Journal of Manufacturing Systems 71 (2023) 298–308

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Error homogenization in physics-informed neural networks for modeling in manufacturing

Clayton Cooper, Jianjing Zhang, Robert X. Gao*

Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH 44106, USA

ARTICLEINFO

Keywords:
Physics-informed neural network
Ensemble
Partial differential equation
Error homogenization Allen-Cahn equation

ABSTRACT

Physics-informed neural networks (PINNs) have demonstrated effectiveness in solving partial differential equations (PDEs) associated with manufacturing scenarios, due to their physically interpretable training logic. One limitation has been that PINNs often exhibit heterogenous error maps and high-error "hot spots" throughout the solution domain, which reduce not only the solutions' accuracy but also their overall consistency with the physical laws. This study addresses this limitation by presenting an efficient and error-aware PINN ensembling technique for error homogenization in solving manufacturing problems. Specifically, a PINN is first established by constraining its training process through a manufacturing specific PDE and corresponding boundary conditions to ensure physical consistency. Next, the loss landscape in the neighborhood of three PINNs trained with varying network parameter initialization is sampled to generate a PINN ensemble. Finally, the outputs of the ensemble members are combined through an inverse error-weighted average to yield the prediction of the PDE solution. Evaluation using the Allen-Cahn PDE, which describes phase separation in the solidification of metallic alloys, shows that the developed method reduces the average prediction error by 63% and error standard deviation by 30% across the solution space, demonstrating its effectiveness for PINN error reduction and homogenization. Additionally, the method has also demonstrated 96% reduction in the computational time as compared to conventional ensembling methods.

1. Introduction

To facilitate digitization as an essential element of the cyber-physical manufacturing systems, accurate and robust solution of partial differential equations (PDEs) that govern manufacturing systems and processes such as heat transfer, chemical diffusion, and material flow, etc., has attracted increasing attention [1–3]. Representative PDEs shown in Fig. 1 illustrate the broad applicability of PDEs on shop floors, as well as how the prediction of upstream PDE solutions cascades into subsequent processes and the final product throughout the manufacturing system. This implies that ensuring accurate solution of PDEs at each of the production steps can lead to reduced error propagation in the entire system, more robust production planning, a priori identification of noncompliant product and production bottlenecks, simultaneous optimization of product quality and production time, and, ultimately, avoidance of facility-wide operational disruptions caused by part rework [4]. Given this motivation, recent scientific literature has shown a variety of PDE modeling approaches that support new manufacturing paradigms such as digital twinning [5].

Standard PDE solution methods rely on analytical or numerical techniques such as finite element analysis (FEA) and computational fluid dynamics (CFD), which exhibit good agreement with experimental observations and a robust connection to the underlying process physics [6, 7]. However, these benefits come at the cost of a generally high computational complexity as additional dimensions or nodes are considered, inability in handling real-world process disturbances, and spatiotemporal resolution constrained by the meshing scheme and/or element size [8,9]. Concurrent with the rise of industrial big data and increased computing power, data-driven approaches to solving PDEs have been investigated to overcome these drawbacks [10, 11].

Data-driven modeling refers to the adaptive learning of input-output relationships using system observations through neural networks and/or other methods [14,15]. Compared to analytical methods, data-driven methods have several advantages. First, they have lower computational complexity as a function of the dimensionality and number of nodes when compared to the typically exponential growth of analytical solutions. This makes them an attractive choice for solving higher-dimensional PDEs where a high resolution is desired [16].

Second, due to their mesh-free nature, data-driven models enable unique inference at all points of the PDE solution domain rather than being constrained to a grid of simulation nodes and the associated interpolation between them. This allows data-driven methods to predict future PDE behavior without simulating all time steps up to the time of interest. Third, data-driven models can consider process disturbances as model input and learn to express

* Corresponding author.
E-mail address: robert.gao@case.edu (R.X. Gao).

https://doi.org/10.1016/j.jmsy.2023.09.013

Received 2 July 2023; Received in revised form 26 August 2023; Accepted 23 September 2023
Available online 29 September 2023

the effects of disturbances in the model output, making them responsive to real-world manufacturing scenarios.

Numerous studies on data-driven techniques have been conducted in recent years, demonstrating their success in manufacturing contexts [17–19]. Nonetheless, one limitation of these techniques is the general lack of compliance with physical domain laws. This is because neither their training processes nor prediction logic (e.g., forward computation in a neural network) is constrained by physics, which leads to the possibility of incorrect predictions outside of the training data range arising from spurious prediction logic obtained during training [20]. A recent example is illustrated in [21], wherein a data-driven COVID-19 detection algorithm learned to predict infection based on benign geometric features of chest radiographs rather than lung tissue damage, resulting in poor performance on never-before-seen data and potentially compromised diagnostic outcomes. Issues such as these are becoming increasingly relevant given that explainable artificial intelligence, which encapsulates data-driven modeling, has been codified in law [22,23], and is becoming commonplace within scientific literature in order to verify prediction logic and synthesize new knowledge about the systems being characterized [24,25]. In order for data-driven methods to be accepted by the manufacturing community as trustworthy, it must be assured that domain knowledge is respected in the solutions provided by data-driven models.

A promising approach to enforcing domain knowledge in PDE solutions is the physics-informed neural network (PINN) [26]. Initially developed to solve nonlinear PDEs, PINNs penalize network predictions that violate constraints on the PDE and correspondingly guide the training procedure towards model parameters which minimize deviation between PDE constraints and PINN output. This physically interpretable training logic of PINNs has demonstrated success in solving PDEs and is gaining traction in the manufacturing community as a reliable tool for physics-driven modeling [27]. Although competing machine learning-based methods of solving PDEs have been published, they lack either explicit connection to the PDE physics or require training data which may not be available. For example, neural process (NP) has been proposed for PDE modeling given its high modeling accuracy and built-in uncertainty quantification [28]. This approach is useful in situations wherein simulated or experimental data is available to train the NP encoder and decoder to accurately model the system under study. However, this may not always be the case due to, e.g., high computational cost of simulation, difficulty in observing the phenomenon of interest (such as alloy solidification in casting operations as discussed in this study), or the phenomenon of interest occurring infrequently and yielding little data for training. In comparison, such data availability issues do not apply to PINNs since PINNs model known system dynamics based on a priori physical understanding, with or without additional system observations.

PINN-based PDE solutions include the modeling of hyperelastic stress, which dictates the fabrication of elastomers and biological tissues, with 0.09 Pa root mean squared error as compared to 0.18 Pa using a non-PINN method [29]. A PINN was used to identify material parameters, including Young's modulus and Poisson's ratio, for a structural health monitoring application based on material deformation PDEs, which successfully identified the parameters of interest within 1% error [30]. Meanwhile Chen et al. successfully predicted natural convection temperature evolution using a PINN with less than 0.1% error, with implications for heat treatment prediction as well as chemical processing in manufacturing [31]. Liao et al. modeled the temperature evolution of an additively manufactured part using a PINN and observed 47 K root mean squared error as compared to experimental data [32].

While the PINNs from these studies exhibit good average error over the solution space, it has also been observed that these PINNs exhibit regions of high local error, or "hot spots," throughout the solution domain [32–34]. This means that moving a small distance away from the solution point in a space or time may cause large changes in the error magnitude and thus reflects an unstable model output that diminishes user confidence in the PINN predictions. Often underreported in literature, such instability is also difficult to compensate for using typical error correction techniques such as spatiotemporally-aware bias terms. As such, PINNs remain prone to heterogenous error distributions and methods of alleviating the error hot spots are needed.

A promising method of accomplishing this homogenization is ensembling methods such as model averaging reduce the prediction variance (a proxy measure of error heterogeneity) and, for nonlinear models, the average error

