

Enhancing the accuracy of physics-informed neural network surrogates in flash calculations using sparse grid guidance*

Yuanqing Wu ^{a, #}, Shuyu Sun ^{b, #,*}

^a School of Computer Science and Technology, Dongguan University of Technology, Dongguan, Guangdong 523808, China

^b Computational Transport Phenomena Laboratory (CTPL), Division of Physical Sciences and Engineering (PSE), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia



ARTICLE INFO

Keywords:

Physics-informed neural networks
Flash calculations
Sparse grids
Compositional flows

ABSTRACT

Flash calculations pose a significant performance bottleneck in compositional-flow simulations. While sparse grids have helped mitigate this bottleneck by shifting it to the offline stage, the accuracy of the surrogate model based on physics-informed neural networks (PINN) is still inferior to that of the sparse grid surrogate in many cases. To address this issue, we propose the sparse-grid guided PINN training algorithm. This approach involves rearranging the collocation points using sparse grids at each epoch to capture changes in the residual space. By doing so, the PINN surrogate achieves the required accuracy using the fewest collocation points possible, thereby avoiding potential performance bottlenecks. Moreover, the training time complexity of the sparse-grid guided PINN training is significantly lower compared to the normal training while maintaining the same level of accuracy. Consequently, the sparse-grid guided PINN training method enhances the accuracy of the PINN surrogate with minimal computational overhead. During the experiments, a flash calculation of methane-propane mixture is conducted using a PINN surrogate, guided by the principles of sparse grids. The collective experimental observations underscore the clear advantages of employing sparse-grid guided PINN training, showcasing superior outcomes in terms of convergence, stability, and accuracy.

1. Introduction

It is well known that flash calculations are a performance bottleneck in compositional-flow simulations [1–5], and how to accelerate them has recently become a hot topic [6–8]. Among all the solutions, the sparse-grid method [9–11,12] can achieve much greater acceleration than other methods, with experiments showing an acceleration of over 2000 times. In their solution, a sparse-grid surrogate is generated during the offline stage of the compositional-flow simulations, which is then interpolated to provide approximate results for flash calculations during the online stage of the simulations. This approach shifts the computational burden from the online stage to the offline stage. During the offline stage, flash calculations are performed to generate the sparse-grid surrogate. It is worth noting that a larger number of components results in a higher dimension of the sparse-grid surrogate. In such cases, the cost of generating the sparse-grid surrogate is also increased. Consequently, another performance bottleneck arises, although it is much smaller than the bottleneck encountered during the online stage.

To further address the bottleneck in the offline stage, a physics-informed neural network (PINN) surrogate is introduced in [13]. More about the PINN can be referenced to [14–16]. It has been identified that stability analysis [17,18] and the successive substitution routine [19] are two performance bottlenecks in flash calculations. The use of the PINN surrogate helps circumvent these bottlenecks, significantly reducing the generation time. However, in their experiments, the accuracy of the PINN surrogate is not as good as that of the sparse-grid surrogate. Thus, improving the accuracy of the PINN surrogate becomes a necessary task. One possible solution is to employ a larger number of collocation points in the sampling space. However, this approach also increases the training time, which may introduce another performance bottleneck in the offline stage. Therefore, it is crucial to determine the minimum number of points required to achieve the desired accuracy.

Properly arranging the collocation points in the sampling space is one solution to this issue [20,21]. There are primarily two approaches: the non-adaptive way and the adaptive way [22]. In the non-adaptive

* Corresponding author.

E-mail addresses: wuyuanq@126.com (Y. Wu), shuyu.sun@kaust.edu.sa (S. Sun).

This work is funded by General Program of Natural Science Foundation of Shenzhen (No. 20200801100615003).

way, the collocation points can be arranged in the sampling space either non-randomly or randomly based on a probability distribution, such as the uniform distribution. These points cover the entire sampling space, ensuring good generality. However, the distribution of points does not coincide with the error distribution in the sampling space, making it non-adaptive. In the adaptive way, more points are placed in zones with higher errors or larger value changes. These points capture the most information of the sampling space. Conversely, in other zones, fewer points are needed. As a result, the adaptive approach can significantly reduce the number of points required, potentially circumventing the performance bottleneck in the offline stage.

In the adaptive way, the collocation points are reselected at the beginning of each epoch to reflect the latest error conditions of the sampling space. The errors are defined as the residuals of the governing equations in the PINN. Consequently, complex computations may be required to calculate the errors. At the start of a new epoch, the errors on the points used in the previous epoch are computed. Then, the n points with the smallest errors are removed from the point set, and n new points are randomly sampled in the sampling space according to a probability distribution. This process generates a new point set for the new epoch. However, this operation faces two main issues. Firstly, the error distribution does not guide the generation of the point set effectively, meaning that some points may not accurately represent zones with larger errors. Secondly, the point set needs to be regenerated at each epoch, which may potentially become a performance bottleneck.

In the previous introduction, we discussed the use of sparse grids [23, 24] as a surrogate for approximating flash calculations. Initially, full grids were used, but this approach presented some challenges. However, the application of sparse grids proved to be an effective solution to these challenges. Building upon this concept, this study aims to explore the use of sparse grids in addressing the data point-set generation issue. The structure of the paper is as follows:

Introduction: Provides an overview of the topic.

Pressure-Temperature Flash Calculations: Offers a brief description of flash calculations.

Surrogates of Physics-Informed Neural Networks: Details the concept of surrogates using physics-informed neural networks.

Sparse-Grid Guided PINN Training: Explores the implementation of sparse-grid guided training for PINNs.

Experiments: Presents the experiments conducted to validate the proposed approach.

Conclusion: Summarizes the findings and concludes the study.

2. Pressure-temperature flash calculations

In compositional-flow simulations, flash calculations play a crucial role in determining the phase conditions and compositions of the phases in different physical and chemical environments, and therefore have many applications [25]. Two-phase conditions are commonly observed in hydrocarbon reservoirs, where the heavy phase is referred to as the oil phase (denoted by subscript "o") and the light phase as the gas phase (denoted by subscript "g"). The fluids consist of m components, each with a composition represented by x_i , where i ranges from 1 to m . Additionally, the pressure (p) and temperature (T) are given and remain constant throughout. These fixed-pressure and fixed-temperature flash calculations are known as pressure-temperature (PT) flash calculations [17,19]. The results of the flash calculations provide information about the phase conditions and compositions. In two-phase flash calculations, the phase conditions can include a single-oil phase, a single-gas phase, or an oil-gas phase where both phases coexist in the fluids. The compositions of the oil phase are denoted as x_{oi} ($i = 1, \dots, m$), while the compositions of the gas phase are denoted as x_{gi} ($i = 1, \dots, m$). Furthermore, if the total mole amount of substances is denoted as " F ", the flash calculations yield the mole amounts of substances in the oil phase (denoted as " L ") and the gas phase (denoted as " V "). Fig. 1 provides a brief overview

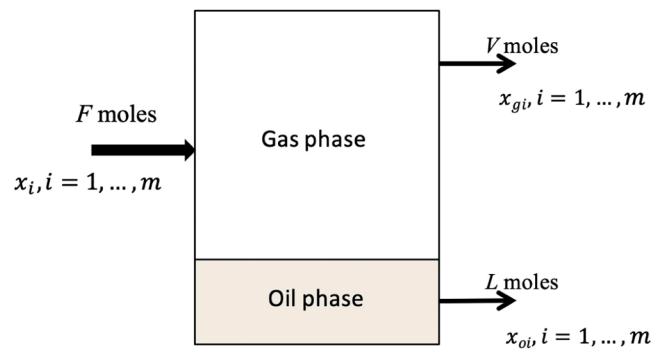


Fig. 1. A brief description of two-phase PT flash calculations.

of two-phase PT flash calculations.

When equilibrium is achieved and the oil-gas phase appears, flash calculations are governed by a set of equations that includes mass conservation equations (1) - (3) and fugacity equilibrium equations (4). The equations are as follows:

$$Fx_i = x_{oi}L + x_{gi}V, \quad i = 1, \dots, m, \quad (1)$$

$$\sum_{i=1}^m x_{oi} = 1, \quad (2)$$

$$\sum_{i=1}^m x_{gi} = 1, \quad (3)$$

$$f_{oi}(T, p, x_{oi}) = f_{gi}(T, p, x_{gi}), \quad i = 1, \dots, m. \quad (4)$$

The equations are stipulated by the following inequalities.

$$x_{oi} \geq 0, \quad i = 1, \dots, m, \quad (5)$$

$$x_{gi} \geq 0, \quad i = 1, \dots, m, \quad (6)$$

$$L \geq 0, \quad (7)$$

$$V \geq 0. \quad (8)$$

In the above equations, " f_{oi} " represents the oil-phase fugacity of the i -th component, while " f_{gi} " represents the gas-phase fugacity of the i -th component. The fugacity is defined as

$$f_{oi} = x_{oi}\varphi_{oi}p, \quad i = 1, \dots, m, \quad (9)$$

$$f_{gi} = x_{gi}\varphi_{gi}p, \quad i = 1, \dots, m, \quad (10)$$

with φ_{oi} and φ_{gi} represent the fugacity coefficients of the oil phase and the gas phase, respectively. To solve for the compositions in the phases, various complex iterative computational procedures, especially the conventional successive substitution method, are performed. These computations create a performance bottleneck in flash calculations. Additionally, stability analysis is conducted before these computations to determine the phase conditions (to ensure the four inequalities stated above to hold strictly), which also contributes to the performance bottleneck. It is important to note that in stability analysis, when determining a single-oil phase, the value of L is set to 1, while for a single-gas phase, L is set to 0. Additionally, in the single-oil phase, the values of x_{oi} and x_{gi} are equal to x_i and 0, respectively. Conversely, in the single-gas phase, x_{gi} and x_{oi} are equal to x_i and 0, respectively. For further information on flash calculations, please refer to [9].

