www.springer.com/41230
https://doi.org/10.1007/s41230-024-4016-7
Research & Development
Vol. 22 No. 2 March 2025
CHINA FOUNDRY

Temperature fields prediction for the casting process by a conditional diffusion model

***Jin-wu Kang, Jing-xi Zhu, and Qi-chao Zhao**
School of Materials Science and Engineering, Key Laboratory for Advanced Materials Processing Technology, Tsinghua University, Beijing 100084, China

Abstract: Deep learning has achieved great progress in image recognition, segmentation, semantic recognition and game theory. In this study, a latest deep learning model, a conditional diffusion model was adopted as a surrogate model to predict the heat transfer during the casting process instead of numerical simulation. The conditional diffusion model was established and trained with the geometry shapes, initial temperature fields and temperature fields at t_i as the condition and random noise sampled from standard normal distribution as the input. The output was the temperature field at t_{i+1}. Therefore, the temperature field at t_{i+1} can be predicted as the temperature field at t_i is known, and the continuous temperature fields of all the time steps can be predicted based on the initial temperature field of an arbitrary 2D geometry. A training set with 302 2D shapes and their simulated temperature fields at different time steps was established. The accuracy for the temperature field for a single time step reaches 97.7%, and that for continuous time steps reaches 69.1% with the main error actually existing in the sand mold. The effect of geometry shape and initial temperature field on the prediction accuracy was investigated, the former achieves better result than the latter because the former can identify casting, mold and chill by different colors in the input images. The diffusion model has proved the potential as a surrogate model for numerical simulation of the casting process.

Keywords: diffusion model; U-Net; casting; simulation; heat transfer

CLC numbers: TP391.9 **Document code:** A **Article ID:** 1672-6421(2025)02-139-12

1 Introduction

Numerical simulation is a popularly used analysis method to solve ordinary and partial differential equations (ODE, PDE) using numerical methods such as the finite element method (FEM), finite difference method (FDM), and finite volume method (FVM), boundary element method (BEM), etc. [1-5]. Numerical simulation requires the discretization of the CAD models into elements, which can be challenging for complex objects, the establishment of constitutional models, which can be hard for some unknown mechanisms behind phenomena, the solution of large scale equation sets, which usually takes a long time and may be hard to converge. In addition, the rapid development of deep learning (DL) method offers an alternative to replace or enhance numerical simulation. In these years, many kinds of DL models sprout, such as deep neural network (DNN), convolutional neural network (CNN), recurrent neural network (RNN), and generative adversarial networks (GANs), etc. These models have been widely used in image processing such as image segmentation, image recognition, etc. [6-11]. In 2022, Ho et al. [12] proposed diffusion models and these models achieved great success in image processing.

In recent years, some researchers have been trying to introduce the DL method to perform prediction tasks as a surrogate for numerical simulation. Zhang et al. [13] used a physics informed neural networks (PINNs) model to solve the geometry identification problems of voids or inclusions in a matrix that incorporated the PDEs of solid mechanics in the loss function. Tang et al. [14] established a 3D recurrent residual U-Net model. As it was trained on the simulated dynamic 3D saturation and pressure fields of oil-water two phase flow for a set of random 'channelized' geomodels, the

***Jin-wu Kang**
Ph. D., Associate Professor. His research interests mainly focus on the modeling and simulation of casting, solidification, and additive manufacturing. To date, he has published about 200 papers and received four academic awards from the Beijing city, Ministry of Education of China, etc. for his distinguished research.
E-mail: kangjw@tsinghua.edu.cn
Received: 2024-01-25; Revised: 2024-05-02; Accepted: 2024-07-06

saturation and pressure evolution for 3D channelized models can be predicted. Yi et al. [15] employed a two-branch U-Net model to predict the thermal gradient in the formed tacks for laser powder bed additive manufacturing. Yang et al. [16] used convolutional recurrent neural networks to predict various microstructure evolution phenomena. Diffusion models also show potential in computational chemistry [17].

These research works demonstrate the potential of deep learning models to address the scientific computation problems. However, up to date, they are still dealing with very simple problems, and there is no research on the application of diffusion model in scientific computing area.

Casting is one of the main manufacturing methods. The numerical simulation of the casting process is important for the understanding and optimization of casting process [18-19]. Heat transfer is the fundamental of the casting process, which dictates solidification behavior, stress and strain development, microstructure evolution, and the occurrence of defects such as shrinkage porosity, cracks, deformation, etc. Therefore, heat transfer was selected for deep learning studies in this study. The conditional diffusion model was introduced as a surrogate model to simulate the casting process. The problem of predicting the temperature fields of castings with different shapes at different solidification times was investigated.

2 Diffusion model for the simulation of casting process

2.1 Conditional diffusion model

Diffusion models are a class of likelihood-based models that make use of two Markov chains: a forward chain that gradually injects noise into the signal, and a reverse chain that successively reduces noise to the original data [12]. The former is typically to transform any data distribution into a simple prior distribution (e.g., standard Gaussian), while the latter Markov chain reverses the former by learning transition kernels parameterized by deep neural networks. The forward process is a noise-adding process, called diffusion process, and the reverse process is called denoising process. In inferring, new data points are subsequently generated by first sampling a random vector from the prior distribution, followed by ancestral sampling through the reverse Markov chain.

During the noise-adding process, the noise is added as follows at t step:

$$y_t=\sqrt{1\text{-}\beta_t}\,y_{t-1}+\sqrt{\beta_t}\,\varepsilon \tag{1}$$

where ε is the noise sampled from the standard normal distribution N(0, 1), β_t is weight constant changing linearly with t in the range (0, 1). Thus, the new state is the function of the value of the last step $y_{t\text{-}1}$ and the noise ε.

During denoising process, firstly, the deep neural networks, a Res-U-Net [20], was trained by the input of y_t, and the mean and variance of noise is the output, thus the noise responding to $y_{t\text{-}1}$ and y_t is obtained by sampling the given mean and variance. Then, this noise is removed from y_t to acquire $y_{t\text{-}1}$.

$$y_{t-1}=\frac{1}{\sqrt{\bar{\alpha}_t}}\left(y_t-\frac{1-\bar{\alpha}_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(y_t,\,t)\right) \tag{2}$$

where $\varepsilon_\theta(y_t, t)$ is the noise obtained from the trained Res-U-Net model for step t. The noise is correlated to y_t and t, α_t=1−β_t, $\bar{\alpha}_t$ is as follows:

$$\bar{\alpha}_t=\prod_{i=1}^{t}\alpha_i \tag{3}$$

The noise for each step is different, in order to consider the difference of time steps, a unique location code is set for each time step, which is related to t. It is expressed as follows:

$$E_t=[\sin(w_0t),\ \sin(w_1t),..,\ \sin(w_{i\text{-}1}t),..,\ \sin(w_{d_{\text{model}}\text{-}1}t)] \tag{4}$$

where $w_i=\dfrac{x(r,\,s)}{1000^{i/(d_{\text{model}}-1)}}$, d_{model} is the model number $h\times w\times ch$, h, w, and ch are the height, width, and channel number of a image, $x(r, s)$ is the pixel at (r, s) position in the image. Thus for each pixel in each channel, there is a value corresponding to a time step.