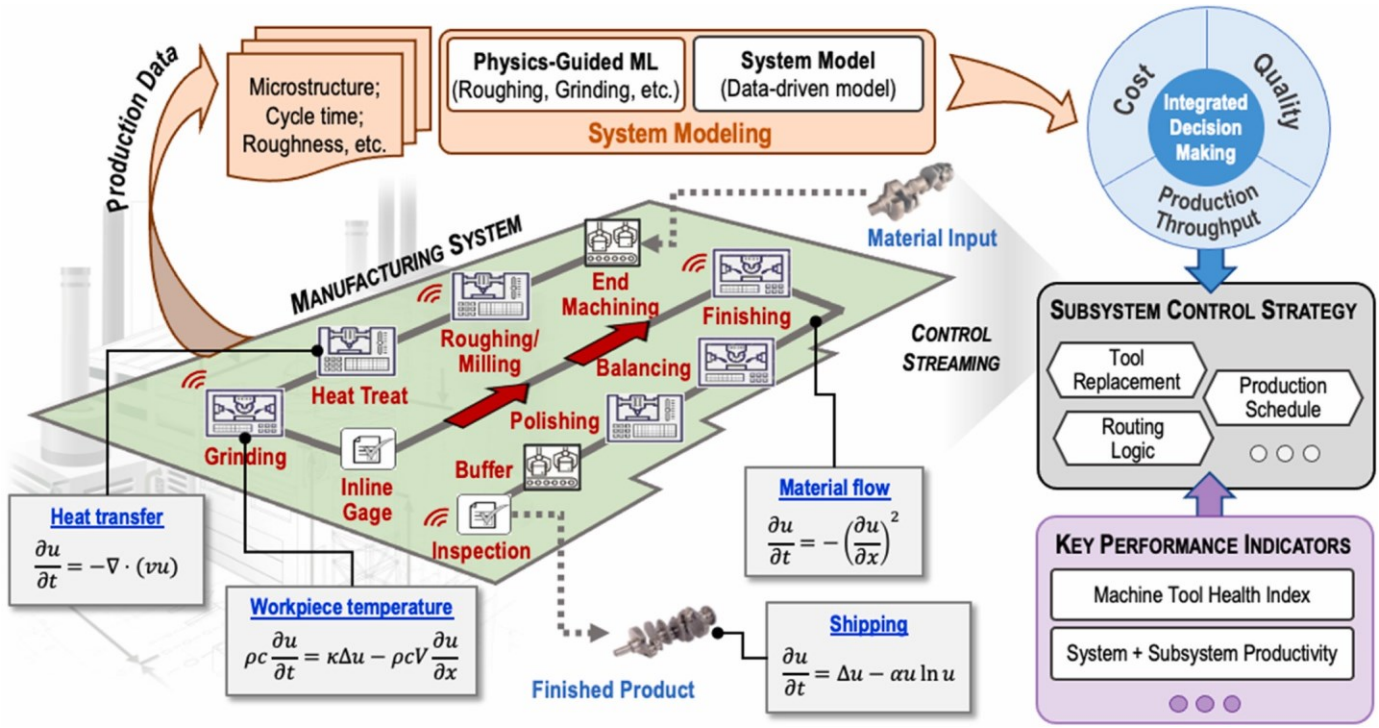

Fig. 1. Common partial differential equations in a CPS where solution accuracy of each process affects error propagation throughout the system and the final product. PDEs selected from [3,12,13].

[35,36]. However, training a set of individual neural networks to achieve ensembling benefits can be computationally expensive, as neural networks often require several minutes, hours, or days to train [37]. The presented study

is motivated by this observation and proposes an efficient ensemble-based method of PINN error homogenization. Specifically, three PINNs are first trained to define a loss landscape, from which new PINNs are sampled rather than trained from scratch. A sampling procedure is then defined based on the loss landscape to allow for the control of anticipated ensemble member error, and correspondingly, the accuracy of the ensemble as a whole. The PDE solution prediction is subsequently found as a weighted average of ensemble member outputs, with the weights being inversely proportional to each member's loss. The expected reduction in the error heterogeneity is shown to be dependent on the average pairwise correlation between ensemble member outputs. An overview of the proposed method is shown in Fig. 2.

The presented study has made the following contributions:

1) Established an efficient ensembling method for PINNs, which yields as many unique models as desired after training only three networks using the standard training process,
2) Developed a systematic and error-aware PINN sampling procedure based on the loss landscape topography with a tunable parameter controlling the expected error of the newly sampled PINNs,
3) Demonstrated the error homogenization of the proposed PDE solving method by comparing the ensemble error distribution with that of a single PINN using the Allen-Cahn equation for modeling alloy solidification in casting as an example.

The rest of this paper is organized as follows: Section 2 introduces the theoretical background of PINNs, the proposed computationally- efficient ensembling method through loss landscape sampling, and the weighted aggregation of ensemble members. Section 3 describes the experimental validation of the proposed method using the Allen-Cahn equation and outlines the PINN training procedure. Section 4 presents and discusses the results of the proposed sampling and ensembling method in solving the Allen-Cahn equation and demonstrates the pairwise correlation between sampled PINNs that yielded the ensemble performance as shown. Conclusions from this study and future research directions are discussed in Section 5.

2. PINN theoretical background

The structure of a PINN for the case of a one-dimensional PDE, $u(t,x)$, is shown in Fig. 3. Spatiotemporal coordinates t and x are passed as inputs to a fully-connected neural network, f, which outputs a proposed PDE solution $\tilde{u}$ as a function of the inputs and the network weight and bias parameters, ϑ. The forward calculation is represented in (1):

$$\tilde{u}(t,x;f) = f(t,x;\vartheta) \forall t \in [t_{\downarrow}, t_{\uparrow}], x \in [x_{\downarrow}, x_{\uparrow}] \quad (1)$$

Where the ↓ and ↑ subscripts denote lower and upper bounds, respectively.

A PINN is considered physics-informed because of its multi-term loss function, $\mathbf{L}$:

$$\mathbf{L}(u) = \omega_u \mathbf{L}_u(u) + \omega_b \mathbf{L}_b(u) + \omega_d \mathbf{L}_d(u) \quad (2)$$

where $\mathbf{L}_u$ is proportional to the PINN's deviation from the PDE

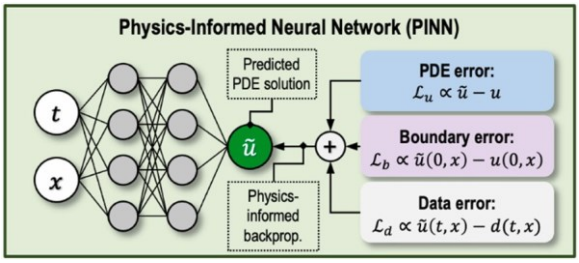

Fig. 3. Physics-informed neural network architecture for 1-D PDE with spatiotemporal input coordinates t and x and ground-truth solution u. The network output $\tilde{u}$ replicates the ground-truth solution at each coordinate pair. Adherence to the PDE is assured during the model training process through physics-informed backpropagation, with the overall network loss comprised of terms that consider deviations from the PDE equations, boundary conditions, and experimental data, respectively.

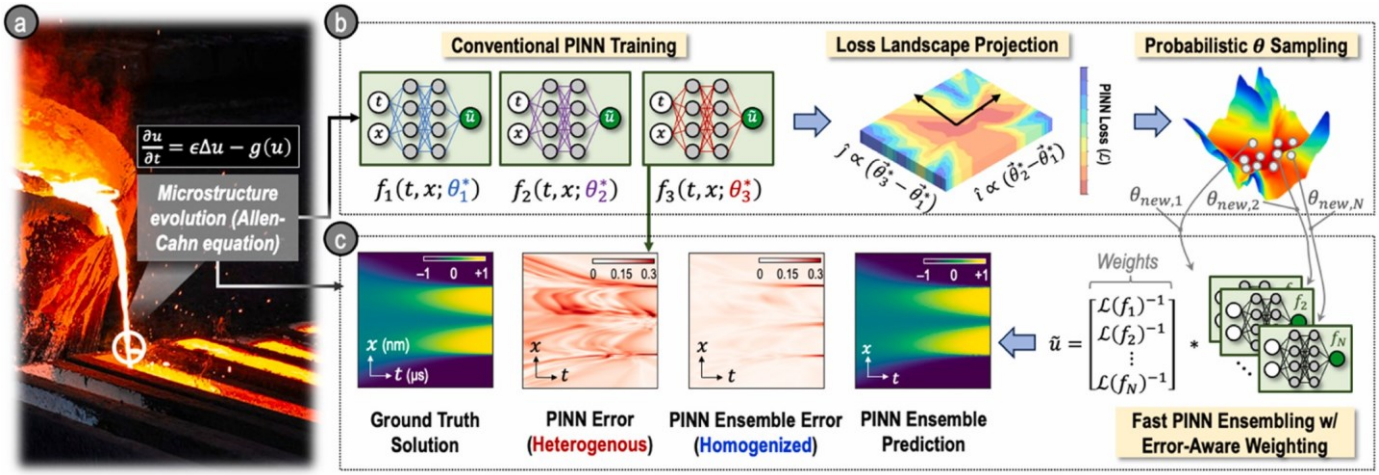

Fig. 2. Overview of proposed error homogenization methodology: a) illustration of metal alloy casting governed by the Allen-Cahn PDE; b) loss landscape visualization and sampling method to retrieve PINN parameters (ϑ_{new}) after training only three models using conventional backpropagation; c) inverse error-weighted ensembling using ϑ_{new} sampled from loss surface and comparison of single-PINN and PINN ensemble solutions to ground-truth solution found by numerical simulation.
Casting image adapted from [38,39].

constraints at non-boundary or "colocation" points, $\mathbf{L}_b$ is proportional to the PINN's deviation from the PDE constraints on the spatial and temporal boundaries, $\mathbf{L}_d$ is proportional to the difference between the PINN prediction and any experimentally observed datapoints, and each ω_* is a weighting term. The losses are expressed as:

$$\mathbf{L}_u(u) = \frac{1}{n_u} \sum_{i=1}^{N_u} |\mathbf{F}(u(t_i,x_i;f))| \quad (3)$$

$$\mathbf{L}_b(u) = \frac{1}{n_b} \sum_{i=1}^{N_b} |\mathbf{B}(u(t_i,x_i)) - \mathbf{B}(u(t_i,x_i;f))| \quad (4)$$

$$\frac{1}{} \sum^{}$$

$\mathbf{L}_d(u) = \frac{1}{n_d}\sum_{i=1}^{n_d} |d_i - u(t_i, x_i; f)|$ (5) where **F** is the nonlinear differential operator for the PDE [38], **B** represents the boundary conditions, d_i is the i^{th} experimental observation of PDE-governed system, and n_* is the number of interior, boundary, or observed points in the PINN training batch. When used as the scalar loss term during backpropagation, the sum of these terms enforce that the PINN-predicted PDE solution respects the governing physics at the colocation points, on the boundaries, and with respect to experimental observations [26], [40].

