JOURNAL OF SIMULATION
https://doi.org/10.1080/17477778.2024.2394063

RESEARCH ARTICLE

A visual digital twin framework based on residual-based fourier neural operator online simulation method

Guodong Sa[a,b], Chenhui Wu[b,c], Zhenyu Liu[b,c], Daxin Liu[b,c], Bin Li[d], Wugang Wang[d] and Jianrong Tan[b,c,e]

[a]Ningbo Innovation Center, Zhejiang University, Ningbo, China; [b]School of Mechanical Engineering, Zhejiang University, Hangzhou, China; [c]State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China; [d]Key Laboratory of Healthy & Intelligent Kitchen System Integration of Zhejiang Province, Ningbo, China; [e]Zhejiang Lab, Hangzhou, China

ABSTRACT

To enjoy a wonderful cooking experience in a smart kitchen, users and designers need a visual interaction platform. This paper proposes a DT framework incorporating data processing, flow field online simulation, equipment monitoring, interaction and visualization. Specifically, the DT online simulation and visualization of the kitchen fume flow field serve as the foundation for appliance design and control. Additionally, users can gain deeper insight into the state and change trends of the kitchen. To address the online simulation of the flow field, a RFNO online simulation method is proposed. In addition, this paper proposes an Echarts-based 2D and UE5-based 3D flow field visualization method to enable dynamic visualization and interaction of the flow field. The proposed DT framework was successfully verified in the case of the smart kitchen, demonstrating its efficiency and effectiveness.

ARTICLE HISTORY
Received 21 July 2023
Accepted 15 August 2024

KEYWORDS
Digital twin; Reduced-order model; flow field visualization; fourier neural operator

1. Introduction

The DT is described as a simulation process that integrates multidisciplinary and multiscale simulation through multiple sensors' collaborative sensing of physical reality (Tuegel et al., 2011). Smart home DT technology is a cutting-edge approach that leverages sensors, IoT devices, and Artificial Intelligence algorithms to collect and analyze data in real-time. By doing so, it creates a virtual model of the home environment that can be used for remote monitoring, prediction, and control. This technology has the potential to greatly improve home security, energy efficiency, and environmental sustainability by providing real-time insights into the behaviour of the home environment.

Designers of smart homes face several challenges, including the need for real-time and convenient data exchange and high data accuracy. To address these challenges, researchers have developed various approaches to enhance the efficiency and reliability, and user-friendliness of smart home DT systems. Gopinath et al. (2019) redesigned the smart home with DT technology, which improved efficiency and user-friendliness compared to traditional IoT systems. Maryasin (2019) used Ontologies to facilitate data exchange between embedded models and service programmes of the DT, as well as between the DT and external agents, users, or other software. Yang et al. (2023) used InfoMat to map user location, walking trajectory, and dynamic activity to the twin space. However, these DT systems only target single-point data, making it difficult to reflect field data. Real-time calculation and high accuracy for field data in smart home DT systems have not yet been fully addressed. Future research should focus on developing more advanced algorithms and techniques to improve the accuracy and real-time processing of field data in smart home DTs.

From the user's perspective, a smart home DT must achieve optimal levels of interactivity and visualization. In this regard, Xu et al. (2023) conducted an interactive visual analysis of urban traffic data and devised an intuitive and user-friendly interface for users. Similarly, Kriushichev et al. (2020) developed Vector Viz, which enables data to be represented using a particle system, streamlines, or a contour on a plane. However, the interactive visualization of the smart home flow field remains an unexplored area. Therefore, further research and development are necessary to enhance the interactive visualization techniques of smart home DTs, thereby improving user experience and operational efficiency.

In the context of smart homes, the kitchen is a crucial area that requires special attention. Given that cooking is a daily activity in most households, it is important to consider the potential health risks associated with the production of fumes, particularly in China where cooking is deeply rooted in tradition and culture. Numerous studies have demonstrated that fumes can have adverse health effects, including

CONTACT Daxin Liu liudx@zju.edu.cn State Key Laboratory of CAD&CG, Zhejiang University, 6-430, Kaiwuyuan, Zijingang Campus, Xihu District, Hangzhou, Zhejiang 310027, China

respiratory diseases (Casas et al., 2012) and even lung cancer (Buonanno et al., 2017). As such, developing effective monitoring, control, visualization, and interaction mechanisms for the kitchen scene is imperative to mitigate these risks and ensure a healthy living environment.

The smart kitchen DT platform plays a crucial role in enabling integrated home designers to select the appropriate hood model for different kitchens. However, without the ability to perform real-time calculations and visualization of the flow field, the platform's effectiveness is limited. This can lead to reduced design efficiency, increased costs, and improper design, which can compromise the effectiveness of fume discharge and reduce kitchen comfort while posing safety hazards. Moreover, users may find it challenging to understand the generation and propagation rules of fume and visually manage the kitchen fume environment. This can lead to ineffective measures to control fumes and a lack of awareness of the degradation of air quality, which can negatively reduce comfort, and cooking experience, and impact health. Therefore, real-time simulation and visualization of the kitchen flow field are essential to address these challenges and ensure optimal kitchen performance and user experience.

1.1. Online simulation of the fume flow field

The computational problem of the flow field is a common challenge in engineering, often addressed through simulation techniques (Tezduyar et al., 1996). Despite significant advancements in simulation software, such as Ansys Fluent, the computational demands of flow field simulations remain time-consuming and often fail to meet real-time requirements. To address this challenge, a fast computational method known as ROM has been developed. ROM establishes a mapping between input parameters, such as physical parameters or initial conditions, and the output flow field, enabling real-time computation of flow fields.

Commonly used ROMs for flow fields include POD-Galerkin (Rowley & Dawson, 2017), PINN (Lu, Jin, et al., 2021), GNS (Pfaff et al., 2020), Neural Operator-based learning. POD-Galerkin is based on the Navier-Stokes equations and uses POD-extracted orthogonal modes to satisfy the Galerkin expansion. However, it has several disadvantages such as non-convergence, non-conservation of energy, and too long transition states. PINN utilizes the gradient of the partial differential equation (PDE) and the loss between the real value and predicted solutions to accelerate the calculation of flow fluid. However, it is slow and difficult to achieve the convergence of a complex PINN. GNS, received formatted graphical data as input which fits the flow field grid, enabling the simulation of complex fluids. Han et al. (2022) combined GNS with the transformer to improve the error accumulation problem. However, the hardware required for training is very demanding. Neural Operator-based learning, such as DeepOnet (Lu, Jin, et al., 2021) and FNO (Z. Li et al., 2020), have good generalization capability and require only quasi-linear computational complexity. Convolutional Neural Network (CNN) (Gu et al., 2018) is effective for images with local features, but for the flow field online simulation problem, which is essentially solving PDEs, the global convolution in Fourier space is more effective than CNN. FNO-based ROM has significant advantages in the real-time computation of the kitchen flow field. However, increasing the number of layers of FNO may decrease accuracy and even cause the gradient to disappear, limiting the flow field simulation accuracy.

1.2. Visualization of the fume flow field

Visualization techniques are commonly employed as a supplementary tool to flow field simulation, to provide professional designers with insights into the characteristics of the flow field, such as pressure, velocity, and volume fraction (Kamat & Martinez, 2001). However, these techniques often overlook the behaviour of the flow field in a given scene, such as the flow and diffusion of fumes in a kitchen. Users may be more concerned with the visual representation of the flow field rather than its specific properties. This can make it difficult for users to effectively manage kitchen appliances.

Flow field visualization is an important aspect of understanding fluid dynamics in various fields such as engineering, physics, and meteorology. Vector diagrams, flow line diagrams, and line convolutional integration (Cabral & Leedom, 1993) are some of the commonly used visualization methods for flow fields. These methods help in representing the direction and magnitude of fluid flow clearly and concisely. With the advancement of graphics technology, it has become possible to choose appropriate flow field visualization methods for different scenarios. For instance, web visualization is suitable for 2D flow fields, while UE5 (Epic Games, 2021) is well-equipped for 3D flow field visualization and interaction. However, integrating sensor data and flow field data import, processing, and visualization in the smart kitchen DT automatically is a challenging task. Moreover, it is essential to ensure the accuracy and reliability of the data to enable effective decision-making in the smart kitchen environment.

Integrating the application scenes into the flow field visualization model and interactive features is the key to meeting the requirements of users and designers,

where interactive features include the user-system interaction and the system-hardware interaction.

The proposed visual DT framework for the smart kitchen integrates data processing, flow field online simulation, equipment monitoring and control, interaction, and visualization. Offline CFD-based simulations are often time-consuming, so the RFNO online simulation method is proposed to achieve rapid and accurate calculations. Additionally, this paper proposes the Echarts-based 2D flow field web visualization and the UE5-based 3D flow field particle interactive visualization method to enable dynamic visualization and interaction of the flow field, which is particularly beneficial for users as existing simulation results are often geared towards professionals.

The remainder of this article is structured as follows: Chapter 2 introduces the DT framework of the smart kitchen, Chapter 3 introduces the online simulation method, Chapter 4 introduces the interaction and visualization method, and Chapter 5 introduces cases and applications.

2. The DT framework of the smart kitchen

The DT of a smart kitchen is a crucial tool for designers and users alike. For designers, it enables greater efficiency by allowing them to quickly test and validate design solutions in a virtual environment, saving time and resource costs. Additionally, the DT can produce highly accurate prediction results, providing designers with a deeper understanding of the kitchen's operation.

For users, the DT provides better decision support by helping them understand the state of the kitchen and make informed decisions to improve the kitchen flow field and reduce health hazards from fumes. The DT also provides more intuitive and vivid visualization, allowing users to better understand and perceive information in the virtual environment and feel the improvement of the kitchen space by kitchen products, enhancing their overall happiness. Overall, the DT of the smart kitchen is an essential tool for improving efficiency, accuracy, and user experience.

The smart kitchen DT framework depicted in Figure 1 is a comprehensive system that integrates various technologies and components to create a virtual replica of a physical kitchen environment. This framework enables real-time monitoring and analysis of kitchen operations, as well as the ability to simulate various scenarios for improved efficiency and productivity. The framework is designed to leverage cutting-edge technologies such as IoT, machine learning, and data analytics to provide a seamless and intuitive user experience.

The IoT layer is responsible for collecting data from various sensors, such as temperature sensors, PM2.5 sensors, and VOC sensors, among others. This data is then sent to the IoT Hub, which serves as the primary connection point for IoT devices. The IoT Hub sends the online data to the ROM layer as input, while the sensor data is sent to the DT Platform via the REST API and Plug-in. The data is then rendered in the DT Platform.

The flow field simulation data is stored in a twin database in the ROM layer. This simulation data is used as offline data for training the ROM. Once the ROM is constructed using the data in the database, online data is inputted, and flow field data is outputted into the DT Platform via the Plug-in.

