless results. To approximate the *NVT* flash calculation by the deep learning model, we optimize the weight parameters to fit the equilibrium mole fraction of vapor and liquid components. In the following, 90% of the data are used to train our network model, while the remaining 10% data are used for validation unless otherwise noted.

Table 1. Performance of the Deep Learning Model for Different Data Samples

source	N_{train}	N_{test}	t_{train} (s)	t_{test} (s)	ε_{a}	ε_{r}
Bakken (301 × 301)	81540	9061	2352	9.2	0.02362	0.02885
EagleFord1 (51 × 51)	2340	261	42	4.1	0.02946	0.03582
EagleFord1 (71 × 71)	4536	505	134	5.9	0.01930	0.02410
EagleFord1 (101 × 101)	9180	1021	235	6.3	0.01378	0.01723
EagleFord1 (151 × 151)	20520	2281	512	7.1	0.01301	0.01647
EagleFord1 (201 × 201)	36361	4040	1232	7.9	0.01261	0.01580
EagleFord1 (301 × 301)	81540	9061	2531	8.6	0.01252	0.01571
EagleFord2 (301 × 301)	81540	9061	2468	8.9	0.01684	0.01964

Table 1 presents the training data size (N_{train}), testing data size (N_{test}), training time (t_{train}) and testing time (t_{test}) of the deep neural network with different data sets. Here t_{test} represents the time that the trained model spent on estimating equilibrium mole fractions for the flash problem of the same data size. For instance, it takes 7.9 s for the trained model to predict the mole fractions of the EagleFord1 oil on a 201 × 201 grid. In addition, the mean absolute error (ε_{a}) and relative error (ε_{r}) are presented in Table 1 as well. Clearly, for the

EagleFord1 oil, as the number of input data becomes larger, the training time significantly increases, while the testing time does not change too much. Furthermore, we observe that both absolute and relative prediction errors continue to decrease with the data size increasing. Under the same data volume, it seems the more components are involved, the larger prediction error the deep neural network model exhibits. However, the Bakken oil makes an exception and yields greater error than the Eagle Ford oils. This might be attributed to the underneath correlations between different components differ from the investigated fluid mixtures so that the trained network model yields different accuracy. Essentially, the composition of the Bakken oil is quite different from the compositions of the two Eagle Ford samples, the latter of which exhibit some similarities to some extent. This may explain why our observation disagrees with the expectation. Table 2 compares the computational time of *NVT* flash calculations to the testing time of the deep neural network for the EagleFord1 oil with different data sizes. It can be seen that the testing time is much less than the computational time of flash calculations. When the data size reaches 301 × 301, the trained model makes predictions 244 times faster than the iterative flash calculation.

Table 2. Comparison of Computational Time of the Iterative Flash Calculation and Testing Time of the Deep Neural Network for the EagleFord1 Oil with Different Data Sizes

data size	t_{flash} (s)	t_{test} (s)
51 × 51	61	4.1
71 × 71	122	5.9
101 × 101	237	6.3
151 × 151	547	7.1
201 × 201	1021	7.9
301 × 301	2100	8.6

Figure 4 shows the loss function decreases with iterations, indicating the predictions of the trained model are approaching the results of the *NVT* flash calculation. The inset figure displays the local variation of the total loss function after a sharp decrease within the first 100 iterations, which is indistinguishable in the main figure. It is observed that the difference between deep learning predictions and iterative flash solutions continues to decrease, but the decline rate significantly slows down after about 40 or 50 iterations. For the EagleFord1 oil, the increasing data size, from 51 × 51 to 201 × 201, helps to accelerate the reduction of the loss function. However, such a speedup effect vanishes with more data used to train the model. In comparison, the loss function of the EagleFord2 oil exhibits a faster decline rate with the same data size (e.g., 301 × 301) as the EagleFord1 oil until it reaches a certain threshold, below which the loss function remains almost steady.