The conditional diffusion model used in this study is shown in Fig. 1. Compared with the vanilla diffusion model, the conditional diffusion model leveraged extra information, for example, initial shape of the casting which is used as the model condition to help training and sampling. It adopts a Res-U-Net [20] by adding residual blocks, which can improve the prediction result [21], as shown in Fig. 2. The Res-U-Net model is aimed during the reverse process to estimate the noise added during the forward process. In this model, there are mainly down sampling branch and up sampling branch which are symmetrical. Between the down sampling and up sampling, there are skip connections for the same level from down sampling to up sampling. In the down sampling, there are several layers of residual blocks and pooling, where the residual block containing the combination of convolutions, normalization, and activation. So, the image size is gradually reduced by pooling. In the up sampling, deconvolution is taken to gradually expand the image size to the size of input image, and residual blocks are taken for each layer as well. For the conditional diffusion model, there are two inputs of Res-U-Net model, one is condition, and the other is the image with noise added y_t. The two inputs are concatenated into 6 channels firstly and then treated by a convolution. The following is residual blocks. The residual block consists of three paths, one containing two convolutions, one shortcut convolution, and the treating of time step, which is also a $h\times w\times ch$ array, the same size as the treating of input images. In the residual blocks the processing is repeated several times (n_Resnet). The embedding of time step information t begins with the coding of t and input by Eq. (4), and followed by two consecutive neural networks. Thus, the obtained $h\times w\times ch$ array is fed into the residual blocks in layers of both of the down sampling and up sampling.

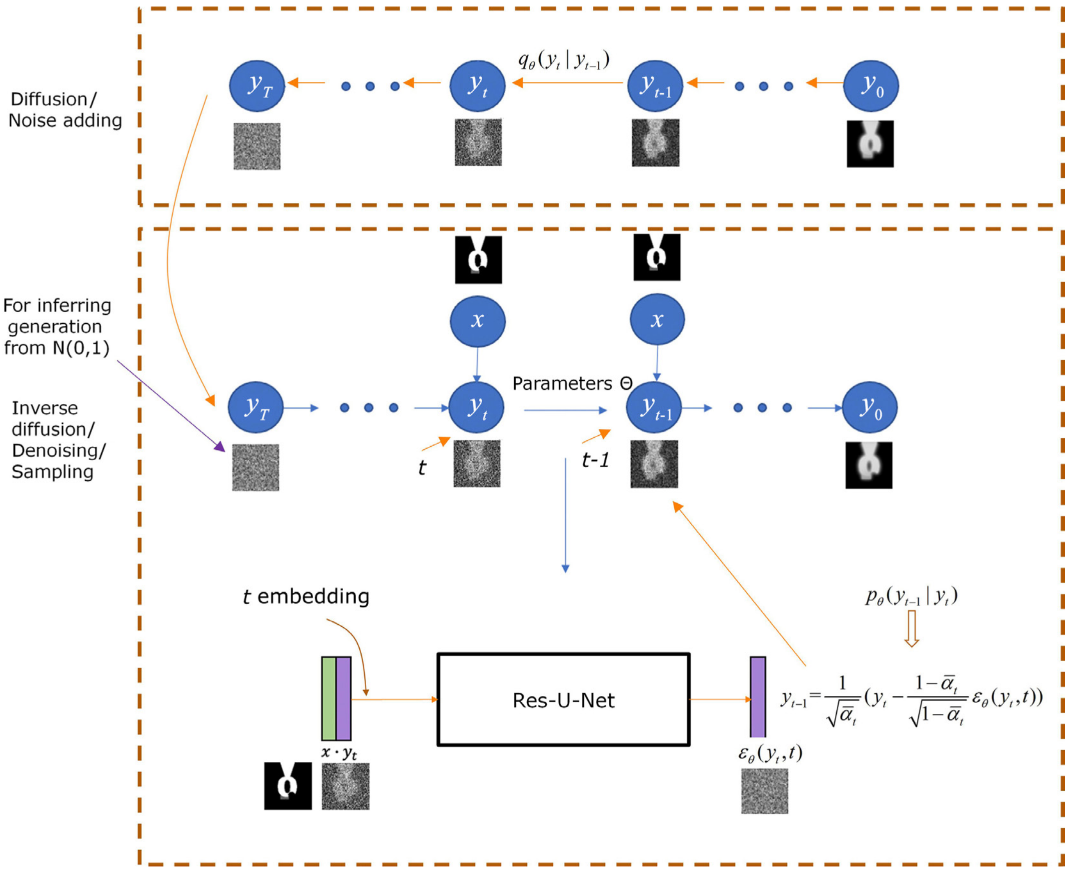


Fig. 1: Conditional diffusion model

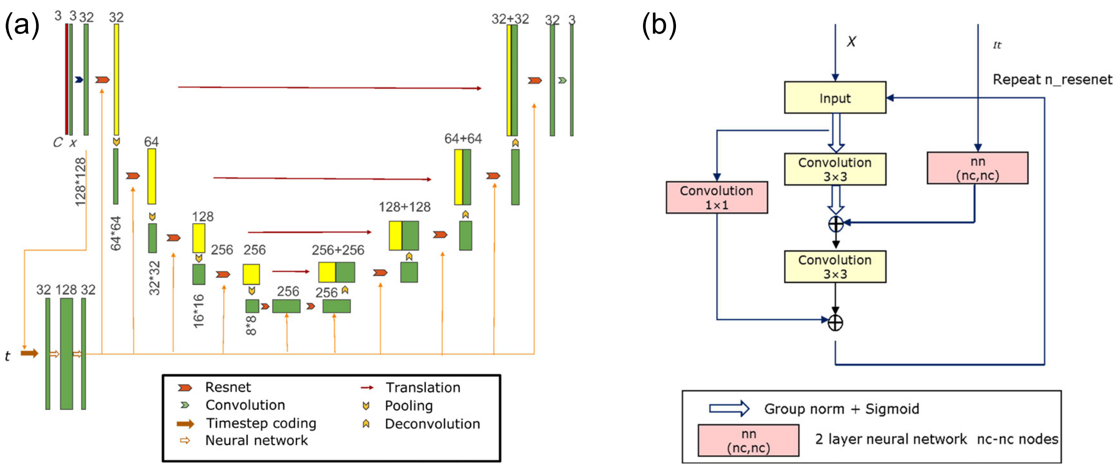


Fig. 2: Res-U-Net model and residual block: (a) Res-U-Net model; (b) residual block

2.2 Loss function

The difference between the predicted noise and actual noise was used for the loss function [12]:

$$E(x,x_0,\varepsilon)\sum_{t=1}^{T}\frac{1}{T}\left\|\varepsilon-\varepsilon_\theta(x,\sqrt{\beta_t}x_0+\sqrt{1-\beta_t}\varepsilon,\beta_t)\right\|_p^p \quad (5)$$

where t is the time step, $t \in (0, T)$, T is the maximum step, x is the input condition, θ is a parameter set to be trained in the Res-U-Net, p represents the levels of norms with value of 1 or 2 for the first norm or second norm. The L1 Loss (p=1) is applied which is proved to reduce the voracity of images according to Ref. [22], which is suitable for the prediction of temperature fields.

