


## Article

# Evaluation of Machine Learning Applications for the Complex Near-Critical Phase Behavior Modelling of CO<sub>2</sub>–Hydrocarbon Systems

Daulet Magzymov <sup>1,2,\*</sup> , Meruyert Makhatova <sup>1,3</sup>, Zhasulan Dairov <sup>1</sup> and Murat Syzdykov <sup>1</sup><sup>1</sup> Atyrau Oil and Gas University, Baimukhanov St. 45A, Atyrau 060027, Kazakhstan<sup>2</sup> University of Houston, 5000 Gulf Freeway Bldg 9, Houston, TX 77204, USA<sup>3</sup> Colorado School of Mines, 1500 Illinois St, Golden, CO 80401, USA

\* Correspondence: dmagzymov@gmail.com

**Abstract:** The objective of this study was to evaluate the capability of machine learning models to accurately predict complex near-critical phase behavior in CO<sub>2</sub>–hydrocarbon systems, which are crucial for enhanced oil recovery and carbon storage applications. We compared the physical Peng–Robinson equation of state model to machine learning algorithms under varying temperatures, pressures, and composition, including challenging near-critical scenarios. We used a direct neural network model and two hybrid model approaches to capture physical behavior in comprehensive compositional space. While all the models showed great performance during training and validation, the Direct Model exhibited unphysical behavior in compositional space, such as fluctuations in equilibrium constants and tie-line crossing. Hybrid Model 1, integrating a single Rachford–Rice iteration for physical constraints, showed an improved consistency in phase predictions. Hybrid Model 2, utilizing logarithmic transformations to better handle nonlinearities in equilibrium constants, further enhanced the accuracy and provided smoother predictions, particularly in the near-critical region. Overall, the hybrid models demonstrated a superior ability to balance computational efficiency and physical accuracy, closely aligning with the reference of the Peng–Robinson equation of state. This study highlights the importance of incorporating physical constraints into machine learning models for reliable phase behavior predictions, especially under near-critical conditions.

**Keywords:** phase behavior; equation-of-state; hybrid modelling; flash calculation; machine learning



**Citation:** Magzymov, D.; Makhatova, M.; Dairov, Z.; Syzdykov, M. Evaluation of Machine Learning Applications for the Complex Near-Critical Phase Behavior Modelling of CO<sub>2</sub>–Hydrocarbon Systems. *Appl. Sci.* **2024**, *14*, 11140. <https://doi.org/10.3390/app142311140>

Academic Editor: Christos Bouras

Received: 20 October 2024

Revised: 12 November 2024

Accepted: 15 November 2024

Published: 29 November 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The injection of carbon dioxide into oil and gas reservoirs for long-term storage or enhanced oil recovery is an advanced technique to mitigate greenhouse emissions and improve hydrocarbon recovery rates [1,2]. When injected, CO<sub>2</sub> can achieve a certain degree of miscibility with oil, thereby complicating the mass and component exchange of CO<sub>2</sub> and hydrocarbons at the displacement front [3,4]. Therefore, the accurate modeling of CO<sub>2</sub>–oil/gas phase behavior is critical to assessing the success of CO<sub>2</sub> injections. Compositional modeling is needed to capture the CO<sub>2</sub>–hydrocarbon phase equilibria and the flow of multiple phases.

Compositional modeling involves an analysis of the phase behavior of fluids based on their pressure, temperature, and overall composition. This process includes determining the phase stability and conducting flash calculations to establish the number and composition of coexisting phases at equilibrium [5]. These calculations are necessary for each discretization block and time step [6] to provide the physical properties for the flow equations. Depending on the space and time discretization, some simulation runs may require millions of phase equilibrium calculations, which can significantly impact the run time [7] and can consume up to 50% of CPU run time [8], due to the iterative nature of phase equilibrium calculations.

The solution to a flash problem must satisfy mass and component balances and ensure that each component has the same fugacity in the coexisting equilibrium phases, in order to minimize the Gibbs energy of the system [9]. The Rachford–Rice equation is a well-established method for flash calculations [10], which incorporates the equilibrium ratio or K-value. This value is determined through an iterative process using an equation of state (EoS), with the Peng–Robinson (PR) [11] and the Soave–Redlich–Kwong (SRK) [12] cubic EoSs being the most commonly used in petroleum engineering.

Many researchers have developed alternatives to conventional phase equilibrium calculations to speed up phase computations. The reduction method was first attempted by Michelsen in 1986 [13], which involved reducing the calculation to three nonlinear equations and setting the binary interaction coefficients to zero. Later, reduction methods were developed by Hendriks and Van Bergen, Firoozabadi et al., Okuno et al., and Petitfrere and Nichita [14,15]. Another approach developed in an attempt to reduce computational time involved utilizing precalculated tie-lines and was introduced by Voskov and Tchelepi [16] and Belkadi et al. [17].

Recently, machine learning techniques have been used to address computational time challenges in phase equilibrium calculations. Gaganis and Varotsis developed a non-iterative solution for phase stability using support vector classifiers trained on EoS-generated data points [8]. They claimed that this solution was as accurate as the thermodynamic model itself. In a subsequent study, Gaganis and Varotsis [18] expanded the model to include a split phase by applying an Artificial Neural Network (ANN) to provide the equilibrium coefficients. Kashinath et al. [19] demonstrated that their algorithm reduced the phase equilibrium calculations by using a relevance vector machine (RVM) and ANN, resulting in a speeding up of up to 90% computational time. Their work was built on Gaganis and Varotsis [20] and comprised three stages: supercritical phase determination, phase stability, and split phase problems tailored to CO<sub>2</sub> flooding simulations, with two classifiers for the first two stages and an ANN for the splitting step. In another study, Wang et al. (2019) employed an ANN instead of the SVM method to perform both the phase stability tests and phase splitting. An ANN for a stability test is used to predict saturation pressure to identify stable phases, and another ANN provides initial mole fractions and equilibrium ratios for flash calculations.

Zhang et al. [21] introduced a self-adaptive deep learning algorithm that can predict the total number of phases and their molar composition. The algorithm demonstrates adaptability to changes in input dimensions. In a similar vein, Li et al. in 2019 presented a deep learning model that approximates flash calculations at a given mole (N), volume (V), and temperature (T) to address both phase stability and phase splits [22].

Sheth et al. [23] presented a machine learning workflow that combines a classifier and an ANN regressor to label single phases by predicting the critical temperature. This resulted in a 10-fold reduction in iterations applied to a CO<sub>2</sub> injection example. Additionally, machine learning algorithms were utilized to accelerate flash calculations, considering capillary pressure, with the aim of extending the practical application to shale reservoirs [24,25].

Recently, machine learning techniques have been integrated with fundamental physical laws, a concept first introduced by Raissi and Karniadakis [26] and further developed by numerous researches [27,28]. Raissi et al. [29] and Cai et al. [30] introduced physics-informed neural networks (PINNs), which incorporate physical principles and constraints into the loss function, allowing for the consideration of initial and boundary conditions. Fraces and Tchelepi [31] and Magzymov et al. [32] have further applied physics-based models to study fluid flow in porous media, specifically to solve the Buckley–Leverett equation. In alignment with the approach of Raissi et al. [29], Ihunde et al. [33] integrated the underlying physics and physical constraints into ANNs for phase behavior calculations. In this case, the loss function was expanded to include mass and component balance constraints.