In the DT Platform, the visualization platform has been utilized to create a twin model of the kitchen, which includes the geometric model of the kitchen as well as the modeling and rendering of various kitchen appliances such as hoods, ovens, and refrigerators. This model is depicted in Figure 2. The visualization forms have been selected in the visualization layer, allowing

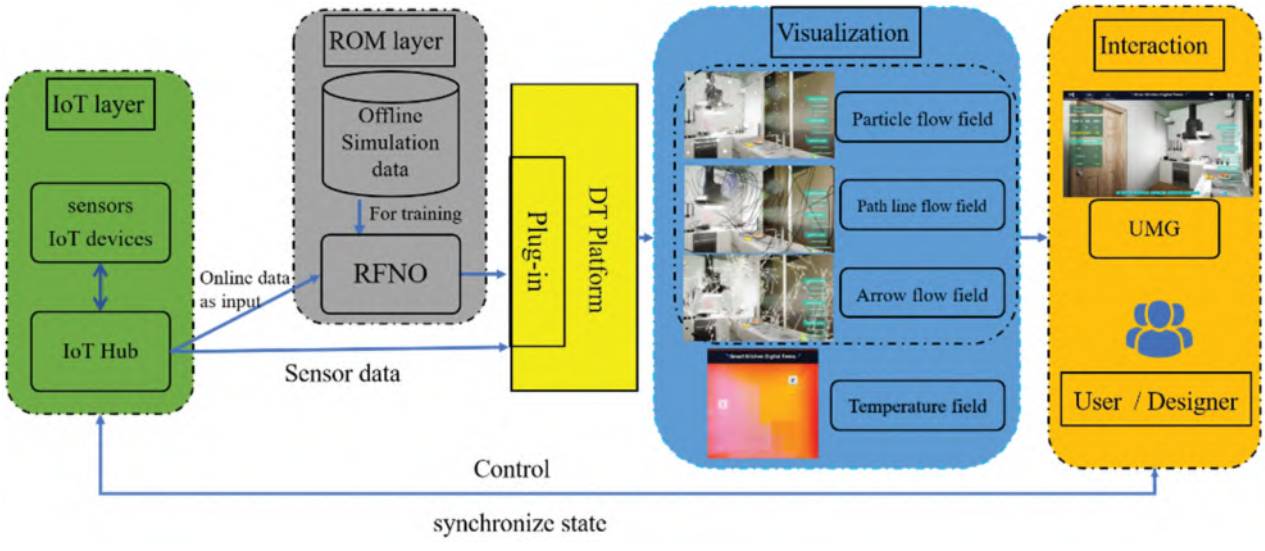


Figure 1. DT framework of the smart kitchen.

Figure 2. Realistic scene and twin scene of the smart kitchen DT.

users to observe the temperature field and choose from particle mode, arrow mode, and path line mode for the flow field. The interactive interface has been implemented in the interaction layer using Unreal Motion Graphics for users and designers. This allows users to interact with the Unreal Motion Graphics to obtain information from sensors such as temperature and air quality, import the flow field, and choose the appropriate way to visualize the flow field. Additionally, users can adjust the hood's gear to observe the change of fume and control the stove fire and hood status according to the state of the cooking fume.

3. The RFNO online simulation method of the flow field

Online simulation of the flow field is an important aspect of the smart kitchen DT, but traditional simulation methods are slow and can't meet the real-time mapping demand. Existing ROMs such as FNO have good performance in terms of accuracy and speed, but t its accuracy is limited.

In flow field online simulation, the datasets are very scarce. To obtain a more accurate approximation of the flow field, increasing the number of layers of FNO can improve accuracy when the number of data samples is difficult to increase. However, in practice, as the number of layers increases, the accuracy of the model becomes higher until a certain level where the accuracy may decrease and even the gradient disappears, as shown in Figure 3. The error gradually decreases from 4 layers to 7 layers, but the gradient disappears at 8 layers. This is because deeper FNO is more prone to gradient disappearance or explosion problems, making the model difficult to train.

To address the issue, a novel approach called RFNO is proposed. The framework of RFNO is illustrated in Figure 4, where the input is denoted as $[a, X, Y]$, representing the initial conditions or physical parameters, and X, Y representing the location information. The input is then encoded and passed through N residual modules, each containing 3 Fourier layers, before being decoded to obtain the solution of the flow field.

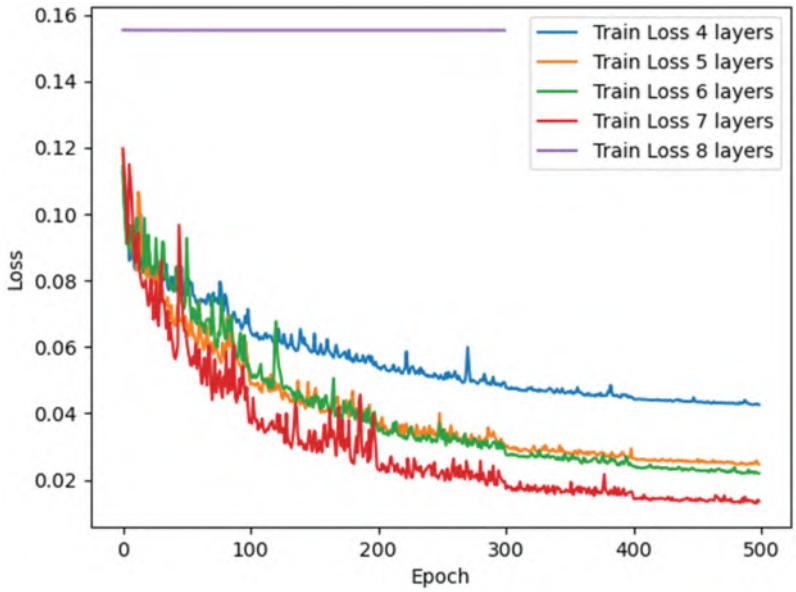


Figure 3. Comparison of the training loss of different layers' FNO for a 2D kitchen. The specifics of the 2D kitchen are outlined in Section 5.3.1.

The iterative updates process from the Fourier layer i to the Fourier layer $i+1$ can be expressed as follows:

$$H_{i+1} := \sigma(WH_{\mathrm{i}} + \mathcal{F}^{-1}(L \cdot (\mathcal{F}H_i))) \tag{1}$$

The initial conditions or physical parameters and position information are first encoded as H. For flow field prediction, the input is the initial condition, while for mapping physical parameters obtained from IoT to the flow field, the input is the physical parameters that are used to guide training under zero initial conditions. The position information confers localized inductive bias, which helps the model converge faster and generalize to new data more easily. By applying the Fourier transform $\mathcal{F}$, the global convolution in Fourier space is effective in extracting flow field information. The linear transform L filters out the high-frequency modes, and the Fourier inverse transform $\mathcal{F}^{-1}$ transforms it back to the source space, adding the output to the linear transform W. A nonlinear mapping is then performed by the activation function σ to help recover the missing high-frequency modes and acyclic boundaries.

To achieve higher accuracy by increasing the number of layers, residual learning (He et al., 2016) is introduced to complement the FNO by adding directly connected channels between different Fourier layers, as shown in Figure 5.

For FNO, without NR_i

$$R_{i+1} = Fou(R_i) \tag{2}$$

where R_i represents input, R_{i+1} represents output.

For RFNO

$$R_{i+1} = Fou(R_i) + NR_i \tag{3}$$

The gradient of FNO is

$$\frac{\partial R_{i+1}}{\partial R_i} \tag{4}$$

The gradient of RFNO is

$$\frac{\partial R_{i+1}}{\partial R_i} = \frac{\partial (Fou(R_i) + NR_i)}{\partial R_i} = \frac{\partial Fou(R_i)}{\partial R_i} + N, N \neq 0 \tag{5}$$

The item $\frac{\partial (NR_i)}{\partial R_i} = N$ in the RFNO ensures that even if the gradient in the network is small or even disappears, the overall gradient still has a non-zero value. The information and gradient can still propagate efficiently, avoiding the problem of disappearing FNO gradient, which will help the network to learn more efficiently and improve the accuracy of the model.

4. The interaction and visualization method

In the context of DTs, the ability to interact with and visualize the model is a crucial aspect. By facilitating user interaction with the DT model, users can gain a more intuitive understanding of the model's operation and can modify and optimize the model accordingly. In the smart kitchen, the DT model can simulate the operation of a hood. Through interaction, users

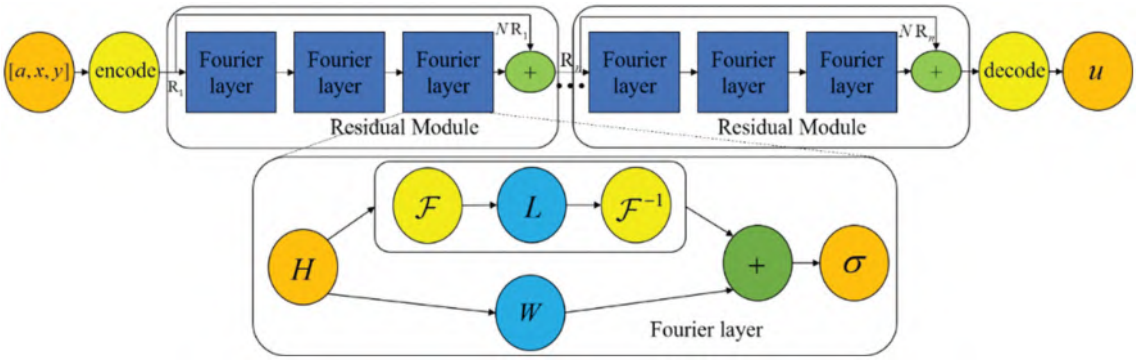


Figure 4. The framework of RFNO.

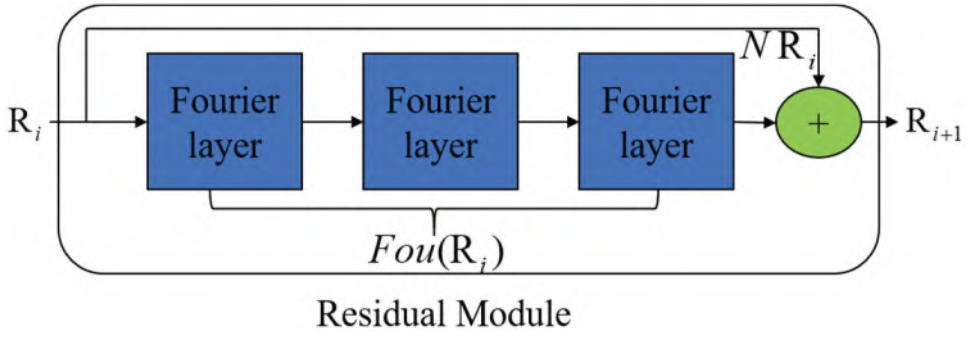


Figure 5. Residual module with 3 fourier layers.

can adjust the hood gear and optimize its performance.

The use of visualization in DTs is essential for presenting data and information in a manner that is easily understood and analyzed by users. Real-time data is particularly useful in enabling users to promptly understand the state and changes of physical systems. By utilizing charts and other data presentation methods within the DT model, users can gain a clearer understanding of trends and patterns in data changes. In the context of a smart kitchen, visualization technology can be particularly useful for designers in selecting appropriate kitchen appliances for different kitchens, thereby improving development efficiency. For users, visualization can help deepen their understanding of the state and trends of the kitchen, and enable them to more intuitively appreciate the improvements made by kitchen appliances.

The DT interaction and visualization in this paper are realized through Echarts and UE5. Echarts is an open-source visualization chart library that utilizes JavaScript, making it suitable for rapid 2D flow field visualization. UE5 is a highly advanced real-time 3D creation tool that boasts top-notch graphics technologies, such as the Niagara particle system, which enables the effective visualization of 3D flow fields. While 2D visualization is faster in terms of computation speed, it often lacks detail and can become distorted when the viewpoint changes. In contrast, 3D visualization can adapt to changes in viewpoint and provide a rich level of detail, but computation speed is often slower.

The flow field visualization method proposed in this paper establishes a bridge for data between the simulation platform and the visualization platform, integrating them to achieve dynamic visualization and interaction. Simulation data is processed and imported into Echarts and UE5, enabling real-time visualization. The powerful graphics processing unit provides fast rendering speed. So it won't be a bottleneck for real-time visualization of DTs

4.1. Echarts-based 2D flow field web visualization method

The previous chapter has introduced ROM, which effectively addresses the problem of real-time generation of 2D flow field data. In this section, we present a methodology for achieving 2D visualization that considers real-time performance and optimal visualization outcomes. The process is outlined in Figure 6 and comprises the following steps:

- Firstly, the data is generated based on the ROM.
- Secondly, the data is processed using a Python script and sent to Echarts in JSON format, where it is stored in an array $[x, y, dx, dy]$.
- Thirdly, the starting point of the rendering is defined as $[x - dx/2, y - dy/2]$, and the endpoint is $[x + dx/2, y + dy/2]$. The rendering is performed using Canvas.