In general, the PINN is trained by sampling a training set of colocation and boundary (t,x), calculating **L** for the sampled locations plus any available experimental observations, and repeatedly backpropagating any error to iteratively find the parameters which minimize the loss on the training data, ϑ^*:

$$\vartheta^* = \underset{\vartheta}{\operatorname{argmin}} \mathbf{L}(\tilde{u}) \quad (6)$$

Using PINNs to solve PDEs has several advantages over numerical methods and conventional supervised machine learning approaches:

1) PINNs can be expanded to accommodate higher-dimensional PDEs with moderate increase in computational complexity. It has been shown that the complexity of numerical methods grows exponentially while PINN complexity growth is sub-exponential [41], making PINNs relatively lightweight from the perspectives of runtime and parameter storage requirements as compared to numerical approaches.
2) PINNs do not require experimental data for training. Because the derivative terms in the loss function are found using automatic differentiation, $\mathbf{L}_u$ and $\mathbf{L}_b$ can be obtained without observational data [42]. This is an advantage of PINNs over conventional supervised learning paradigms in situations where data is scarce or difficult to simulate.
3) PINNs have physically interpretable training logic. Because PINN weights are adjusted to minimize violations of the governing PDE and boundary conditions, the training process of PINN is more physically interpretable as compared to purely data-driven supervised learning, which relies on observational data alone. This contributes to improved trustworthiness of the obtained prediction logic of PINNs.
4) PINN gradients are exact. The gradients found using automatic differentiation are exact within machine precision [43]. This contrasts with numerical methods, where finite difference approximations of derivatives are used that introduce error into the model and may degrade the quality of the final solution.
5) PINNs have infinitesimal spatiotemporal resolution. Unlike numerical methods, PINNs are mesh-free. As a result, the (t,x) locations queried for training and inference can be more flexibly selected in the input domain and do not have to be equally spaced. This enables PINNs to generate PDE solutions in a continuous manner both spatially and temporally, while numerical solutions are constrained to interpolating on the node mesh generated prior to executing the simulation.

While standalone PINNs exhibit these advantages and have demonstrated good performance as reported in recent studies [44,45], one noted limitation is the variability in their error maps. This can render a PINN unreliable as moving a small distance in space or time may cause unstable model behavior and error to increase by orders of magnitude, as shown for a single PINN in Fig. 2c. To overcome this limitation, it is desired to generate an ensemble of PINNs and combine their unique predictions together to homogenize the prediction error.

3. Proposed ensembling and PINN error homogenization scheme

Conventional generation of neural network ensembles is time- consuming since each model must be retrained from scratch. For instance, an N-member ensemble wherein each member is trained for 10,000 epochs will require 10,000N total epochs of training. If each epoch is performed in 0.03 s as in this study, each additional ensemble member adds 5 min to the ensemble training time. Thus, even a modestly-sized 50-member ensemble will require over 4 h to train. Considering this substantial computational cost, an efficient ensembling method based on the concept of loss landscape sampling is developed, which can efficiently yield unique PINNs for the cost of only three training sessions.

3.1. Loss landscape definition

Consider conventionally trained PINNs f_1, f_2, and f_3 with vectorized parameters $\vec{\theta}^*_1$, $\vec{\theta}^*_2$ and $\vec{\theta}^*_3$, respectively. The vectorization is done by concatenating all weights and biases into a single vector. When projected into a two-dimensional space, these three points in parameter space define a plane where each in-plane point represents a unique set of parameters, $\vec{\vartheta}_{new}$. Such a projection is represented as:

$$\vec{\theta}_{new}(\alpha,\beta) = \vec{\theta}^*_1 + \alpha\cdot\hat{i} + \beta\cdot\hat{j} \quad (7)$$

where $\hat{i} = \vec{\vartheta}_2 - \vec{\vartheta}_1$, $\hat{j} = (\vec{\vartheta}_3 - \vec{\vartheta}_1) - \cos(i, (\vec{\vartheta}_3 - \vec{\vartheta}_1))\cdot i$, and

$\cos(a,b)$ is the cosine of the angle between a and b [45]. The cosine term appears in the equation for $\hat{j}$ in order to remove the component of $\vec{\vartheta}_3 - \vec{\vartheta}_1$ which is parallel to $\hat{i}$, thus ensuing that the $\hat{i}$ and $\hat{j}$ bases are orthogonal and relationship between each (α,β) and $\vec{\vartheta}_{new}$ is bijective. Once $\vec{\vartheta}_{new}$ is found, it can be substituted directly into the weight and bias terms of f for making predictions.

3.2. PINN parameter sampling from loss landscape

With the loss landscape defined for the three-PINN neighborhood, low-loss regions of **S** can be identified and ϑ_{new} sampled from them, with each new set of parameters corresponding to a new PINN to be ensembled. Loss surface **S** is first transformed into 2-D empirical PDF p defined for region $\alpha \in [\alpha_\downarrow, \alpha_\uparrow] \times \beta \in [\beta_\downarrow, \beta_\uparrow]$ via (8):

$$p(\alpha,\beta;\mathbf{S}) = \frac{e^{C[\max(\mathbf{S}) - \mathbf{S}(\alpha,\beta) - \min(\mathbf{S}))]}}{\int_{\alpha_\downarrow}^{\alpha_\uparrow}\int_{\beta_\downarrow}^{\beta_\uparrow} e^{C[\max(\mathbf{S}) - \mathbf{S}(\alpha,\beta) - \min(\mathbf{S}))]}\, d\alpha\, d\beta} \quad (8)$$

where $c > 0$ is a concentration factor inversely proportional to the spread of the sampled $\vec{\vartheta}_{new}$ away from $\vec{\theta}^*_1$, $\vec{\theta}^*_2$ and $\vec{\theta}^*_3$, and the denominator converges so long as **S** is finite within the integrated region. A large value of c will cause the newly sampled parameter vectors to be close to trained PINNs in the (α,β) plane and the opposite is true for small values of c. This is because a large c exponentially scales the peaks of the PDF at each $\vec{\vartheta}^*$ to be much larger than the surrounding probabilities, making it more likely to draw samples from the peak locations. Additionally, since a small c encourages samples to be taken further away from the loss minima and these samples correspond to high loss, c is inversely proportional to the expected ensemble loss.

Parameter vectors are sampled from p by vectorizing the PDF to find the cumulative density function (CDF) and using the inverse CDF to convert N samples, $k \sim U(0,1)$, to (α,β) locations and the corresponding ϑ_{new}. Since the

parameters of each of the N ensemble PINNs are sampled in constant time, the serialized ensemble generation time is $O(1) * N = O(N)$. The same is true of the ensemble prediction complexity since each of the N PINNs has $O(1)$ computation time. The PINN sampling and computation processes each require approximately 1 µs on a 2.6 GHz CPU, meaning that the runtimes of the sampling and prediction process for the whole ensemble are each on the order of N µs. If the sampling and prediction processes are performed in parallel on n_c cores, the complexity of each process is reduced to $O(\max[N/n_c, 1])$, and so $O(1)$, or 1 µs, ensemble generation time and run time is achieved for the ensemble so long as $n_c \geq N$.

3.3. Inverse error-weighted ensembling

Once N new sets of parameters are sampled from the loss surface, the corresponding PINNs are assembled into an ensemble, Q. The PDE solution outputted by Q is the weighted average of each of the constituent PINNs, with weights inversely proportional to the model's loss as shown in (9):

$$\tilde{u}(t,x;Q) = Q(t,x;f_1,f_2,\ldots,f_N) = \sum_{i=1}^{N} f_i(t,x;\vartheta_{new,i}) \cdot w_i$$

$$w_i = \frac{p(\alpha^i,\beta^i;\mathbf{S})}{\sum_i p(\alpha_i,\beta_i;\mathbf{S})} \quad (9)$$

This weighting ensures that $\sum w_i = 1$, and an ensemble member with a smaller loss value will have a larger w_i than the PINN with a larger loss value. As a result, the most accurate PINNs have more influence over the final result to keep the prediction error low while still allowing the other PINNs to contribute and homogenize the error via their unique prediction distributions.

3.4. Reduction in PINN error heterogeneity due to ensembling

Assuming that the error maps of the PINNs in the ensemble have mean pairwise Pearson correlation coefficient $\bar{\rho} \geq 0$ [46,47], the expected standard deviation of the ensemble error is estimated as:

$$\sigma_Q = \sqrt{\sum_{i,j} w_i w_j C_{ij}}$$

$$C_{ij} = \rho \cdot \sigma_{\tilde{u}(t,x;f_i)} \cdot \sigma_{\tilde{u}(t,x;f_j)} \quad (10)$$

where $\sigma_\star$ denotes the standard deviation of $\star$ and C_{ij} is the estimated covariance of the error of ensemble members f_i and f_j as a fraction (ρ) of the maximum covariance, $\sigma_{\tilde{u}(t,x;f_i)} \cdot \sigma_{\tilde{u}(t,x;f_j)}$ [48]. For PINNs sampled from the low-loss regions of $\mathbf{S}$, the assumption of positive error correlation is deemed reasonable given the close proximity of the sampled parameter vectors and thus close proximity of the model outputs [46]. In the worst possible scenario, all sampled PINNs are exactly correlated, leading to $\bar{\rho} = 1$ and $\sigma_Q = \sigma_{f_i} = \sigma_{f_j} \forall i,j$, i.e., no reduction in error standard deviation via ensembling. However as $\bar{\rho} \to 0$, $\sigma_Q \to 0$, indicating that lower average pairwise correlation represents lower variability in the prediction error. As the standard deviation of the prediction error is decreased, the ensemble's error homogeneity is increased.