Other recent studies have expanded their focus to integrating and employing machine learning (ML) for phase equilibrium in compositional simulations. Specifically, the

use of an ML workflow for identifying single phases in CO<sub>2</sub> injection simulations has notably decreased the time required for phase stability calculations, resulting in accelerated computations [34]. Furthermore, Wang et al. [35] and Chen et al. [36] examined the application of ANN-based phase stability and phase split calculations in numerical examples of gas injection and gas condensate reservoir cases, highlighting an 80% reduction in the computational costs for the stability tests.

The objective of this paper was to assess the accuracy and physical reliability of machine learning approaches for flash calculations and phase behavior predictions, particularly in complex near-critical conditions for CO<sub>2</sub>–hydrocarbon mixtures. This involved evaluating both a direct neural network model and two hybrid models against the established Peng–Robinson equation of state under various conditions. First, we generated a synthetic dataset to train the machine learning algorithms. Second, we discussed the structures and features of the direct and hybrid machine learning algorithms used in this paper. Third, we demonstrated the performance of capturing physical phenomena in compositional space and compared the model predictions with the physical model. Last, we discussed the implications and conclusions of this study.

## 2. Materials and Methods

### 2.1. Data Generation and Sampling

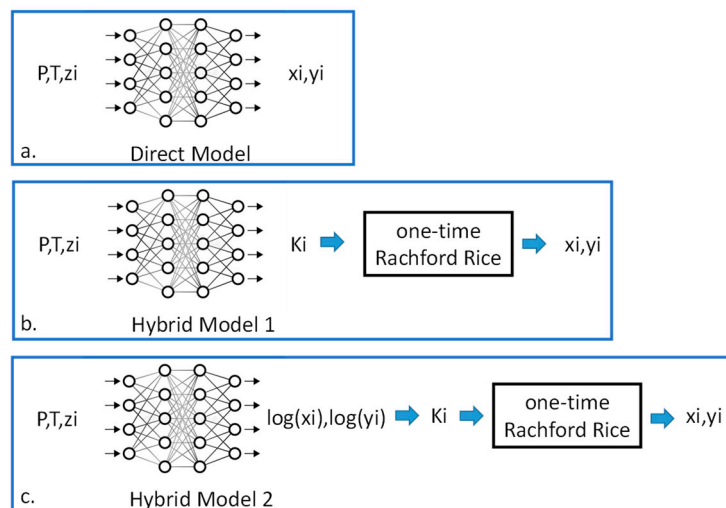
A total of 4000 data points were generated from a mixture of carbon dioxide (CO<sub>2</sub>), n-butane (C4), and n-decane (C10) to model the phase behavior under varying thermodynamic conditions. We split the dataset into 80% training, 10% validation, and 10% testing. The dataset covered pressures from 80 atm to 40 atm and temperatures ranging from 57 °C to 67 °C. The overall composition of the CO<sub>2</sub>–C4–C10 system was sampled across this range, including the regions close to the critical point. This ensured a comprehensive representation of both typical and near-critical phase behavior scenarios, which are notoriously difficult to model accurately due to the sharp changes in physical properties near the critical point. The inputs to the models consisted of five parameters: temperature, pressure, and the mole fractions of the three components (CO<sub>2</sub>, n-butane, and n-decane). We used a neural network architecture of 2 hidden layers and 15 neurons in each hidden layer. Table 1 shows the inputs that were used to generate the dataset. Appendix A shows the sample of data shown as input columns and output columns.

**Table 1.** Physical model inputs used to generate ML input dataset.

	Pc (atm)	Tc (K)	$\omega$
carbon dioxide (CO <sub>2</sub> )	72.8	304.2	0.225
n-butane (C4)	37.5	425.2	0.193
n-decane (C10)	20.8	617.6	0.49
interaction coefficients	carbon dioxide (CO <sub>2</sub> )	n-butane (C4)	n-decane (C10)
carbon dioxide (CO <sub>2</sub> )	0	0.115	0.115
n-butane (C4)	0.115	0	0.012
n-decane (C10)	0.115	0.012	0

### 2.2. Model Scenarios

We tested three machine learning-based models against a physical model using the generated dataset. Each model’s ability to capture both standard and near-critical point phase behavior was evaluated, as near-critical conditions introduce particular challenges for modeling (see Figure 1). P is pressure, T is temperature,  $z_i$  is overall composition,  $x_i$  is liquid fraction,  $y_i$  is vapor fraction, and  $K_i$  is equilibrium ratio.



**Figure 1.** Model scenarios: (a) Direct Model, (b) Hybrid Model 1, and (c) Hybrid Model 2.

1. *Direct Model*: In this scenario, a neural network was trained to directly predict the liquid phase composition  $x_i$  and vapor phase composition  $y_i$  for each component. The neural network was trained on the full dataset, with the inputs being temperature, pressure, and overall composition. The outputs were the mole fractions of the components in the liquid and vapor phases at equilibrium. Special attention was given to the model's performance near the critical point, where rapid changes in composition occur.

2. *Hybrid Model 1—neural network for  $K_i$  and Rachford–Rice*: The first hybrid approach combined machine learning with physical modeling. The neural network predicted the equilibrium coefficients  $K_i = y_i/x_i$  for each component, where  $y_i$  and  $x_i$  represent the vapor and liquid mole fractions, respectively. Once the  $K_i$  values were predicted, the one single Rachford–Rice iteration was used to compute the phase compositions and generate tie-lines in the compositional space. This method retained physical consistency while benefiting from the speed of machine learning. The performance around the critical point was specifically monitored due to the high sensitivity of the  $K_i$  values in these conditions.

3. *Hybrid Model 2—neural network for logarithmic compositions*: The second hybrid approach added another layer of transformation to the model outputs. Here, the neural network predicted  $\log(x_i)$  and  $\log(y_i)$ , the logarithmic mole fractions of the components in the liquid and vapor phases. These outputs were exponentiated to recover  $x_i$  and  $y_i$ , which were then used to compute the equilibrium coefficient  $K_i$ . As in Hybrid Model 1, the Rachford–Rice iteration was applied to ensure consistency and generate accurate tie-lines in the compositional space. This method was also tested rigorously near the critical point, where minor differences in the predicted compositions can lead to significant errors.

### 2.2.1. Near-Critical Point Behavior

Given the challenges posed by the near-critical point conditions, special care was taken to evaluate how well each machine learning model captured the phase behavior in this region. Near the critical point, small changes in pressure or temperature can cause dramatic shifts in phase behavior, making this a particularly challenging test for machine learning models. The models' ability to accurately predict the onset of the two-phase regions and the correct composition of each phase was closely examined.

### 2.2.2. Phase Behavior Modeling and Validation

The performance of the machine learning models was compared against a physical model using the Peng–Robinson equation of state for the phase equilibrium calculations. The predicted phase compositions and tie-lines were validated against the results from the physical model. The evaluation criteria focused on accuracy in standard and near-critical conditions.

### 2.2.3. Flash Calculations and the Rachford–Rice Equation

To determine the phase compositions at equilibrium, flash calculations were performed. The Rachford–Rice equation was used to compute the vapor fraction  $n^V$  by solving the following:

$$f(n^V) = \sum_{i=1}^N (y_i - x_i) = \sum_{i=1}^N \frac{z_i(K_i - 1)}{1 + (K_i - 1)n^V} = 0 \quad (1)$$

where  $z_i$  is the overall mole fraction of the component,  $K_i$  is the equilibrium ratio,  $n^V$  is the vapor fraction,  $x_i$  is the liquid composition, and  $y_i$  is the vapor composition. The first guess of  $n^V$  is set to the mean of  $(\frac{1}{1-K_{max}}, \frac{1}{1-K_{min}})$ . Furthermore, we solved the K-values and  $n^V$  iteratively using both the Newton–Raphson method and successive substitution, starting with updating  $n^V$  at the fixed K-value followed by further updating the K-values. Function  $f(n^V)$  decreases monotonically with asymptotes  $1/(1 - K_i)$ . Final convergence was achieved when the K-value error was in the range of the accepted tolerance.