3. Surrogates of physics-informed neural networks

If flash calculations are treated as a function with $F = 1$, the inputs of the function can be considered as the pressure p , the temperature T , and

the compositions of the substances x_i ($i = 1, \dots, m$). Consequently, the outputs of the function can be considered as the phase compositions x_{oi} , x_{gi} ($i = 1, \dots, m$), and L . Since $L + V = F$, it is not necessary to include V as an output. Given that the composition values range between zero and one, the pressure p and the temperature T should be nondimensionalized to \tilde{p} and \tilde{T} , respectively, using the following equations:

$$\tilde{p} = \frac{p - p_{\min}}{p_{\max} - p_{\min}}, \quad (11)$$

$$\tilde{T} = \frac{T - T_{\min}}{T_{\max} - T_{\min}}. \quad (12)$$

Here, p is assumed to be in the range $[p_{\min}, p_{\max}]$, and T is assumed to be in the range $[T_{\min}, T_{\max}]$. When the PINN surrogate is applied, the inputs of the function form the input layer of the PINN, while the outputs form the output layer of the PINN. Fig. 2 illustrates a diagram of the PINN surrogate.

The PINN surrogate shown in Fig. 2 consists of an input layer, an output layer, and three hidden layers. Due to the nature of feed-forward neural networks and the challenge of accurately representing cases where $L = 0$ or $L = 1$, the PINN struggles to accurately capture the single-phase conditions. Therefore, the assumption of an oil-gas phase condition is made by the PINN. To ensure that the sum of compositions in each phase is equal to one and that the compositions remain positive, softmax layers are incorporated. The sigmoid layer ensures that the value of L remains within the range of $[0,1]$. Mask layers are used to maintain consistency in the compositions between the input and output layers. Specifically, when x_i is zero, x_{oi} and x_{gi} are also set to zero. It is worth mentioning that the hyperparameters of the PINN, such as the number of hidden layers and neurons per layer, can be adjusted based on the specific requirements of the problem. However, it is possible for the PINN surrogate to approximate the single-phase conditions by introducing a tolerance factor ϵ ($0 \leq \epsilon \leq 0.1$). This tolerance factor allows for an approximation of the phase condition as follows:

$$\text{Phase condition is } \begin{cases} \text{a single-oil phase, if } L > 1 - \epsilon, \\ \text{a single-gas phase, if } L < \epsilon, \\ \text{an oil-gas phase, otherwise.} \end{cases} \quad (13)$$

The physical information of the oil-gas phase, described by Equations (1) - (4), is integrated into the network by including their residuals in the loss function. Fortunately, Eqs. (2) and (3) are naturally fulfilled when

using softmax functions, resulting in zero residuals. Therefore, only the residuals of Eq. (1) and Eq. (4) are considered in the loss function. To formulate the loss function, the residuals of the equations are assembled into an auxiliary vector known as the residual vector, one by one. For cases where $x_i > 0$, the residual for Eq. (1) is defined as:

$$\left| \log \frac{x_{oi} * L + x_{gi} * (1 - L)}{x_i} \right|. \quad (14)$$

However, if $x_i = 0$, there are two residuals arising from Eq. (1), which are defined as follows:

$$\sqrt{x_{oi}^2}, \quad (15)$$

$$\sqrt{x_{gi}^2}. \quad (16)$$

The calculation of the residual for Eq. (4) is slightly more complex. When $x_i = 0$, there is no residual associated with Eq. (4). Therefore, we will only discuss the case where $x_i > 0$. If the oil-gas phase is present, the residual is defined as follows:

$$\left| \log \frac{x_{oi} * \varphi_{oi}}{x_{gi} * \varphi_{gi}} \right|, \quad (17)$$

On the other hand, if there is only a single phase, the residual is defined as:

$$\sqrt{(x_{oi} - x_i)^2 + x_{gi}^2}, \quad (18)$$

$$\sqrt{(x_{gi} - x_i)^2 + x_{oi}^2}, \quad (19)$$

for the single-oil phase and single-gas phase, respectively.

However, the approximation of phase conditions using ϵ heavily relies on its selection. To mitigate the subjectivity, the residuals of Eq. (4) in the single-phase conditions can be modified as follows. When a single-oil phase is present, the residual is defined as:

$$\left| \log \frac{x_{oi} * \varphi_{oi}}{x_{gi} * \varphi_{gi}} \right| * (1 - L) + \sqrt{(x_{oi} - x_i)^2 + x_{gi}^2} * L. \quad (20)$$

In this scenario, a higher output value of L indicates a higher likelihood of a single-oil phase. Therefore, the weight of the oil-gas-phase residual is smaller, while the weight of the single-oil-phase residual is

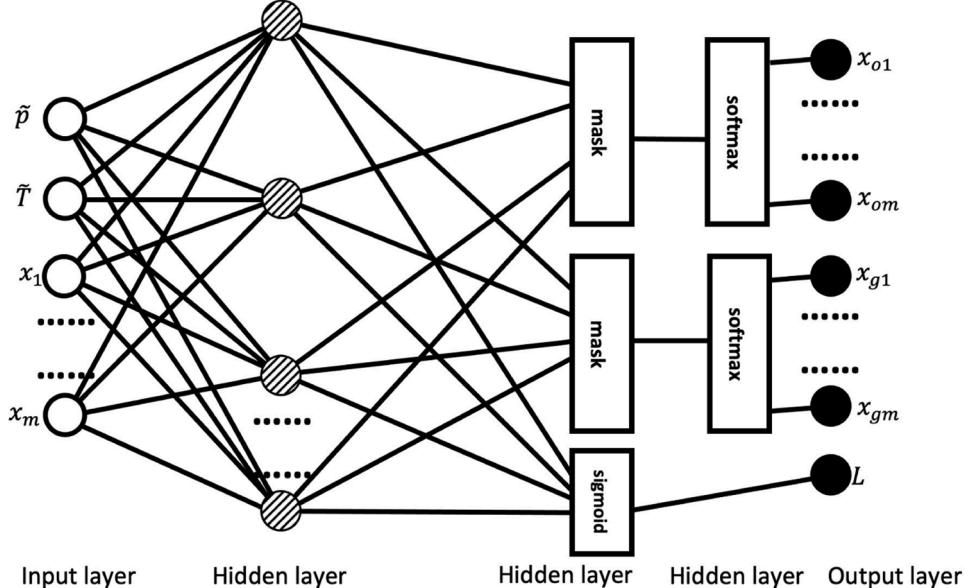


Fig. 2. A diagram of the PINN surrogate.

larger. When a single-gas phase is present, the residual is defined as:

$$\left| \log \frac{x_{oi} * \varphi_{oi}}{x_{gi} * \varphi_{gi}} \right| * L + \sqrt{\left(x_{gi} - x_i \right)^2 + x_{oi}^2} * (1 - L). \quad (21)$$

After these operations, the loss function can be defined as the mean squared error between the residual vector and the zero vector.

The phase condition determined by the expression (13) is used solely for deciding the residual expression during the training process. However, once the PINN surrogate is trained, it can be used to determine the phase condition using the PINN-induced phase decision method (PPDM) proposed by Wu and Sun in their work [13]. The PPDM leverages the capabilities of the trained PINN surrogate to make accurate phase predictions based on the phase compositions.

4. Sparse-grid guided PINN training

4.1. PINN training

Sparse grids are mathematical techniques used for the numerical approximation and efficient representation of functions, especially in high-dimensional spaces. They offer a means to reduce the computational complexity associated with traditional methods like full grids or adaptive grids. The core concept behind sparse grids is to leverage the hierachal structure of the problem domain by selectively refining certain regions of interest while coarsening others. This selective refinement enables a more efficient allocation of computational resources.

On the other hand, the training of the PINN involves multiple epochs. In the training conducted by Wu and Sun [13], the arrangement of the collocation points remains unchanged in each epoch. However, in the sparse-grid guided point-set generation algorithm, the collocation points are regenerated for every epoch to capture changes in the equation residuals between successive epochs. By employing the idea of sparse grids, the points are strategically placed to capture important features on regions of interest in the equation-residual space.

In the equation-residual space used for constructing sparse grids, there are m dimensions representing different variables such as non-dimensionalized pressure (\tilde{p}) and compositions (x_i) for each component ($i = 1, \dots, m - 1$). The temperature is typically considered fixed in compositional-flow simulations and is not considered as a dimension in the equation-residual space. Based on the analysis in Section 3, every point in the equation-residual space is associated with $2m$ residuals. When $x_i = 0$, there are two residuals, one from Eq. (15) and the other from Eq. (16). When $x_i > 0$, there is one residual from Eq. (14). Additionally, depending on the phase condition, there is an additional residual from Eq. (17) for the oil-gas phase, Eq. (20) for the single-oil phase, or Eq. (21) for the single-gas phase. Given that there are m components, there are a total of $2m$ residuals associated with each point in the equation-residual space.

Fig. 3 illustrates the training process of the PINN guided by sparse grids. Initially, an initial collocation point set is used to train the PINN in the first epoch. For simplicity, these points can be arranged in a uniform grid pattern. Once the PINN is trained, it serves as a surrogate to approximate the values of L and the phase compositions, enabling the calculation of equation residuals in the equation-residual space. Sparse grids are then constructed for each type of residual, resulting in a total of $2m$ -dimensional sparse grids. These sparse grids are combined to create a unified sparse grid. The points within this unified sparse grid form the sampling space for the next epoch of PINN training. If the PINN achieves convergence, meaning it reaches a satisfactory level of accuracy, the training process is concluded. However, if convergence is not achieved, the training proceeds to the next epoch, continuing the iterative process until convergence is reached.