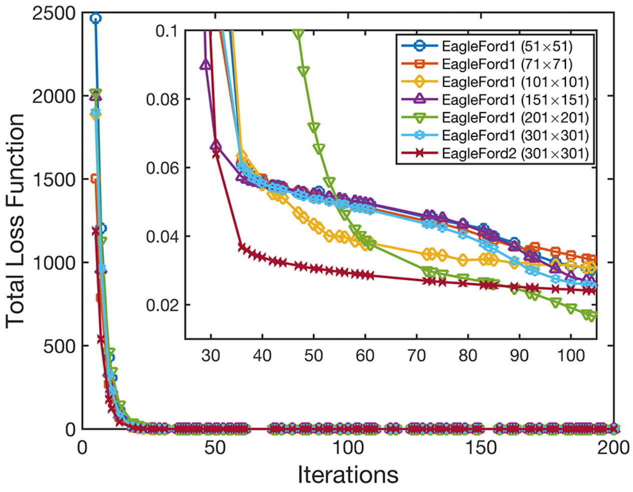


Figure 4. Variation of total loss function for EagleFord1 and EagleFord2 oils. The inset figure zooms in the first 100 iterations with a loss function less than 0.1.

4.2. Phase Number Characterization. Previous research follows the conventional framework of flash calculation, which consists of two separate stages: stability test and phase split calculation. In contrast, one key effort of this study is training the deep neural network to automatically determine the number of phases without additional stability testing. Under this guideline, the proposed neural network model completes both stability test and phase split calculation simultaneously, and predicts the number of phases together with the molar compositions of vapor and liquid phases, as shown in Figure 2. To validate the prediction accuracy of our model on phase stability problems, we compare the number of phases computed by the deep learning model and iterative flash calculation. Figure 5 displays the phase number of the EagleFord1 oil as a function of the temperature at two overall concentrations: 1988 (top) and 2648 mol/m^3 (bottom), respectively. Note that we only plot the results slightly across the phase boundary in Figure 5. Overall, our deep learning model successfully captures the phase transition process from the vapor−liquid region to the single vapor region. The single-phase state, either single vapor or liquid, is determined based on the fact that the mole fractions of components in the existing phase should approach overall mole fractions. Specifically, if the mixture is a single vapor (liquid) phase at the given condition, the mole fractions of the vapor (liquid) components should be very close to the overall mole fractions z_i, and the mole fractions of the liquid (vapor) components

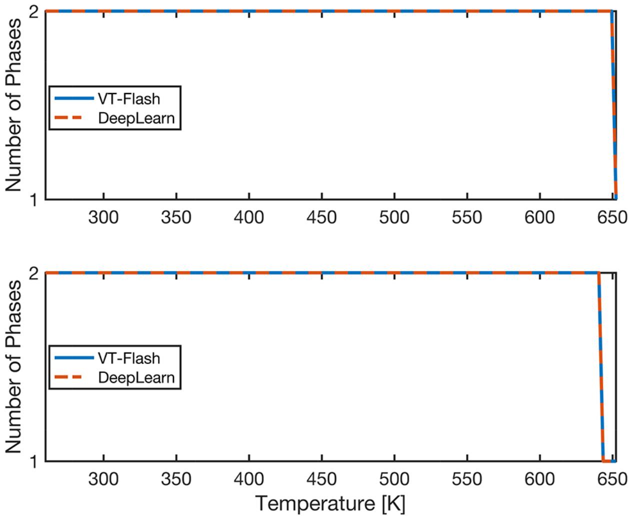


Figure 5. Number of phases, predicted by the deep learning model (dashed line) and iterative flash calculation (solid line), as a function of temperature under the specified overall concentrations of 1988 (top) and 2648 mol/m^3 (bottom), respectively.

approach zero. On the other hand, if the mixture has two phases at the given condition, the phase identification can be specified by comparing the phase density or phase molecular weight to that of the feed composition. The heavier phase is the liquid phase, while the lighter phase is the vapor phase. It can be seen that the predictions of the deep learning model near the phase boundary agree with iterative flash solutions very well in these two cases.