2.3 Accuracy evaluation

The MAE (mean absolute error) between the predicted and actual values of all the pixels of the output image was used as the accuracy index, as follows:

$$\overline{L1}=\frac{\sum_i|\hat{x}_i-x_i|}{\mathrm{sum}(i)} \quad (6)$$

$$acc=\frac{\mathrm{sum}(|\hat{x}_i-x|)<10\ ^\circ\mathrm{C}}{\mathrm{sum}(i)}\times 100\% \quad (7)$$

where x_i is the true value, $\hat{x}_i$ is the predicted value, i represents a single pixel, sum(i) is the total number of pixels, which is 128×128.

3 Preparation of training and validation sets

The premise of the neural network is training. Therefore, it is necessary to prepare a training set. In order to cover as more as possible casting shapes, a wide range of geometries of castings should be included in the training set to cover most of casting features.

3.1 Preparation of geometrical models

It is difficult to directly construct the CAD models of numerous geometries using CAD software. As the document editing software, Microsoft Word provides databases of characters from many languages and many kinds of symbols, hundreds of these characters and symbols were typed into a document file with the same size and character spacing and line spacing. Then, their images were separately captured as the shapes to represent 2D basic casting shapes. The foreground color of the image was set as white to be the casting and the background as black to represent mold, as shown in Fig. 3.

To further expand the casting geometries, a random erosion method (REM) was proposed. A circle was used as an erosion source, which moves in the image field to erode the casting geometries as it meets casting pixels. The position of a circle was placed in the image randomly and the erosion was performed many times to obtain many new shapes of castings, as shown in Fig. 3. Then, the topological features were checked to get rid of the isolated areas by the two-pass method. The image size was set as 128×128, and the casting was no bigger than the image size, roughly in the range of 0.3–0.8 of the image. The casting size can be adjusted by setting the actual size corresponding to pixels.

Based on the obtained casting models, risers and chills were added to reflect the actual casting process. The top and bottom of these casting models were judged, a riser was formed by extending the top of the casting to the top border of the image with a taper of 2° marking with white, the same color as the casting. The bottom of the casting was expanded downward to 20 pixels which were treated as chill by marking grey color. This results in the casting having a riser on the top and chill on the bottom, which facilitates sequential solidification and feeding during the solidification process, as shown in Fig. 3. The surrounding area of the casting and chill was the molds. By this way, 900 geometrical shapes were established.

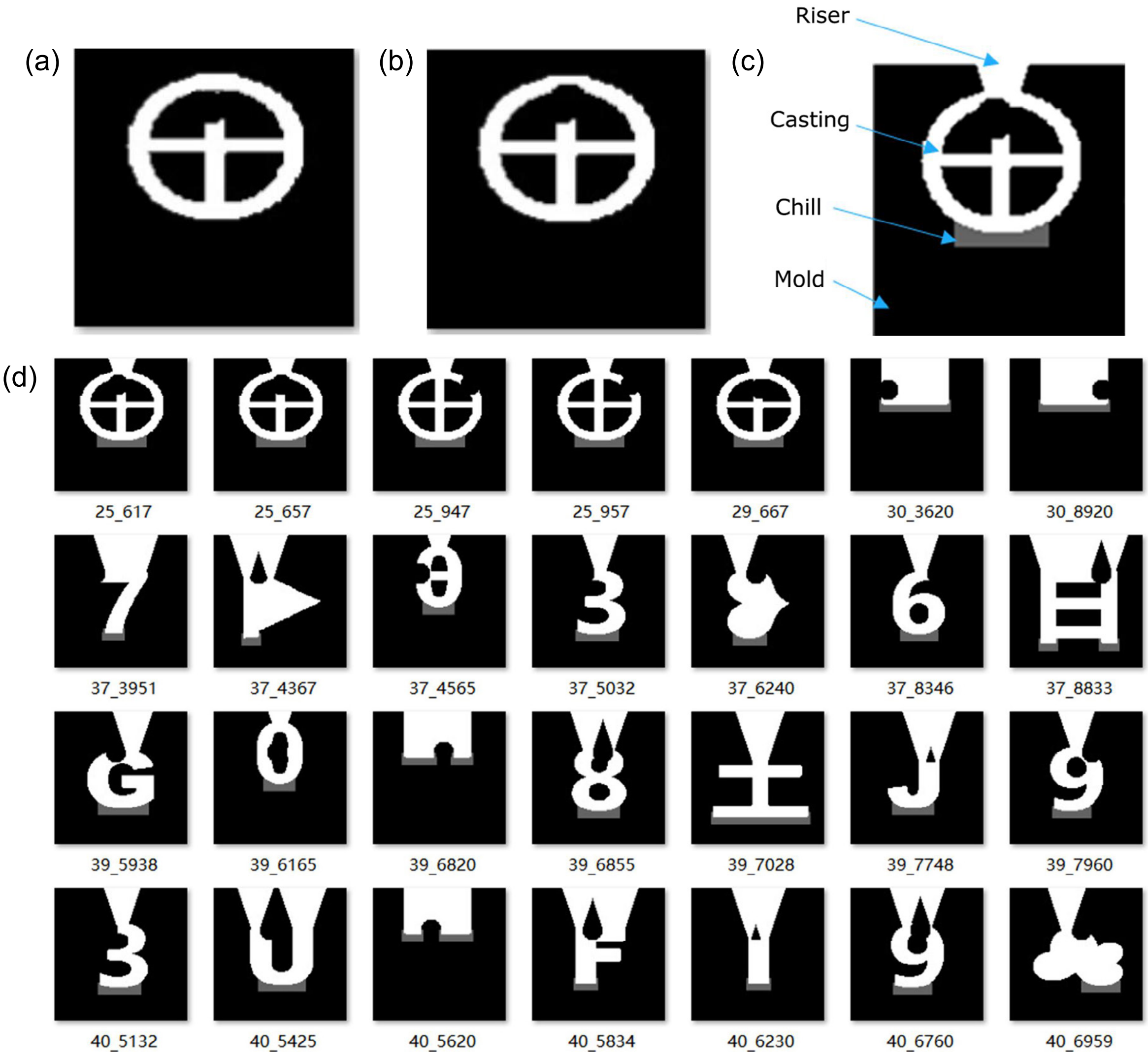


Fig. 3: Sand mold casting model after adding risers and chills: (a) basic shape of casting; (b) eroded casting; (c) model of casting, chill, riser, and mold; (d) examples of casting models

3.2 Preparation of simulated results

The images of castings with chills and molds were used as the input for the heat transfer simulation, and each pixel can be treated as a finite difference mesh. The size of each pixel was supposed to be 0.005 m×0.005 m to roughly represent the popularly used casting size in the range of 200–500 mm. The casting, chill, and mold were given the typical materials used in casting production, aluminum alloy, steel, and sand mold, respectively. The boundary conditions between the casting top surface and the mold with environment were set as convection and radiation. The initial temperature of castings was set as the temperature of the liquid melt, and those of chill and mold were set as the environment temperature. In this case, the material properties and initial conditions are listed in Table 1. The mold-air and casting-air boundaries were treated as natural convection with heat transfer coefficient of 20 $W^{-1}·m^{-2}·K^{-1}$ and 25 $W^{-1}·m^{-2}·K^{-1}$, respectively. The heat resistance of boundaries among casting, chill, and mold was supposed to be 10^{3} $W^{-1}·m^{-2}·K^{-1}$, the heat transfer at these boundaries are treated by the heat resistance model, i.e., the total heat resistance between a pair of boundary meshes was the resistance of each side and the boundary resistance.