This process enables the efficient and effective visualization of 2D flow field data, which is critical for optimizing kitchen appliance design and control. Using Echarts and Python script enhances the speed and accuracy of data processing while using Canvas enables fast and reliable rendering.

4.2. UE5-based 3D flow field particle interactive visualization method

The process of importing flow field data is illustrated in Figure 7. Initially, the data is deposited into a text file, where each line represents a point of information, including position, velocity, and pressure. Subsequently, the text data is read and stored in a texture for efficient graphics processing unit. The data is then passed to the Niagara particle system via the texture. In UE5, an "Actor" is created to facilitate the import and storage of data in the texture. By leveraging this methodology, the visualization of 3D flow fields can be achieved in a highly efficient and effective manner.

The use of texture for data transfer offers several advantages, including the ability to store a large amount of information. Specifically, a 2K texture can support the storage of 4,194,304 groups of data, with each group containing RGBA 4 channels' information, thereby satisfying the information storage and transfer requirements for most flow fields.

Once the data is saved in the texture, it is sampled in the Niagara particle system using the "TextureSample" function. The "Vector2D" serves

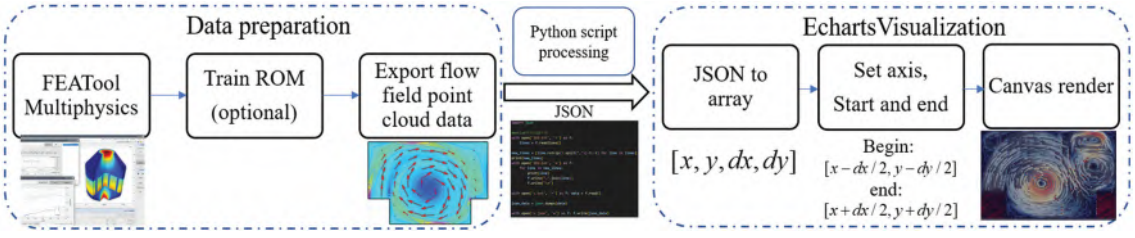


Figure 6. The process of 2D flow field data visualization.

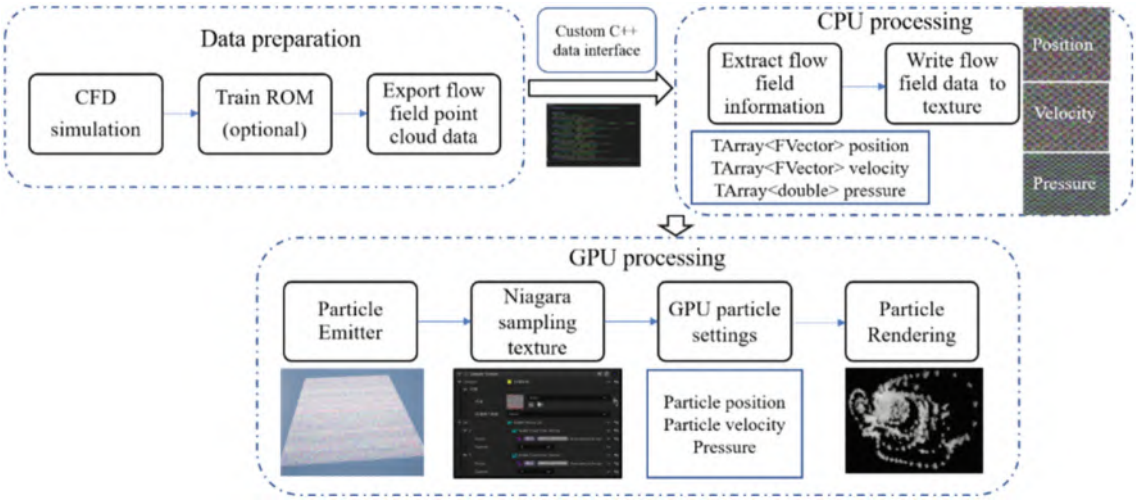


Figure 7. The process of 3D flow field data visualization.

as the input for "TextureSample", and the texture data is sampled using the "SampleTexture2D" function within the module. Specifically, the position information in the texture is set to the position of the particle, while the velocity information in the texture is set to the velocity of the particle. A "Sprite Renderer" is added under the "Render" module, and the rendering material can be specified. Additionally, the velocity can be visualized using the arrows in the "Mesh Renderer" with the direction of the arrows corresponding to the velocity.

To enable users to visualize the effect of the hood, the Niagara Fluids plug-in can be utilized to simulate the movement of fumes under the influence of different hood gears. The kitchen hood simulation process is detailed in Figure 8. By setting the "Wind Direction" of the Niagara system to z and exposing the "Wind Speed" to the user, adjustments to the "Wind Speed" enable the user to experience the movement of fume under different hood conditions. This approach facilitates the effective optimization of kitchen appliance design and control, leading to an enhanced user experience.

5. Cases and applications

5.1. Public datasets

Z. Li et al. (2020) used the 2-d Navier-Stokes equation for a viscous, incompressible fluid in vorticity form on the unit torus to produce data samples:

$$\begin{aligned} &\partial_t w(x,t) + u(x,t)\cdot \\ &\quad \nabla w(x,t) = \nu\Delta w(x,t) + f(x), \quad x \in (0,1)^2, t \in (0,T] \\ &\quad \nabla \cdot u(x,t) = 0, x \in (0,1)^2, \qquad t \in [0,T] \\ &\quad w(x,0) = w_0(x), \qquad\qquad x \in (0,1)^2 \end{aligned} \tag{6}$$

where $u(x,t)$ is the velocity field, $w =\quad \times u$ is the vorticity, and $w(x,0)$ is the initial vorticity, $f(x)$ is the forcing function, and ν is the viscosity coefficient. The $(\frac{T}{2}, T]$ vorticity is predicted by learning the vorticity of $(0, \frac{T}{2}]$. The prediction accuracy is measured using Root Relative Mean Squared Error(RRMSE)(Z. Li et al., 2020).

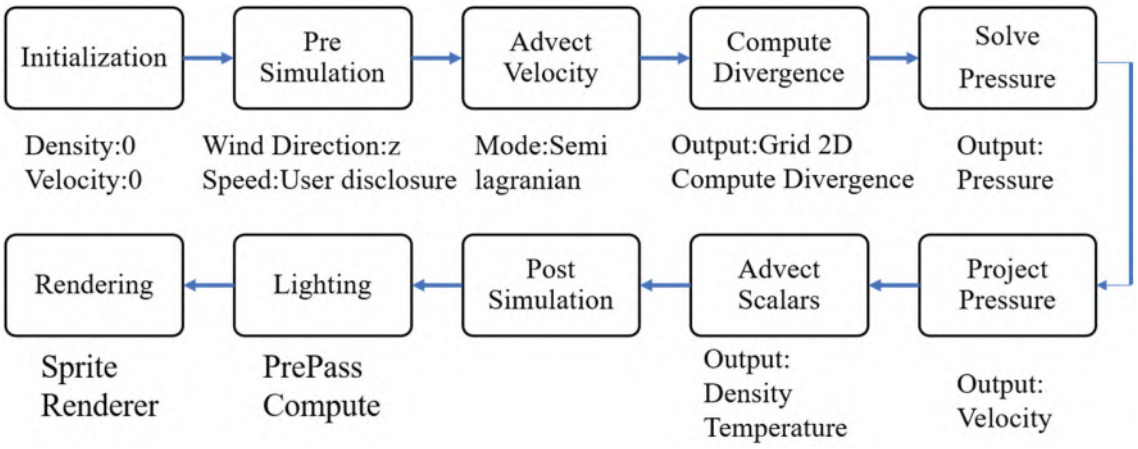


Figure 8. The process of simulating kitchen hoods to absorb fume.

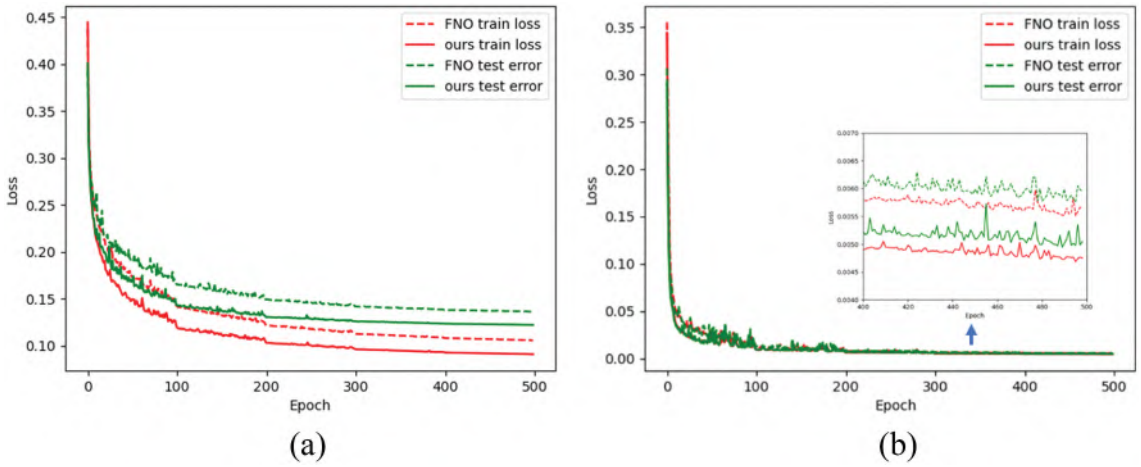


Figure 9. Train loss and test error comparison of FNO and ours, where viscosity =1e-5 (a) and viscosity =1e-3 (b).

$$RRMSE = \sqrt{\left(\frac{\sum_{i=1}^{i=n}(y_i - \hat{y}_i)^2}{n\sum y_i^2}\right)} \tag{7}$$

where y_i is the true value of the sample, $\hat{y}_i$ is the predicted value. n is the sample size.

The depicted results in Figure 9 exhibit the training loss and test error. Table 1 presents a comparative analysis between RFNO and FNO, wherein it is observed that for a viscosity $v = 1e\text{-}5, T = 20$, predicting $(10, 20]$ using $(0, 10]$, RFNO outperforms FNO by reducing the training error by 14.00% and the test error by 10.42%, the corresponding comparative analysis result is shown in Figure 10. Similarly, for a viscosity $v = 1e\text{-}3$, $T = 50$, predicting $(25, 50]$ using $(0, 25]$, RFNO yields a significant reduction in the training error by 14.29% and the test error by 15.25% as compared to FNO, the corresponding comparative analysis result is shown in Figure 11. Compared with the ground truth(the pseudo-spectral method, calculation time is 2s), both RFNO and FNO have a fast calculation time of only 0.01s.

5.2. Case: Flow around a cylinder

The effectiveness of RFNO is demonstrated by the flow around a cylinder(3D). Mesh is divided into 10,362 mesh points and 8640 triangles. To realize the mapping of the inlet velocities to the solution, the flow field data are generated using OpenFOAM. In this section, 400 sets of samples with inlet velocities ranging from 1e-2 to 4 were generated using OpenFOAM, with a convergence criterion of residuals reduced to 1e-4, and the velocity field of flow around a cylinder at an inlet velocity of 0.45 is shown in Figure 12. After these samples were shuffled and normalized, 280 groups were used for training, and 120 groups were used for testing. The number of FNO layers used for comparison with the RFNO was 8 and 10 layers.