4.3. Prediction Evaluation. The accuracy of our deep learning model shows as the small prediction errors in Table 1. Here we visually compare molar compositions computed by the deep learning and iterative flash approaches to further illustrate this accuracy. Figure 6 displays the mole fractions of C_1, C_{5-7}, and C_{10+} in the vapor phase of the Bakken oil as a function of the temperature at $C = 343$ mol/m^3. The open and solid symbols represent the deep learning estimation and iterative flash solution, respectively. Even though the mole fraction profiles computed by both approaches exhibit some opposite variation tendency, the deep neural network is considered to yield good predictions under certain confidence, especially for the heaviest component, C_{10+}. Figure 7 shows the mole fractions of C_2, C_{5-6}, C_{7+}, and C_{13+} remain constant at the specified temperature interval, implying the EagleFord1 oil is in the single-phase state at the given overall concentration. It can be seen that the deep learning predictions agree very well with the iterative flash solutions. Figure 8 presents the mole fractions of C_1, C_3, nC_4, and CO_2 in the liquid phase of the EagleFord1 oil at another temperature interval when $C = 1810$ mol/m^3. The mole fraction of C_1 predicted by the neural network model deviates from the results of *NVT* flash calculation more distinctly than those of the other light components. Overall, the optimized deep neural network exhibits a good performance on the prediction accuracy. Admittedly, in some cases there are some slight differences.

The effect of data size on the prediction accuracy is validated in Figure 9. With the temperature increasing, the mole fraction of C_1 remains constant at its overall mole fraction of 0.5816, denoted by the black line, which means the EagleFord1 oil is single phase under the given conditions. As expected, the less data we use to train the model, the more deviation takes place in the outcomes of the deep neural network. It is found that

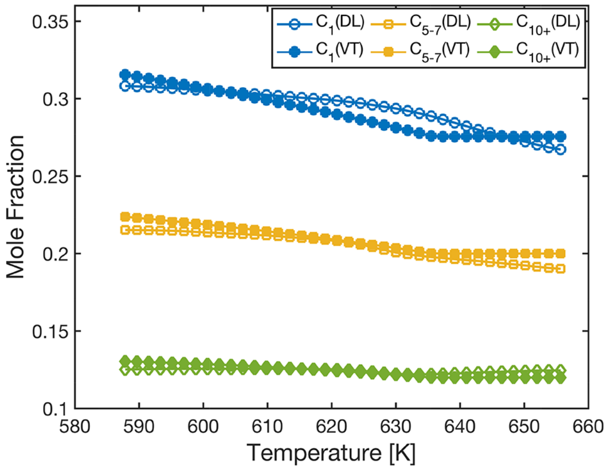


Figure 6. Mole fractions of C_1, C_{5-7}, and C_{10+} in the vapor phase of Bakken oil predicted by the deep learning approach (open symbols) and iterative flash calculation (solid symbols).

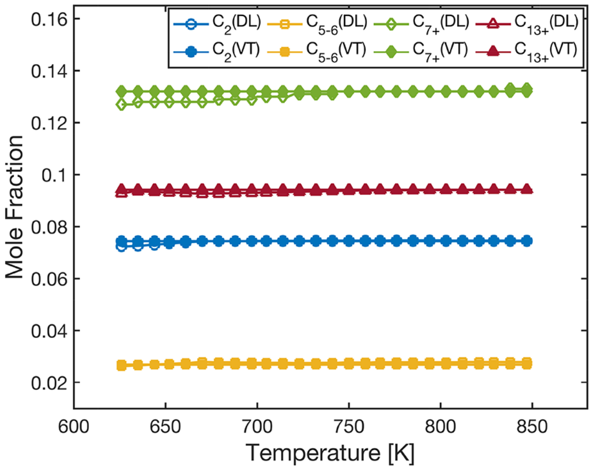


Figure 7. Mole fractions of C_2, C_{5-6}, C_{7+}, and C_{13+} in single-phase EagleFord1 oil. The open and solid symbols represent the deep learning prediction and iterative flash solution, respectively.

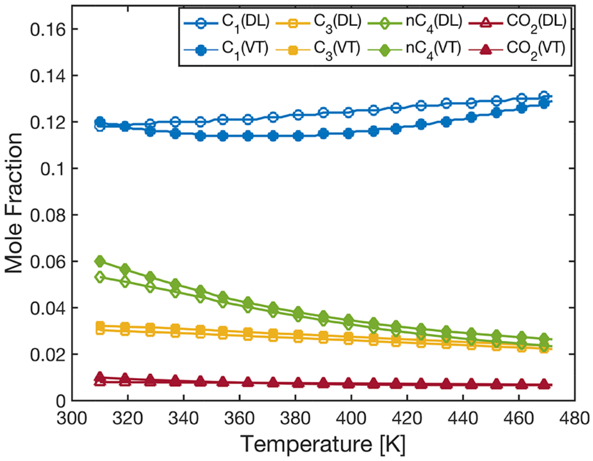


Figure 8. Mole fractions of C_1, C_3, nC_4, and CO_2 in the liquid phase of EagleFord1 oil. The open and solid symbols represent the deep learning prediction and iterative flash solution, respectively.