4. Experimental evaluation

To evaluate the proposed ensembling method for PINN error homogenization, metal casting is chosen as a representative case study. With over $110B in economic impact and nearly 500k employees in the United States alone, metal casting represents one of the world's largest and most diverse manufacturing sectors [49]. It is also one of the most critical foundations of manufacturing operations, since nearly 90% of all manufactured durable goods rely on cast parts as either functional components or in-process tools [50]. Given this reach, rigorous quality standards for castings are obligatory in order to ensure product conformity and end-user safety. Of cast parts' myriad quality dimensions, microstructural composition is the most critical since it determines several key performance characteristics such as tensile strength, creep resistance, corrosion susceptibility, and fatigue life, as predicted by the process-structure-property-performance (PSPP) paradigm [51–53]. Therefore, if cast parts' microstructure can be accurately characterized, conformity to design can be verified, part performance measures can be predicted accurately and reliably without the need for large-scale destructive testing, saving material resources.

Phase separation in casting operations, which has a strong influence on mechanical properties, is governed by the Allen-Cahn equation. Depicted in Fig. 4, the equation describes reaction-diffusion systems, including phase separation in multi-component alloy systems with order-disorder transitions. It has become a tool to model and study phase transitions and interfacial dynamics in materials science and is often used to model alloy composition at the molecular level and thus serves as a basis for digital twinning of bulk deformation processes in manufacturing, such as forging and casting [54].

The general form of the Allen-Cahn equation is expressed as:

$$\frac{\partial u}{\partial t} = \epsilon \Delta u - g(u) \quad (11)$$

where ϵ is a constant, Δ is the Laplace operator denoting the sum of all unmixed 2nd-order partial derivatives in the spatial coordinates, and g is an arbitrary function of u. This study uses the 1-D form of (11) with periodic Dirichlet and Neumann boundary conditions as studied in [26] to maintain comparability with PINN literature:

$$\frac{\partial u}{\partial t} = 0.0001 \frac{\partial^2 u}{\partial x^2} - (5u + 5u^3); x \in [-1,1], t \in [0,1]$$

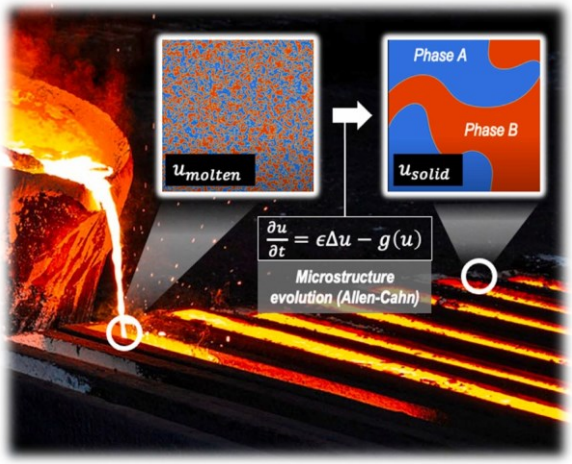

Fig. 4. Allen-Cahn equation-governed microstructural evolution of a cast steel alloy at a phase boundary. The initial phase distribution immediately after pouring is highly mixed and disorderly but quickly separates into larger and well-defined regions of each solid phase according to the equation as time progresses.
Adapted from [37, 55].

$$u(0,x) = x^2\cos(\pi x); u(t,-1) = u(t,1); \frac{\partial}{\partial t}(t,-1) = \frac{\partial}{\partial t}(t,1) \quad (12)$$

To find the ground truth solution to (12), i.e., u, the PDE is simulated using the Chebfun package for MATLAB [56]. The solution surface is depicted for a μm-ns scale in Fig. 5 and serves as the PDE solution which the singleton PINN and PINN ensemble will aim to replicate as $\tilde{u}(t,x;f)$ and $\tilde{u}(t,x;Q)$, respectively. The value of u is an indicator variable reflecting the material state as either phase A or phase B, and is thus unitless. Intermediate values indicate finely mixed phases as shown as u_{molten} in Fig. 4, however these values disappear as time progresses and the material solidifies, thereby reflecting the instantaneous stepwise changes between phases in a solid material. As $t\to\infty$, u becomes comprised of entirely step functions with discrete domain $u \in \{-1,1\}$ and the intermediate values disappear.

4.1. PINN architecture and training

The PINN architecture in this study comes from [26] and is defined as a 5-layer fully-connected neural network with 100 neurons per hidden layer, hyperbolic tangent activation, Adam optimization, 0.001 initial learning rate, and loss function as defined in (2) with $\omega_u = \omega_b = 0.5$ and $\omega_d = 0$ since experimental data was not used for training given the difficulties of observing phase separation in real-time. The network receives (t,x) tuples from the domain defined in (12) as input and outputs $\tilde{u}(t,x;f)$ as described in (1).

The PINN training procedure is initialized by sampling 1000 boundary and 50,000 colocation points from the (t,x) domain using the Latin Hypercube sampling strategy [57]. The sampled points are then split into 16 stratified batches of equal size and used to perform mini-batch gradient descent for 10,000 epochs. Batching was used in order to broaden the "valleys" of the loss minima found during training, leading to better model generalization and larger low-loss regions from which to sample ensemble members compared to full-batch training [58]. At the end of training, the model parameters which yielded the lowest loss are stored as ϑ_i^* for the i^{th} trained PINN.

The three PINNs needed to explore the loss surface are each trained with random parameter initializations and unique training datasets. Training each model took approximately 5 min on a single-threaded 2.6 GHz CPU.

4.2. Error quantification and homogenization measurement

To measure the overall accuracies of the singleton PINN and PINN ensemble, the physics-informed loss as defined in (2) is evaluated on a validation dataset comprised of 500 boundary and 25,000 colocation points of the PDE input domain which are disjoint from the training data. A lower loss value corresponds to a less error-prone prediction of the PDE solution.

Error homogeneity for a given $\tilde{u}$, denoted ξ, is quantified as the standard deviation of the error at each of the validation coordinates as described in (13):

$$\xi(\star) = \mathrm{E}_v(\mathbf{L}(\tilde{u}(t,x;\star) - \mathrm{E}_v[\mathbf{L}(\tilde{u}(t,x;\star))]) \quad (13)$$

where v represents the validation dataset. A lower value of ξ indicates a predictive model with greater error homogeneity and fewer error hot spots in the solution domain.

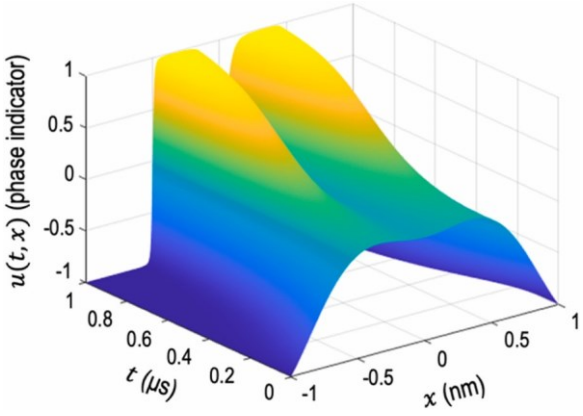

Fig. 5. Simulated 1-D Allen-Cahn equation solution. Vertical axis represents instantaneous phase denoted by − 1 (phase A) or + 1 (phase B) with intermediate values occurring during phase transition. x axis denotes spatial location, t axis denotes time since phase separation began.

5. Results and discussion

5.1. PINN training and loss landscape visualization

The training and validation losses for PINNs f_1, f_2, and f_3 are shown in Fig. 6. The average loss ($\mathbf{L}$) decay over time, as calculated across all training data, indicates that the physics-informed model training procedure asymptotically guided the PINN parameters ϑ_1, ϑ_2, and ϑ_3 to achieve loss minima and thus good agreement with the PDE constraints. The similarities of the colocation and boundary losses indicate good replication of the PDE solution over the entire solution domain, and the similarities between the training and validation losses shows the PINNs are not overfit.

Note that the noise in the loss curves is due to the mini-batch training. Since the model parameters are updated each time a batch is processed, some parameter updates may be backpropagated that bolster the prediction accuracy on one batch while worsening the accuracy of others. If the average loss across all mini-batches at the end of an epoch is greater than the previous epoch, the loss shown in the plot will increase. The effects of this phenomenon are negligible, however, since the overall loss trend is asymptotic, and the training procedure is developed to capture the best network parameters achieved at any point during training. The mini-batches therefore help ensure broad loss minima as discussed in [58] with little degradation of final model accuracy.

Fig. 7 depicts the training loss landscape $\mathbf{S}$ near the three trained networks. The deep craters represent ϑ_1^*, ϑ_2^*, and ϑ_3^* and every other point represents a linear combination of these three parameters. The loss surface is notably smooth throughout the (α,β) domain, with loss rapidly increasing in the vicinity of the trained models and even further near the edges of the plot. The effects of this loss topography on the PINN parameter sampling procedure are investigated relative to the sampling concentration factor in the following section.