The training results are shown in Table 2. From the results, it can be seen that the error of RFNO is much smaller, with a reduction of 14.67% for the training data and 19.78% for the test data, which indicates that RFNO can approximate the real flow field more accurately than FNO. Meanwhile, the computation time of

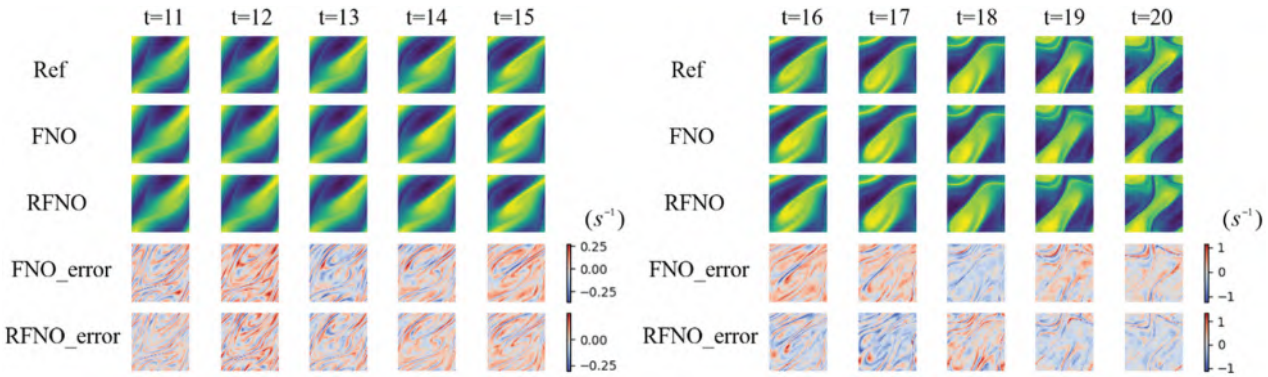


Figure 10. v = 1e−5, T=20. The results of the $(10, 20]$ vorticity field were obtained by solving the 2D Navier-Stokes equation using RFNO and FNO. The subfigures in the figure are arranged in a top-to-bottom sequence, depicting the ground truth, predictions by FNO and RFNO, and the respective errors of FNO and RFNO.

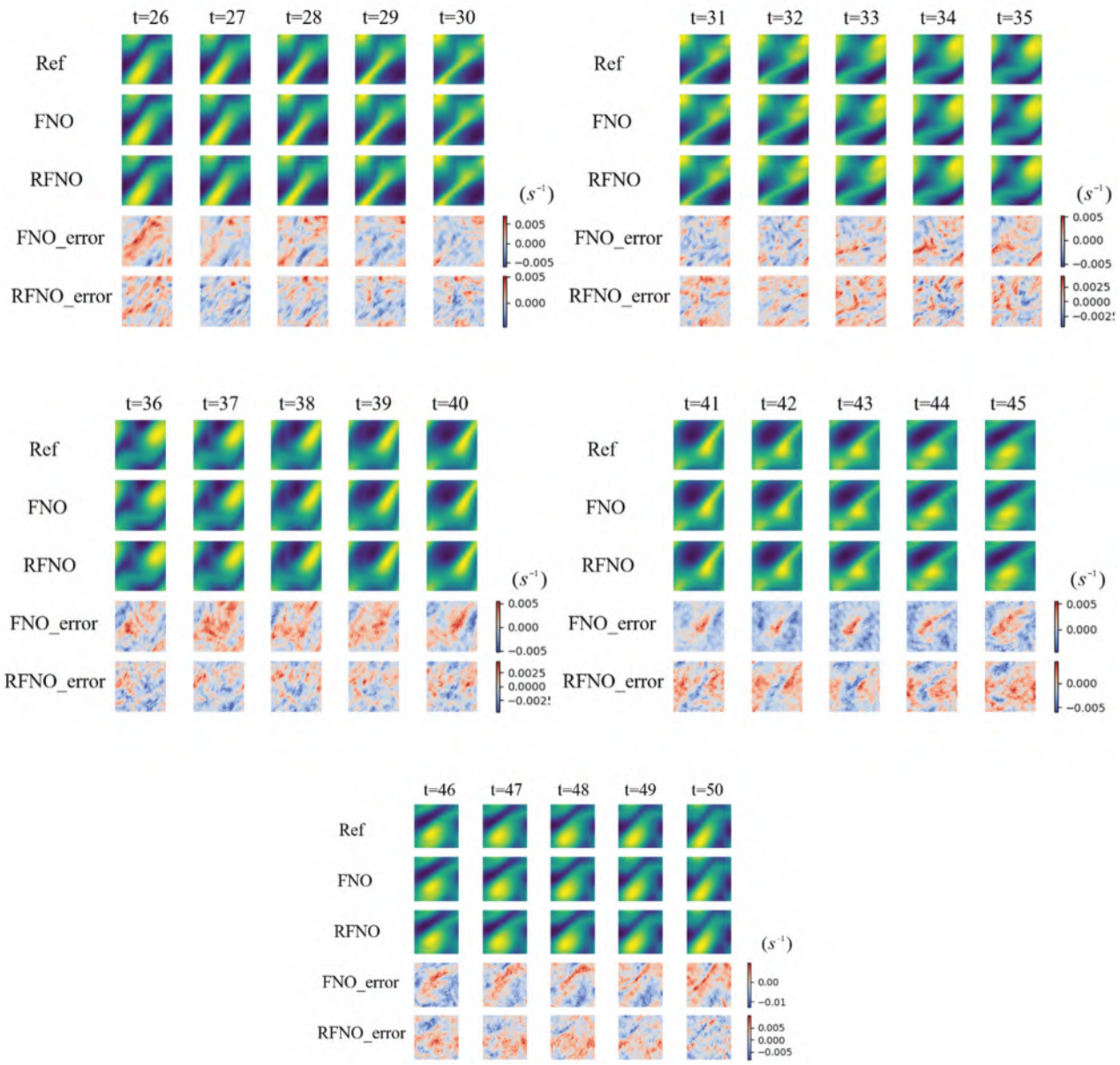


Figure 11. v = 1e−3, T=50. The results of the $(25, 50]$ vorticity field were obtained by solving the 2D Navier-Stokes equation using RFNO and FNO. The subfigures in the figure are arranged in a top-to-bottom sequence, depicting the ground truth, predictions by FNO and RFNO, and the respective errors of FNO and RFNO.

Table 1. Accuracy and time comparison of different methods.

Calculation method	Training error	Test error	Calculation time (s)
FNO(v=1e-5, T=20)	10.57%	13.63%	0.01
RFNO(v=1e-5, T=20)	9.09%	12.21%	0.01
FNO(v=1e-3, T=50)	0.56%	0.59%	0.01
RFNO(v=1e-3, T=50)	0.48%	0.50%	0.01

both RFNO and FNO models is only 0.01s, which is much smaller than the 8.50 sec required by the OpenFOAM simulation tool, and realizes the real-time mapping of the inlet conditions to the solution of the flow field. Compared with the ground truth (OpenFOAM, calculation time is 8.5s), both RFNO and FNO have a fast calculation time of only 0.01s.

The training data loss and test data error for both models are shown in Figure 13. We observe that the error rises when the number of FNO layers increases from 8 to 10, which shows the problem of accuracy limitation of FNO. In contrast, RFNO (with 4 residual modules and a total of 12 layers) can still be trained efficiently and achieve higher accuracy.

Figure 14 (a) shows the velocity field solved by the CFD-based method, (b) shows the velocity field solved by RFNO, and (c) shows the difference between the CFD-based method and the RFNO velocity field. It can be seen that RFNO can map it with high accuracy.

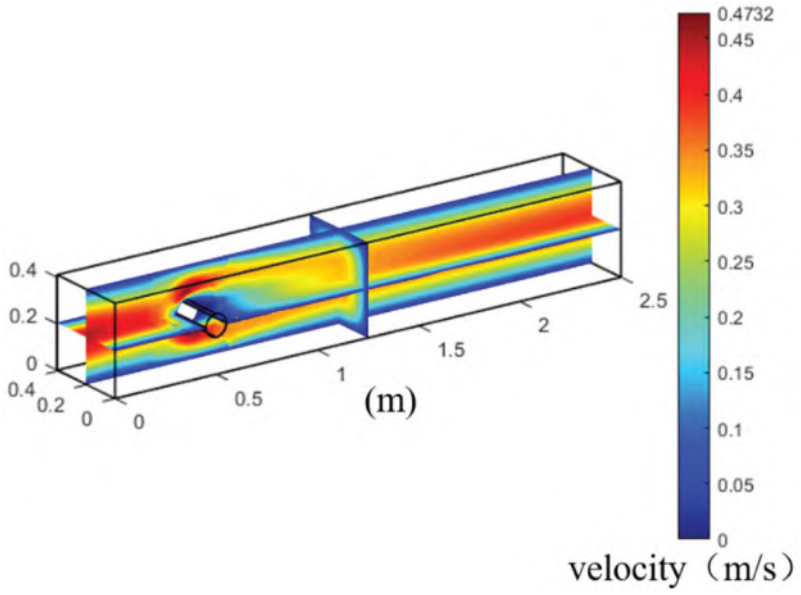


Figure 12. The velocity field of flow around a cylinder.

Table 2. Comparison of the accuracy and time of different methods for flow around a cylinder.

Calculation method	Training data error	Test data error	Calculation time (s)
FNO	6.02%	7.43%	0.01
RFNO	5.25%	5.96%	0.01

5.3. Application in the DT of smart kitchen

5.3.1. For designers: Simulation using FEA and experiment using PIV

Given the regular and simple structure of the kitchen geometric model, the focus is primarily on the 2D section where the fume is generated. The data samples are generated using FEATool Multiphysics, and the 2D model shape and mesh division are depicted in Figure 15. The kitchen shape measures 3.9 m by 2.6 m, with recesses on each side of the kitchen counter. The left counter has dimensions of 0.7 m by 0.8 m, while the right counter measures 0.55 m by 0.8 m. The grid size is set at 0.04 m, resulting in a total of 12,755 cells. The initial condition is set to zero, while the boundary conditions are such that the left counter serves as the kitchen cooking place with an inlet velocity of 0.2 m/s in the vertically upward direction, while the other positions are set to the wall with no slip.

To simulate the changes in fume flow properties under different cooking conditions, the viscosity parameters are adjusted accordingly. The resulting vorticity field when viscosity = 1e-3 is presented in Figure 16.

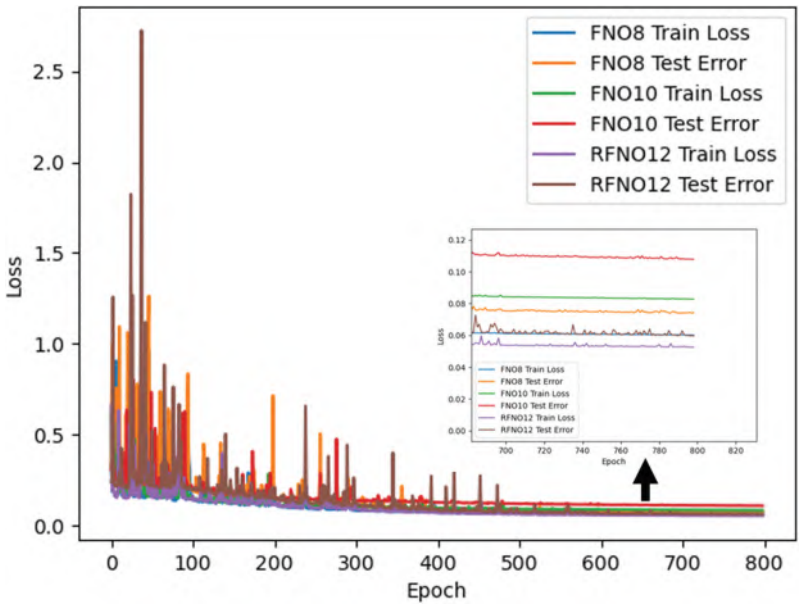


Figure 13. Comparison of training loss and test error for FNO and RFNO.