The initial temperature field was kept as images serving as the input. The temperature is converted into color gray scale by Eq. (8):

$$T=(T-T_{min})\times255/T_{max} \quad (8)$$

where T_{max}, T_{min} are the maximum and minimum temperatures, respectively. For the gray scale, the range is from 0 to 255.

The numerical simulation was carried out using a self-developed code based on Python, which is a 2D calculation program. It was programmed to batch-process all the casting geometries for numerical simulation. The output is also images in gray scale, with each pixel corresponding to one mesh and color representing temperature. The temperature fields at 10 min, 20 min, 30 min, 40 min, 50 min, and 60 min are obtained and used for the training of the diffusion models. They are denoted as T_i at t_i, i=0, 1, 2, 3, 4, 5.

Table 1: Materials properties

Materials	Thermal conductivity ($W·m^{-1}·K^{-1}$)	Specific heat ($J·kg^{-1}·K^{-1}$)	Density ($kg·m^{-3}$)	Initial temperature (°C)	Latent heat ($J·kg^{-1}$)
Cast Al alloy	140	880	2,680	700	425,000
Sand mold	1.25	469	1,590	25	–
Chill	48	468	7,800	25	–

4 Training

The basic program downloaded from Ref. [22] is written in Pytorch. The channels of input and output channels are modified to one for images at gray scale. The operating system is Linux. The hardware and software specifications are summarized in Table 2. The hyperparameters used for training are summarized in Table 3.

Three training schemes were set, as shown in Table 4. Scheme 1 takes the geometry models as input, and the output is the temperature fields at different times, a diffusion model for each time step. Thus, there are six diffusion models for the six time steps, i.e., each diffusion model for the prediction of each temperature field T_i. For Scheme 2, the input is the initial temperature field, a diffusion model for each time step, therefore, there are six diffusion models for six time steps as well. For Scheme 3, the input is the temperature at t_i, and the output is the temperature field at t_{i+1}, so, only one diffusion model corresponds to the six time steps. The Scheme 3 can be applied in two cases, one is the input of the already known temperature field at t_i, then the temperature field at t_{i+1} can be predicted. The other case is to continuously predict all of the temperature fields at different times by inputting the initial temperature T_0 at t_0, by which the temperature T_1 at t_1 is predicted, then it is fed into the diffusion model as input to predict the temperature field T_2 at t_2. Thus, by this way, the

Table 2: Experimental environment

Items	Contents
GPU1	NVIDIA GeForce GTX 1080 T_i
GPU2	NVIDIA GeForce GTX 1080 T_i
Operation system	Ubuntu 18.04
Libraries	Pytorch 1.13.1
Python version	Python 3.7.16

Table 3: Hyperparameters used for training

Items	Values
Res-U-Net CNN kernels	3×3
*n*_Res_block	2
Batch size	12
Optimizer	Adam
Diffusion maximum time step, t_{Max}	2,000
Drop out	0.2
Weight of noise, β_t	[0.0001–0.02]
Iteration number	1e5

Table 4: Scheme of training and prediction accuracy

Scheme	Training					Input → output	Inferring			
	Conditioned input	Output	Set size	Number of diffusion models	Time		Loss	Accuracy (%)	Time for a case (s)	Set size
Scheme 1	Geometry	T_i	302	6	12 h 24 min	Geometry $\rightarrow \hat{T}_i$	5.4245	90.33	55	8
Scheme 2	T_0	T_i	302	6	11 h 46 min	$T_0 \rightarrow \hat{T}_i$	10.3099	77.51	53	8
Scheme 3	T_i	T_{i+1}	1,806	1	6 h 42 min	$\hat{T}_i \rightarrow \hat{T}_{i+1}$	3.0256	97.68	52	200
						$T_0 \rightarrow \hat{T}_1 \rightarrow \dots \hat{T}_i \rightarrow \hat{T}_{i+1} \rightarrow \dots$	7.1777	69.13	52	10

final temperature field T_5 at t_5 is obtained. This case is for the prediction of a series of temperature fields at different times of a new casting. For Schemes 1 and 2, 302 geometries were used for training, and for Scheme 3, 1,806 samples were used with 302 geometries timing 6 time steps for each geometry.

5 Prediction

As the diffusion models were trained, they were used to predict the temperature fields for any different castings. The loss, accuracy, training and prediction time of each scheme are listed in Table 4. It can be seen that the prediction time for a single geometry model is less than one minute. The accuracy is positively related to the loss. The Scheme 3 owns the highest accuracy for the case of single step prediction, reaching 97.7%. The continuous prediction based on Scheme 3 is of the lowest accuracy as 69.1%. The decaying of loss with the iteration number of Scheme 3 is shown in Fig. 4. After 20,000 iterations, the loss continuously decreases to a small value, and the accuracy reaches a high level.

Examples of the isolated predicted results are shown in Fig. 5. The comparison of the predicted and simulated results of different time steps illustrates relatively good agreement, with just very few pixels with high temperature differences over ± 20 °C.

Examples of continuously predicted results by Scheme 3 are shown in Fig. 6. The input is just the initial temperature field, and the temperature fields of 10–60 min are predicted step by step with the predicted results as the input for the next step. The casting and mold temperature are extracted from the predicted results so as to observe the temperature fields at different scales. From the predicted results, it can be seen that the castings cool down and molds are heated up gradually. The cooling and heating curves of some points evenly distributed in the casting, the mold or chill are plotted in Fig. 6(h). The predicted and simulated results concluding the distribution and the curves are in good agreement. Although the accuracy for the continuously predicted results is only 69.1%. The accuracy variation with time steps is shown in Fig. 7. It can be seen the accuracy of all components including casting, chill and mold declines with the increasing of time steps: for the first two steps the accuracy is higher than 90%, while it decreases rapidly to 50% at the final step (60 min). Thus, their average accuracy is 69.1%. However, the accuracy of the predicted temperature field of the casting roughly increases with time, reaching 91.9% at 30 min, and 87.6% at 50 min. Thus, the discrepancy lies mainly in chill and mold.