5.2. Effect of sampling parameters on parameter selection and ensemble loss

The relationship between concentration factor and sampling density is depicted for three values of c in Fig. 8a-c. It is seen that a small concentration factor deemphasizes the loss value when sampling (α,β) and leads to samples being taken all over the loss landscape, including from high-loss regions. This corresponds to a large distance between points, as well as a large distance from each point to the nearest trained PINN, as shown in Fig. 8d. Conversely, as c is increased, the peaks of the PDF near the trained models grows exponentially taller according to (8) and the sampled (α,β) move closer to the loss minima generated by the trained PINNs. As $c\to\infty$, the sampled models will grow increasingly closer to the trained models, as quantified in Fig. 8d.

To simultaneously observe the relationship between c, N, and $\mathbf{L}$, the ensemble's average loss is evaluated on a grid of (c,N) points and visualized in Fig. 9. Relatively high loss values are observed for low concentration factors and model counts, shown by the blue region in the

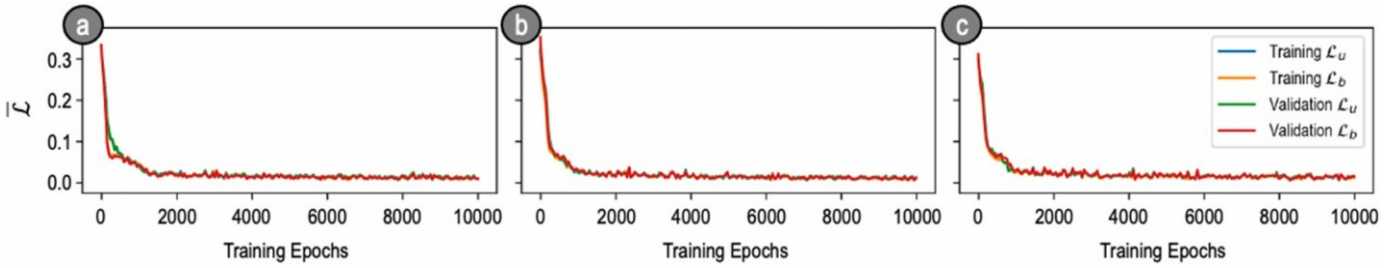

Fig. 6. Training and validation loss curves for: a) f_1, b) f_2, and c) f_3 showing that each conventionally trained PINN converged during training.

lower left corner. While increasing N generally reduces the ensemble loss, a low concentration factor consistently degrades ensemble performance, as prominently observed in the small blue region between N = 20 and N = 40 on the left axis where c is small. This evidence suggests

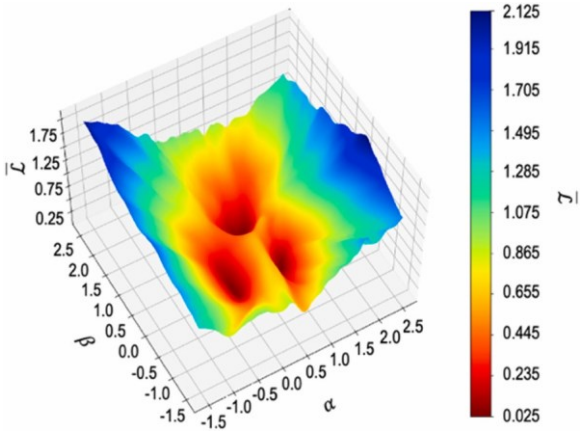

Fig. 7. Loss landscape near ϑ_1^*, ϑ_2^*, and ϑ_3^* evaluated on an (α,β) grid. The height and corresponding color denote the average loss value of the resulting PINN at each (α,β) point. The deep craters shown in dark red represent the three trained models, with f_1 at (0,0), f_2 at (1,0) and f_3 at approximately (0.6, 1.

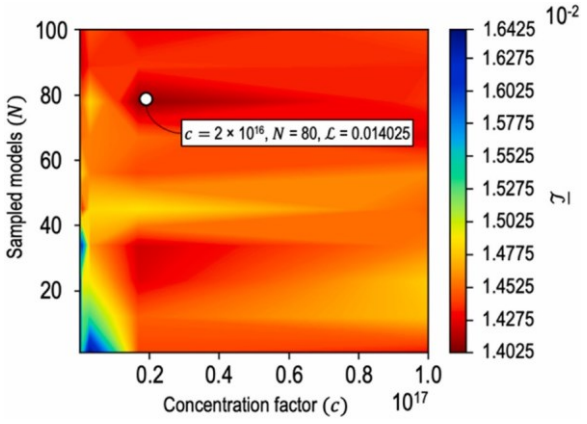

Fig. 9. Loss value as a function of concentration factor and number of models sampled for ensemble.

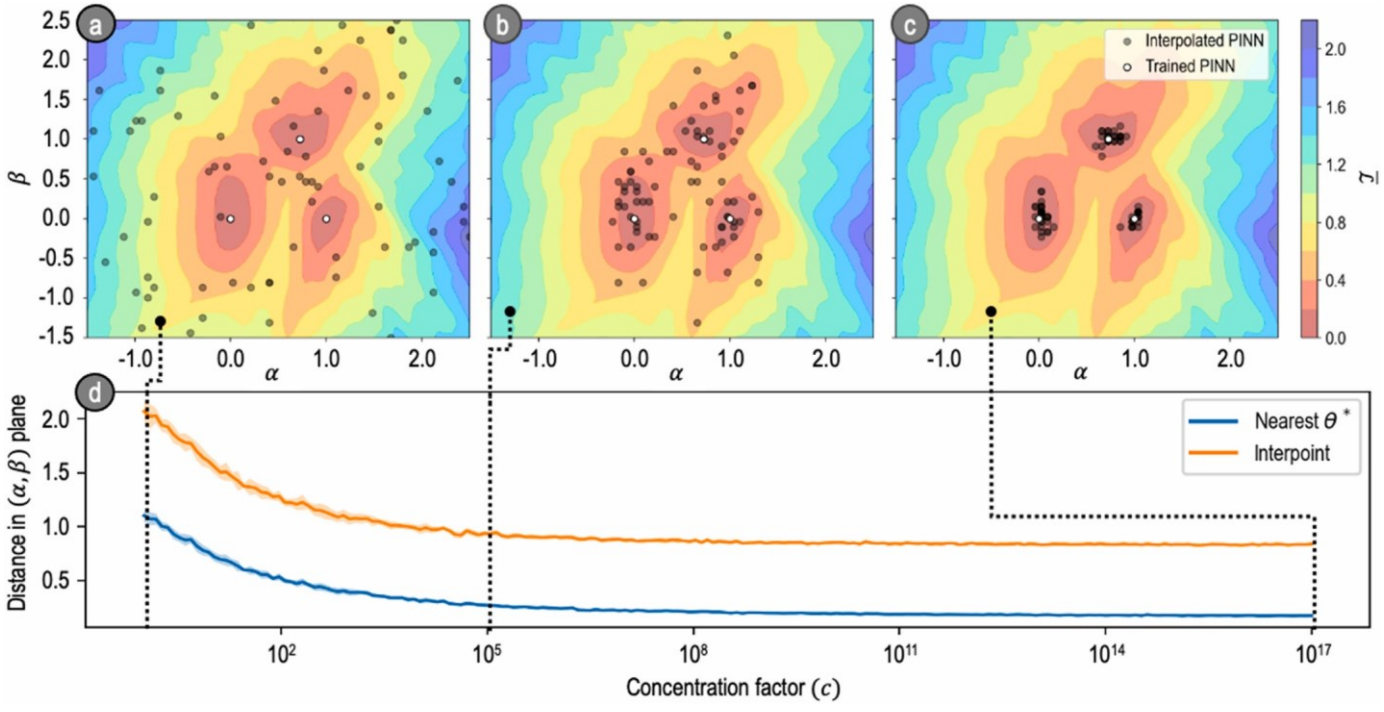

Fig. 8. Effect of concentration factor on PINN parameter sampling shown at a) $c = 1$, b) $c = 10^5$, and c) $c = 10^{17}$. Depicted in d) is average distance of sampled points to nearest conventionally trained model and to all other points over range of c; shading indicates one standard deviation.

the theoretical inverse relationship between c and $\mathbf{L}_{\_\_}$ holds in practice. For the given search space, the loss minimum occurs at ($c = 2 \times 10^{16}$, $N = 80$), where $\mathbf{L}_{\_\_} = 0.014025$. These ensembling parameters are used to produce the ideal ensemble solution for comparison to the singleton PINN.

5.3. Comparison of predicted PDE solutions

Predicted PDE solutions and error maps for a common validation dataset are shown in Fig. 10 for f_1, which exhibited the lowest post- training validation loss, and the PINN ensemble with optimal c and N as identified above. Visual inspection indicates that the ensemble has better error homogeneity than the singleton as shown by the reduced number of error hot spots in the solution space and decreased intensity of coloration in these regions. Furthermore, there is less error "waviness" in the ensemble solution, especially for $t < 0.8$ and $x \in [-0.5, 0.5]$. This leads to more stable predictions and error estimates in this region compared to the single model.