### 2.2.4. Machine Learning Implementation

As outlined in Figure 1, we used three different approaches for the implementation. The Direct Model used machine learning inputs and outputs as follows:

$$\text{Neural Network } (P, T, z_i) = [x_i, y_i] \quad (2)$$

For Hybrid Model 1, we generated  $K_i$  from the neural network and then used one Rachford–Rice iteration. Such an approach ensures that mass balance is honored for  $z_i$ ,  $x_i$ ,  $y_i$ , and the vapor–liquid fractions at all times.

$$\text{Neural Network } (P, T, z_i) = [K_i] \quad (3)$$

$$\text{Rachford – Rice } (K_i) = [x_i, y_i] \quad (4)$$

For Hybrid Model 2, we used the neural network to generate  $\ln(x_i)$  and  $\ln(y_i)$ . We then calculated  $K_i$  and used one Rachford–Rice iteration. This way, we address the highly nonlinear nature of the  $K_i$  variables, large and small values of  $x_i$  and  $y_i$ , and mass balance consistency between  $z_i$ ,  $x_i$ ,  $y_i$ , and the vapor–liquid fractions.

$$\text{Neural Network } (P, T, z_i) = [\ln x_{i_{nn}}, \ln y_{i_{nn}}] \quad (5)$$

$$K_i = \frac{e^{\ln y_{i_{nn}}}}{e^{\ln x_{i_{nn}}}} \quad (6)$$

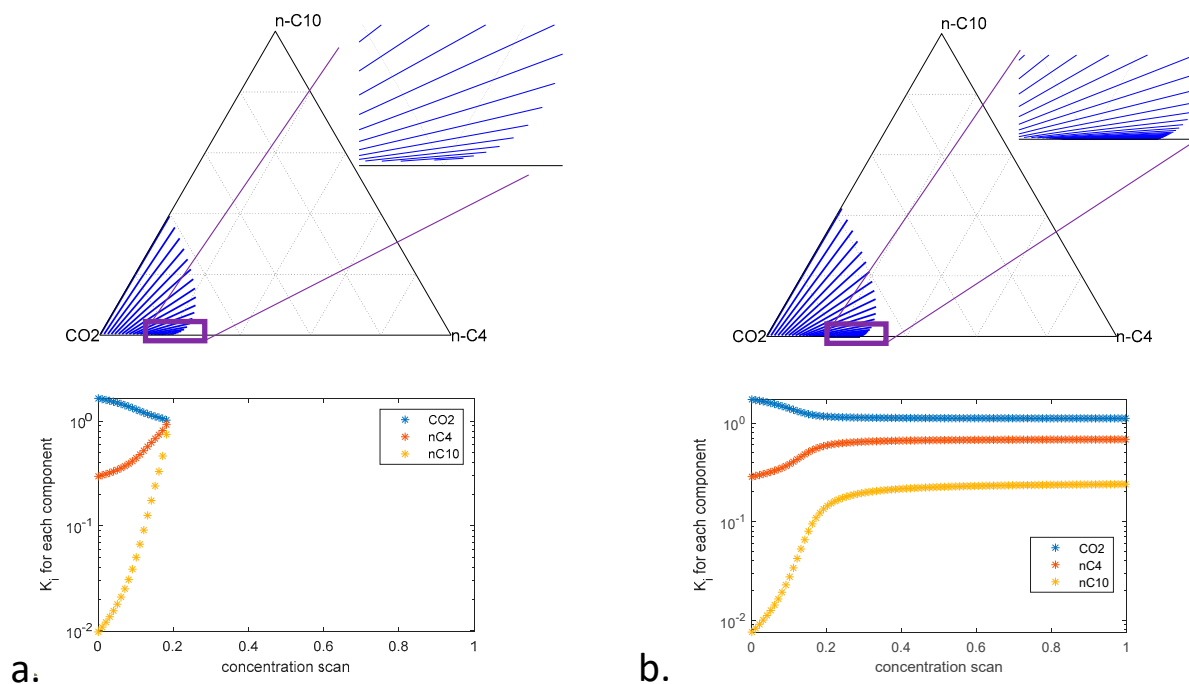
$$\text{Rachford – Rice } (K_i) = [x_i, y_i] \quad (7)$$

The next section demonstrates the results for the Direct, Hybrid 1, and Hybrid 2 Models. Each performance is compared to the ground reference from the physical model of the Peng–Robinson equation of state. We compare the physical model and machine learning approaches in both the near-critical region, as well as far away from critical conditions. The results compare the entire compositional space in ternary diagrams in contrast to the typically checked individual flash calculations that do not capture the holistic performance of machine learning models.

## 3. Results and Discussions

The Peng–Robinson equation of state was applied to the CO<sub>2</sub>, n-butane, and n-decane system to calculate the phase behavior. The model accurately captures the equilibrium between the liquid and vapor phases, particularly under typical reservoir conditions and near-critical points. The generated tie-lines and close-up view depict the distribution of phases in the compositional space. As expected, the Peng–Robinson EoS performs well in identifying two-phase regions, with distinct tie-line slopes representing phase equilibrium.

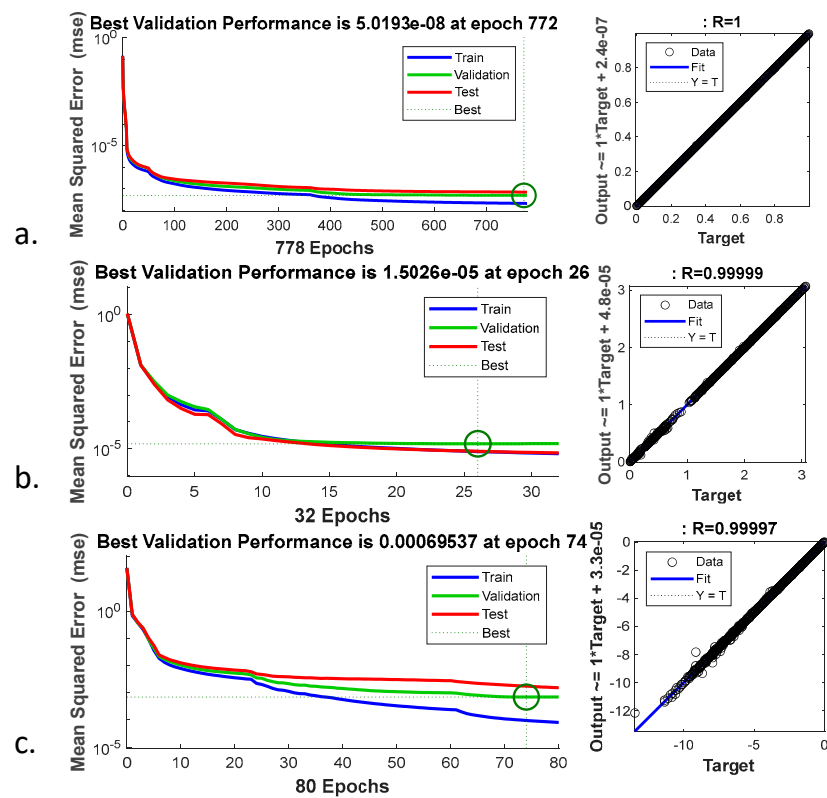
Figure 2 represents the phase behavior and equilibrium constants calculated using the Peng–Robinson equation of state for a CO<sub>2</sub>, n-butane, and n-decane mixture under two conditions: near-critical and regular. In Figure 2a, the near-critical conditions at 80 atm and 60 °C show the tie-lines becoming shorter and denser, indicating the difficulty of distinguishing between the liquid and vapor phases as the system approaches the critical point. The equilibrium constant  $K_i$  approaches one, indicating the critical point. In Figure 2b, under regular conditions at 75 atm and 60 °C, the tie-lines are longer and more evenly spaced, indicating well-defined phase behavior. The equilibrium constant  $K_i$  displays smoother transitions across the concentration scan, with CO<sub>2</sub> showing higher  $K_i$  values at lower concentrations, while n-decane's heavier nature results in a lower  $K_i$ .