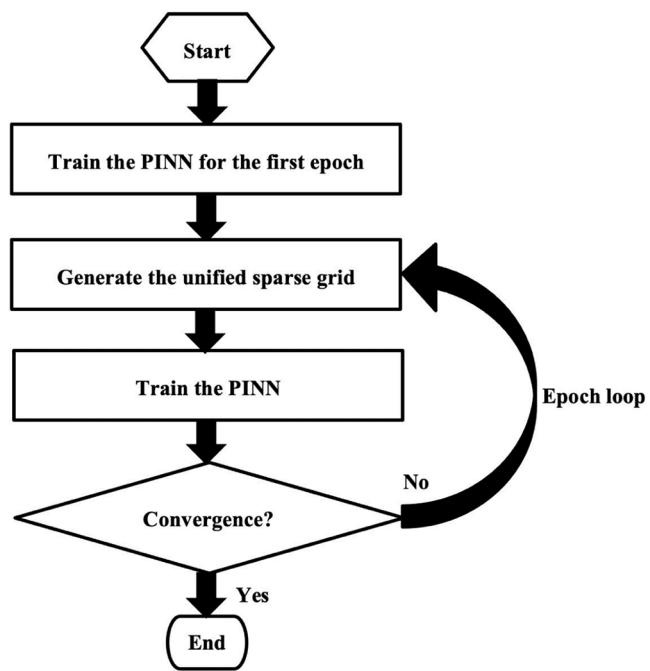


Fig. 3. A Flowchart of the PINN training guided by sparse grids.

4.2. Sparse grid construction

In the construction of the sparse grid, the process begins with placing an initial point called the God point at the center of the equation-residual space. This point also serves as the only point on the first level of the sparse grid, denoted as $S_{(1,1)(1,1)}$ in a two-dimensional equation-residual space. The subscripts in this notation represent the point level and index in each dimension, respectively. It is important to note that all levels and indices start from one, and all indices are odd.

For any point in the sparse grid, the associated equation residual is calculated. Additionally, the equation residual can also be approximated by considering all its ancestors. The difference between the true value and the approximated value of the equation residual is referred to as the surplus of the point. If the absolute value of the surplus exceeds a threshold ϵ , indicating that the approximation accuracy is not sufficient, the point needs to be refined, and all its children should be possibly added to the sparse grid. Conversely, if the surplus is below ϵ , no further refinement is required. The level of the children is one higher than that of the parent point. This refinement process is carried out iteratively, level by level, until either the absolute values of all surpluses are below ϵ or the maximum level is reached.

Fig. 4 illustrates a three-level sparse grid in a two-dimensional space. The solid point represents the God point. After refinement, four children are added to the sparse grid, represented by the striped points. The points $S_{(2,1)(3,1)}$ and $S_{(1,2)(1,1)}$ do not undergo further refinement as their approximation accuracy is sufficient. On the other hand, the points $S_{(1,2)(1,3)}$ and $S_{(2,1)(1,1)}$ are further refined, but only one child is added for each of them, indicating that the approximation accuracy of the other children is satisfactory. Notably, the hollow point $S_{(2,2)(1,3)}$ is the common child of the points $S_{(1,2)(1,3)}$ and $S_{(2,1)(1,1)}$.

It is worth highlighting that when the dimensionality exceeds two, the configuration of the equation-residual space may not be regular due to the requirement that the sum of all compositions should be equal to one. Consequently, some points may fall outside of the space. However, they are still added as ghost points in the sparse grid to maintain the hierarchical structure, provided that at least one of their children lies within the space. For further details on the construction of sparse grids, please refer to [12].

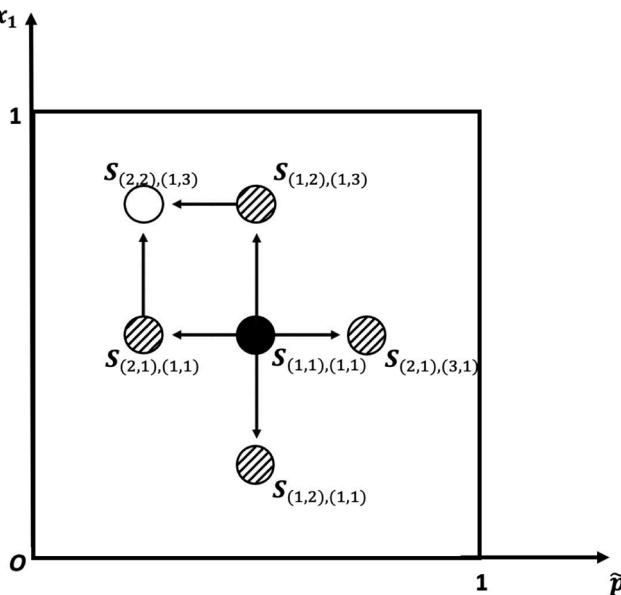


Fig. 4. A sketch of a three-level sparse grid in a two-dimensional space.

5. Experiments

In the experiments, a PINN similar to the one depicted in Fig. 2 is utilized as a surrogate. The PINN is parametrized with six hidden layers, and each of the first four hidden layers consists of 20 neurons. The activation function employed in the hidden layers is the Rectified Linear Unit (ReLU). It is important to note that numerous hyperparameter values have been thoroughly explored, and the selections made in the experiments represent the optimal choices. The pressure range spans from 19 bar to 21 bar, while the temperature is fixed at 220 K. The threshold value for ϵ is set to 0.1. The training process is executed for a maximum of 20 epochs, with a batch size of 128. Prior to each training run, the samples are shuffled. The "HeNormal" initializer, proposed by Datta in 2020 [26], is employed to initialize the network parameters. To enhance the model's generalization and prevent overfitting, L2 regularization or L2 penalty is applied. The Adam optimizer is utilized to optimize the network during the training process. The PINN implementation is carried out using Keras and TensorFlow 2.8, and the code is executed on a Mac system.

In the PT flash calculation, two substances, methane and propane, are considered. In Table 1, the molecular weights of methane and propane are 16.04 g/mol and 44.09 g/mol, respectively. The critical temperatures for methane and propane are 190.0 K and 370.0 K, while the critical pressures are 46 bar and 42 bar, respectively. The acentric factors for methane and propane are 0.01 and 0.15, respectively. Additionally, the binary interaction parameter (BIP) for the substances is 0.036. The surplus threshold (c) for the sparse grids is set to 10^{-5} , indicating that points with an absolute surplus value larger than 10^{-5} will undergo further refinement in the sparse grid construction process.

5.1. Experiments with the seven-level sparse grid

5.1.1. An experiment without hierachal priority

In the sparse grid construction, the maximal level of the sparse grid is

Table 1
Physical and chemical parameters of the substances.

Substance	Molecular weight (g/mol)	Critical temperature (K)	Critical pressure (bar)	Acentric factor
Methane	16.04	190.0	46	0.01
Propane	44.09	370.0	42	0.15

set to seven. The equation-residual space is discretized using a uniform grid of 31 points in the pressure dimension and 51 points in the first-composition dimension, with methane as the first component. These points form the initial sample point set in the equation-residual space. The experimental results are shown below.

In Fig. 5, it can be observed that the training process of the PINN surrogate achieves convergence within 20 epochs. To assess the accuracy of the PINN surrogate, a uniform grid of 100 points is sampled in the equation-residual space, specifically in the pressure dimension and the first-composition dimension. True flash calculations are performed on this grid to obtain the true phase conditions and phase compositions. The PINN surrogate is then applied to the grid to obtain the approximated values.

Fig. 6 illustrates the comparison between the true phase diagram and the approximated phase diagram obtained from the PINN surrogate. Additionally, the phase error in the equation-residual space is shown. Three types of phase conditions are identified: the red points represent the single-oil phase, the blue points represent the single-gas phase, and the green points represent the oil-gas phase. In Fig. 6(c), any point where the approximated phase condition differs from the true phase condition is marked as a black point, indicating a discrepancy between the two.

The composition error is defined as the absolute difference between the true value and the approximated value, as shown in Fig. 7. From the figures, it can be observed that the largest errors occur in the regions where the phase condition is incorrectly determined by the PPDM. Fortunately, these regions are relatively small. In other regions, the errors are generally very small. It is worth noting that when the single-phase conditions are correctly determined, the errors are zero. Additionally, in the two-component condition, the error diagrams of x_{o1} and x_{o2} should be the same, as well as the error diagrams of x_{g1} and x_{g2} . However, some differences can still be observed between them. For instance, if a point is associated with a single-gas phase but is incorrectly determined as an oil-gas phase, the error of x_{o1} and the error of x_{o2} are not equal to each other, although their sum is one. The points in the upper-right region of Fig. 6(c) fall into this category.

It is important to note that the training results of the PINN can vary due to the randomness associated with the initial weights and biases of the network, as well as the randomness introduced by the batch selection. In this study, the results with the smallest errors were chosen to be presented, ensuring the demonstration of the best-performing outcomes.

5.1.2. An experiment with hierachal priority

The hierachal attribute of sparse grids can significantly contribute to their high performance. However, the previous experiment did not