While the ensemble generally outperforms the singleton in terms of error presence and intensity, both approaches yield high error at $x = \pm 0.5$ for $t > 0.8$, which is where the PDE solution becomes "sharp" as depicted in Fig. 5. This inability to accurately model sharp regions of PDEs is a well-documented shortcoming of PINNs and is inherited by the ensemble [59]. This behavior is believed to be caused by inadequately fine sampling along steep gradients of the PDE solution, which impedes the network's ability to accurately reconstruct such regions. As such, the error hot spots in the sharp regions of the PDE solution may be mitigated through the use of adaptive sampling methods when training the ensemble PINNs, such as residual-based adaptive refinement wherein sharp regions of the PDE solution are automatically detected via their loss values and subsequently oversampled to improve PINN performance in these regions [60,61].

The average prediction loss and loss standard deviation agrees with visual inspection. As depicted in Table 1, the ensemble prediction has 62.7% lower loss and 29.6% lower loss standard deviation than the single model, indicating that the ensemble is both more accurate than the singleton and has greater error homogeneity. An analysis of the ensemble follows in the subsequent section to better understand the source of these performance improvements.

To observe the robustness of the proposed method's stochastic parameter selection for the ensemble's PINN members, 100 ensembles are sampled independently from the loss landscape defined in Fig. 7 using the proposed loss landscape sampling technique with c and N parameters as identified in Fig. 9 and assessed on the PDE solution domain. The distributions of average error, $\mathbf{L}_{\_\_}$, and error standard deviation, ξ, are shown in Fig. 11 along with the global means of each parameter across the 100 ensembles.

Table 1
Error comparison between singleton PINN and PINN ensemble.

Model	Average Error ($\mathbf{L}_{\_\_}$)	Error Std. Deviation (ξ)
Single PINN	0.0379	0.0287
PINN Ensemble	0.0141	0.0202
Absolute Difference	-0.0238	-0.0085
Relative Difference	-62.7%	-29.6%

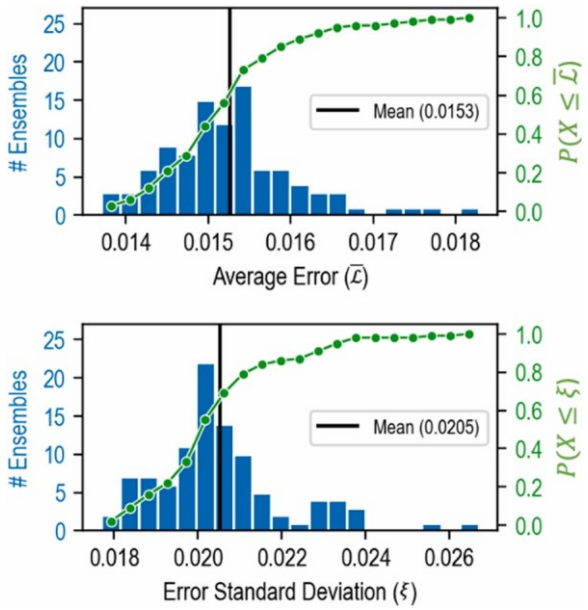

Fig. 11. Histograms of ensemble average errors and error standard deviations for 100 ensembles sampled from loss surface defined by ϑ_1^*, ϑ_2^*, and ϑ_3^* with $c = 2 \times 10^{16}$ and $N = 80$. The green line plot depicts the cumulative probability of sampling an ensemble with an $\mathbf{L}_{\_\_}$ or ξ value (X) lower than the corresponding value along the horizontal axis. As shown by the cumulative probability, there is a 62% chance that an ensemble will have a below-average $\mathbf{L}_{\_\_}$ and a 76% chance that an ensemble will have a below-average ξ.

Both distributions are approximately centered about their means as evidenced by the clear peaks located near the vertical black lines. Additionally, both distributions are light-tailed as well, with less than a 1% chance of sampling an ensemble with $\overline{\mathcal{L}}$ or ξ more than + 3 standard deviations from the mean. The means themselves are in good agreement with the results shown in Table 1. The $\mathbf{L}_{\_\_}$ global mean is observed to be 0.0153, which is 8.5% higher than single ensemble's $\mathbf{L}_{\_\_}$ of 0.0141. Meanwhile the ξ global mean is observed to be 0.0205, or 1.4% higher than the single ensemble's ξ. These discrepancies are to be expected since, as evidenced by the cumulative probabilities shown in green, there is a greater than 60% chance that an ensemble drawn at random will have a below-average error or standard

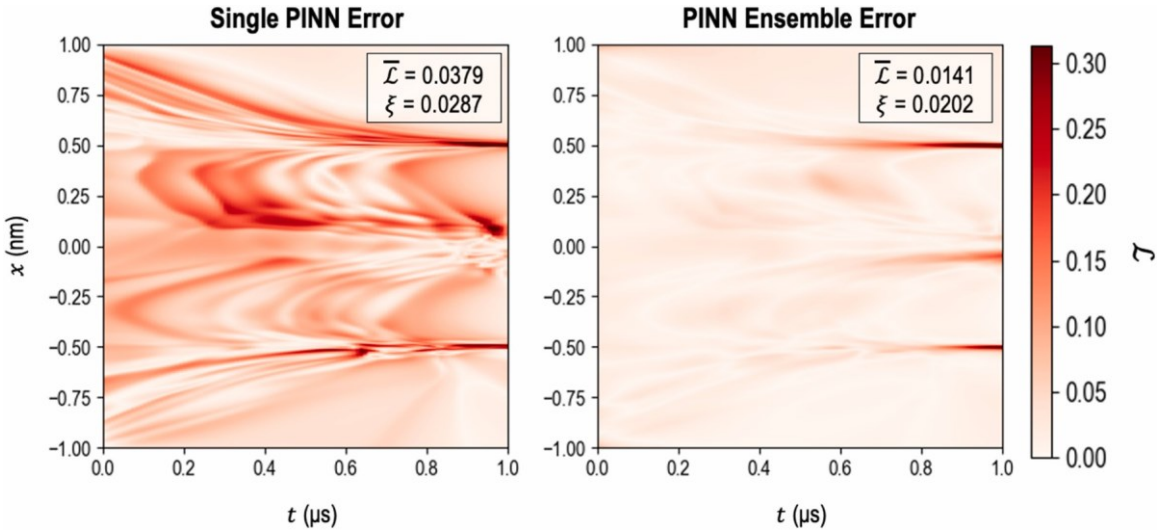

Fig. 10. Error comparison of single PINN and PINN ensemble solving Allen-Cahn equation specified in (12).

deviation. Overall, these results suggest the ensemble member sampling procedure is probabilistically unlikely to yield detrimental outliers and instead prefers models with **L** ___ and ξ at or below the global means of these metrics.

5.4. Ensemble member error map correlation

Average pairwise correlations of the ensemble member error maps are plotted as a histogram in Fig. 12. As desired in (10), the mean correlation is far from unity, at 0.06, indicating that the models do not reinforce one another's error and, when aggregated, tend towards zero error heterogeneity. Additionally, the maximum pairwise correlation is only 0.24, indicating very little error reinforcement in even the worse- case scenario amongst the sampled models. 15 of the 80 models (18.8%) have negative average pairwise correlations, indicating error correction capabilities via error map anticorrelation.

Overall, the small pairwise correlations of the ensemble members support the reduction in mean loss and loss standard deviation. It should be noted that the experimental error standard deviation (71% of single PINN) does not match the theoretical error standard deviation based on the average pairwise correlation (6% of single PINN). This is likely because (10) assumes independent PINNs whereas the PINNs used in this study are not independent due to the clustered sampling procedure used, which preferred models nearer the trained PINNs. Further investigation of this relationship is identified as a promising area for future work in fast neural network ensembling.

6. Conclusions

The presented study aims to fill an existing research gap for solving PDEs in manufacturing using PINNs: heterogenous error distributions over the PDE solution space. By leveraging the mathematical properties of ensembling, an efficient ensembling scheme and model parameter sampling procedure have been developed to homogenize PINN prediction errors while also reducing the mean error across the solution space. At the cost of only three trained models, unique PINNs have been generated to form an ensemble, and the expected error of which is controlled by the concentration factor of the loss surface-derived PDF.

It is shown that the expected reduction in error variance is inversely proportional to the average pairwise correlation between ensemble member outputs. Verification of the method using the Allen-Cahn PDE shows strong error homogenization and reduction capabilities, which reduced error standard deviation by 29.6% and average error by 62.7%. This is due to near-zero average pairwise correlation within the 80-member ensemble used for prediction, which leads to increased homogeneity of the ensemble error compared to a singleton PINN. Further, the ensemble is built using only 15 min of model train time (3 trained models), compared to 400 min if conventional ensembling is used (80 trained models), representing a 96% reduction in the computational time.

The key findings of this study are summarized as:

1) The error hot-spot phenomenon exhibited by singleton PINNs has been effectively reduced through aggregative ensembling. Specifically, the average error and error standard deviation have shown to be reduced by 62.7% and 29.6%, respectively, by a PINN ensemble as compared to a single conventionally trained PINN.
2) Loss landscape sampling of PINN parameters is an effective and computationally efficient replacement for conventional PINN training, with the ensemble formation time being reduced by 96% in this study when compared to conventional network training.
3) The developed inverse error-weighted parameter sampling and model aggregation methods enable the ensemble error to be controlled during the ensemble formation process. This control been demonstrated by observing the relationship between ensemble error, concentration factor, and the number of ensemble members sampled.