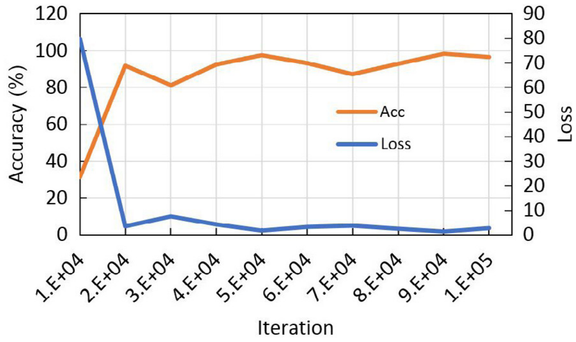


Fig. 4: Loss and accuracy variation with iterations of Scheme 3

6 Discussion

6.1 Effect of inputs

The accuracies of Schemes 1 and 2 are different, as the first one takes the geometry model as input, while the later one takes the initial temperature as input. The former one illustrates higher accuracy. In the geometry model, the three components of different materials, casting, mold, and chill can be clearly identified, while in the initial temperature field, chills are not identified because they take the same temperature as the mold. Thus, the identification of each component in different colors in the input images is more important.

6.2 Error caused by the discrepancy of T_i and T_{i+1}

For Schemes 1 and 2, the accuracy roughly decreases for longer time prediction, as shown in Figs. 8 and 9. However, for Scheme 3, the prediction accuracy from true T_i to T_{i+1} roughly increases with longer time prediction, which is higher than that of Schemes 1 and 2. The reason is that for Schemes 1 and 2, regardless of T_0 or geometry to the T_i, as i increases,

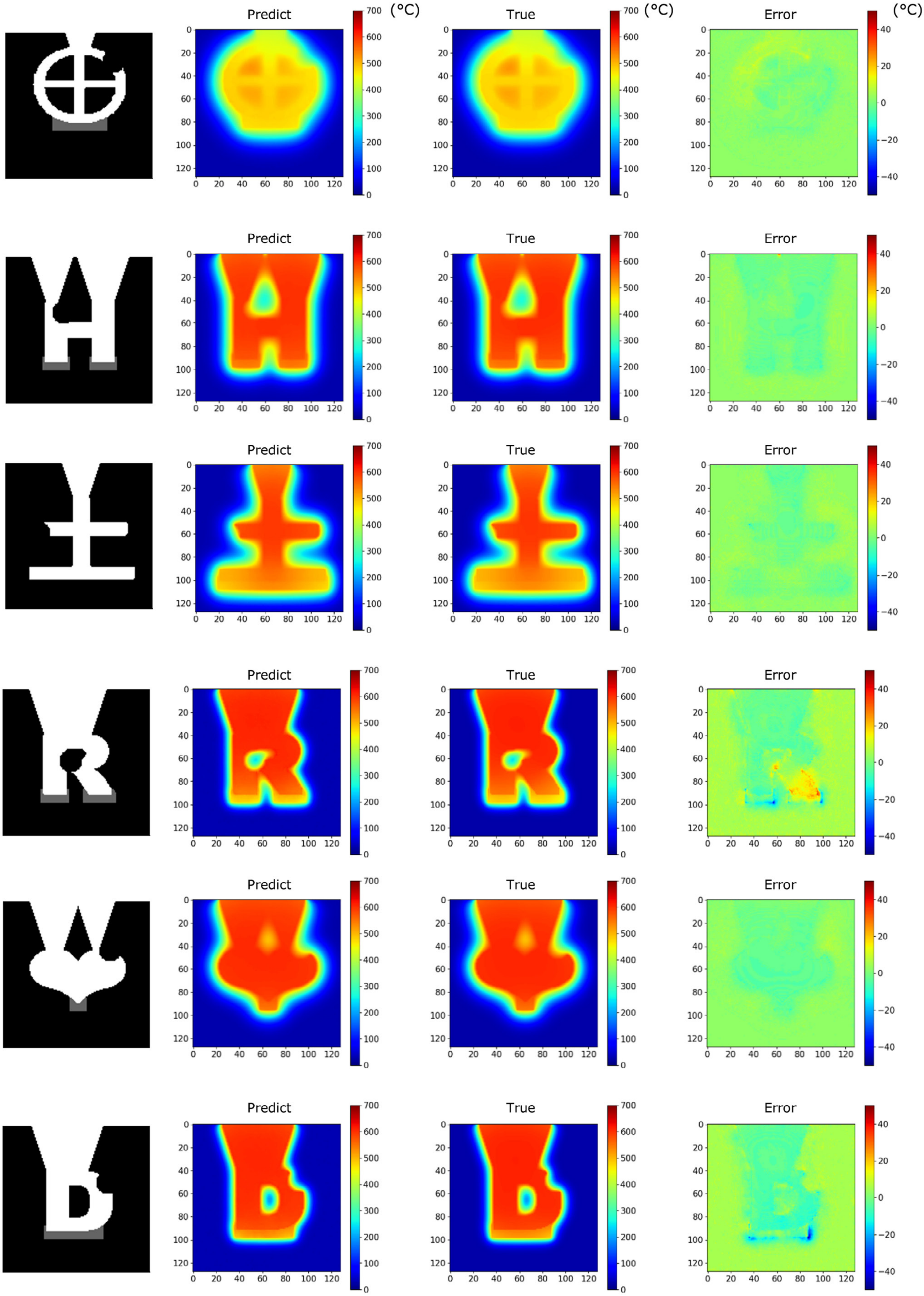
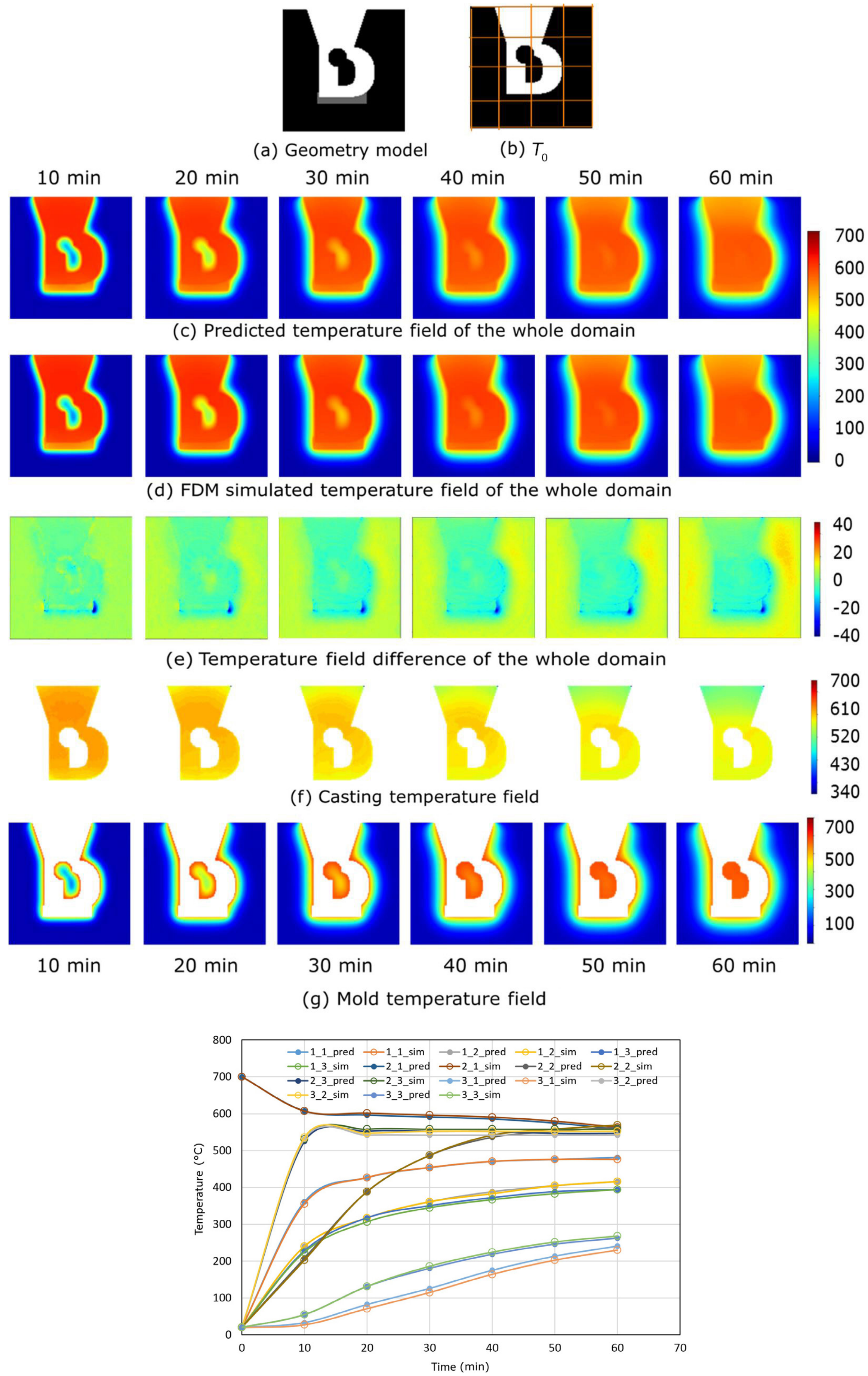


