



# Uncertainty-Aware Digital Twins: Robust Model Predictive Control Using Time-Series Deep Quantile Learning

**Yi-Ping Chen**

Department of Mechanical Engineering,  
Northwestern University,  
Evanston, IL 60208  
e-mail: chenyp@u.northwestern.edu

**Ying-Kuan Tsai**

Department of Mechanical Engineering,  
Northwestern University,  
Evanston, IL 60208  
e-mail: yingkuan.tsai@northwestern.edu

**Vispi Karkaria**

Department of Mechanical Engineering,  
Northwestern University,  
Evanston, IL 60208  
e-mail: vispikarkaria2026@u.northwestern.edu

**Wei Chen<sup>1</sup>**

Department of Mechanical Engineering,  
Northwestern University,  
Evanston, IL 60208  
e-mail: weichen@northwestern.edu

*Digital twins, virtual replicas of physical systems that enable real-time monitoring, model updates, predictions, and decision-making, present novel avenues for proactive control strategies for autonomous systems. However, achieving real-time decision-making in digital twins considering uncertainty necessitates an efficient uncertainty quantification (UQ) approach and optimization driven by accurate predictions of system behaviors, which remains a challenge for learning-based methods. This article presents a simultaneous multistep robust model predictive control (MPC) framework that incorporates real-time decision-making with uncertainty awareness for digital twin systems. Leveraging a multi-step-ahead predictor named time-series dense encoder (TiDE) as the surrogate model, this framework differs from conventional MPC models that provide only one-step-ahead predictions. In contrast, TiDE can predict future states within the prediction horizon in one shot, significantly accelerating MPC. Furthermore, quantile regression is employed with the training of TiDE to perform flexible and computationally efficient UQ on data uncertainty. Consequently, with the deep learning quantiles, the robust MPC problem is formulated into a deterministic optimization problem and provides a safety buffer that accommodates disturbances to enhance the constraint satisfaction rate. As a result, the proposed method outperforms existing robust MPC methods by providing less conservative UQ and has demonstrated efficacy in an engineering case study involving directed energy deposition (DED) additive manufacturing. This proactive, uncertainty-aware control capability positions the proposed method as a potent tool for future digital twin applications and real-time process control in engineering systems. [DOI: 10.1115/1.4069104]*

**Keywords:** digital twin, robust model predictive control, real-time decision-making, time-series, deep neural network, quantile learning

## 1 Introduction

**1.1 Problem Definition.** The concept of digital twins [1,2] has shown promising revolutions in autonomous industries such as manufacturing [3–5] and predictive maintenance [6]. It brings the idea of building bidirectional interactions between the physical system and its virtual counterpart. This enables online decision-making processes to be conducted automatically utilizing the state prediction provided by the virtual systems, and reacts proactively in response to the feedback from the physical systems [7]. One embodiment of online decision-making for digital twins is via model predictive control (MPC) [8], which optimizes system performance by predicting future behavior and adjusting control inputs in real-time based on the model prediction. To account for disturbances in MPC, a family of uncertainty-aware MPC

approaches has been proposed to enhance constraint satisfaction rates in the presence of anticipated uncertainty. Methods such as stochastic MPC [9] and robust MPC [10] aim to approximate uncertainty propagation through the known system dynamics, quantify the distribution of predicted states, and solve the MPC problem by explicitly incorporating uncertainty bounds with the constraints. As the accurate description of the system dynamics may be unavailable a priori, with the recent advancements in machine learning and neural networks (NNs) [4], learning-based or data-driven predictive controllers [11,12] have gained significant attention. However, although NNs can emulate the system dynamics accurately, applying NNs in uncertainty-aware MPC presents significant challenges, as quantifying and estimating uncertainty distributions can be both complex and computationally expensive. Consequently, integrating NN-based models with efficient uncertainty quantification (UQ) methods for decision-making remains an open research question critical to the advancement of digital twin technologies.

**1.2 Uncertainty-Aware Model Predictive Control.** Uncertainty-aware MPC methods can be broadly categorized into robust and stochastic approaches [8]. Robust MPC focuses on

<sup>1</sup>Corresponding author.  
Contributed by the Design Automation Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received January 15, 2025; final manuscript received June 17, 2025; published online August 4, 2025. Assoc. Editor: Pingfeng Wang.

optimizing control inputs to perform effectively under worst-case scenarios, ensuring system stability and constraint satisfaction even under bounded uncertainties [10]. Stochastic MPC leverages probabilistic models to incorporate uncertainties into the optimization process and formulate probabilistic (or chance) constraints in an optimal control problem [9], aiming to achieve a balance between performance and reliability. Techniques like min-max MPC formulate the cost function as the maximum of cost values with the samples generated based on disturbance models [13]. However, min-max MPC often comes with significant computational overhead, posing challenges for real-time implementation [14–16]. To make these optimal control problems more computationally tractable, tube-based techniques have been explored to solve the robust and stochastic MPC problems by explicitly identifying the uncertainty regions in state and control action spaces [16–19]. However, many of these approaches still assume prior knowledge of system dynamics or disturbance characteristics, limiting their applicability in real-world scenarios with incomplete or evolving information. In recent years, data-driven modeling, which can capture complex temporal dependencies and nonstationary dynamics, has gained attention for MPC frameworks. By leveraging these advances, the integration of data-driven methods into MPC frameworks offers new opportunities to improve both performance and adaptability under uncertain conditions.