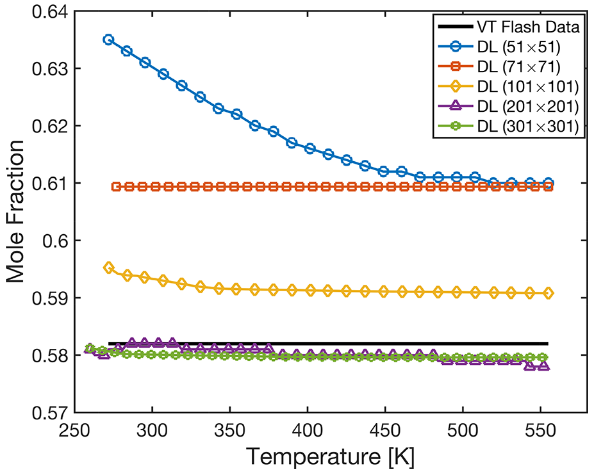


Figure 9. Mole fraction of C_1 predicted by the deep learning model under different data sizes when the EagleFord1 oil is at the single-phase state. The black line represents the solutions of iterative flash calculation.

the deep neural network with the data size of 201 × 201 gives pretty good approximations, which exhibit some slight fluctuations in comparison to the predictions based on the 301 × 301 data size. When the data size increases from 201 × 201 to 301 × 301, the insignificant accuracy improvement, shown in Figure 9, is not worth doubling the computational time as shown in Table 1. Thus, a moderate number of data should be used to train the model in order to balance the accuracy and efficiency of the deep learning model.

5. CONCLUSIONS

In this study, we develop a deep neural network to approximate *NVT* flash calculation so as to speed up phase equilibrium calculations. A dynamic model, comprising the mole and volume evolution equations, is iteratively solved to generate training data for the proposed neural network model. Three real reservoir fluids, including a five-component Bakken oil and two Eagle Ford oils with 8 and 14 components, are investigated, showing that both the dynamic flash model and deep learning model can handle complex fluid mixtures. To increase the prediction accuracy, we reformulate the loss function and employ the dropout technique to reduce the overfitting problem inherent in neural networks. Furthermore, batch normalization is applied to accelerate the training process of the deep learning model. It is noted that the network configuration is designed based on our previous research, which studied various effects on the performance of deep neural networks for phase equilibria problems. Different from the preceding research by machine learning models, our deep learning model gets rid of the constraint of the conventional flash framework where the stability test precedes phase split calculation, so that it can determine the number of phases without additional stability testing. In addition, the proposed network model is able to identify the single-phase state, either single vapor or liquid. A variety of examples are presented to exhibit the accuracy and efficiency of the proposed neural network model. We find that training a deep learning model becomes much more expensive above a threshold data volume, however, with less enhanced accuracy. In our tests, the trained model yields much faster predictions, at most 244 times for the investigated cases than the iterative *NVT* flash, and meanwhile preserve good accuracy. It is worth mentioning that this study paves the path for our future research on acceleration of phase equilibrium calculations. In addition to further improving the accuracy and robustness, the proposed neural network model will be extended to varying composition cases that often take place in EOR processes. Some important effects in unconventional reservoirs, including capillary pressure and nanopore confinement, will also be taken into account in the future work to validate the capability of deep neural networks in confined phase equilibrium problems.

■ ASSOCIATED CONTENT

Ⓢ Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.iecr.9b00527.

Molar compositions and compositional parameters of the Bakken, EagleFord1, and EagleFord2 oils (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: shuyu.sun@kaust.edu.sa.

ORCID

Shuyu Sun: 0000-0002-3078-864X

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors greatly thank the King Abdullah University of Science and Technology (KAUST) for research funding through Grant BAS/1/1351-01-01 and the National Natural Science Foundation of China for support (No. 51874262).

■ REFERENCES

(1) Peng, D.-Y.; Robinson, D. B. A new two-constant equation of state. *Ind. Eng. Chem. Fundam.* **1976**, *15*, 59−64.