Fig. 5: Comparison between simulated and predicted temperature fields for different geometrical models (isolated prediction by Scheme 3)



(h) Comparison of temperature vs time curves of casting, mold and chill with locations at the grid nodes in Fig. 6(b)

Fig. 6: Recurrently predicted results of an example and their comparison with simulated results

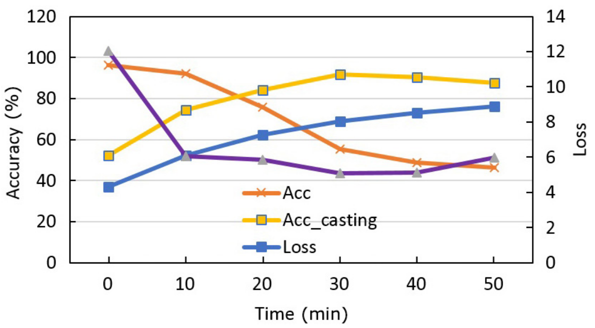


Fig. 7: Accuracy decreases with time steps for the continuous prediction by Scheme 3

the time difference between the input and T_i increases, and the temperature difference between the input and output covers a bigger and bigger time span. However, for Scheme 3, the time difference between the input T_i and output T_{i+1} is always one step.

The isolated predicted results at different time steps of the same geometry exhibit significant accuracy differences in Scheme 3, as shown in Fig. 9. There is significant error around the area with chills from T_0 to T_1, but the error is very small from T_4 to T_5. This means the prediction by diffusion model is independent from the previous steps, which is different from the successive calculation from the initial temperature field in numerical simulation.

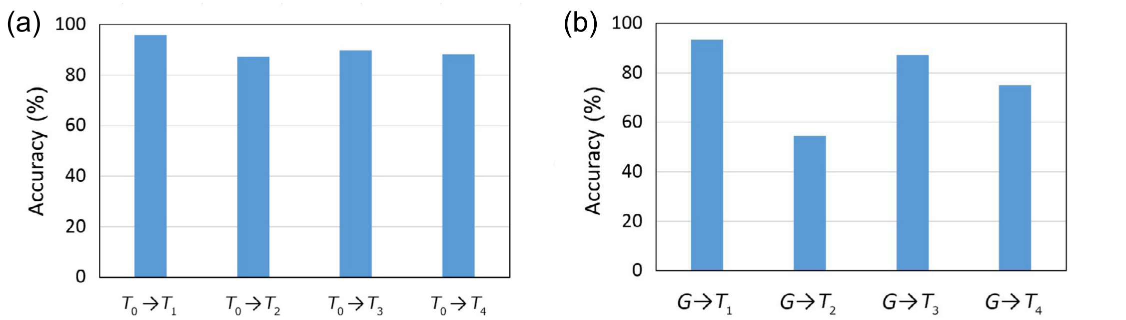


Fig. 8: Accuracy of temperature prediction for different time steps of different models Schemes 1 (a) and 2 (b)

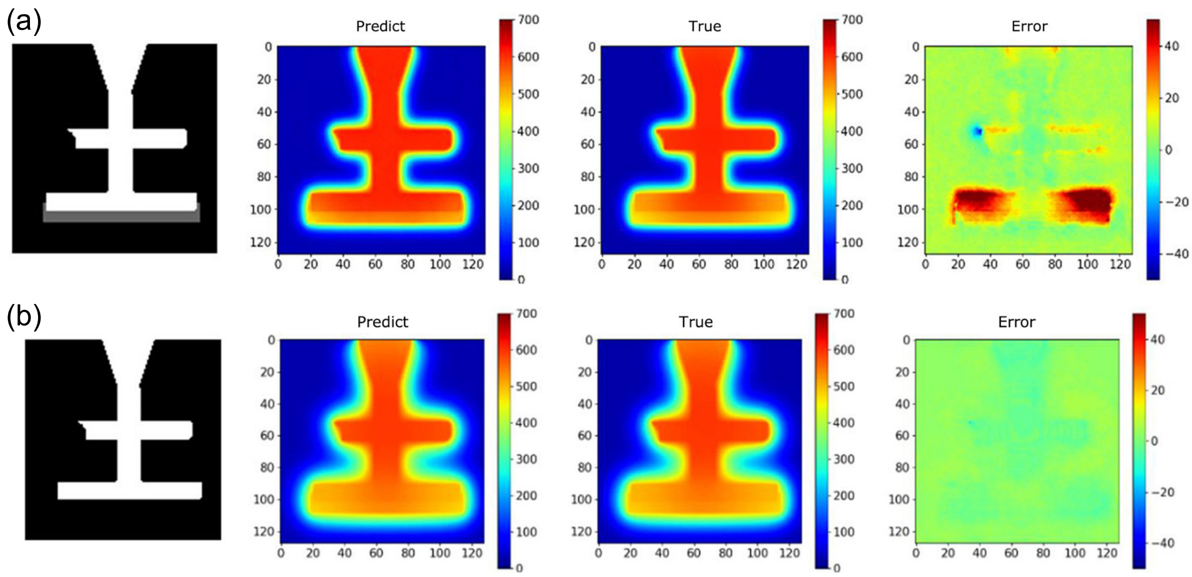


Fig. 9: Comparison of the error for the isolated predicted results by Scheme 3 for the same geometry, big error only occurs for the first step: (a) prediction of T_1 from the initial temperature; (b) prediction of T_5 from true T_4

6.3 Error at sharp corners

The results with a big error of some typical geometries are shown in Fig. 10. It can be seen that the error mainly focuses on the top of riser, sharp corners of castings, and chills. Sharper corners are hard to describe correctly because the resolution of the geometry image is only 128×128, thus the relationship of the sharp corner pixels can't be sufficiently represented.