These findings are envisioned to help improve the trustworthiness and feasibility of PINNs as a modeling tool for manufacturing processes and systems that are characterized by the increasingly interconnected operations. By improving the prediction error homogeneity as evidenced by the ensemble's reduced error standard deviation and increasing prediction accuracy, more reliable modeling predictions with greater stability in each of the spatiotemporal directions can be expected. Such increased model reliability contributes to greater confidence in the modeling outcome, more robust production forecasts, and reduced waste of material and human capital arising from the unexpected machine downtime and related noncompliant quality issues.

Given the broad applicability of PDEs in manufacturing systems, which encompasses topics ranging from material flow to machining, heat treatment, finishing, inspection, and shipping, the developed model ensemble method is envisioned to especially benefit the communities consisting of diverse small- and medium-sized manufacturers (SMMs) where each of the SMMs is specialized in a particular manufacturing process. The result is a unified modeling solution across the spectrum of diverse manufacturing processes that fosters collaboration across SMMs to collectively improve material and energy efficiencies and optimize operation in a coordinated, global manner. The synergy will enhance SMMs' productivity and promote the development of smart and connected communities in which they reside, promoting continued growth and sustainability.

Future research will adapt the ensembling method to reduce prediction error in the sharp regions of the PDE solution using location- aware aggregation methods. Research will also be undertaken to better characterize

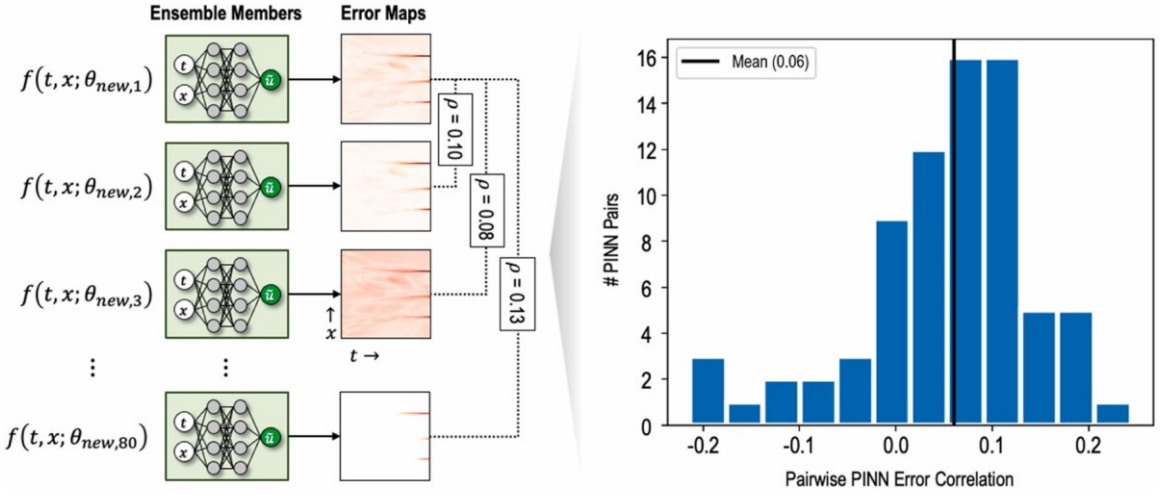

Fig. 12. Average inter-model prediction error correlation for ensemble of 80 PINNs sampled from loss landscape.

the effect of experimental observations on the ensemble accuracy and homogenization capacity. The root causes of the PINN error hot spots outside of the sharp PDE regions will also be comprehensively investigated from theoretical and experimental perspectives such that improved PINN training procedures can be developed and the hot spots can be avoided altogether. Finally, acceleration of the PINN training procedure will be investigated so that the initial model training cost is reduced.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

C. Cooper acknowledges support from the National Science Foundation (NSF) Graduate Research Fellowship under grant No. 1937968. R. Gao and J. Zhang acknowledge support for this research by the NSF under grant No. CNS-2125460, CMMI-2040288, and the NSF Engineering Research Center (ERC, HAMMER) under grant No. EEC- 2133630.

References

[1] Xu X, Lu Y, Vogel-Heuser B, Wang L. Industry 4.0 and Industry 5.0—Inception, conception and perception. J Manuf Syst 2021;vol. 61:530–5. https://doi.org/10.1016/j.jmsy.2021.10.006.

[2] Leng J, Wang D, Shen W, Li X, Liu Q, Chen X. Digital twins-based smart manufacturing system design in Industry 4.0: a review. J Manuf Syst 2021;vol. 60: 119–37. https://doi.org/10.1016/j.jmsy.2021.05.011.

[3] Van Den Berg R, Lefeber E, Rooda K. Modeling and control of a manufacturing flow line using partial differential equations. IEEE Trans Contr Syst Technol 2008;vol. 16(1):130–6. https://doi.org/10.1109/TCST.2007.903085.

[4] Sahoo S, Lo C-Y. Smart manufacturing powered by recent technological advancements: a review. J Manuf Syst . 2022;vol. 64:236–50. https://doi.org/ 10.1016/j.jmsy.2022.06.008.

[5] Shojaeinasab A, et al. Intelligent manufacturing execution systems: a systematic review. J Manuf Syst . 2022;vol. 62:503–22. https://doi.org/10.1016/j. jmsy.2022.01.004.

[6] Knapp GL, Coleman J, Rolchigo M, Stoyanov M, Plotkowski A. Calibrating uncertain parameters in melt pool simulations of additive manufacturing. Comput Mater Sci 2023;vol. 218:111904. https://doi.org/10.1016/j. commatsci.2022.111904.

[7] Liao S, Golgoon A, Mozaffar M, Cao J. Efficient GPU-accelerated thermomechanical solver for residual stress prediction in additive manufacturing. Comput Mech 2023. https://doi.org/10.1007/s00466-023-02273-3.

[8] Gawade V, Singh V, Guo W "Grace". Leveraging simulated and empirical data- driven insight to supervised-learning for porosity prediction in laser metal deposition. J Manuf Syst 2022;vol. 62:875–85. https://doi.org/10.1016/j. jmsy.2021.07.013.

[9] Michopoulos JG, Iliopoulos AP, Steuben JC, Birnbaum AJ, Lambrakos SG. On the multiphysics modeling challenges for metal additive manufacturing processes. Addit Manuf 2018;vol. 22:784–99. https://doi.org/10.1016/j.addma.2018.06.019.

[10] Guo S, et al. Machine learning for metal additive manufacturing: towards a physics-informed data-driven paradigm. J Manuf Syst . 2022;vol. 62:145–63. https://doi.org/10.1016/j.jmsy.2021.11.003.

[11] Hu S, Li X, He H, Cui S, Parashar M. Big data for cyber-physical systems. IEEE Trans Big Data 2020;vol. 6(4):606–8. https://doi.org/10.1109/TBDATA.2020.3033101.

[12] Wang Z, Li Y, Yu T, Zhao J, Wen PH. Prediction of 3D grinding temperature field based on meshless method considering infinite element. Int J Adv Manuf Technol 2019;vol. 100(9–12):3067–84. https://doi.org/10.1007/s00170-018-2801-4.

[13] Armbruster D. The production planning problem: clearing functions, variable lead times, delay equations and partial differential equations. In: Armbruster D, Kempf KG, editors. Decision policies for production networks. London: Springer London; 2012. p. 289–302. https://doi.org/10.1007/978-0-85729-644-3_12.

[14] Wang J, Ma Y, Zhang L, Gao RX, Wu D. Deep learning for smart manufacturing: methods and applications. J Manuf Syst . 2018;vol. 48:144–56. https://doi.org/10.1016/j.jmsy.2018.01.003.

[15] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;vol. 521(7553):436–44. https://doi.org/10.1038/nature14539.

[16] Berner J, Grohs P, Kutyniok G, Petersen P. The modern mathematics of deep learning. First ed.. In: Grohs, Kutyniok G, editors. Mathematical aspects of deep learning. Cambridge University Press; 2022. p. 1–111. https://doi.org/10.1017/ 9781009025096.002. 1st ed

[17] Gao RX, Wang L, Helu M, Teti R. Big data analytics for smart factories of the future. CIRP Ann 2020:1–25. https://doi.org/10.1016/j.cirp.2020.05.002.

[18] Ding H, Gao RX, Isaksson AJ, Landers RG, Parisini T, Yuan Y. State of AI-based monitoring in smart manufacturing and introduction to focused section. IEEE/ ASME Trans Mechatron 2020;vol. 25(5):2143–54. https://doi.org/10.1109/ TMECH.2020.3022983.

[19] Arinez JF, Chang Q, Gao RX, Xu C, Zhang J. Artificial Intelligence in advanced manufacturing: current status and future outlook. J Manuf Sci Eng 2020;vol. 142 (11):110804. https://doi.org/10.1115/1.4047855.

[20] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. hysics-informed machine learning. Nat Rev Phys 2021;vol. 3(6):422–40. https://doi.org/10.1038/ s42254-021-00314-5.

[21] DeGrave AJ, Janizek JD, Lee S-I. AI for radiographic COVID-19 detection selects shortcuts over signal. Nat Mach Intell 2021. https://doi.org/10.1038/s42256-021- 00338-7.

[22] Goodman B, Flaxman S. European union regulations on algorithmic decision- making and a 'Right to explanation. AI Mag 2017;vol. 38(3):50–7.