(2) Soave, G. Equilibrium constants from a modified Redlich-Kwong equation of state. *Chem. Eng. Sci.* **1972**, *27*, 1197−1203.

(3) Whitson, C. H.; Brulé, M. R. *Phase Behavior*, 1st ed.; Henry, L., Ed.; Doherty Memorial Fund of AIME, Society of Petroleum Engineers: Richardson, TX, 2000.

(4) Michelsen, M. L. Simplified flash calculations for cubic equations of state. *Ind. Eng. Chem. Process Des. Dev.* **1986**, *25*, 184−188.

(5) Jensen, B. H.; Fredenslund, A. A simplified flash procedure for multicomponent mixtures containing hydrocarbons and one non-hydrocarbon using two-parameter cubic equations of state. *Ind. Eng. Chem. Res.* **1987**, *26*, 2129−2134.

(6) Hendriks, E. M. Reduction theorem for phase equilibrium problems. *Ind. Eng. Chem. Res.* **1988**, *27*, 1728−1732.

(7) Hendriks, E. M.; Van Bergen, A. Application of a reduction method to phase equilibria calculations. *Fluid Phase Equilib.* **1992**, *74*, 17−34.

(8) Firoozabadi, A.; Pan, H. Fast and robust algorithm for compositional modeling: Part i-stability analysis testing. *SPE Annual Technical Conference and Exhibition, Dallas, TX, Oct 1−4, 2000*; Society of Petroleum Engineers: 2000.

(9) Pan, H.; Firoozabadi, A. Fast and robust algorithm for compositional modeling: Part ii-two-phase flash computations. *SPE Annual Technical Conference and Exhibition, New Orleans, LA, Sept 30−Oct 3, 2001*; Society of Petroleum Engineers: 2001.

(10) Li, Y.; Johns, R. T. Rapid flash calculations for compositional simulation. *SPE Reservoir Eval. Eng.* **2006**, *9*, 521−529.

(11) Rachford, H., Jr; Rice, J. Procedure for use of electronic digital computers in calculating flash vaporization hydrocarbon equilibrium. *JPT, J. Pet. Technol.* **1952**, *4*, 19−3.

(12) Nichita, D. V.; Graciaa, A. A new reduction method for phase equilibrium calculations. *Fluid Phase Equilib.* **2011**, *302*, 226−233.

(13) Gaganis, V.; Varotsis, N. An improved BIP matrix decomposition method for reduced flash calculations. *Fluid Phase Equilib.* **2013**, *340*, 63−76.

(14) Gorucu, S. E.; Johns, R. T. Comparison of reduced and conventional two-phase flash calculations. *SPE J.* **2015**, *20*, 294−305.

(15) Haugen, K. B.; Beckner, B. A critical comparison of reduced and conventional eos algorithms. *SPE J.* **2013**, *18*, 378−388.

(16) Michelsen, M. L.; Yan, W.; Stenby, E. H. A comparative study of reduced-variables-based flash and conventional flash. *SPE J.* **2013**, *18*, 952−959.

(17) Petitfrere, M.; Nichita, D. V. A comparison of conventional and reduction approaches for phase equilibrium calculations. *Fluid Phase Equilib.* **2015**, *386*, 30−46.

(18) Wang, P.; Stenby, E. H. Non-iterative flash calculation algorithm in compositional reservoir simulation. *Fluid Phase Equilib.* **1994**, *95*, 93−108.

(19) Rasmussen, C. P.; Krejbjerg, K.; Michelsen, M. L.; Bjurstrøm, K. E. Increasing computational speed of flash calculations with applications for compositional, transient simulations. *SPE Annual Technical Conference and Exhibition, Denver, CO, Oct 5−8, 2003*; Society of Petroleum Engineers: 2003.

(20) Voskov, D. V.; Tchelepi, H. A. Tie-simplex based mathematical framework for thermodynamical equilibrium computation of mixtures with an arbitrary number of phases. *Fluid Phase Equilib.* **2009**, *283*, 1−11.

(21) Voskov, D. V.; Tchelepi, H. A. Compositional space parameterization for flow simulation. *SPE Reservoir Simulation Symposium, Houston, TX, Feb 26−28, 2007*; Society of Petroleum Engineers: 2007.