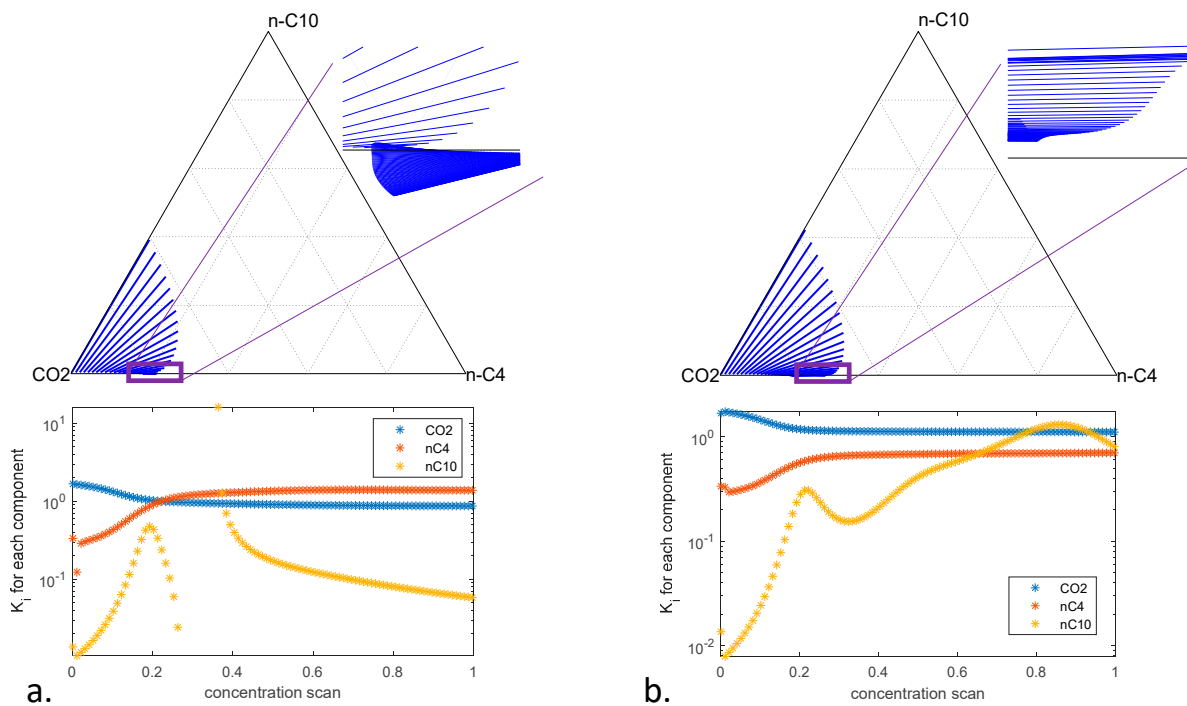
**Figure 2.** Reference case generated using Peng–Robinson equation of state: (a) near-critical conditions of 80 atm and 60 °C; and (b) regular conditions of 75 atm and 60 °C.

Figure 3 shows the training, validation, and testing performance of the neural network models using 4000 flash calculation samples for the Direct Model, Hybrid Model 1, and Hybrid Model 2. All the models demonstrate an excellent fit, with near-perfect matching between the predicted and actual values. However, the Direct Model (a) requires the highest number of training iterations. While the overall accuracy of all the models is impressive, a more detailed analysis of how each model performs in compositional space will be presented in the following section.

The results shown in Figure 4 reveal that while the direct model produced a nearly perfect match during the training, validation, and testing, there were notable issues when examining the model's behavior in compositional space. Specifically, for decane, we observed unphysical fluctuations in the equilibrium constants ( $K_i$ ) under both near-critical and regular conditions. These fluctuations suggest that the model struggles to maintain consistent physical behavior, particularly for heavier components like decane, likely because the equilibrated  $y_i$  values are very small and challenging to capture by the Direct Model. Additionally, a closer examination of the tie-lines shows that they cross each other, which is another indication of unphysical behavior. This crossing of tie-lines should not occur in realistic phase equilibrium scenarios and highlights the limitations of the Direct Model, despite its high accuracy in the training process. These inconsistencies underscore the need for hybrid approaches to ensure both computational efficiency and physical validity.