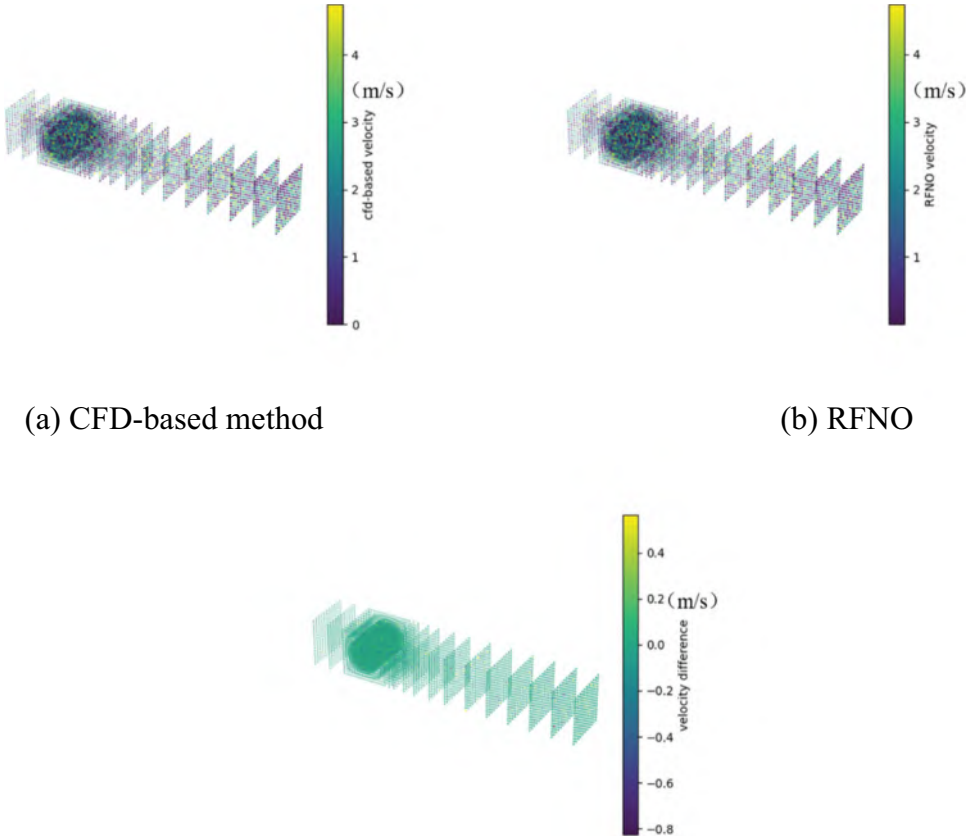


Figure 14. Comparison of velocity fields solved by the two methods.

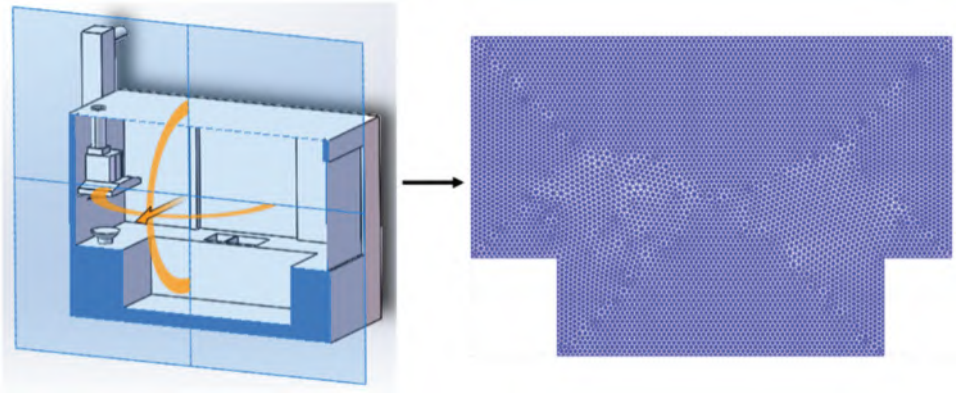


Figure 15. Kitchen geometric shape and meshing.

This study employs varying viscosities to approximate the flow field of kitchen fumes under different cooking conditions. A total of 500 data samples with viscosities ranging from 1e-4 to 1e-1 are utilized for training, with 350 sets used for training and 150 sets used for testing. The model output is the vorticity field.

Figure 17 illustrates the results for the kitchen fume flow field, with the error increasing when the number of layers is increased to 8 for 500 samples and the gradient even disappearing for 100 samples in FNO. In contrast, RFNO (with 3 residual modules and 9 layers) can still be effectively trained, achieving higher accuracy. As demonstrated in Table 3, the training error for 500 samples is reduced by 14.29% and the testing error is reduced by 7.69%, the corresponding results can be found in Figure 18. These results indicate that RFNO has better flow field approximation and generalization. In practice, the trained model can provide a solution to the flow field in just 0.01s, while FEA takes 14.86s. The proposed RFNO model can thus meet the demand

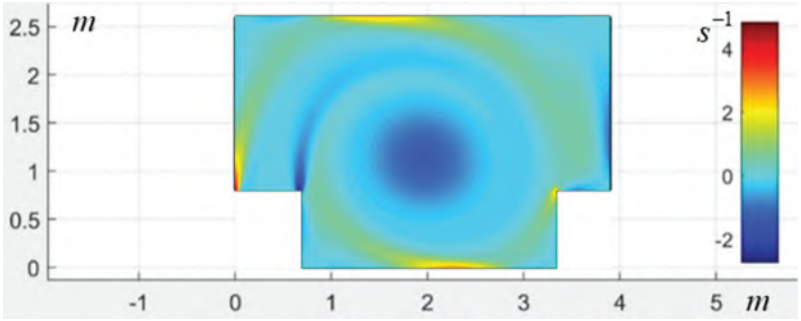


Figure 16. Vorticity field of 2D kitchen.

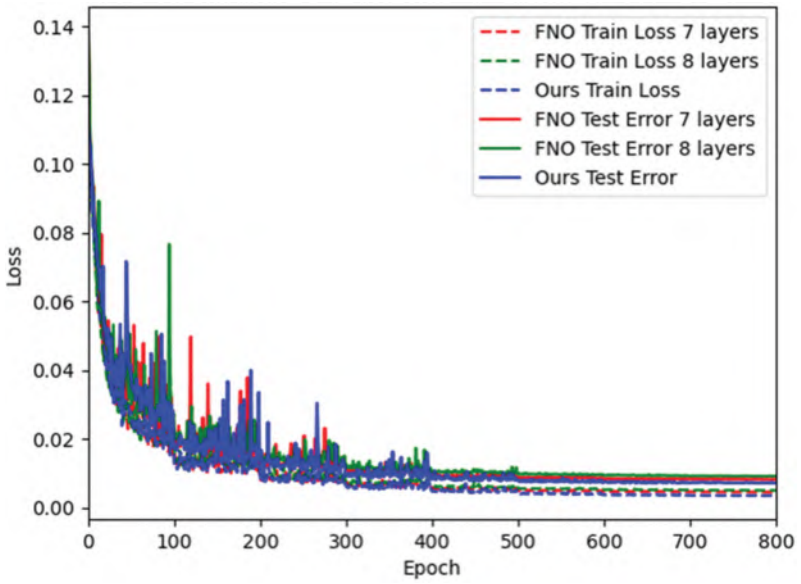


Figure 17. Training loss and test error comparison of FNO and RFNO.

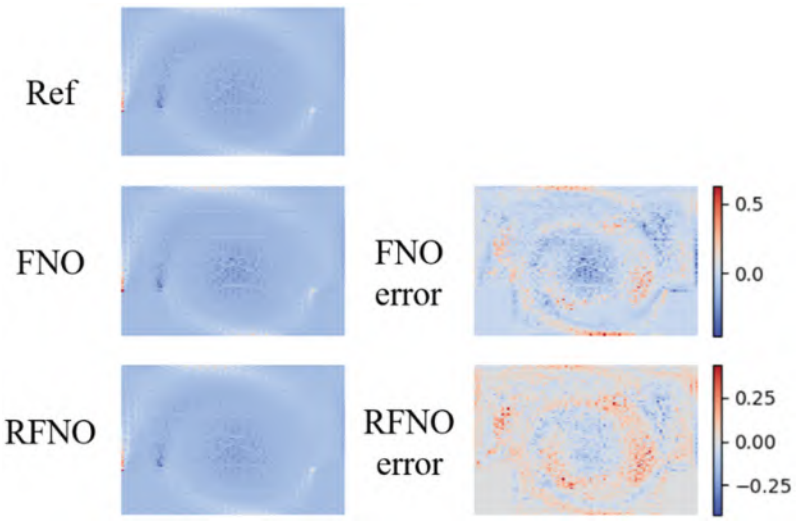


Figure 18. The subfigures in the figure depict the ground truth, results by FNO and RFNO, and the respective errors of FNO and RFNO.

Table 3. Accuracy and time comparison of different methods.

Calculation method	Training error	Test error	Calculation time (s)
FNO(500 samples)	0.56%	0.91%	0.01
RFNO(500 samples)	0.48%	0.84%	0.01

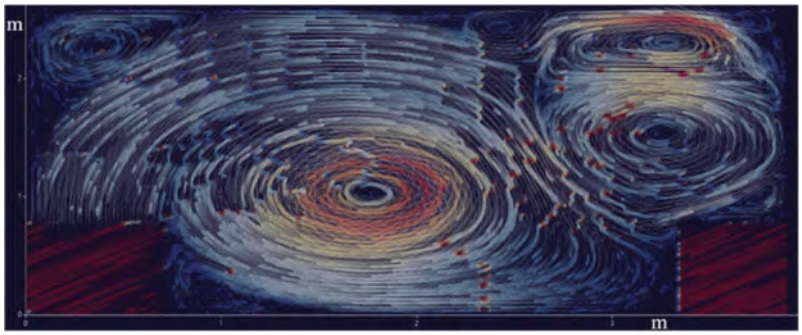


Figure 19. 2D flow field visualization of the kitchen.

for real-time mapping of the smart kitchen DT. Compared with the ground truth(FEA, calculation time is 14.86s), both RFNO and FNO have a shorter calculation time of only 0.01s.

The 2D visualization of the flow field is presented in Figure 19.

Existing online simulation methods rarely verify the validity of experimental data. Given the complex composition of the fume released at high temperatures, the simulation may not be entirely accurate. Therefore, experimental data is needed to verify the efficacy of RFNO.

PIV is a non-invasive method for measuring velocity with high spatial resolution, making it well-suited for determining the fume flow field. In this study, high-powered 2D PIV is employed to obtain the fume flow field under frontal high-gear interference conditions, and RFNO is used to construct a ROM to predict the fume flow.

The distance between the wall-mounted range hood and the pan was set at 0.6 m, with the hood in high gear and the source of smoke being stable. A total of 400 sets of fume data were collected from the front. The resolution of the vorticity field is100×100. By analyzing the PIV data, it is found that the PIV equipment will capture the non-fume flow field data when shooting, and these data greatly interfere with the prediction of the fume flow field, so it is necessary to denoise the data, and the resolution of the flow field velocity after denoising is 72×72.

We predict time steps of $[30, 35]$ by $[0, 30]$. Velocity distributions were obtained by employing an adaptive cross-correlation algorithm combined with a high-precision sub-pixel interpolation algorithm. This methodology has been verified to effectively control systematic error to within 2% (Cao et al., 2014).

The results of the experiment are presented in Figure 20, with Table 4 showcasing a comparative analysis of the training and testing errors for FNO (12 layers). and RFNO (with 4 residual modules and 12 layers). The training error is reduced by 53.27%, while the testing error is reduced by 35.59%, demonstrating the superior accuracy of RFNO in experimental settings. This study successfully implements a real data-driven prediction of the fume flow field using PIV.