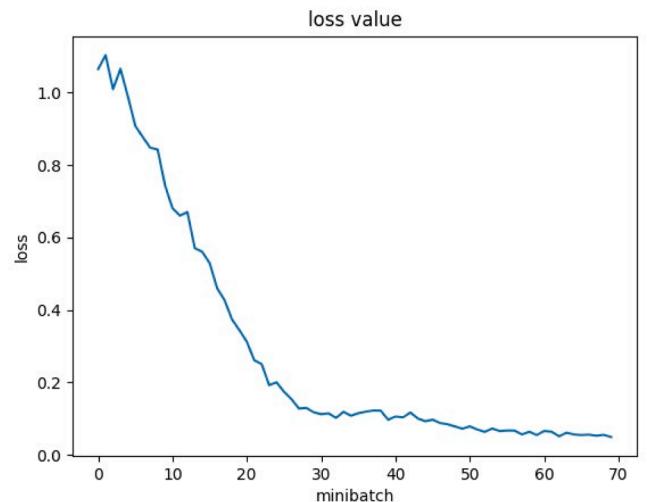
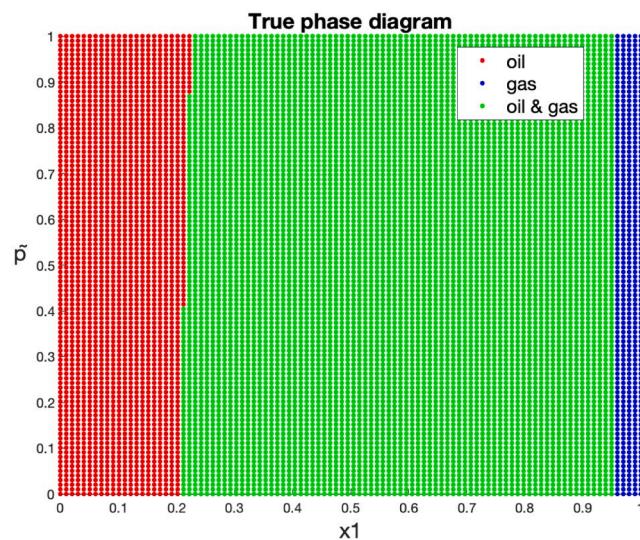
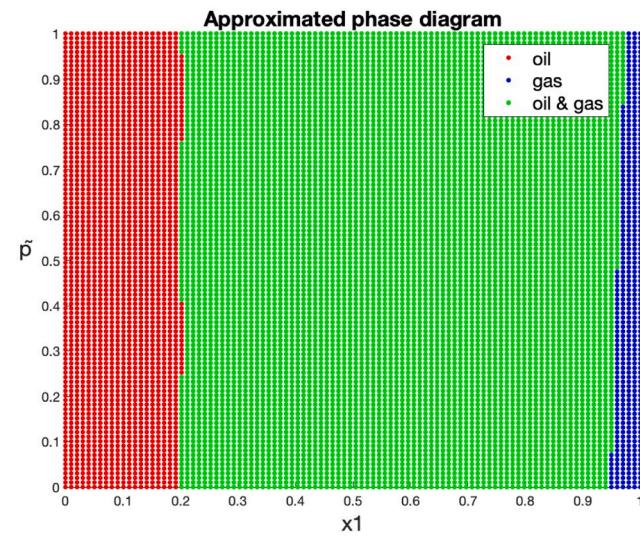


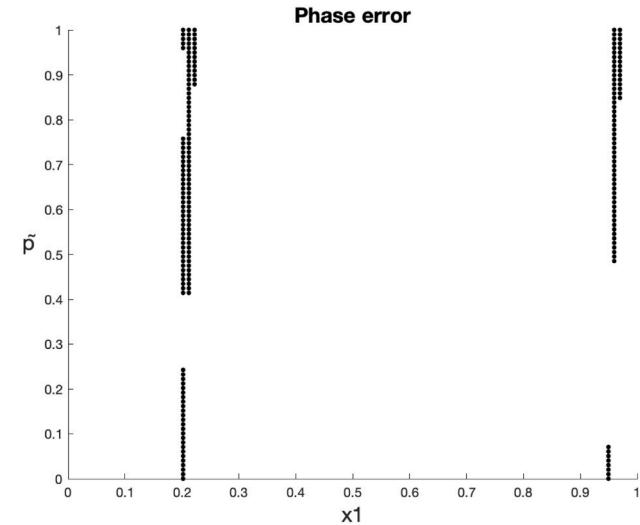
Fig. 5. Loss of the PINN training.



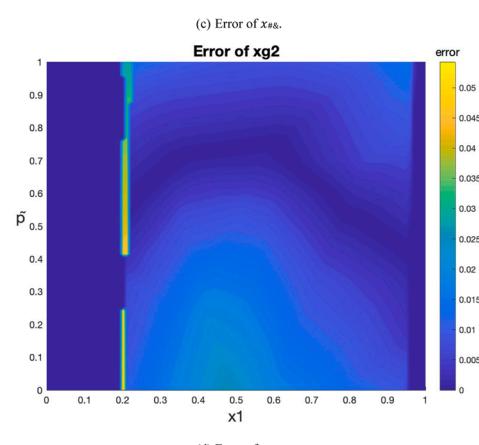
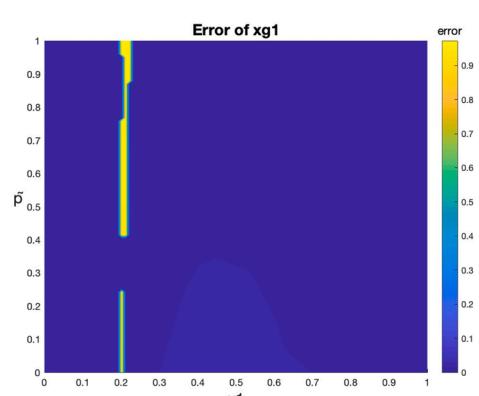
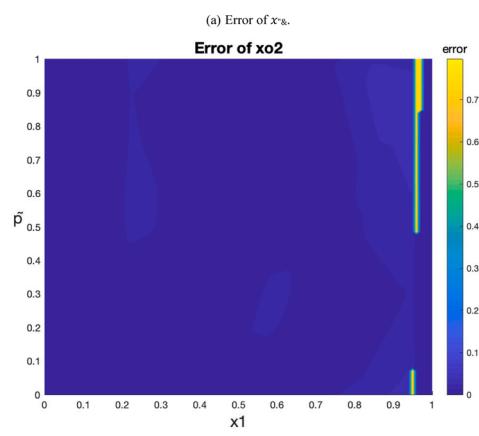
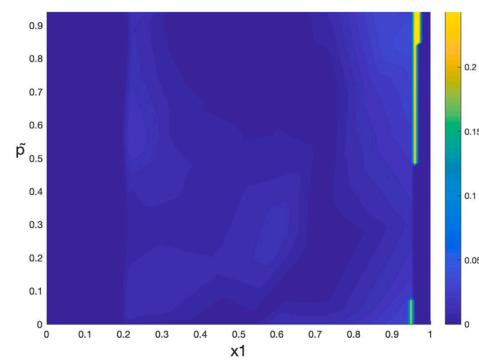
(a) A true phase diagram.



(b) An approximated phase diagram.



(c) Phase identification error.

Fig. 6. Phase diagrams of the equation-residual space.**Fig. 7.** Errors of the phase compositions.

utilize this attribute. In order to incorporate the hierarchical attribute, the points on lower levels should have a higher probability of being trained earlier. To achieve this, the points are organized in a "dataset" where points with lower levels are placed before those with higher levels. The following Python code snippet can be used to implement this:

```
dataset = dataset.shuffle(buffer_size=BATCH_SIZE).batch(BATCH_SIZE)
```

Here, "BATCH_SIZE" refers to the batch size, which was set to 128 in the experiments. After making these changes, the experiment in [Section 5.1.1](#) is repeated, and the results are presented below.

The results of the experiment in [Fig. 9](#) and [Fig. 10](#) indicate that the application of the hierarchical priority, as implemented in this experiment, does not significantly improve the approximation accuracy of the PINN surrogate compared to the previous experiment. This suggests that the hierarchical attribute may not be necessary for training in this particular scenario. Furthermore, it is observed that the loss curve in [Fig. 8](#) is more stable compared to that in [Fig. 5](#). However, it is worth noting that [Fig. 8](#) requires more batches to complete the 20 epochs of training, indicating the generation of a larger number of sparse grid points during the training process. This observation suggests that the application of the hierarchical priority may not be as impactful or necessary in this case. Overall, based on these findings, it can be concluded that the hierarchical priority may not be essential for achieving high approximation accuracy in this specific PINN training scenario.

5.1.3. An experiment without a sparse-grid guide

In the above experiments, sparse grids were utilized to generate collocation points in the equation-residual space. To further highlight the significance of sparse-grid guided PINN training, an additional experiment was conducted without the use of sparse grids. It was observed that the total number of collocation points for the experiment without hierarchical priority amounted to 8554, resulting in an average of 427.7 collocation points per epoch. In this particular experiment, the number and arrangement of collocation points remained consistent throughout each epoch. Specifically, 459 points were selected for each epoch, slightly exceeding the average of 427.7 points. Within the point set, 9 points were uniformly sampled in the pressure dimension, while 51 points were uniformly sampled in the first-composition dimension. [Fig. 11](#) illustrates the number of collocation points for the three experiments.

The experimental results are presented below. In [Fig. 12](#), it is evident that the loss decreases during the training process. However, it is noticeable that convergence is not achieved by the end of the 20th epoch. Additionally, the loss curve exhibits significant fluctuations,

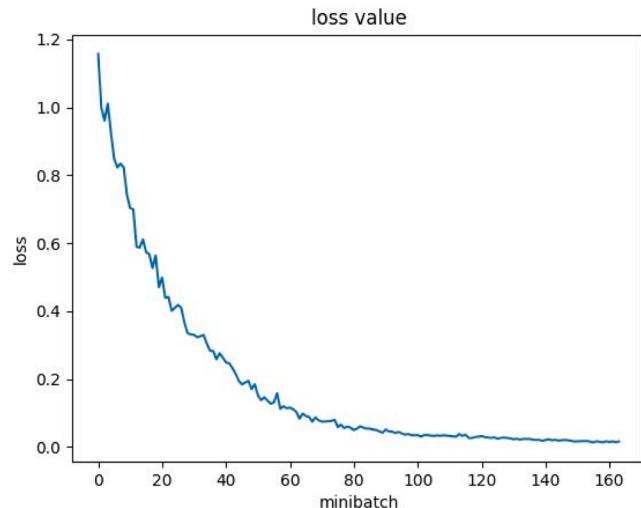
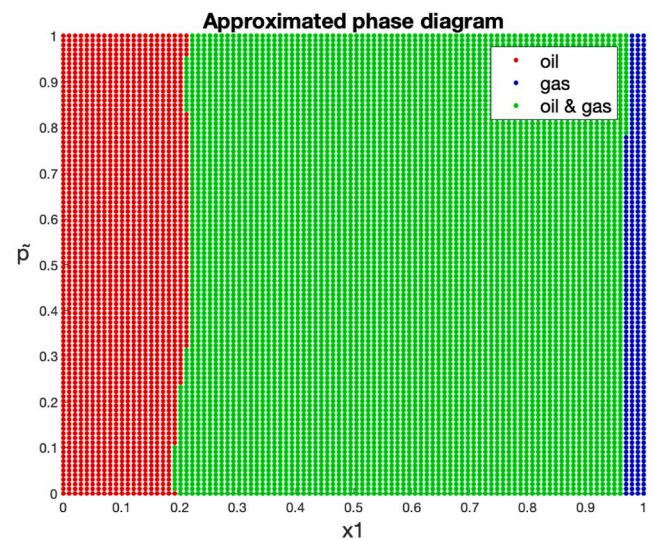
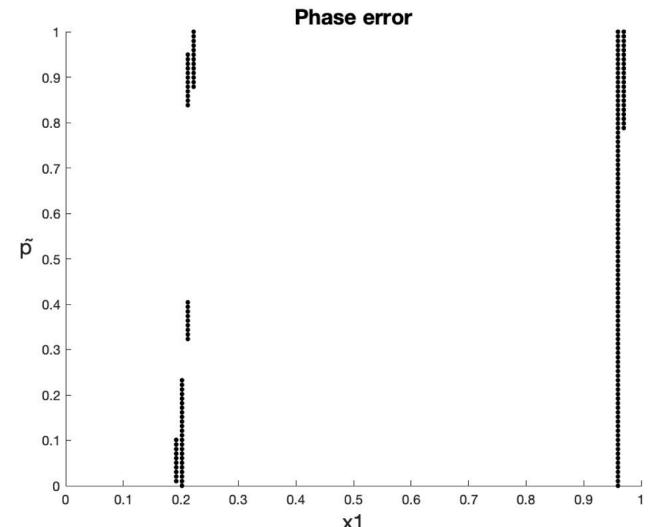


Fig. 8. Loss of the PINN during training.