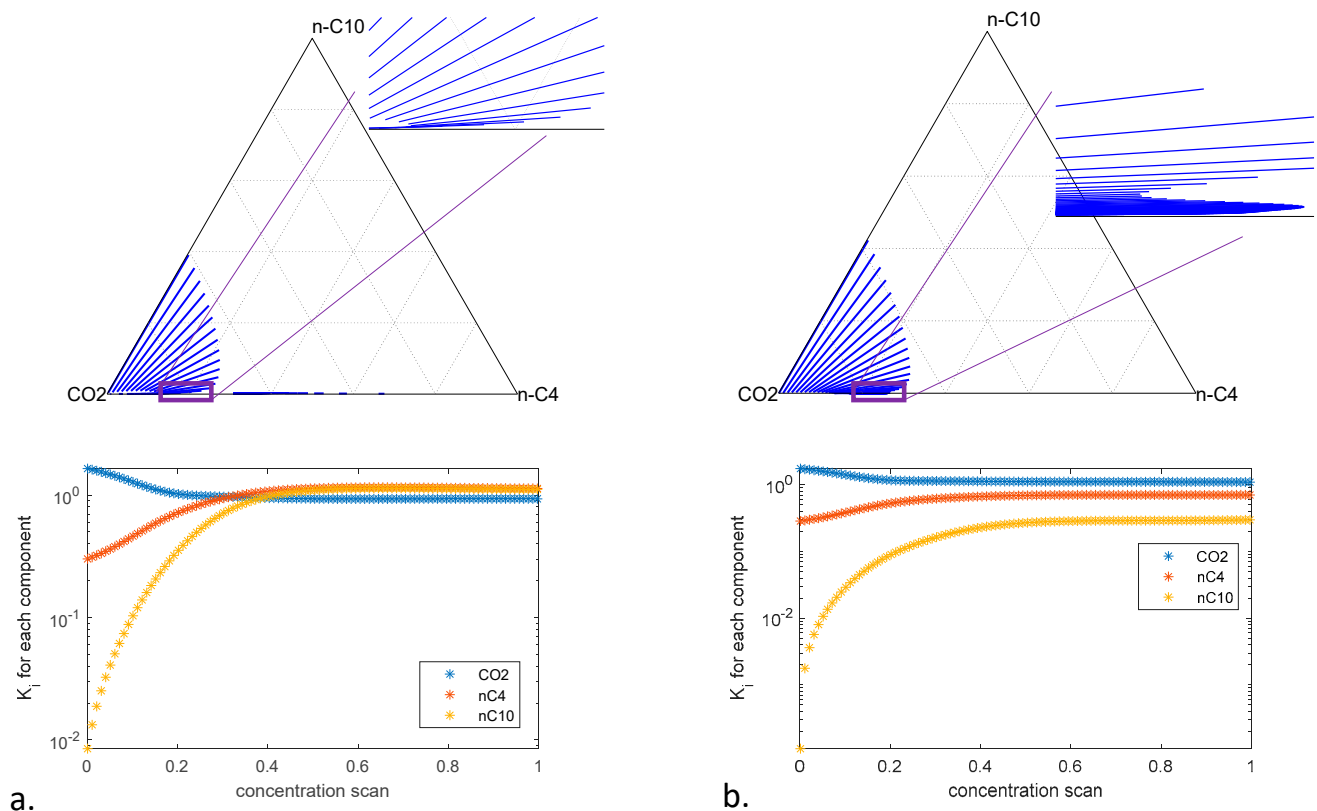


**Figure 3.** Machine learning (neural network) training, validation, and testing using 4000 flash calculation samples (pressure [80 atm to 40 atm], temperature [57 °C to 67 °C], and overall composition of CO<sub>2</sub>-C<sub>4</sub>-C<sub>10</sub> [0 to 1]) for (a) Direct Model, (b) Hybrid Model 1, and (c) Hybrid Model 2.



**Figure 4.** Ternary diagram generated using the direct machine learning model ( $x_i, y_i$ ): (a) near-critical conditions of 80 atm and 60 °C; and (b) regular conditions of 75 atm and 60 °C.

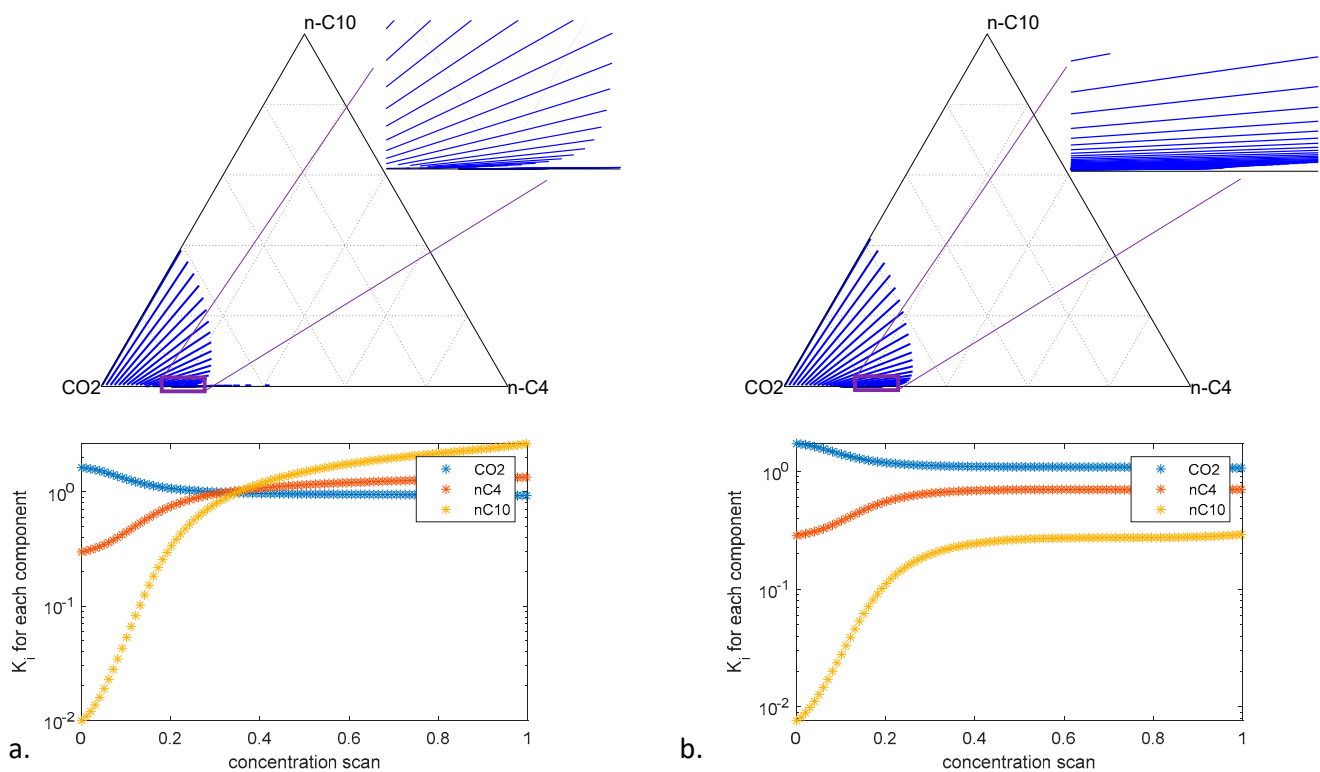
In Figure 5, we observe the phase behavior generated by Hybrid Model 1, where the neural network predicts the  $K_i$  values followed by a single Rachford–Rice iteration to calculate the equilibrium compositions ( $x_i$  and  $y_i$ ). The results show smooth and consistent  $K_i$  values across both near-critical (a) and regular (b) conditions, particularly for the heavier component, decane (nC10), which previously displayed unphysical fluctuations in the Direct Model. The tie-lines in both conditions are physically consistent and do not cross each other, confirming that the mass balance and phase behavior are properly maintained. This improved accuracy is due to the integration of the Rachford–Rice equation, which honors the underlying physical constraints of the system. By using a single Rachford–Rice iteration after the neural network generates the  $K_i$  values for the given pressure, temperature, and composition, Hybrid Model 1 successfully integrates machine learning with traditional thermodynamic principles, producing more reliable results in the compositional space.



**Figure 5.** Ternary diagram generated using Hybrid Model 1— $K_i \rightarrow RR \rightarrow x_i, y_i$ : (a) near-critical conditions of 80 atm and 60 °C; and (b) regular conditions of 75 atm and 60 °C.

Figure 6 presents the results from Hybrid Model 2, where the neural network predicts the logarithmic mole fractions  $\log(x_i)$  and  $\log(y_i)$  and then calculates  $K_i$ . As with Hybrid Model 1, a single Rachford–Rice iteration is then used to ensure that the phase compositions honor mass balance and phase equilibrium. The overall behavior of Hybrid Model 2 is similar to that of Hybrid Model 1, with smooth and consistent tie-lines and no unphysical behavior such as tie-line crossing. However, by using  $\log(x_i)$  and  $\log(y_i)$  in the process, in Hybrid Model 2 we are attempting to linearize the highly nonlinear  $K_i$  variable, potentially improving the accuracy of the predictions, particularly for challenging near-critical conditions. The logarithmic transformation helps the model handle the variability of  $K_i$  more effectively, ensuring that the final phase compositions are more physically consistent after the Rachford–Rice correction. This approach integrates machine learning with physical principles while addressing the inherent nonlinearity of the phase equilibrium variables.