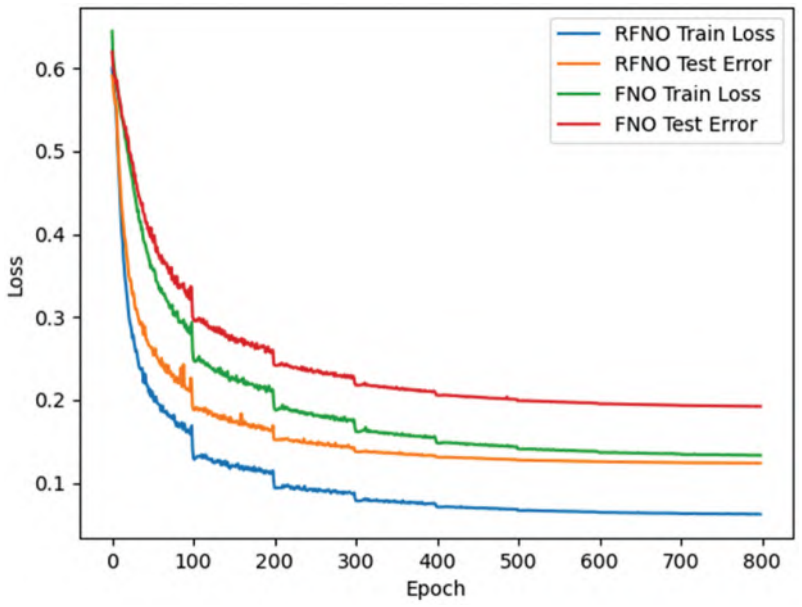


Figure 20. Training loss and test error comparison of FNO and RFNO.

Table 4. Accuracy comparison of different methods.

Calculation method	Training error	Test error
FNO	13.31%	19.22%
RFNO	6.22%	12.38%

5.3.2. For users: Sensor data-driven online prediction of air quality fields and inverse control of equipment

(1) The training of sensor data-driven online prediction model for air quality fields

To understand the impact of the kitchen flow field on the user, home environments require an analysis of the air quality field. The prediction of air quality in a smart home is very critical, and realizing accurate prediction of air quality can help to assist in decision-making and protect the health of home occupants. In this section, an attempt will be made to use the proposed RFNO prediction method for air quality prediction to analyze the impact of the kitchen flow field on home users and take appropriate measures.

To achieve the prediction, 20 sensors are installed in the home physical scenario and each sensor collects four types of data (temperature, humidity, PM2.5 concentration, and VOC concentration). The distribution of 20 sensors is shown in Figure 21. The four sensors in Figure 21 (a) (b) (c) (d) are the closest to the kitchen, which have a large variation of data and have a direct impact on the cooking staff, and will be focused later.

These multi-sensor compositions of air quality field data construct a low-resolution field. The core of this type of field prediction is temporal dependence within each sensor data time series and spatial relationships between sensors at different locations. To collect data under different cooking conditions, cooking experiments are conducted. The cooking experiments were conducted for two days, with 18,000 seconds of data on the first day and 15,000 seconds of data on the second day, with 80 sensor data per second. The data of the first day is used as training data and the data of the second day is used as test data. Specifically, the data were divided by 30 seconds and 60 seconds, which predicted the data of the last 5 seconds and the last 10 seconds. The loss of the training data of 30 seconds and 60 seconds and the decrease of the error of the test data are shown in Figure 22. The model gradually converged at 200 steps, and the accuracy of the model was stabilized at a low level. It can be considered that the prediction of the RFNO in the air quality field is effective.

The RRMSE of the error prediction results are shown in Table 5. From the prediction results, it can be seen that the model achieves good results for different time durations, especially when predicting values after 5 seconds with higher accuracy.

Figure 23, Figures 24, Figures 25, 26 show the RFNO (30-second prediction for 5 seconds) prediction of home PM2.5, temperature, humidity, and VOC data compared to the actual results, respectively. The four subgraphs of each graph show the predicted results of the four sensors closest to the kitchen (Figure 21), in order of the kitchen breathing zone (the location of the cooking staff's nasal passages), the kitchen outlet (high), the kitchen exit (low), and the living room, and these results are highly variable and the predictions are more representative of the direct impact on the cooking staff and the home environment. Due to space constraints, the other 16 sensor predictions, as well as the 30-second prediction for 10 seconds, 60-second prediction for 5 seconds, and 60-second prediction for 10 seconds are not shown here.

Overall, as shown by the comparison curves, the RFNO method proposed in this paper has high accuracy and realizes the sensing data-driven digital twin online prediction.

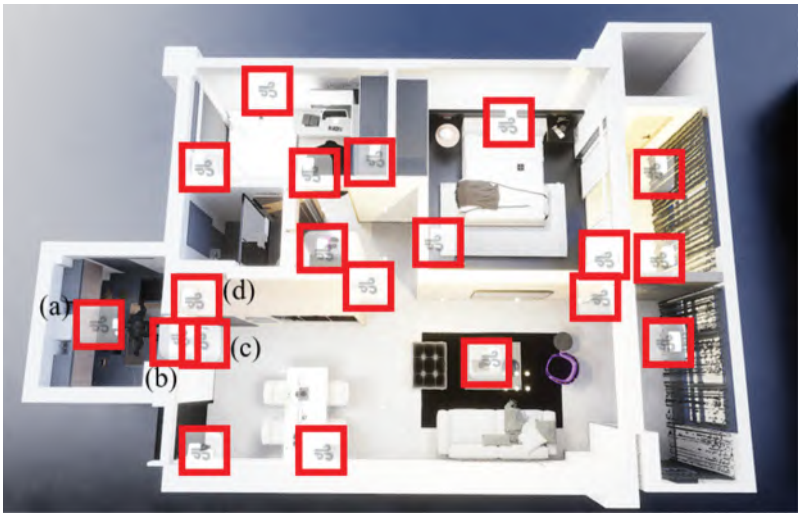
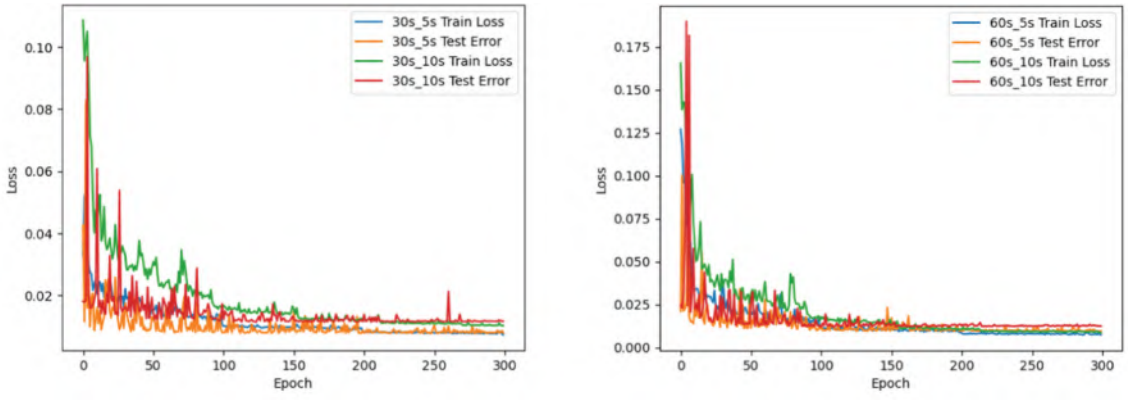


Figure 21. The distribution of 20 sensors.



(a) 30 seconds to predict 5 or 10 seconds (b) 60 seconds to predict 5 or 10 seconds

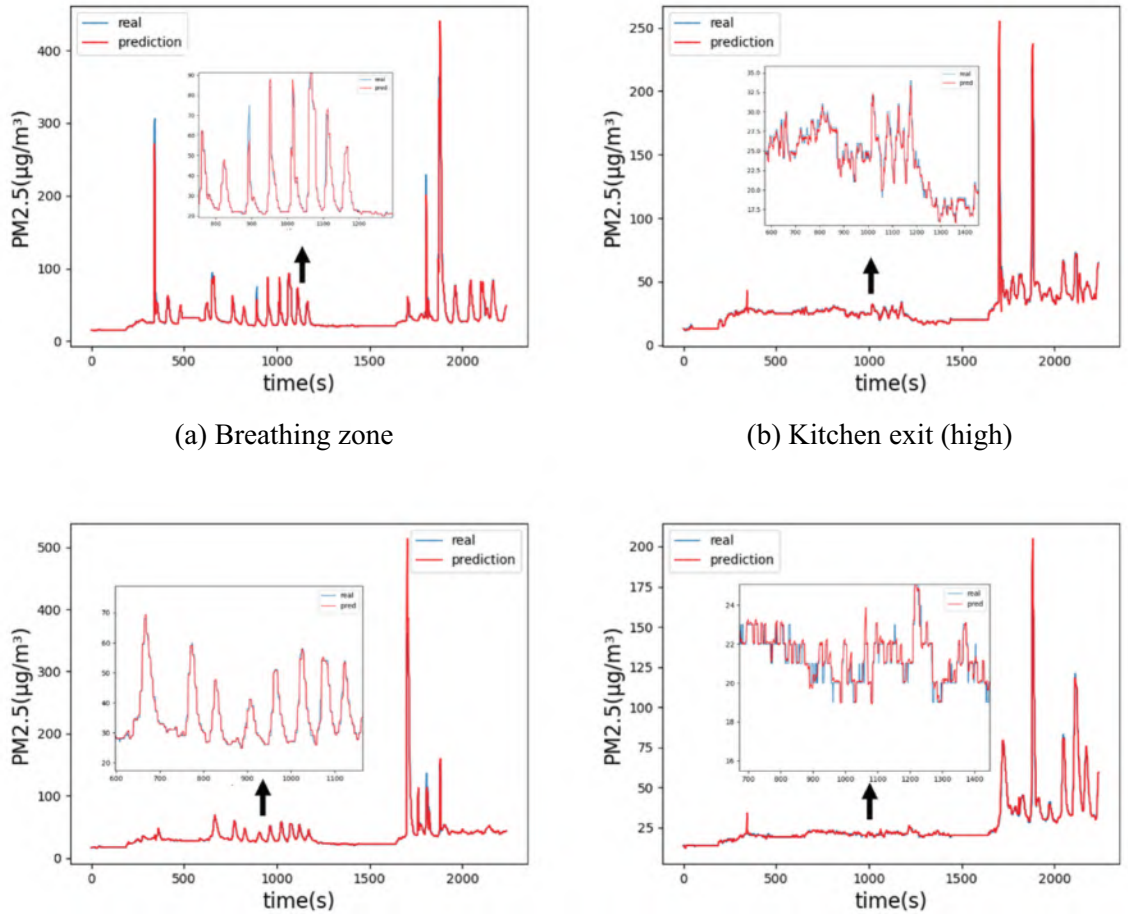
Figure 22. Training loss and test error for different time durations.

Table 5. Cooking experiment predictions.