(a) An approximated phase diagram.



(b) Phase identification error.

Fig. 9. Phase diagrams of the equation-residual space.

indicating that the training lacks stability compared to the previous experiments. [Fig. 13](#) illustrates a pronounced phase error, and [Fig. 14](#) demonstrates larger errors compared to previous experiments. It is worth noting that despite using slightly more collocation points per epoch compared to the experiment without hierarchical priority, the results do not show improvement. These observations collectively highlight the superiority of sparse-grid guided PINN training in terms of convergence, stability, and accuracy.

5.2. Experiments with the nine-level sparse grid

In this section, the maximal level of the sparse grid is increased to nine, and the three experiments with the seven-level sparse grid are repeated. As a result, the number of collocation points also increases, as shown in [Fig. 15](#).

The experimental results are presented below. In [Fig. 16](#), it is observed that all three loss curves achieve convergence by the end of the 20th epoch. However, the experiment without a sparse-grid guide exhibits more fluctuations. Furthermore, when comparing the loss curves to those obtained from the experiments with a seven-level sparse grid, both the convergence and stability of the curves are improved due to the

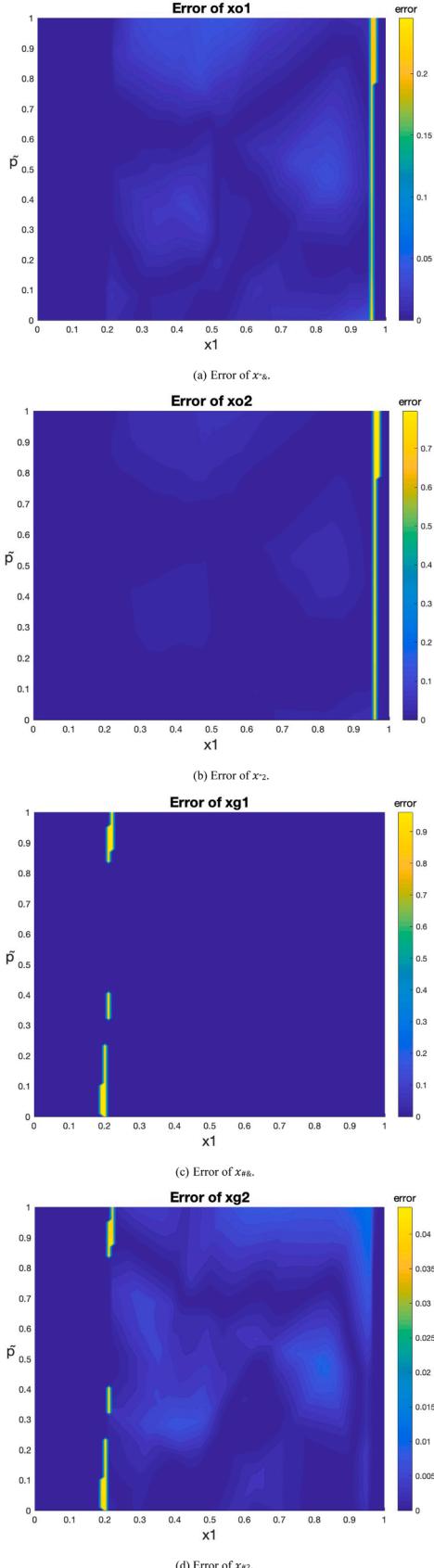


Fig. 10. Errors of the phase compositions.

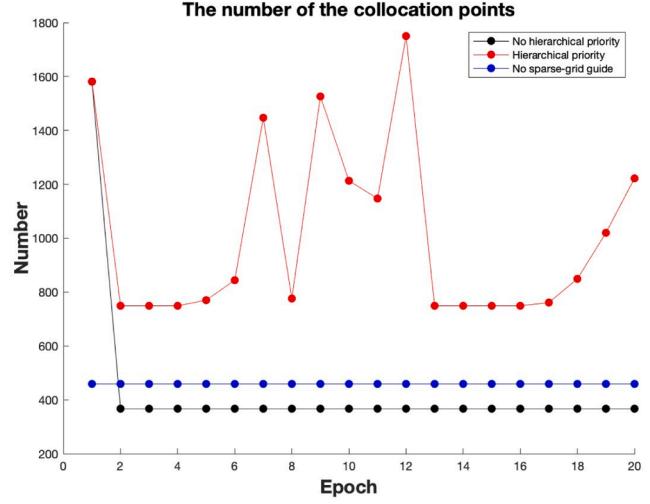


Fig. 11. The number of the collocation points for the experiments without hierachal priority, with hierachal priority, and without a sparse-grid guide.

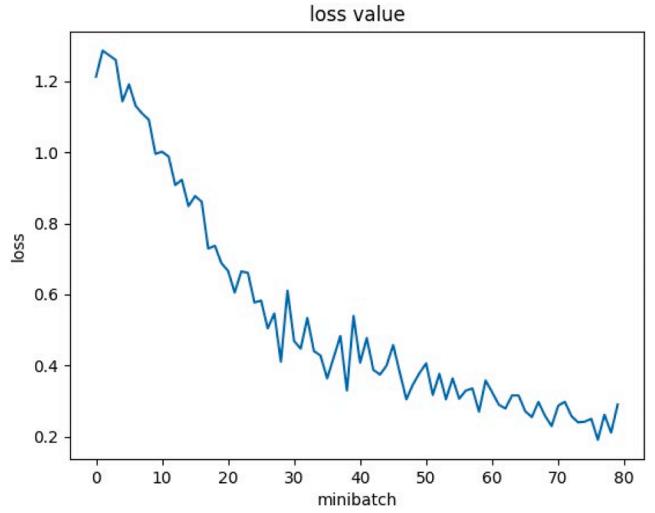


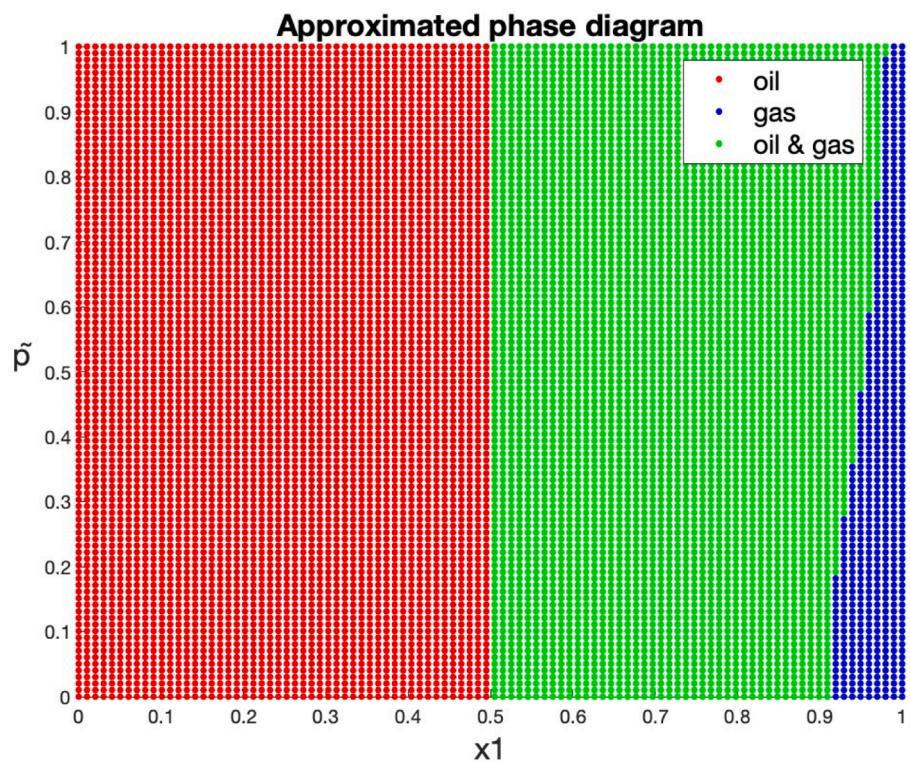
Fig. 12. Loss of the PINN during training.

increased number of collocation points. In Fig. 17, it is evident that the phase errors are similar across all three experiments. Additionally, in Figs. 18–20, it can be seen that all errors are very small. Therefore, it can be concluded that with a nine-level sparse grid and an adequate number of collocation points, the training exhibits good stability, convergence, and accuracy in phase decision and composition prediction. In such cases, the use of sparse-grid guided PINN training may not be necessary. In other words, the priority of the sparse-grid guided PINN training becomes apparent when a smaller number of collocation points are available.