Chills affect the results because their geometry features are not clearly demonstrated in the input in Scheme 3. In the diffusion model, either geometry or initial temperature is used which means the absence of the initial temperature or geometry, so the chills information is missing. In future, it is necessary to consider two or more conditions to include both geometry and initial temperature features.

The significant error occurs at the top of riser as the riser spans the whole width of the domain in Fig. 10(g) perhaps because of the lack of these cases in the training set. No more cases are found as the width of the riser top is less than that of the domain.

6.4 Error at sand mold and chill

The accuracy decrease of Scheme 3 with continuous prediction was checked and the temperature error is shown in Fig. 11. It can be seen the temperature error of castings shows no significant

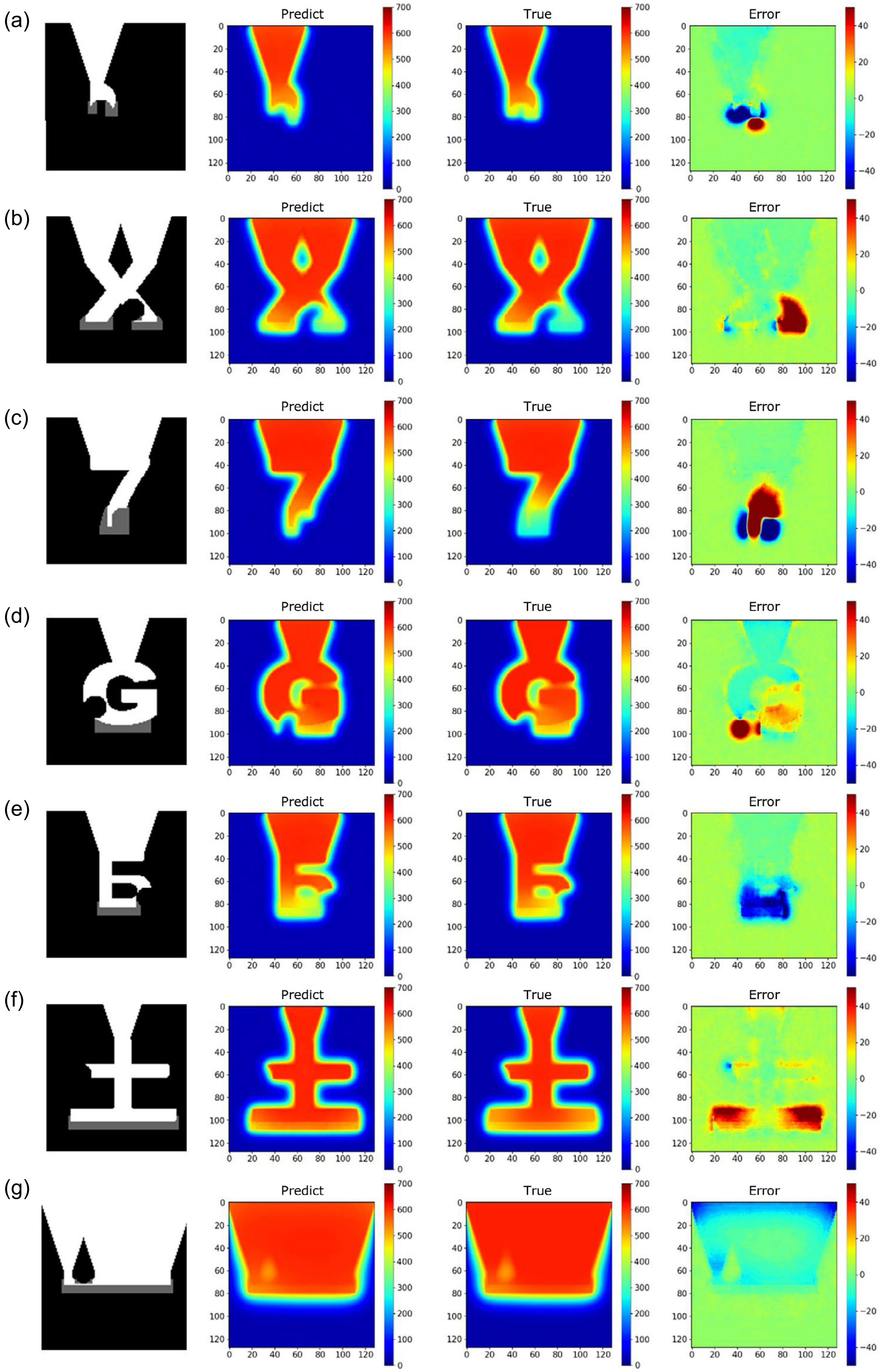


Fig. 10: Error occurs mainly around the top of the riser, sharp corners and chills for the predicted results by Scheme 3

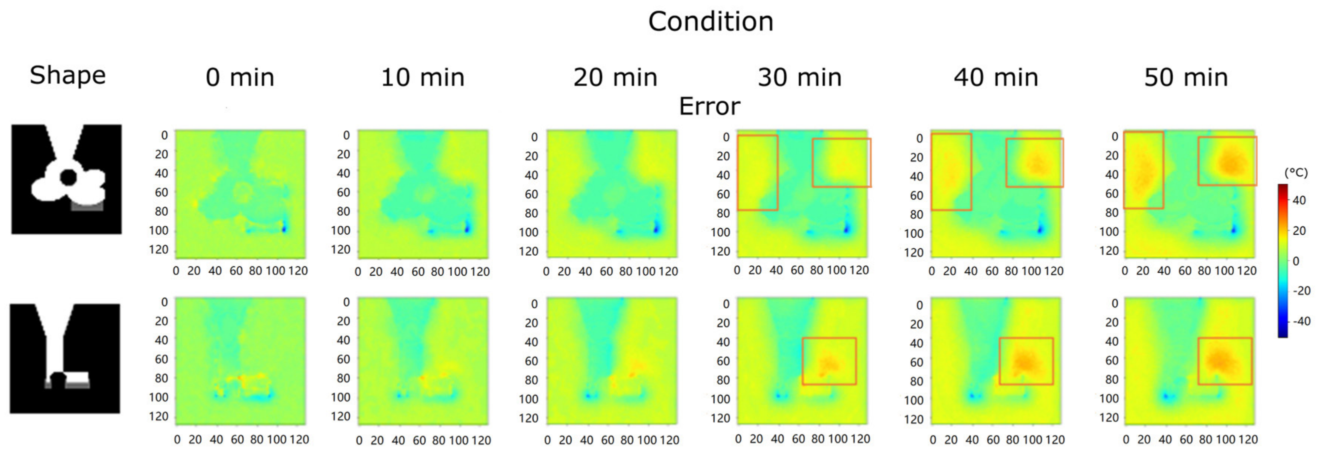


Fig. 11: Temperature error of recurrently predicted results at different time instants for Scheme 3

jump, while, that of sand molds increases after 20 min in some areas, which leads to the decrease of the whole accuracy. This complies with the accuracy curves of Fig. 7. It is reasonable that there is error accumulation during continuous predictions. But the different error accumulations of the casting and mold are at different degrees because the prediction by diffusion model is based on the addition of noise on each image pixel separately instead of the casting/mold border heat transfer in numerical simulation. Thus, the prediction of casting and mold are independent to some extent in the diffusion model. However, it is not clear why the error of mold increases significantly with error accumulation.