[23] European Commission, 2021. Proposal for a regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain union legislative acts. 2021.

[24] Ahmed I, Jeon G, Piccialli F. From artificial intelligence to explainable artificial intelligence in industry 4.0: a survey on what, how, and where. IEEE Trans Ind Inf 2022. https://doi.org/10.1109/TII.2022.3146552. pp. 1–1.

[25] Cooper C, Zhang J, Huang J, Bennett J, Cao J, Gao RX. Tensile strength prediction in directed energy deposition through physics-informed machine learning and Shapley additive explanations. J Mater Process Technol 2023;vol. 315:117908. https://doi.org/10.1016/j.jmatprotec.2023.117908.

[26] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys . 2019;vol. 378:686–707. https://doi. org/10.1016/j.jcp.2018.10.045.

[27] S. Cuomo, V.S. di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, 2022. Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next. arXiv, Jun. 07, 2022. Accessed: Nov. 17, 2022. [Online]. Available: 〈http://arxiv.org/abs/2201.05624〉 .

[28] Wang Y, Wang K, Cai W, Yue X. NP-ODE: neural process aided ordinary differential equations for uncertainty quantification of finite element analysis. IISE Trans 2021; vol. 54(3):211–26. https://doi.org/10.1080/24725854.2021.1891485.

[29] D.W. Abueidda, S. Koric, E. Guleryuz, and N.A. Sobh, 2022. "Enhanced physics- informed neural networks for hyperelasticity." arXiv, May 24, 2022. Accessed: Jun. 07, 2022. [Online]. Available: 〈http://arxiv.org/abs/2205.14148〉 .

[30] D. Anton and H. Wessels, 2023. "Physics-Informed Neural Networks for Material Model Calibration from Full-Field Displacement Data." arXiv, Dec. 15, 2022. Accessed: Jan. 11, 2023. [Online]. Available: 〈http://arxiv.org/abs/2212.07723〉 .

[31] Chen W, Wang Q, Hesthaven JS, Zhang C. Physics-informed machine learning for reduced-order modeling of nonlinear problems. J Comput Phys 2021;vol. 446: 110666. https://doi.org/10.1016/j.jcp.2021.110666.

[32] Liao S, Xue T, Jeong J, Webster S, Ehmann K, Cao J. Hybrid thermal modeling of additive manufacturing processes using physics-informed neural networks for temperature prediction and parameter identification. Comput Mech 2023. https://doi.org/10.1007/s00466-022-02257-9.

[33] S. Basir, 2002. "Investigating and Mitigating Failure Modes in Physics-informed Neural Networks (PINNs)." arXiv, Sep. 20, 2022. Accessed: Sep. 27, 2022. [Online]. Available: 〈http://arxiv.org/abs/2209.09988〉 .

[34] A.S. Krishnapriyan, A. Gholami, S. Zhe, R.M. Kirby, and M.W. Mahoney, 2022. "Characterizing possible failure modes in physics-informed neural networks." arXiv, Nov. 11, 2021. Accessed: Dec. 07, 2022. [Online]. Available: 〈http://arxiv.org/abs/2109.01050〉 .

[35] Breiman L. Bagging predictors. Mach Learn 1996;vol. 24(2):123–40. https://doi.org/10.1007/BF00058655.

[36] J. Nixon, D. Tran, and B. Lakshminarayanan, 2020. "Why Aren't Bootstrapped Neural Networks Better?," in Proceedings of NeurIPS 2020, Vancouver, BC, Canada, 2020.

[37] Z. Fang, S. Wang, and P. Perdikaris, 2023. "Ensemble learning for Physics Informed Neural Networks: a Gradient Boosting approach." arXiv, Feb. 25, 2023. Accessed: Mar. 02, 2023. [Online]. Available: 〈http://arxiv.org/abs/2302.13143〉 .

[38] "Additional Casting Methods," Kovatch Castings Knowledge Base, Mar. 2019. 〈https://www.kovatchcastings.com/knowledge-base/additional-casting-methods/〉 (accessed Jun. 27, 2023).

[39] S. Fu˘cík and A. Kufner, 1980. Nonlinear differential equations. in Studies in applied mechanics, no. 2. Amsterdam; New York: New York: Elsevier Scientific Pub. Co.; distribution for the USA and Canada, Elsevier/North-Holland, 1980.

[40] Rojas R. The Backpropagation Algorithm. Neural Networks. Berlin, Heidelberg: Springer Berlin Heidelberg,; 1996. p. 149–82. https://doi.org/10.1007/978-3-642- 61068-4_7.

[41] P.A. Cioica-Licht, M. Hutzenthaler, and P.T. Werner, 2022. "Deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear partial differential equations." arXiv, May 28, 2022. Accessed: Jun. 07, 2022. [Online]. Available: 〈http://arxiv.org/abs/2205.14398〉 .

[42] Bartholomew-Biggs M, Brown S, Christianson B, Dixon L. Automatic differentiation of algorithms. J Comput Appl Math . 2000;vol. 124(1–2):171–90. https://doi.org/10.1016/S0377-0427(00)00422-2.

[43] Baydin AG, Pearlmutter BA, Radul AA, Siskind JM. Automatic differentiation in machine learning: a survey. J Mach Learn Res 2018;vol. 18(153):1–43.

[44] Sharma P, Chung WT, Akoush B, Ihme M. A review of physics-informed machine learning in fluid mechanics. Energies . 2023;vol. 16(5):2343. https://doi.org/ 10.3390/en16052343.

[45] Fernandez de la Mata F, Gij´ on A, Molina-Solana M, G´ omez-Romero J. Physics- ´ informed neural networks for data-driven simulation: advantages, limitations, and opportunities. Phys A: Stat Mech Appl 2023;vol. 610:128415. https://doi.org/10.1016/j.physa.2022.128415.

[46] T. Garipov, P. Izmailov, D. Podoprikhin, D. Vetrov, and A.G. Wilson, 2023. "Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs." arXiv, Oct. 30, 2018. Accessed: Jan. 11, 2023. [Online]. Available: 〈http://arxiv.org/abs/1802.10026〉 . [47] B. Everitt and A. Skrondal, 2021. The Cambridge dictionary of statistics. 2010. Accessed: Dec. 08, 2021. [Online]. Available: 〈http://www.books24×7.com/marc.asp?bookid=36106〉 .

[48] Schmelling M. Averaging correlated data. Phys Scr . 1995;vol. 51(6):676–9. https://doi.org/10.1088/0031-8949/51/6/002.

[49] U.S , 2019. Metalcasting Industry Impact on U.S. Jobs & The Economy - Economic Impact Table," American Foundry Society, 2019.

[50] U.S , 2019. Metalcasting Industry Impact on U.S. Jobs & The Economy - Economic Impact Report," American Foundry Society, 2019.

[51] Olson GB. Computational design of hierarchically structured materials. Science . 1997;vol. 277(5330):1237–42. https://doi.org/10.1126/science.277.5330.1237.

[52] Zhao M-C, Liu M, Song GL, Atrens A. Influence of microstructure on corrosion of As-cast ZE41. Adv Eng Mater . 2008;vol. 10(1–2):104–11. https://doi.org/10.1002/adem.200700246.

[53] R.W. Armstrong, "Hall-Petch Relationship: Use in Characterizing Properties of Aluminum and Aluminum Alloys," p. 30.

[54] Hussain S, Shah A, Ayub S, Ullah A. An approximate analytical solution of the Allen-Cahn equation using homotopy perturbation method and homotopy analysis method. Heliyon 2019;vol. 5(12):e03060. https://doi.org/10.1016/j.heliyon.2019.e03060.

[55] Nils Berglund, 2023. "Tool to create videos of particles or waves in different 2D domains." Apr. 30, 2023. Accessed: Jun. 27, 2023. [Online]. Available: 〈https://github.com/nilsberglund-orleans/YouTube-simulations〉 .

[56] Platte RB, Trefethen LN. Chebfun: a new kind of numerical computing," in progress in industrial mathematics at ECMI 2008. In: Fitt AD, Norbury J, Ockendon H, Wilson E, editors. in Mathematics in Industry, vol. 15. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010. p. 69–87. https://doi.org/10.1007/978-3-642-12110-4_5.

[57] Stein M. Large sample properties of simulations using latin hypercube sampling. Technometrics 1987;vol. 29(2):143–51. https://doi.org/10.1080/00401706.1987.10488205.

[58] N.S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P.T.P. Tang, 2021. "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima." Feb. 09, 2017. Accessed: Jun. 14, 2021. [Online]. Available: 〈http://arxiv. org/abs/1609.04836〉 .

[59] Mao Z, Jagtap AD, Karniadakis GE. Physics-informed neural networks for high- speed flows. Comput Methods Appl Mech Eng 2020;vol. 360:112789. https://doi.org/10.1016/j.cma.2019.112789.

[60] Z. Mao and X. Meng, 2023. "Physics-informed neural networks with residual/ gradient-based adaptive sampling methods for solving PDEs with sharp solutions." arXiv, Feb. 15, 2023. Accessed: Feb. 23, 2023. [Online]. Available: 〈http://arxiv. org/abs/2302.08035〉 .

[61] Lu L, Meng X, Mao Z, Karniadakis GE. DeepXDE: a deep learning library for solving differential equations. SIAM Rev 2021;vol. 63(1):208–28. https://doi.org/10.1137/19M1274067.