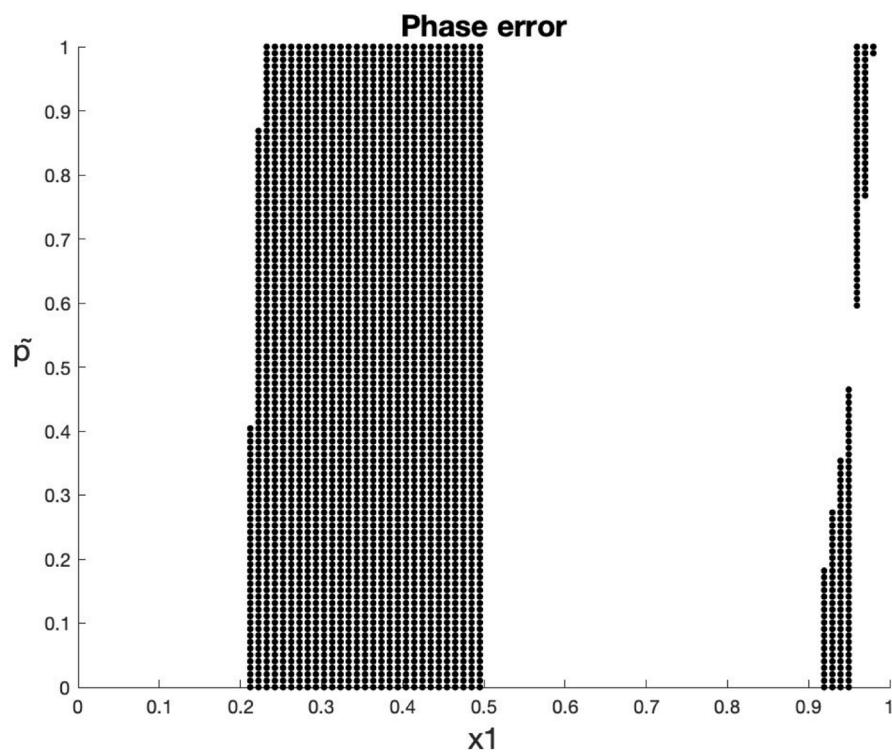
Additionally, it is observed from Fig. 11 and Fig. 15 that the number of collocation points for the experiments without hierarchical priority remains consistent over multiple epochs. This observation suggests that the construction of the sparse grid may not be necessary for every epoch. In such cases, the sparse-grid guided PINN training can be further simplified, reducing the computational overhead associated with the construction of the sparse grid for each epoch.

5.3. Time complexity

From the conducted experiments, it can be concluded that when the number of collocation points is insufficient, rearranging the points



(a) An approximated phase diagram.



(b) Phase identification error.

Fig. 13. Phase diagrams of the equation-residual space.

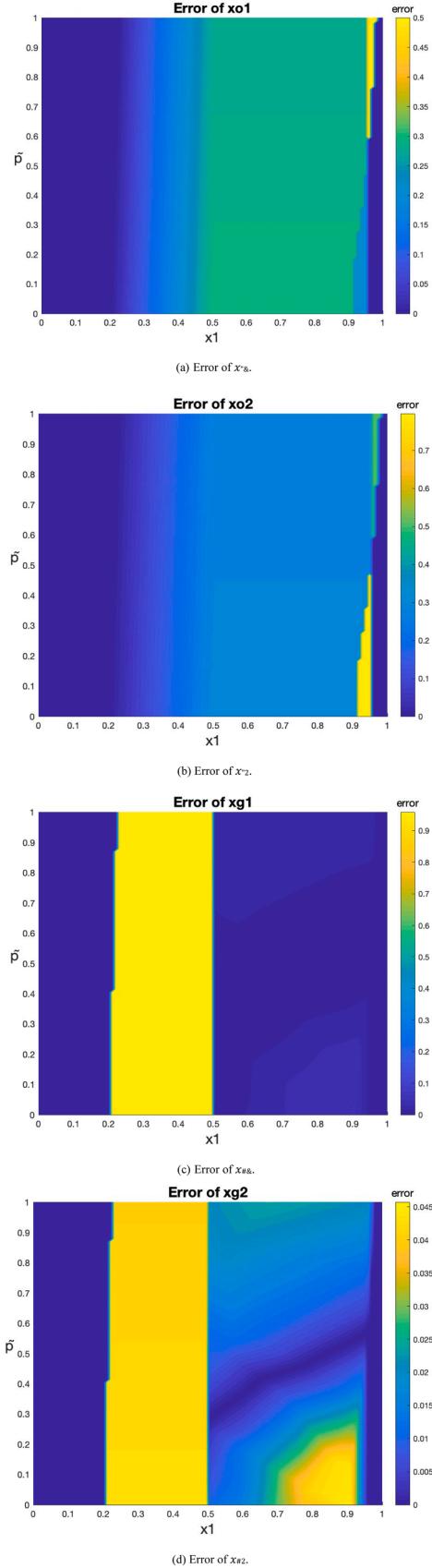


Fig. 14. Errors of the phase compositions.

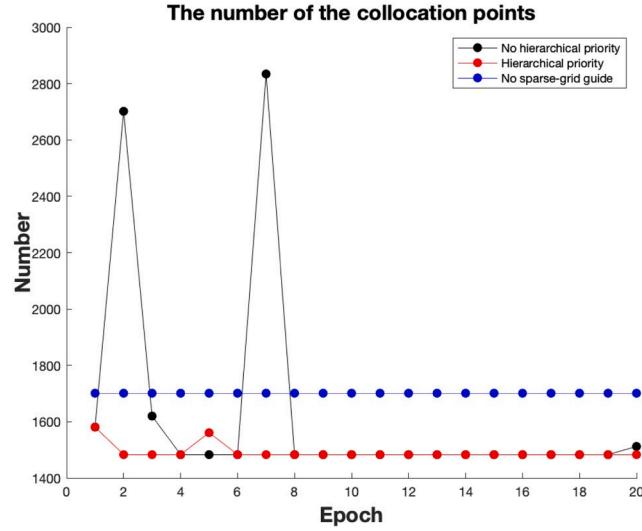


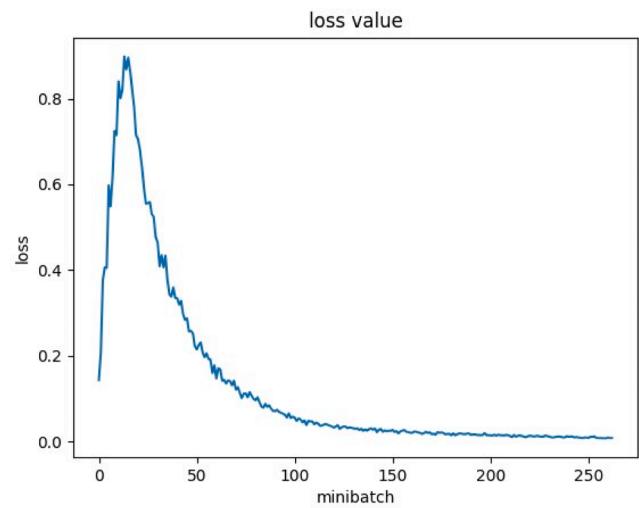
Fig. 15. The number of the collocation points for the experiments without hierachal priority, with hierachal priority, and without a sparse-grid guide.

according to the sparse grid before each epoch can still ensure the accuracy of the PINN. However, when an adequate number of collocation points are available for training the PINN effectively, the assistance of sparse grids is not crucial.

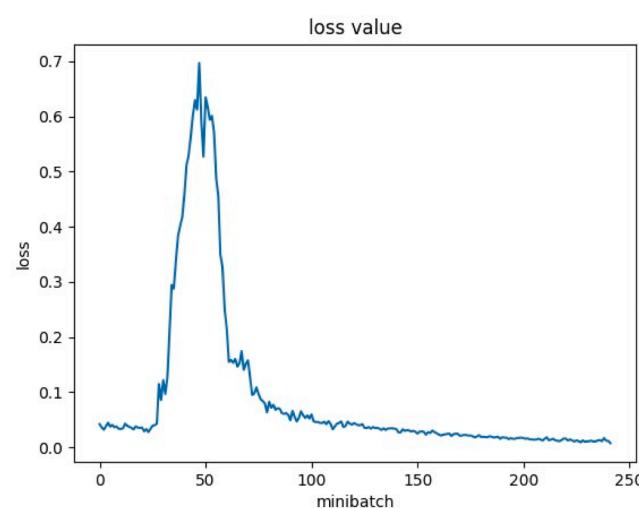
It is important to note that the benefit obtained comes at the cost of performing sparse grid construction at each epoch. The calculation of fugacity is known to be a performance bottleneck among all computations. Therefore, the time required for fugacity calculations serves as a good indicator of time complexity. In the sparse-grid guided PINN training, if the number of collocation points for all epochs is denoted as " u ", then $O(u)$ fugacity calculations need to be performed in both the sparse grid construction and the PINN training. Consequently, the time complexity of the sparse-grid guided PINN training is $O(u)$.

On the other hand, without the guidance of sparse grids, a larger number of collocation points is required at each epoch to achieve similar levels of accuracy. In this scenario, if the number of collocation points for all epochs is denoted as " v ", then $O(v)$ fugacity calculations need to be performed solely in the PINN training, resulting in a time complexity of $O(v)$. Since v is generally much larger than u , the time complexity of the sparse-grid guided PINN training is significantly smaller compared to normal PINN training with similar levels of accuracy.