(22) Voskov, D. V.; Tchelepi, H. A. Compositional space parametrization for miscible displacement simulation. *Transp. Porous Media* **2008**, *75*, 111−128.

(23) Belkadi, A.; Yan, W.; Michelsen, M. L.; Stenby, E. H. Comparison of two methods for speeding up flash calculations in compositional simulations. *SPE Reservoir Simulation Symposium, Woodlands, TX, Feb 21−23, 2011*; Society of Petroleum Engineers: 2011.

(24) Yan, B.; Wang, Y.; Nasrabadi, H.; Killough, J. E.; Wu, K. Accelerating Flash Calculation using Compositional Space for Compositional Simulation. *J. Pet. Sci. Eng.* **2017**, *159*, 1000−1008.

(25) Wu, Y.; Kowitz, C.; Sun, S.; Salama, A. Speeding up the flash calculations in two-phase compositional flow simulations−The application of sparse grids. *J. Comput. Phys.* **2015**, *285*, 88−99.

(26) Wu, Y.; Chen, Z. The application of high-dimensional sparse grids in flash calculations: From theory to realisation. *Fluid Phase Equilib.* **2018**, *464*, 22−31.

(27) Li, Y.-K.; Nghiem, L. X. The development of a general phase envelope construction algorithm for reservoir fluid studies. *SPE Annual Technical Conference and Exhibition, New Orleans, LA, Sep 26−29, 1982*; Society of Petroleum Engineers: 1982.

(28) Whitson, C. H.; Michelsen, M. L. The negative flash. *Fluid Phase Equilib.* **1989**, *53*, 51−71.

(29) Li, Y.; Johns, R. T. A rapid and robust method to replace Rachford-Rice in flash calculations. *SPE Reservoir Simulation Symposium, Houston, TX, Feb 26−28, 2007*; Society of Petroleum Engineers: 2007.

(30) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NIPS 2012)*; Pereira, F., Burges, C. J. C., Bottou, L., Weinberger, K. Q., Eds.; Neural Information Processing Systems Foundation, Inc.: 2012; pp 1097−1105.

(31) Korjani, M.; Popa, A.; Grijalva, E.; Cassidy, S.; Ershaghi, I. A new approach to reservoir characterization using deep learning neural networks. *SPE Western Regional Meeting, Anchorage, AK, May 23−26, 2016*; Society of Petroleum Engineers: 2016.

(32) Sarath, P. K.; Turgay, E.; Kemal, A. Structuring an Integrative Approach for Field Development Planning Using Artificial Intelligence and its Application to an Offshore Oilfield. *SPE Annual Technical Conference and Exhibition, Houston, TX, Sept 28−30, 2015*; Society of Petroleum Engineers: 2015.

(33) Araya-Polo, M.; Alpak, F.; Hunter, S.; Hofmann, R.; Saxena, N. Deep Learning-Driven Pore-Scale Simulation For Permeability Estimation. *ECMOR XVI-16th European Conference on the Mathematics of Oil Recovery, Barcelona, Spain, Sept 3−6, 2018*; European Association of Geoscientists and Engineers: 2018.

(34) Gaganis, V.; Varotsis, N. Non-iterative phase stability calculations for process simulation using discriminating functions. *Fluid Phase Equilib.* **2012**, *314*, 69−77.

(35) Burges, C. J. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discovery* **1998**, *2*, 121−167.

(36) Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273−297.

(37) Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, 1st ed.; Cambridge University Press: New York, 2000.

(38) Gaganis, V.; Varotsis, N. Machine learning methods to speed up compositional reservoir simulation. *SPE Improved Oil Recovery Conference, Copenhagen, Denmark, June 4−7, 2012*; Society of Petroleum Engineers: 2012.

(39) Gaganis, V.; Varotsis, N. An integrated approach for rapid phase behavior calculations in compositional modeling. *J. Pet. Sci. Eng.* **2014**, *118*, 74−87.

(40) Gaganis, V. Rapid phase stability calculations in fluid flow simulation using simple discriminating functions. *Comput. Chem. Eng.* **2018**, *108*, 112−127.

(41) Kashinath, A.; Szulczewski, M. L.; Dogru, A. H. A fast algorithm for calculating isothermal phase behavior using machine learning. *Fluid Phase Equilib.* **2018**, *465*, 73−82.