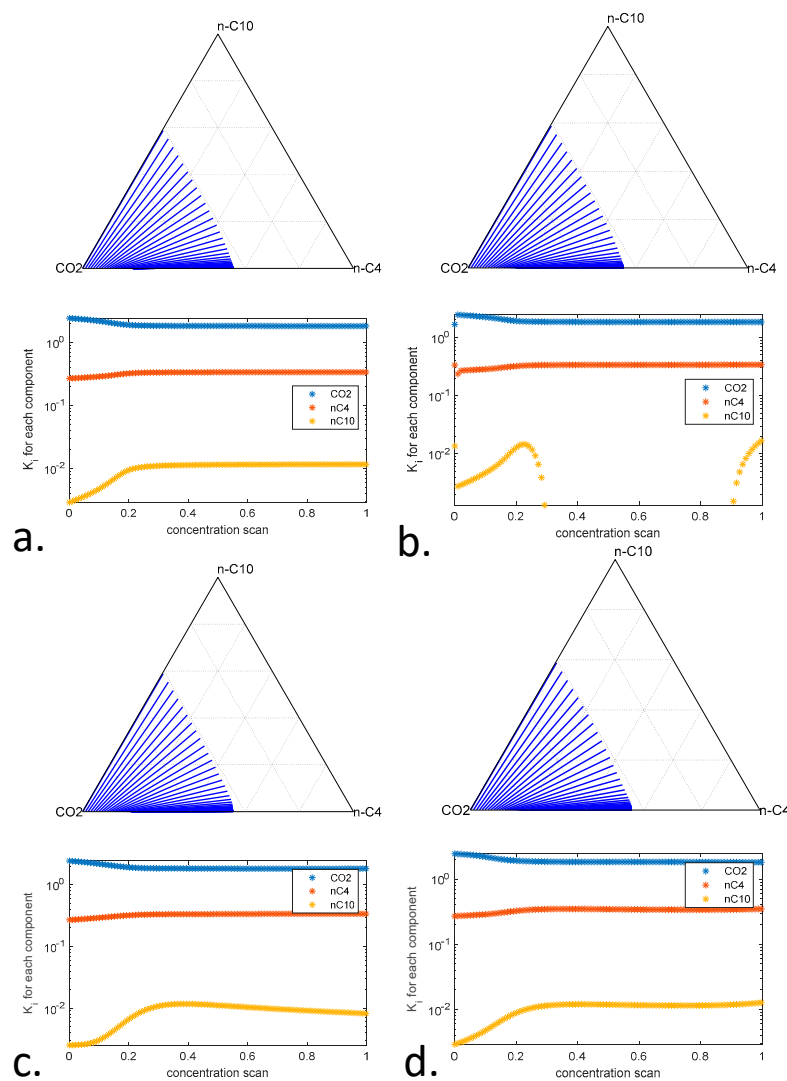




**Figure 6.** Ternary diagram generated using Hybrid Model 2— $\log(x_i), \log(y_i) \rightarrow K_i \rightarrow RR \rightarrow x_i, y_i$ : (a) near-critical conditions of 80 atm and 60 °C; and (b) regular conditions 75 atm and 60 °C.

In Figure 7, we observe the phase behavior for the system under far-from-critical conditions (50 atm and 60 °C). The Peng–Robinson EoS (Figure 7a) serves as the reference case, providing physically consistent phase predictions. The Direct Model (Figure 7b), while still achieving a high accuracy during training, shows some unphysical fluctuations in the equilibrium constants, particularly for heavier components like decane, which highlights the limitations of this approach in capturing true physical behavior. In contrast, both Hybrid Model 1 (Figure 7c) and Hybrid Model 2 (Figure 7d) produce more consistent and physically accurate predictions. However, it is noticeable that Hybrid Model 2 offers even smoother predictions that are more closely aligned with the Peng–Robinson reference case. The use of logarithmic transformations in Hybrid Model 2 improves its ability to handle the nonlinear nature of equilibrium constants, providing an advantage in terms of both physical consistency and prediction smoothness, even in less challenging, far-from-critical conditions.

This study evaluated the overall performance of machine learning algorithms to capture physical behavior in a controlled environment where we directly control the synthetic data, thus eliminating any potential quality data uncertainty and consistency. The implications of this study can be applicable for broader applications that heavily rely on phase behavior modelling, such as compositional reservoir simulations. The models that are discussed in this paper can be used for phase behavior modelling with a single machine learning run once tuned and trained once at the beginning of the simulation, instead of the traditional iteration-based approach for reservoir simulations. Moreover, CO<sub>2</sub> storage project design and prediction need simulations that cover decades and centuries from case studies with fast and reliable modelling, where traditional full field simulations can be challenging due to their high computational intensity. Thus, machine learning-based hybrid models can be a possible solution; however, one needs to ensure the underlying physics is honored during implementation.



**Figure 7.** Ternary diagrams for far-from-critical conditions of 50 atm and 60 °C: (a) Peng–Robinson EoS; (b) Direct Model; (c) Hybrid Model 1; and (d) Hybrid Model 2.

We have studied a well-defined model with known fluid characterizations based on limited pure components. However, more complex systems such as real crude hydrocarbon systems should also be tested in future studies, including the typically unknown plus fractions. Another limitation of this study is that the dataset is based on a deterministic set of input values that do not have uncertainties that are typical in experimental data processing. Thus, the methods should also be tested and tuned with real fluid experimental data that include experimental uncertainty and error bars in future studies.

#### 4. Conclusions

The objective of this study was to evaluate a machine learning application for the complex near-critical phase behavior modelling of CO<sub>2</sub>–hydrocarbon systems. We compared the physical Peng–Robinson equation of state model performance and machine learning algorithms under varying temperatures, pressures, and composition, including challenging near-critical scenarios. We used a direct neural network model and two hybrid model approaches to capture the physical behavior in a comprehensive compositional space for a CO<sub>2</sub>, n-butane, and n-decane system. The key conclusions are as follows:

1. All machine learning algorithms showed great performance during training and validation. However, when the machine learning algorithms were used directly

for comprehensive testing in compositional space, it is evident that direct model implementation was not producing physical features, such as fluctuating K-values and inaccuracies near critical point compositions.

2. Hybrid models: both hybrid models produced more physically constrained results, with Hybrid Model 2 providing the smoothest and most reliable predictions.
3. Near-critical conditions: all models faced challenges in the near-critical region, but Hybrid Model 2 was closest to the physical model in capturing accurate phase behavior.
4. Far-from-critical conditions: under far-from-critical conditions, the overall behavior of all three models produced reasonable results.

In summary, while neural network training is highly accurate, this study demonstrates that physical constraints, such as those incorporated into the hybrid models, are crucial for ensuring reliable phase behavior predictions, especially at near-critical points. Future work could explore expanding the hybrid modeling approach to more complex systems such as real hydrocarbon systems with plus fraction pseudocomponents. Moreover, future studies should address the hybrid model applicability for experimental data, accessing uncertainties and error bars.

**Author Contributions:** Conceptualization, D.M.; Methodology, D.M.; Software, M.M.; Validation, D.M.; Formal analysis, D.M. and M.M.; Investigation, M.M.; Resources, Z.D.; Data curation, M.M.; Writing—original draft, D.M.; Writing—review & editing, M.M.; Visualization, D.M.; Supervision, D.M. and M.S.; Project administration, Z.D. and M.S.; Funding acquisition, M.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research has been funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (grant no. AP13068661).

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Sample Input/Output Data

Table A1. Input.