6.5 Selection of FDM or FEM

The method takes images of geometry and temperature field as input, so, it is actually independent of the simulated results by FDM or finite element method (FEM). As the training set of castings is transferred into images, it can be accepted by the AI model. But for the details of fine area, the numerically simulated results by FEM will be better for training. In this study, FDM is selected because it is easy for the development and the direct conversion of pixels and meshes, one pixel is corresponding to a mesh, verse versa. For FDM in this study, the input is images of shapes instead of mesh data.

7 Conclusions

Several conditioned diffusion models were established as a surrogate for the numerical simulation of the casting process. A set with more than 302 different casting shapes was constructed for the training, validation and prediction. The condition can be the geometry model or the initial temperature field T_0, and the input is a random sample of normal distribution. The prediction results of these diffusion models were obtained and compared, and the comparison between the predicted results and the simulated results was also conducted.

(1) The diffusion model takes the input of pairs of temperature field T_i as input and T_{i+1} as label owns the highest accuracy reaching 97.7% (as the temperature error of each pixel is less than 10 °C) for the isolated prediction of a single step. The continuous prediction of temperature fields just based on T_0 is realized by the serial inputs of the outputs of previous steps and the accuracy is 69.1%, but the error actually occurs in the sand mold, no significant error increases for the temperature field of the castings.

(2) The accuracy of taking the geometry shapes as condition to predict the temperature fields at different time steps reaches 90.3%, while the accuracy of taking the initial temperature T_0 as condition is 77.5%. When taking the temperature fields at T_i (i in the range of 0–5) as the conditions, the increase of the accuracy to 97.7% for isolated prediction is because of the training set is expanded by 6 times and the time difference is limited to 1.

(3) The predicted temperature field results by diffusion model is independent from the previous steps, which is different from the successive calculation from the initial temperature field in numerical simulation. Furthermore, the predicted results for the casting are independent from that of chill or mold to some extent, which means the big error of a component doesn't affect a better predicted result of other components in the geometry model.

(4) The treatment of two or more conditions such as geometry, initial condition, thermal properties and boundary conditions needs further study in the construction of diffusion model. The diffusion model is of the potential as a surrogate model for the numerical simulation of casting process. The application of the trained model in equipment for online simulation and decision making will facilitate the development of intelligent equipment and then intelligent manufacturing.

Acknowledgment

The research was sponsored by Tsinghua-Toyota Joint Research Fund.

Conflict of 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.

References

[1] Velten K. Mathematical modeling and simulation: Introduction for scientists and engineers. Wiley-VCH, 2009.

[2] Zienkiewicz O C, Taylor R L. The finite element method: Vol 1. Basic formulation and linear problems (Fourth Edition). McGraw-Hill, Maidenhead, England, 1989.

[3] Tong Z X, He Y L, Tao W Q. A review of current progress in multiscale simulations for fluid flow and heat transfer problems: The frameworks, coupling techniques and future perspectives. International Journal of Heat and Mass Transfer, 2019, 137: 1263–1289. https://doi.org/10.1016/j.ijheatmasstransfer.2019.04.004

[4] Aliabadi M H. Boundary element methods in linear elastic fracture mechanics. Comprehensive Structural Integrity, Pergamon, Oxford, 2003: 98–140. https://doi.org/10.1016/B978-0-12-822944-6.00162-6

[5] Jia Y, Wang H, Le Q. Transient coupling simulation of multi-physical field during pulse electromagnetic direct-chill casting of AZ80 magnesium alloy. International Journal of Heat and Mass Transfer, 2019, 143: 118524. https://doi.org/10.1016/j.ijheatmasstransfer.2019.118524

[6] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 1958, 65: 386. https://doi.org/10.1037/h0042519

[7] Cichy R M, Kaiser D. Deep neural networks as scientific models. Trends in Cognitive Sciences, 2019, 23(4): 305–317. https://doi.org/10.1016/j.tics.2019.01.009

[8] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. Computer Science, 2014. https://doi.org/10.48550/arXiv.1409.1556

[9] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks. Communications of the ACM, 2020, 63(11): 139–144. https://doi.org/10.1145/3422622

[10] Tsoi A C, Back A. Discrete time recurrent neural network architectures: A unifying review. Neurocomputing, 1997, 15(3–4): 183–223. https://doi.org/10.1016/S0925-2312(97)00161-6

[11] Milletari F, Navab N, Ahmadi S A, et al. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), 2016: 565–571. https://doi.org/10.1109/3DV.2016.79

[12] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems, 2020, 33: 6840–6851. https://doi.org/10.48550/arXiv.2006.11239

[13] Zhang E, Dao M, Karniadakis G E, et al. Analyses of internal structures and defects in materials using physics-informed neural networks. Science Advances, 2022, 8(7): eabk0644. https://doi.org/10.1126/sciadv.abk0644

[14] Tang M, Liu Y, Durlofsky L J. Deep-learning-based surrogate flow modeling and geological parameterization for data assimilation in 3D subsurface flow. Computer Methods in Applied Mechanics and Engineering, 2021, 376: 113636. https://doi.org/10.1016/j.cma.2020.113636

[15] Yi L, Simon P M, Ghansiyal S, et al. Simulation of the temperature gradient in laser-based powder bed fusion using machine learning. Procedia CIRP, 2022, 115: 36–41. https://doi.org/10.1016/j.procir.2022.10.046

[16] Yang K, Cao Y, Zhang Y, et al. Self-supervised learning and prediction of microstructure evolution with convolutional recurrent neural networks. Patterns, 2021, 2(5): 100243. https://doi.org/10.1016/j.patter.2021.100243

[17] Luo S, Su Y, Peng X, et al. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. In: Advances in Neural Information Processing Systems 35-36th Conference on Neural Information Processing Systems, NeurIPS, 2022: 9754–9767

[18] Liu B, Kang J, Zhao H. Study on macro and micro modeling on solidification process of shaped casting. Chinese Journal of Mechanical Engineering, 2022, 15: 252–256

[19] Gránásy L, Tóth G I, Warren J A, et al. Phase-field modeling of crystal nucleation in undercooled liquids – A review. Progress in Materials Science, 2019, 106: 100569. https://doi.org/10.1016/j.pmatsci.2019.05.002

[20] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770–778

[21] Chaurasia A, Culurciello E L. Exploiting encoder representations for efficient semantic segmentation. In: Proceedings of the 2017 IEEE Visual Communications and Image Processing (VCIP), Las Vegas, NV, USA, IEEE Computer Society, 2017: 1–4

[22] Saharia C, Ho J, Chan W, et al. Image super-resolution via iterative refinement. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(4): 4713–4726. https://doi.org/10.1109/TPAMI.2022.3204461