(42) Wang, K.; Luo, J.; Wei, Y.; Wu, K.; Li, J.; Chen, Z. Artificial neural network assisted two-phase flash calculations in isothermal and thermal compositional simulations. *Fluid Phase Equilib.* **2019**, *486*, 59−79.

(43) Li, Y.; Zhang, T.; Sun, S.; Gao, X. Accelerating flash calculation through deep learning methods. *J. Comput. Phys.* **2019**, *394*, 153−165.

(44) Jindrová, T.; Mikyška, J. Fast and robust algorithm for calculation of two-phase equilibria at given volume, temperature, and moles. *Fluid Phase Equilib.* **2013**, *353*, 101−114.

(45) Jindrová, T.; Mikyška, J. General algorithm for multiphase equilibria calculation at given volume, temperature, and moles. *Fluid Phase Equilib.* **2015**, *393*, 7−25.

(46) Nichita, D. V. New unconstrained minimization methods for robust flash calculations at temperature, volume and moles specifications. *Fluid Phase Equilib.* **2018**, *466*, 31−47.

(47) Kou, J.; Sun, S.; Wang, X. An energy stable evolution method for simulating two-phase equilibria of multi-component fluids at constant moles, volume and temperature. *Comput. Geosci.* **2016**, *20*, 283−295.

(48) Cabral, V. F.; Castier, M.; Tavares, F. W. Thermodynamic equilibrium in systems with multiple adsorbed and bulk phases. *Chem. Eng. Sci.* **2005**, *60*, 1773−1782.

(49) Kou, J.; Sun, S. A stable algorithm for calculating phase equilibria with capillarity at specified moles, volume and temperature using a dynamic model. *Fluid Phase Equilib.* **2018**, *456*, 7−24.

(50) Li, Y.; Kou, J.; Sun, S. Thermodynamically Stable Two-Phase Equilibrium Calculation of Hydrocarbon Mixtures with Capillary Pressure. *Ind. Eng. Chem. Res.* **2018**, *57*, 17276−17288.

(51) Luo, S.; Lutkenhaus, J. L.; Nasrabadi, H. Multi-Scale Fluid Phase Behavior Simulation in Shale Reservoirs by a Pore-Size-Dependent Equation of State. *SPE Annual Technical Conference and Exhibition, San Antonio, TX, Oct 9−11,2017*; Society of Petroleum Engineers: 2017.

(52) Travalloni, L.; Castier, M.; Tavares, F. W. Phase equilibrium of fluids confined in porous media from an extended Peng−Robinson equation of state. *Fluid Phase Equilib.* **2014**, *362*, 335−341.

(53) Schnabel, R. B.; Eskow, E. A revised modified Cholesky factorization algorithm. *SIAM J. Optim.* **1999**, *9*, 1135−1148.

(54) Mikyška, J.; Firoozabadi, A. Investigation of mixture stability at given volume, temperature, and number of moles. *Fluid Phase Equilib.* **2012**, *321*, 1−9.

(55) Kou, J.; Sun, S.; Wang, X. Linearly Decoupled Energy-Stable Numerical Methods for Multicomponent Two-Phase Compressible Flow. *SIAM J. Numer. Anal.* **2018**, *56*, 3219−3248.

(56) Zhang, Y.; Lashgari, H. R.; Di, Y.; Sepehrnoori, K. Capillary pressure effect on hydrocarbon phase behavior in unconventional reservoirs. *SPE Low Perm Symposium, Denver, CO, May 5−6, 2016*; Society of Petroleum Engineers: 2016.

(57) Cui, X.; Yang, E.; Song, K.; Huang, J.; Killough, J.; Dong, C.; Liu, Y.; Wang, K. Phase Equilibrium of Hydrocarbons Confined in Nanopores from a Modified Peng-Robinson Equation of State. *SPE Annual Technical Conference and Exhibition, Dallas, TX, Sept 24−26, 2018*; Society of Petroleum Engineers: 2018.

(58) Siripatrachai, N.; Ertekin, T.; Johns, R. T. Compositional simulation of discrete fractures incorporating the effect of capillary pressure on phase behavior. *SPE Improved Oil Recovery Conference, Tulsa, OK, Apr 11−13,2016*; Society of Petroleum Engineers: 2016.

DOI: 10.1021/acs.iecr.9b00527