Prediction method	Training error	Testing error
RFNO(30 seconds to predict 5 seconds)	0.66%	0.84%
RFNO(30 seconds to predict 10 seconds)	1.04%	1.18%
RFNO(60 seconds to predict 5 seconds)	0.63%	0.97%
RFNO(60 seconds to predict 10 seconds)	0.84%	1.25%

(2) The deployment of sensor data-driven online prediction model for air quality fields

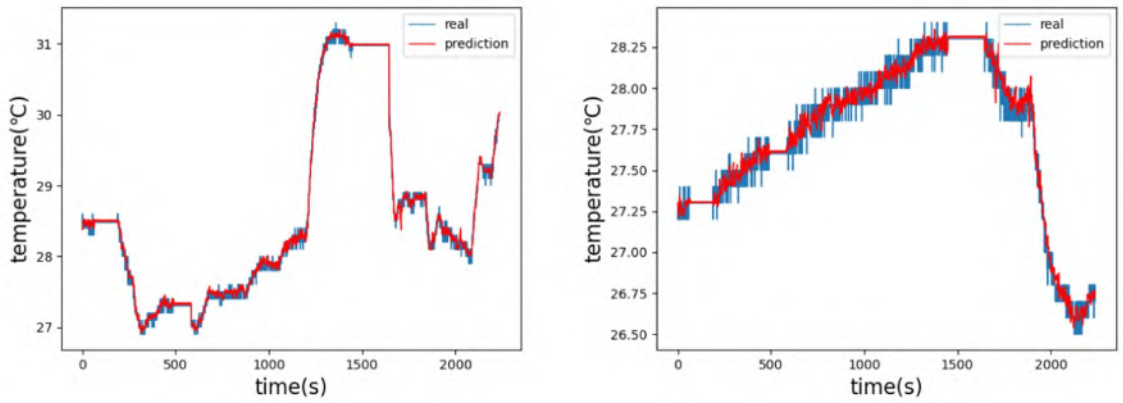
The trained physical field prediction model is integrated into the digital twin system, which can effectively sense the future occurrence of the situation and realize the sensing data-driven online prediction and assisted decision-making of the digital twin. The multi-functional integrated air sensors can monitor the temperature, humidity, VOC, and PM2.5, which can be transmitted to the local computer through the industrial-grade serial port and stored in the SQL



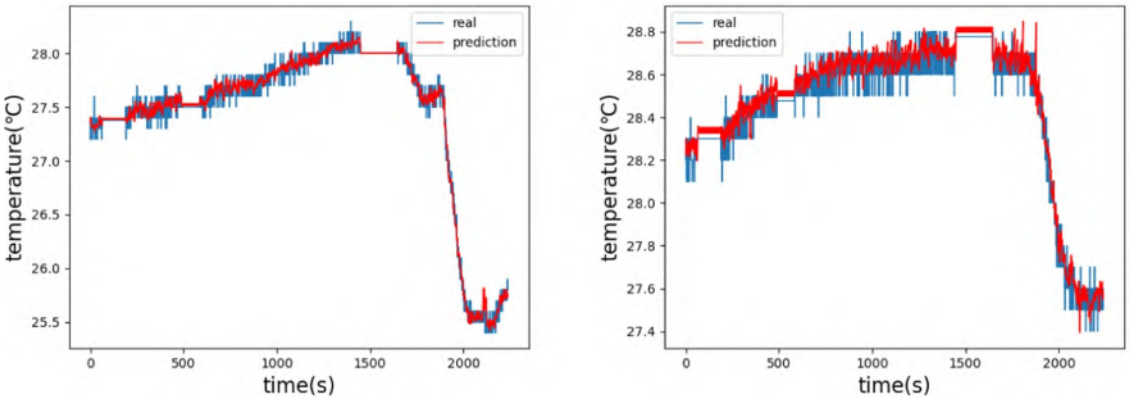
(a) Breathing zone (b) Kitchen exit (high)

(c) Kitchen exit(low) (d) Living room

Figure 23. Comparison of predicted and actual PM2.5 data at four key locations under simulated cooking.

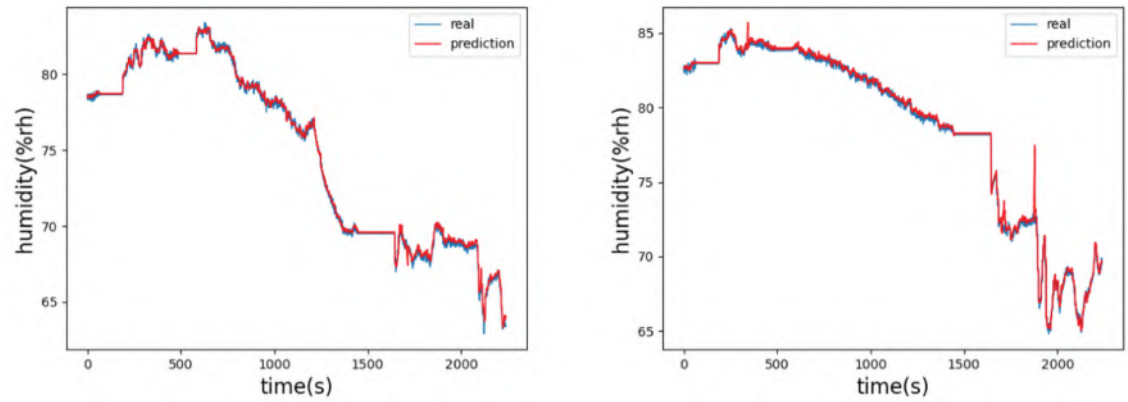


(a) Breathing zone (b) Kitchen exit (high)

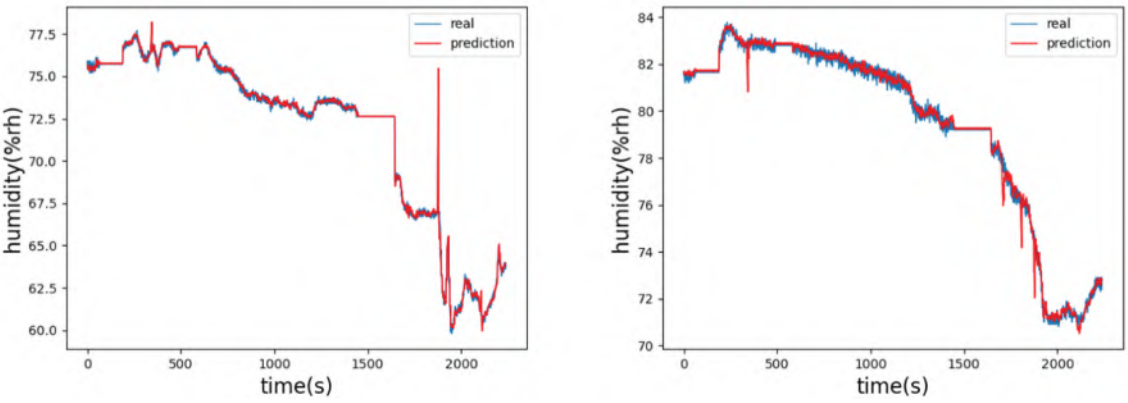


(c) Kitchen exit(low) (d) Living room

Figure 24. Comparison of predicted and actual temperature data at four key locations under simulated cooking.



(c) Kitchen exit(low) (d) Living room



(c) Kitchen exit(low) (d) Living room

Figure 25. Comparison of predicted and actual humidity data at four key locations under simulated cooking.

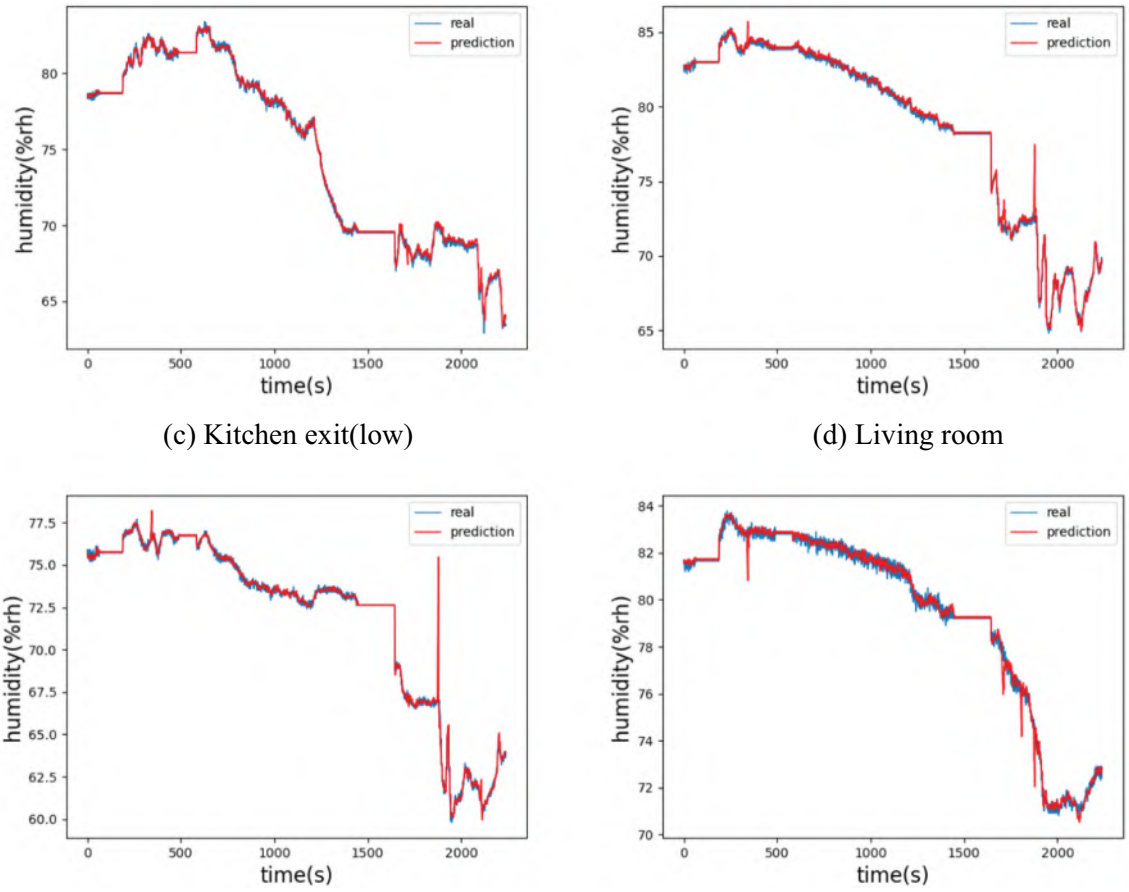


Figure 26. Comparison of predicted and actual VOC data at four key locations under simulated cooking.

Server database, using the trained RFNO prediction model can predict the data after 5 or 10 seconds based on these real-time sensor data, which is integrated and visualized in the Unreal Engine.

Specifically, by connecting to SQL Server via Python Pyodbc to read the database and read the latest sensor data, the trained RFNO model in PyTorch can read the data and output the predicted values. Using the Flask web server to provide the Get request interface, the prediction model can be integrated with the digital twin visualization platform via the REST API. In this paper, the digital twin platform in the Unreal Engine client through the VaRest Get request occurs to the Flask web server to get the data.

(3) Application of sensing data-driven online prediction model for air quality fields

Users can turn on or off the hood, adjust the hood gear, and other functions in the interactive interface. The hood of the twin scene will not change its state with it, and it will first send the corresponding control commands to the topic of the cloud platform, there are five types of hood control messages as follows:

Off: {“PowerSwitchAll”:1, “WorkMode”:0}

Power on: {“PowerSwitchAll”:2, “WorkMode”:0}

Weak Gear: {“PowerSwitchAll”:2, “WorkMode”:1, “FanLevel”:2}

Strong: {“PowerSwitchAll”:2, “WorkMode”:1, “FanLevel”:3}

Auto: {“PowerSwitchAll”:2, “WorkMode”:2}

The hood of the physical scene will execute the corresponding command after receiving the control command, and report the current status after the execution is completed. The twin device receives the status message from the hood device and parses it to update the status, realizing the virtual control and virtual-reality mapping.

In addition the user can control the hood when the deployed model predicts that the value of PM2.5 concentration in the kitchen breathing zone reaches 75 μg/m3 after 10 seconds, and considers that the air quality has reached mild pollution, then the system automatically sends control instructions to the cloud platform to turn on the hood in advance (Figure 29(b)), to make the hood turn off (Figure 29(a)) after PM2.5 is reduced to the normal concentration.