Input						
dataID	P(atm)	T(K)	z_CO <sub>2</sub>	z_nC <sub>4</sub>	z_nC <sub>10</sub>	
1	60.5	336.1	0.64	0.19	0.16	
2	52.0	332.6	0.50	0.31	0.19	
3	47.2	332.7	0.01	0.27	0.71	
4	63.9	334.8	0.25	0.28	0.47	
5	67.4	335.9	0.94	0.04	0.02	
6	73.8	332.0	0.17	0.82	0.01	
7	49.5	336.9	0.27	0.59	0.14	
8	42.1	337.7	0.35	0.08	0.57	
...	...	...	...	...	...	...
3999	47.1	333.4	0.38	0.30	0.32	
4000	79.4	338.6	0.30	0.21	0.49	

**Table A2.** Direct model output.

Direct Model Output						
Data ID	x_CO <sub>2</sub>	x_nC4	x_nC10	y_CO <sub>2</sub>	y_nC4	y_nC10
1	$4.74 \times 10^{-1}$	$2.62 \times 10^{-1}$	$2.64 \times 10^{-1}$	$9.19 \times 10^{-1}$	$7.92 \times 10^{-2}$	$1.89 \times 10^{-3}$
2	$4.29 \times 10^{-1}$	$3.47 \times 10^{-1}$	$2.24 \times 10^{-1}$	$8.99 \times 10^{-1}$	$1.00 \times 10^{-1}$	$1.17 \times 10^{-3}$
3	$3.86 \times 10^{-1}$	$1.84 \times 10^{-1}$	$4.30 \times 10^{-1}$	$9.47 \times 10^{-1}$	$5.12 \times 10^{-2}$	$1.39 \times 10^{-3}$
4	$5.02 \times 10^{-1}$	$2.00 \times 10^{-1}$	$2.98 \times 10^{-1}$	$9.39 \times 10^{-1}$	$5.91 \times 10^{-2}$	$2.13 \times 10^{-3}$
5	$5.16 \times 10^{-1}$	$1.14 \times 10^{-1}$	$3.70 \times 10^{-1}$	$9.64 \times 10^{-1}$	$3.35 \times 10^{-2}$	$2.59 \times 10^{-3}$
6	$7.21 \times 10^{-1}$	$2.76 \times 10^{-1}$	$2.63 \times 10^{-3}$	$8.34 \times 10^{-1}$	$1.66 \times 10^{-1}$	$3.94 \times 10^{-4}$
7	$3.98 \times 10^{-1}$	$4.94 \times 10^{-1}$	$1.08 \times 10^{-1}$	$8.40 \times 10^{-1}$	$1.60 \times 10^{-1}$	$8.24 \times 10^{-4}$
8	$3.35 \times 10^{-1}$	$7.82 \times 10^{-2}$	$5.87 \times 10^{-1}$	$9.75 \times 10^{-1}$	$2.38 \times 10^{-2}$	$1.72 \times 10^{-3}$
...	...	...	...	...	...	...
3999	$3.83 \times 10^{-1}$	$3.01 \times 10^{-1}$	$3.16 \times 10^{-1}$	$9.12 \times 10^{-1}$	$8.65 \times 10^{-2}$	$1.23 \times 10^{-3}$
4000	$5.85 \times 10^{-1}$	$1.37 \times 10^{-1}$	$2.78 \times 10^{-1}$	$9.50 \times 10^{-1}$	$4.60 \times 10^{-2}$	$4.04 \times 10^{-3}$

**Table A3.** Hybrid model 2 output.

Hybrid Model 2 Output						
Data ID	ln(x_CO <sub>2</sub> )	ln(x_nC4)	ln(x_nC10)	ln(y_CO <sub>2</sub> )	ln(y_nC4)	ln(y_nC10)
1	$4.74 \times 10^{-1}$	$2.62 \times 10^{-1}$	$2.64 \times 10^{-1}$	$9.19 \times 10^{-1}$	$7.92 \times 10^{-2}$	$1.89 \times 10^{-3}$
2	$4.29 \times 10^{-1}$	$3.47 \times 10^{-1}$	$2.24 \times 10^{-1}$	$8.99 \times 10^{-1}$	$1.00 \times 10^{-1}$	$1.17 \times 10^{-3}$
3	$3.86 \times 10^{-1}$	$1.84 \times 10^{-1}$	$4.30 \times 10^{-1}$	$9.47 \times 10^{-1}$	$5.12 \times 10^{-2}$	$1.39 \times 10^{-3}$
4	$5.02 \times 10^{-1}$	$2.00 \times 10^{-1}$	$2.98 \times 10^{-1}$	$9.39 \times 10^{-1}$	$5.91 \times 10^{-2}$	$2.13 \times 10^{-3}$
5	$5.16 \times 10^{-1}$	$1.14 \times 10^{-1}$	$3.70 \times 10^{-1}$	$9.64 \times 10^{-1}$	$3.35 \times 10^{-2}$	$2.59 \times 10^{-3}$
6	$7.21 \times 10^{-1}$	$2.76 \times 10^{-1}$	$2.63 \times 10^{-3}$	$8.34 \times 10^{-1}$	$1.66 \times 10^{-1}$	$3.94 \times 10^{-4}$
7	$3.98 \times 10^{-1}$	$4.94 \times 10^{-1}$	$1.08 \times 10^{-1}$	$8.40 \times 10^{-1}$	$1.60 \times 10^{-1}$	$8.24 \times 10^{-4}$
8	$3.35 \times 10^{-1}$	$7.82 \times 10^{-2}$	$5.87 \times 10^{-1}$	$9.75 \times 10^{-1}$	$2.38 \times 10^{-2}$	$1.72 \times 10^{-3}$
...	...	...	...	...	...	...
3999	$3.83 \times 10^{-1}$	$3.01 \times 10^{-1}$	$3.16 \times 10^{-1}$	$9.12 \times 10^{-1}$	$8.65 \times 10^{-2}$	$1.23 \times 10^{-3}$
4000	$5.85 \times 10^{-1}$	$1.37 \times 10^{-1}$	$2.78 \times 10^{-1}$	$9.50 \times 10^{-1}$	$4.60 \times 10^{-2}$	$4.04 \times 10^{-3}$

**Table A4.** Hybrid model 1 output.

Hybrid Model 1 Output			
Data ID	K_CO <sub>2</sub>	K_nC4	K_nC10
1	$1.94 \times 10^{-0}$	$3.02 \times 10^{-1}$	$7.13 \times 10^{-3}$
2	$2.10 \times 10^{-0}$	$2.89 \times 10^{-1}$	$5.23 \times 10^{-3}$
3	$2.45 \times 10^{-0}$	$2.78 \times 10^{-1}$	$3.23 \times 10^{-3}$
4	$1.87 \times 10^{-0}$	$2.96 \times 10^{-1}$	$7.16 \times 10^{-3}$
5	$1.87 \times 10^{-0}$	$2.94 \times 10^{-1}$	$7.01 \times 10^{-3}$
6	$1.16 \times 10^{-0}$	$6.01 \times 10^{-1}$	$1.50 \times 10^{-1}$
7	$2.11 \times 10^{-0}$	$3.23 \times 10^{-1}$	$7.63 \times 10^{-3}$

Table A4. Cont.

Hybrid Model 1 Output				
Data ID	K_CO <sub>2</sub>	K_nC4	K_nC10	
8	$2.91 \times 10^{-0}$	$3.04 \times 10^{-1}$	$2.94 \times 10^{-3}$	
...	...	...	...	...
3999	$2.38 \times 10^{-0}$	$2.87 \times 10^{-1}$	$3.89 \times 10^{-3}$	
4000	$1.62 \times 10^{-0}$	$3.35 \times 10^{-1}$	$1.45 \times 10^{-2}$	