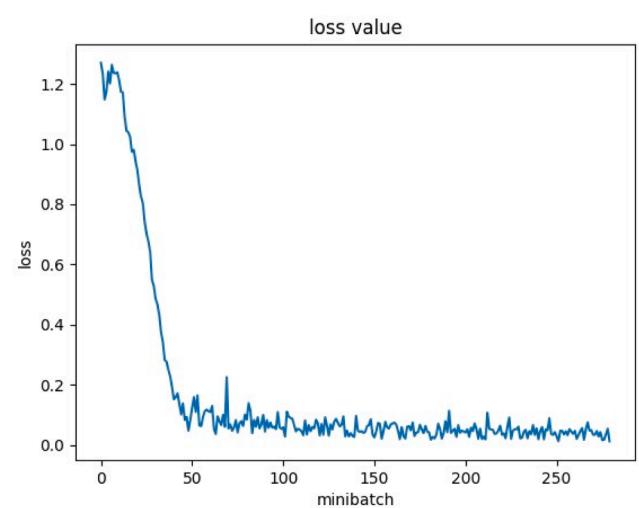
It is noteworthy that alternative algorithms for dynamically allocating collocation points have been proposed [27,28]. In their studies, an initial collocation point set is uniformly selected within the domain. After the first epoch, ' w ' points with the smallest surpluses are removed from the point set, and ' w ' new points are randomly chosen from the domain and added to the set. This process is repeated recursively for each epoch. In contrast to the point generation algorithm employed in our work, where all collocation points are regenerated before each epoch, their algorithm only requires a partial regeneration of points before each epoch. Additionally, their approach is more cost-effective in terms of generating new points because they are generated randomly. This cost-effectiveness also translates into reduced overhead for dynamically allocating collocation points. However, it's worth noting that, due to the random generation of points in their algorithm, their collocation points may not accurately capture phase-transition zones, potentially leading to degraded performance. Consequently, more epochs may be necessary to achieve convergence. Furthermore, their algorithm maintains a fixed number of collocation points throughout the process, whereas our algorithm allows for a variable number of collocation points in different epochs, providing an advantage. This flexibility ensures that fewer points are needed when phase-transition zones are less prominent, reducing the point-generation cost in our work.



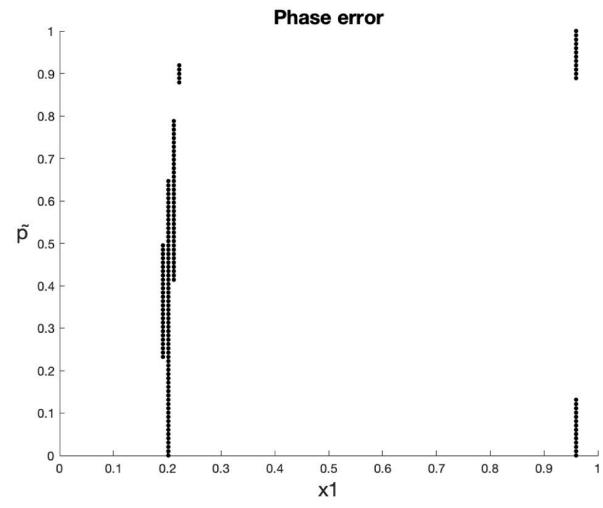
(a) Without hierachal priority.



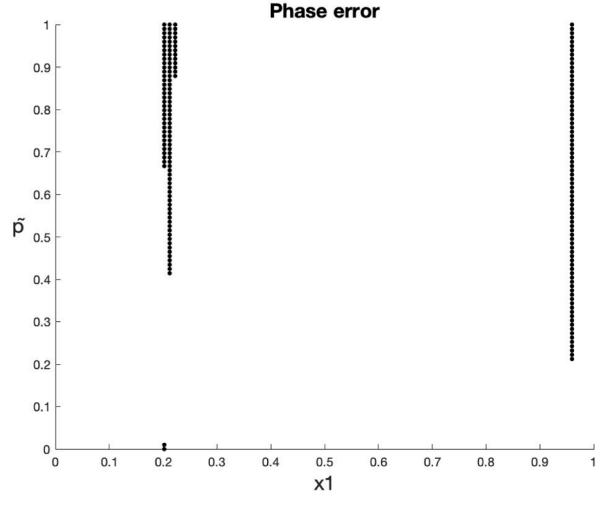
(b) With hierachal priority.



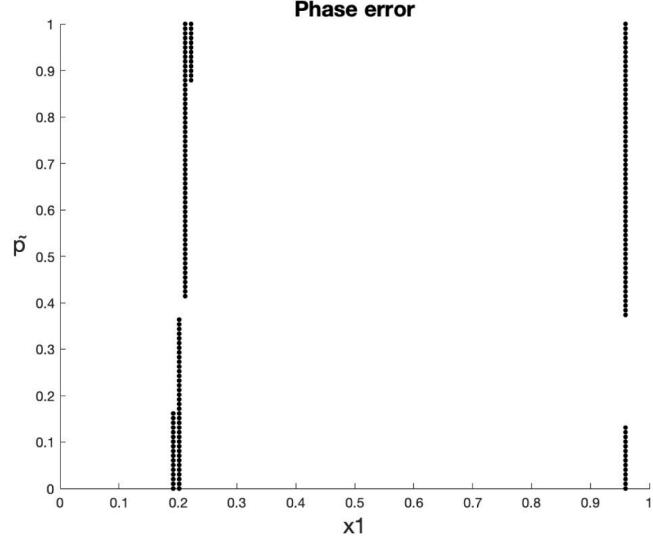
(c) Without a sparse-grid guide.

Fig. 16. Loss of the PINN training.

(a) Without hierachal priority.



(b) With hierachal priority.



(c) Without a sparse-grid guide.

Fig. 17. Phase errors of the equation-residual space.

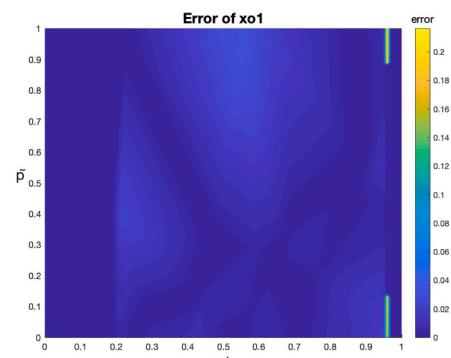
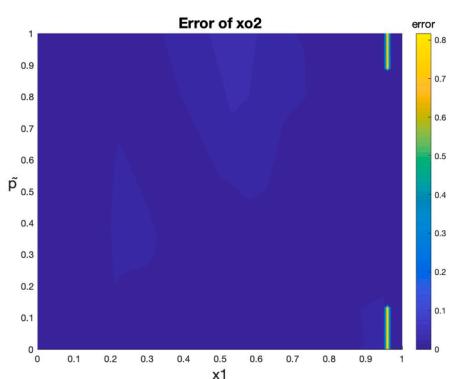
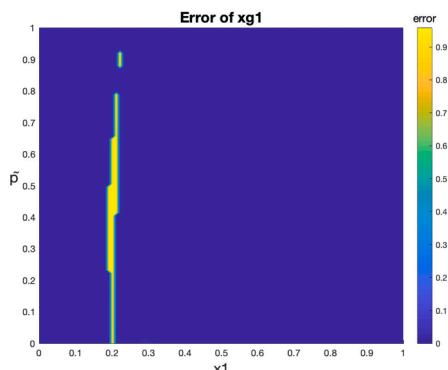
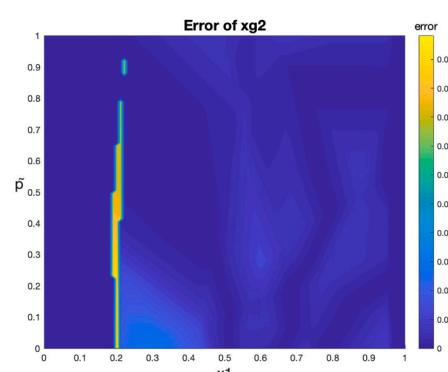
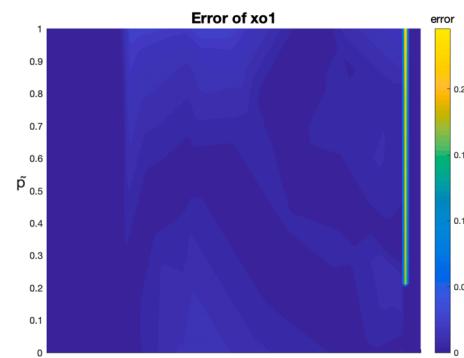
(a) Error of x^* 's.(b) Error of x^{*2} .(c) Error of $x_{\#1}$.(d) Error of $x_{\#2}$.

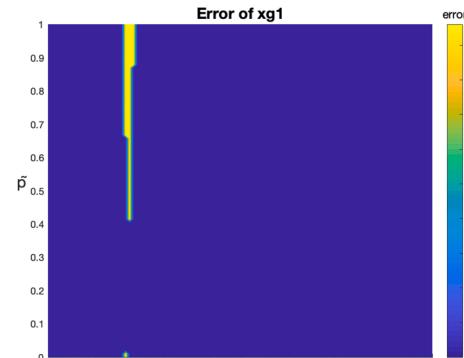
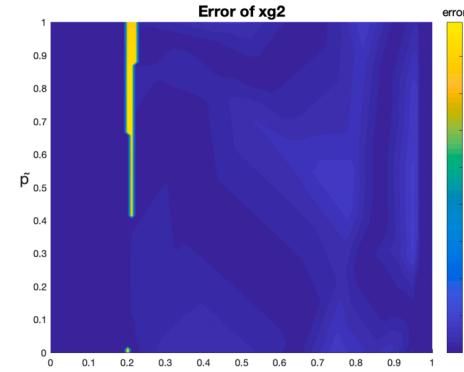
Fig. 18. Errors of the phase compositions for the experiment without hierarchical priority.



(a)



(b)

Error of $x_{\#1}$ 

(c)

Fig. 19. Errors of the phase compositions for the experiment with hierarchical priority.