5.3.3. For users: Visualization and interaction

The visualization of the kitchen DT offers real-time information on the kitchen’s status, enabling users to monitor parameters such as temperature, PM2.5 concentration, and more. This feature not only enhances users’ ability to monitor the kitchen environment at all

Figure 27. Temperature data query for different locations of the kitchen.

Figure 28. 3D flow field visualization(arrow(a), path line(b), particle mode(c)).

times but also facilitates problem identification. As illustrated in Figure 27, the temperature of the panhandle during cooking can reach up to 64°C, posing a risk of scalding upon contact. Additionally, excessively high room temperatures may lead to food spoilage.

Utilizing the fume flow field visualization enables users to visually perceive the dispersion of fumes within the kitchen and prompts them to be attentive to the removal of grease stains or adhesions present on the hood and kitchen walls. Moreover, users can gain a comprehensive understanding of how fumes can potentially lead to poor indoor air quality and adversely impact human health, thereby empowering them to take appropriate measures to improve indoor air quality. By offering users a means to detect potential fire hazards, such as clogged hood pipes and excessive fume concentration, the visualization tool can effectively prevent fires. Additionally, the visualization tool can heighten users' safety awareness by allowing them to recognize the risks associated with fumes more clearly, thereby facilitating the development of appropriate usage habits and enhancing kitchen safety.

The fume flow field visualization is demonstrated in Figure 28, which supports arrow mode (**a**), path line mode (**b**), and particle mode (**c**). The arrow mode depicts the velocity direction, while the concentration of fumes is represented by particle density. In path line mode, the RGB colour values correspond to the X, Y, and Z components of velocity.

The functionality of the kitchen DT is further augmented by its interactive capabilities, which enable users to exercise control over kitchen equipment utilizing the DT interface. This includes the capacity to fine-tune stove fire settings and regulate hood gear, as shown in Figure 29. By leveraging this feature, users can avoid opening the hood too late and the consequent dissemination of fume, while simultaneously mitigating the risk of exposure to hazardous temperatures and other associated dangers during equipment operation.

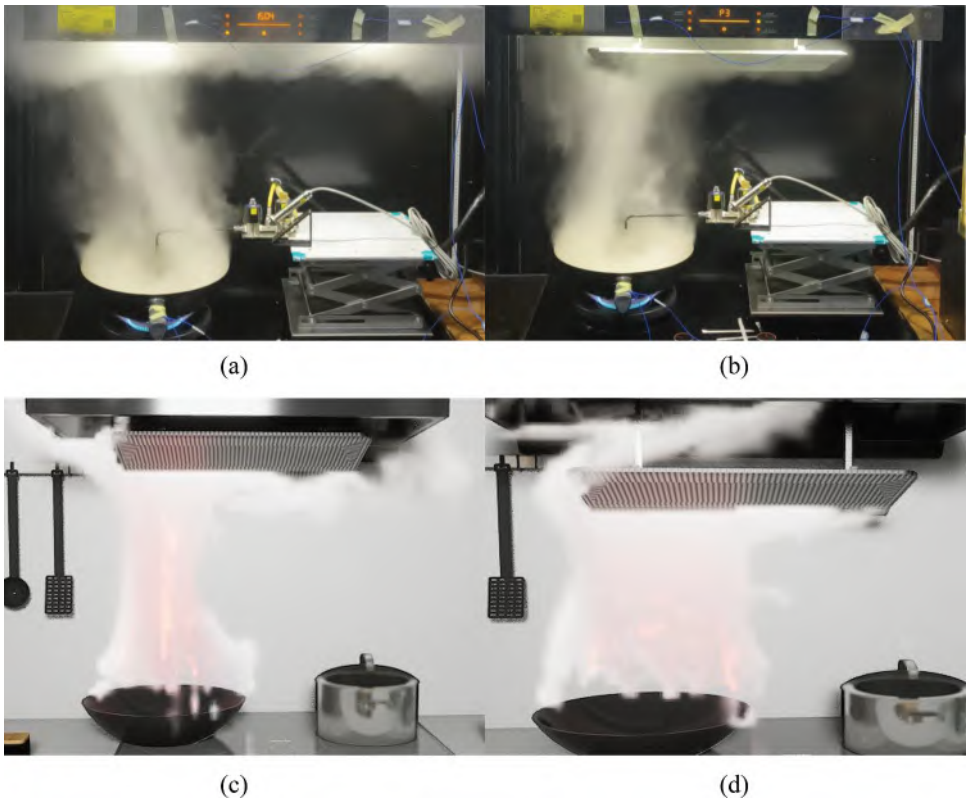


Figure 29. (a) and (b) illustrate the fume flow patterns observed in the actual kitchen, under low and high-gear hood working conditions, respectively. In contrast, (c) and (d) depict the corresponding fume flow patterns in the virtual kitchen, also under low and high gear hood working conditions, respectively.

6. Conclusion

In this paper, we build a DT framework for the smart kitchen, which encompasses data processing, flow field online simulation, equipment monitoring and control, interaction, and visualization. The DT facilitates real-time monitoring and prediction of kitchen air quality and temperature, as well as the calculation and visualization of the kitchen flow field in real-time. The smart kitchen DT offers significant benefits for designers, enabling them to design appropriate kitchen appliances, improve development efficiency, and make more accurate predictions. For users, the DT offers valuable insights into the current state and dynamic evolution of the kitchen environment, thereby elevating the overall user experience to a higher level. To address the critical online simulation and visualization aspects of the DT, we propose an RFNO online simulation method, as well as an Echarts-based 2D flow field web visualization and UE5-based 3D flow field particle interactive visualization method. We validate the proposed approach using public datasets, simulation data, and experimental data.

The RFNO proposed in this paper is a data-driven approach that depends on the quality and quantity of data. We have generated several thousand sets of simulation data through the FEATool Multiphysics toolbox and OpenFOAM software in verifying the effectiveness of the method. The research in this paper tests the validity of the methodology of this paper only on simulation data sets as well as on2D3C's PIV data. Meanwhile, the method of this paper is utilized to analyze the impact of the kitchen flow field on the users through the prediction of air quality from real-time sensor data. However, there are more challenging flow field problems in industrial scenarios. It is very difficult to obtain enough training samples of real flow fields. In the future, more attention needs to be paid to obtaining high-quality flow field samples, such as 3D3C's PIV, to continue exploring the feasibility of applying this method to 3D flow fields.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the Ningbo Key Research and Development Programme under Grant No. 2023Z134, and the National Natural Science Foundation of China under Grant Nos. U22A6001, 52105279.

ORCID

Daxin Liu http://orcid.org/0000-0002-0379-1990

References

Buonanno, G., Stabile, L., Morawska, L., Giovinco, G., & Querol, X. (2017). Do air quality targets really represent safe limits for lung cancer risk? *Science of the Total Environment*, *580*, 74–82. https://doi.org/10.1016/j.scitotenv.2016.11.216

Cabral, B., & Leedom, L. C. (1993, September). Imaging vector fields using line integral convolution. In Proceedings of the 20th annual conference on Computer graphics and interactive techniques, Anaheim, CA (pp. 263–270).

Cao, X., Liu, J., Pei, J., Zhang, Y., Li, J., & Zhu, X. (2014). 2D-PIV measurement of aircraft cabin air distribution with a high spatial resolution. *Building & Environment*, *82*, 9–19. https://doi.org/10.1016/j.buildenv.2014.07.027

Casas, L., Tischer, C., Tiesler, C., Brüske, I., Koletzko, S., Bauer, C. P., Wichmann, H.-E., von Berg, A., Berdel, D., Krämer, U., Schaaf, B., Lehmann, I., Herbarth, O., Heinrich, J., & GINIplus and LISAplus Study Group. (2012). Association of gas cooking with children's respiratory health: Results from GINIplus and LISAplus birth cohort studies. *Indoor Air*, *22*(6), 476–482. https://doi.org/10.1111/j.1600-0668.2012.00784.x

Epic Games. (2021). *UnrealEngine 5 Documentation*. https://docs.unrealengine.com/5.0/en-US/index.html

Gopinath, V., Srija, A., & Sravanthi, C. N. (2019, May). Redesign of smart homes with digital twins. *Journal of Physics Conference Series*, *1228*(1), 012031. IOP Publishing. https://doi.org/10.1088/1742-6596/1228/1/012031

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354–377. https://doi.org/10.1016/j.patcog.2017.10.013

Han, X., Gao, H., Pfaff, T., Wang, J. X., & Liu, L. P. (2022). Predicting physics in mesh-reduced space with temporal attention. arXiv preprint arXiv:2201.09113.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA (pp. 770–778).

Kamat, V. R., & Martinez, J. C. (2001). Visualizing simulated construction operations in 3D. *Journal of Computing in Civil Engineering*, *15*(4), 329–337. https://doi.org/10.1061/(ASCE)0887-3801(2001)15:4(329)

Kriushichev, A. G., Silkina, N. S., & Nenakhova, E. A. (2020, November). Interactive visualization for simulating the movement of liquids and gases. In Global Smart Industry Conference (GloSIC), Chelyabinsk, Russia (pp. 370–374). IEEE.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2020). Fourier neural operator for parametric partial differential equations. arXiv preprint arXiv:2010.08895.

Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, *3*(3), 218–229. https://doi.org/10.1038/s42256-021-00302-5

Maryasin, O. (2019, September). Home automation system ontology for digital building twin. In XXI International Conference Complex Systems: Control and Modeling Problems (CSCMP), Samara, Russia (pp. 70–74). IEEE.

Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., & Battaglia, P. W. (2020). Learning mesh-based simulation with graph networks. arXiv preprint arXiv:2010.03409.

Rowley, C. W., & Dawson, S. T. (2017). Model reduction for flow analysis and control. *Annual Review of Fluid Mechanics*, *49*(1), 387–417. https://doi.org/10.1146/annurev-fluid-010816-060042

Tezduyar, T., Aliabadi, S., Behr, M., Johnson, A., Kalro, V., & Litke, M. (1996). Flow simulation and high performance computing. *Computational Mechanics*, *18*(6), 397–412. https://doi.org/10.1007/BF00350249

Tuegel, E. J., Ingraffea, A. R., Eason, T. G., & Spottswood, S. M. (2011). Reengineering aircraft structural life prediction using a digital twin. *International Journal of Aerospace Engineering*, *2011*, 1–14. https://doi.org/10.1155/2011/154798

Xu, H., Berres, A., Yoginath, S. B., Sorensen, H., Nugent, P. J., Severino, J., Tennille, S. A., Moore, A., Jones, W., & Sanyal, J. (2023). Smart mobility in the cloud: Enabling real-time situational awareness and cyber-physical control through a digital twin for traffic. *IEEE Transactions on Intelligent Transportation Systems*, *24*(3), 3145–3156. https://doi.org/10.1109/TITS.2022.3226746

Yang, Y., Shi, Q., Zhang, Z., Shan, X., Salam, B., & Lee, C. (2023). Robust triboelectric information-mat enhanced by multi-modality deep learning for smart home. *InfoMat*, *5*(1), e12360. https://doi.org/10.1002/inf2.12360

Appendix

All the synonyms that are used in this paper

Synonyms	Explanation
DT	Digital twin
FEA	Finite Element Analysis
FNO	Fourier Neural Operator
GNS	Graph networks
IoT	Internet of Things
JSON	JavaScript Object Notation
PINN	Physics-informed Neural Network
PIV	Particle Image Velocimetry
POD	Proper Orthogonal Decomposition
PM2.5	Particulate matter less than or equal to 2.5 microns in diameter
RFNO	Residual-based Fourier Neural Operator
ROM	Reduced-order model
UE5	Unreal Engine 5
VOC	Volatile organic compounds
CNN	Convolutional Neural Network
CPU	Central Processing Unit
GPU	Graphics Processing Unit
UMG	Unreal Motion Graphics UI Designer