## References

- Li, L.; Khorsandi, S.; Johns, R.T.; Dilmore, R.M. CO<sub>2</sub> enhanced oil recovery and storage using a gravity-enhanced process. *Int. J. Greenh. Gas Control.* **2015**, *42*, 502–515. [[CrossRef](#)]
- Magzymov, D.; Dindoruk, B.; Johns, R.T. Carbon Capture, Utilization, and Storage in the Context of Petroleum Industry: A State-of-the-art Review. In Proceedings of the SPE Improved Oil Recovery Conference, Virtual, 25–29 April 2022. [[CrossRef](#)]
- Dindoruk, B.; Johns, R.T.; Orr, F.M., Jr. Measurement and Calculation of Minimum Miscibility Pressure: A State of the Art Review. *SPE Reserv. Eval. Eng.* **2021**, *24*, 367–389. [[CrossRef](#)]
- Sinha, U.; Dindoruk, B.; Soliman, M. Prediction of CO<sub>2</sub> Minimum Miscibility Pressure (MMP) Using Machine Learning Techniques. In Proceedings of the SPE Improved Oil Recovery Conference, Virtual, 31 August–1 September 2020; p. SPE 200326-PA.
- Michelsen, M.L. The isothermal flash problem. Part I. Stability. *Fluid Phase Equilibria* **1982**, *9*, 1–19. [[CrossRef](#)]
- Young, L.C.; Stephenson, R.E. A generalized compositional approach for reservoir simulation. *Soc. Pet. Eng. J.* **1983**, *23*, 727–742. [[CrossRef](#)]
- Michelsen, M.L.; Mollerup, J.M. *Thermodynamic Models: Fundamentals and Computational Aspects*; Tie-Line Press: Kokkedal, Denmark, 2004.
- Gaganis, V.; Varotsis, N. Non-iterative phase stability calculations for process simulation using discriminating functions. *Fluid Phase Equilibria* **2012**, *314*, 69–77. [[CrossRef](#)]
- Danesh, A. *PVT and Phase Behaviour of Petroleum Reservoir Fluids*; Elsevier Science Ltd.: Oxford, UK, 1998.
- Rachford, H.H.; Rice, J.D. Procedure for Use of Electronic Digital Computers in Calculating Flash Vaporization Hydrocarbon Equilibrium. *J. Pet. Technol.* **1952**, *4*, 19. [[CrossRef](#)]
- Robinson, D.B.; Peng, D.Y. *The Characterization of the Heptanes and Heavier Fractions for the GPA Peng–Robinson Programs*; Research Report RR-28; Gas Processors Association: Tulsa, OK, USA, 1978.
- Soave, G. Equilibrium constants from a modified Redlich-Kwong equation of state. *Chem. Eng. Sci.* **1972**, *27*, 1197–1203. [[CrossRef](#)]
- Michelsen, M.L. Simplified flash calculations for cubic equations of state. *Ind. Eng. Chem. Process. Des. Dev.* **1986**, *25*, 184–188. [[CrossRef](#)]
- Hendriks, E.; Van Bergen, A. Application of a reduction method to phase equilibria calculations. *Fluid Phase Equilibria* **1992**, *74*, 17–34. [[CrossRef](#)]
- Okuno, R.; Johns, R.T.; Sepehrnoori, K. Application of a reduced method in compositional simulation. *SPE J.* **2010**, *15*, 39–49. [[CrossRef](#)]
- Voskov, D.V.; Tchelep, H.A. Compositional space parameterization: Multicontact miscible displacements and extension to multiple phases. *SPE J.* **2009**, *14*, 441–449. [[CrossRef](#)]
- Belkadi, A.; Yan, W.; Michelsen, M.L.; Stenby, E.H. Comparison of two methods for speeding up flash calculations in compositional simulations. In Proceedings of the SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 21–23 February 2011.
- Gaganis, V.; Varotsis, N. Machine learning methods to speed up compositional reservoir simulation. In Proceedings of the SPE Europec/EAGE Annual Conference, Copenhagen, Denmark, 4–7 June 2012.
- Kashinath, A.; Szulczewski, M.; Dogru, A. A fast algorithm for calculating isothermal phase behavior using machine learning. *Fluid Phase Equilibria* **2018**, *465*, 73–82. [[CrossRef](#)]
- Gaganis, V.; Varotsis, N. An integrated approach for rapid phase behavior calculations in compositional modeling. *J. Pet. Sci. Eng.* **2014**, *118*, 74–87. [[CrossRef](#)]
- Zhang, T.; Li, Y.; Li, Y.; Sun, S.; Gao, X. A self-adaptive deep learning algorithm for accelerating multi-component flash calculation. *Comput. Methods Appl. Mech. Eng.* **2020**, *369*, 113207. [[CrossRef](#)]
- Li, Y.; Zhang, T.; Sun, S. Acceleration of the NVT flash calculation for multicomponent mixtures using deep neural network models. *Ind. Eng. Chem. Res.* **2019**, *58*, 12312–12322. [[CrossRef](#)]
- Sheth, S.; Heidari, M.R.; Neylon, K.; Bennett, J.; McKee, F. Acceleration of thermodynamic computations in fluid flow applications. *Comput. Geosci.* **2022**, *26*, 1–11. [[CrossRef](#)]
- Zhang, T.; Li, Y.; Sun, S.; Bai, H. Accelerating flash calculations in unconventional reservoirs considering capillary pressure using an optimized deep learning algorithm. *J. Pet. Sci. Eng.* **2020**, *195*, 107886. [[CrossRef](#)]
- Chen, F.; Luo, S.; Wang, H.; Nasrabadi, S. A generalized machine learning-assisted phase-equilibrium calculation model for shale reservoirs. *Fluid Phase Equilibria* **2022**, *558*, 113423. [[CrossRef](#)]

26. Raissi, M.; Karniadakis, G.E. Hidden physics models: Machine learning of nonlinear partial differential equations. *J. Comput. Phys.* **2018**, *357*, 125–141. [[CrossRef](#)]
27. Blechschmidt, J.; Ernst, O.G. Three ways to solve partial differential equations with neural networks—A review. *GAMM-Mitteilungen* **2021**, *44*, e202100006. [[CrossRef](#)]
28. Arthurs, C.J.; King, A.P. Active training of physics-informed neural networks to aggregate and interpolate parametric solutions to the Navier-Stokes equations. *J. Comput. Phys.* **2021**, *438*, 110364. [[CrossRef](#)] [[PubMed](#)]
29. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [[CrossRef](#)]
30. Cai, S.; Mao, Z.; Wang, Z.; Yin, M.; Karniadakis, G.E. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mech. Sin.* **2021**, *37*, 1727–1738. [[CrossRef](#)]
31. Fraces, C.G.; Tchelepi, H. Physics informed deep learning for flow and transport in porous media. In Proceedings of the SPE Reservoir Simulation Conference, On-Demand, 4–6 October 2021.
32. Magzymov, D.; Ratnakar, R.R.; Dindoruk, B.; Johns, R.T. Evaluation of machine learning methodologies using simple physics based conceptual models for flow in porous media. *J. Pet. Sci. Eng.* **2022**, *219*, 111056. [[CrossRef](#)]
33. Ihunde, T.A.; Olorode, O. Application of physics informed neural networks to compositional modeling. *J. Pet. Sci. Eng.* **2022**, *211*, 110175. [[CrossRef](#)]
34. Sun, R.; Pan, H.; Xiong, H.; Tchelepi, H. Physical-informed deep learning framework for CO<sub>2</sub>-injected EOR compositional simulation. *Eng. Appl. Artif. Intell.* **2023**, *126*, 106742. [[CrossRef](#)]
35. Wang, K.; Luo, J.; Wei, Y.; Wu, K.; Li, J.; Chen, Z. Practical application of machine learning on fast phase equilibrium calculations in compositional reservoir simulations. *J. Comput. Phys.* **2020**, *401*, 109013. [[CrossRef](#)]
36. Chen, F.; Luo, S.; Wang, S.; Nasrabadi, H. A Novel Machine-Learning Assisted Phase-Equilibrium Calculation Model for Liquid-Rich Shale Reservoirs. In Proceedings of the SPE Reservoir Simulation Conference, Galveston, TX, USA, 28–30 March 2023. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.