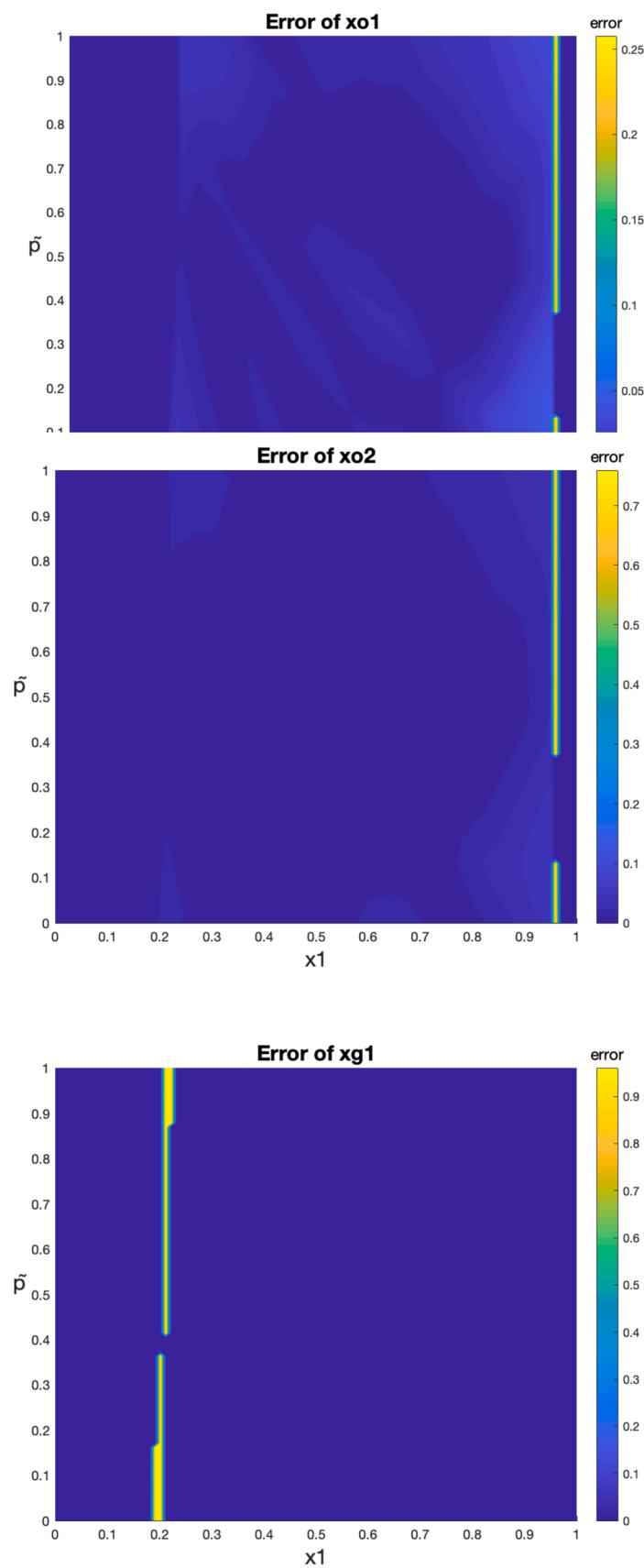


Fig. 20. Errors of the phase compositions for the experiment without a sparse-grid guide.

6. Conclusion

By leveraging sparse grid technology, the performance bottleneck of flash calculations is shifted from the online stage to the offline stage. The PINN surrogate helps alleviate the performance bottleneck in the offline stage, although its accuracy may not be as good as the sparse grid surrogate. To increase accuracy, more collocation points are required in the training, which can become another performance bottleneck. However, the sparse-grid guided PINN training approach allows for achieving the required accuracy using fewer collocation points, at the expense of constructing a sparse grid before each epoch. Fortunately, the time complexity of sparse grid construction is less than or at least comparable to that of the normal training. Considering that the number of collocation points in the sparse-grid guided PINN training is significantly smaller than in normal training, the sparse-grid guided approach achieves similar accuracy with much lower time complexity.

Furthermore, the experiments also demonstrate that when a sufficient number of collocation points are available, both the sparse-grid guided PINN training and normal training can achieve comparable accuracy. In such cases, the application of sparse-grid guided training may not be necessary.

Another observation from the experiments is that the constructed sparse grids remain the same for some consecutive epochs, suggesting that it may not be necessary to reconstruct the sparse grid for every epoch. This indicates potential opportunities to further reduce the computational cost of sparse grid construction. In future work, a criterion could be developed to determine when to reconstruct the sparse grid.

It is important to emphasize that while the sparse-grid guided PINN training is currently applied specifically to flash calculations, its applicability extends to other fields governed by physical equations. The main differences lie in the definition of residuals within the residual space, while the underlying philosophy of the algorithm remains largely unchanged. Exploring the application of this method in other domains is a promising avenue for future research.

CRediT authorship contribution statement

Yuanqing Wu: Conceptualization, Methodology, Software, Investigation, Writing – original draft. **Shuyu Sun:** Visualization, Investigation, Validation, Supervision, Writing – review & editing.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Yuanqing Wu reports financial support was provided by Natural Science Foundation of Shenzhen.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

- [1] B. Becker, B. Guo, I. Buntic, B. Flemisch, R. Helmig, An Adaptive Hybrid Vertical Equilibrium/Full-Dimensional Model for Compositional Multiphase Flow, *Water Resources Research* 58 (1) (2022), e2021WR030990.
- [2] Y. Li, H. Yang, S. Sun, Fully implicit two-phase VT-flash compositional flow simulation enhanced by multilayer nonlinear elimination, *J. Comput. Phys.* 449 (2022), 110790.
- [3] Y. Wu, S. Sun, Equivalence of two models in single-phase multicomponent flow simulations, *Comput. Mathem. Appl.* 71 (6) (2016) 1303–1316.
- [4] Y. Wu, S. Sun, A field-based general framework to simulate fluids in parallel and the framework's application to a matrix acidization simulation, *PLoS ONE* 17 (2) (2022), e0261134.
- [5] L.C. Young, Compositional reservoir simulation: a review, *SPE J.* 27 (05) (2022) 2746–2792.
- [6] F. Chen, S. Luo, S. Wang, H. Nasrabadi, A generalized machine learning-assisted phase-equilibrium calculation model for shale reservoirs, *Fluid Phase Equilib.* 558 (2022), 113423.
- [7] J. Qu, T. Faney, J.C. de Hemptinne, S. Yousef, P. Gallinari, PTFlash: a vectorized and parallel deep learning framework for two-phase flash calculation, *Fuel* 331 (2023), 125603.
- [8] S. Wang, N. Sobecki, D. Ding, L. Zhu, Y.S. Wu, Accelerating and stabilizing the vapor-liquid equilibrium (VLE) calculation in compositional simulation of unconventional reservoirs using deep learning based flash calculation, *Fuel* 253 (2019) 209–219.
- [9] Wu Y. Parallel reservoir simulations with sparse grid techniques and applications to wormhole propagation (Doctoral dissertation). 2015.
- [10] Y. Wu, Z. Chen, The application of high-dimensional sparse grids in flash calculations: from theory to realisation, *Fluid Phase Equilib.* 464 (2018) 22–31.
- [11] Y. Wu, C. Kowitz, S. Sun, A. Salama, Speeding up the flash calculations in two-phase compositional flow simulations—the application of sparse grids, *J. Comput. Phys.* 285 (2015) 88–99.
- [12] Y. Wu, M. Ye, A parallel sparse grid construction algorithm based on the shared memory architecture and its application to flash calculations, *Comput. Mathem. Appl.* 77 (8) (2019) 2114–2129.
- [13] Y. Wu, S. Sun, Removing the performance bottleneck of pressure–temperature flash calculations during both the online and offline stages by using physics-informed neural networks, *Phys. Fluids* 35 (4) (2023), 043326.
- [14] S. Cai, Z. Wang, S. Wang, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks for heat transfer problems, *J. Heat Transfer*. 143 (6) (2021).
- [15] Z. Mao, A.D. Jagtap, G.E. Karniadakis, Physics-informed neural networks for high-speed flows, *Comput. Methods Appl. Mech. Eng.* 360 (2020), 112789.
- [16] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707.
- [17] T. Smejkal, J. Mikyska, Unified presentation and comparison of various formulations of the phase stability and phase equilibrium calculation problems, *Fluid Phase Equilib.* 476 (2018) 61–88.
- [18] T. Smejkal, J. Mikyska, J. Kukal, Comparison of modern heuristics on solving the phase stability testing problem, *Disc. Contin. Dynam. Syst.-S* 14 (3) (2021) 1161.
- [19] H. Zhao, H. Jing, Z. Fang, H. Yu, Flash calculation using successive substitution accelerated by the general dominant eigenvalue method in reduced-variable space: comparison and new insights, *SPE J.* 25 (06) (2020) 3332–3348.
- [20] A. Katharopoulos, F. Fleuret, Not all samples are created equal: deep learning with importance sampling, *Int. conf. machine learn.* (2018) 2525–2534.
- [21] M.A. Nabian, R.J. Gladstone, H. Meidani, Efficient training of physics-informed neural networks via importance sampling, *Comput.-Aided Civil Infrastr. Eng.* 36 (8) (2021) 962–977.
- [22] S. Das, S. Tesfamariam, State-of-the-art Review of Design of Experiments for Physics-Informed Deep Learning, arXiv preprint, 2022. [arXiv:2202.06416](https://arxiv.org/abs/2202.06416).
- [23] M. Griebel, Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences, *Computing* 61 (1998) 151–179.
- [24] Z. Tao, Y. Jiang, Y. Cheng, An adaptive high-order piecewise polynomial based sparse grid collocation method with applications, *J. Comput. Phys.* 433 (2021), 109770.
- [25] W. Xue, Y. Wang, Z. Chen, H. Liu, An integrated model with stable numerical methods for fractured underground gas storage, *J. Clean. Prod.* 393 (2023), 136268.
- [26] L. Datta, A Survey on Activation Functions and Their Relation with Xavier and He Normal Initialization, arXiv preprint, 2020. [arXiv:2004.06632](https://arxiv.org/abs/2004.06632).
- [27] A. Henkes, H. Wessels, R. Mahnken, Physics informed neural networks for continuum micromechanics, *Comput. Methods Appl. Mech. Eng.* 393 (2022), 114790.
- [28] S. Rezaei, A. Harandi, A. Moeineddin, B.X. Xu, S. Reese, A mixed formulation for physics-informed neural networks as a potential solver for engineering problems in heterogeneous domains: comparison with finite element method, *Comput. Methods Appl. Mech. Eng.* 401 (2022), 115616.