**1.3 Data-Driven Model Predictive Control.** Data-driven models are essential for surrogating physics in a digital twin and MPC, particularly in two key scenarios: when the system's underlying physics is overly complex or not fully understood, and when simulations are prohibitively computationally expensive or time-consuming [20]. Under these circumstances, data-driven/learning-based methods can identify the system directly using observational data. For example, neural state space models [21] can replace the system and input matrices in a state space formulation. Recursive neural network and long short-term memory (LSTM) are also popular options since their structures resemble the propagation of the dynamics of the systems [22–24]. However, enabling learning-based methods with uncertainty awareness for real-time applications is still challenging, primarily due to the computational complexity of performing UQ. Popular UQ techniques for NNs, such as ensemble methods, Bayesian NNs, Monte Carlo (MC) dropout, and bootstrapping, fall under the category of sampling-based methods [25]. While these methods can numerically approximate the distribution of NN outputs [26], their reliance on Monte Carlo sampling and multiple forward passes of NNs suffers from significant computational time, rendering them impractical for many engineering applications involving MPC.

In contrast, parametric methods, which estimate the parameters of the uncertainty distribution directly, provide a computationally efficient alternative and are widely applied in learning-based MPC. For instance, Kinky inference has been employed to learn parameters representing the bounds of system states [11]. Similarly, in Ref. [27], a Gaussian process (GP) is utilized as a discrepancy model to capture unknown system dynamics, with its predictive uncertainty serving as a probabilistic bound for nominal predictions. One noteworthy approach is quantile regression, which directly learns user-defined quantiles of the data. Unlike methods that require assumptions about the data distribution, quantile regression offers greater flexibility, making it particularly appealing for MPC application [28].

Another challenge in data-driven MPC lies in the significant computational cost of solving optimization problems online. Unlike linear MPC, which benefits from efficient closed-form or quadratic programming-based solutions via linear quadratic regulation (LQR) [8], data-driven MPC relies on surrogate models, e.g., black-box functions such as neural networks, which require iterative numerical solvers and multiple forward evaluations. This becomes particularly burdensome when using one-step-ahead predictors, which must be rolled out recursively to generate the full trajectory prediction.

**1.4 Model Predictive Control With Multistep-Ahead Predictors.** To alleviate the computational burden, a growing body of research has turned toward multistep-ahead predictors, taking advantage of the recent advances in machine learning. These models generate the entire future trajectory in a single forward pass, reducing the number of function evaluations during optimization and thus significantly lowering computational cost. For example, Park et al. [29] demonstrated that using multistep predictors based on transformer architectures yielded substantial improvements in runtime and prediction accuracy, especially for longer horizons. This benefit is particularly attractive in real-time MPC settings, where computational efficiency is critical.

Moreover, multistep predictors provide structural advantages for UQ [30,31]. Traditional robust MPC approaches, such as tube-based methods, typically require recursive uncertainty propagation, which can lead to overly conservative UQ. Recent work, such as Ref. [30], highlights that multistep predictors allow uncertainty to be learned directly from data at the trajectory level, bypassing the need for recursive propagation and enabling simpler, more intuitive bounds. This is especially valuable in data-driven contexts, where the model structure may not be fully known, and where robust decision-making must account for uncertainty without sacrificing computational tractability. Uncertainty estimation in existing methods for robust MPC with multistep-ahead predictors [30,32] focuses on deriving the worst-case using linear models with set membership identification. Although they provide interpretability and theoretical rigor in estimating error bounds, these approaches are limited in linear models and can hardly be generalized to NN-based applications.

In parallel, advances in time-series forecasting have demonstrated the effectiveness of quantile regression integrated with sequence-to-sequence deep learning models for multistep uncertainty estimation [33,34]. These models can capture complex temporal patterns and provide prediction intervals across time, offering a natural fit for robust decision-making that requires trajectory-wise uncertainty estimates. Taken together, these trends point to multistep-ahead prediction and direct UQ learning, such as quantile regression, as promising directions for improving both efficiency and robustness in data-driven MPC, particularly under real-time and uncertain environments in digital twin applications.

**1.5 Research Objective.** This work introduces a simultaneous multistep robust MPC framework that leverages time-series deep neural networks and deep quantile learning to enable fast, uncertainty-aware decision-making in digital twins of complex engineering systems. While previous works have explored multistep predictions to accelerate MPC [29] and employed quantile regression to quantify uncertainty in single-step MPC [28], this work is, to the best of our knowledge, the first to unify these two paradigms, enabling deep learning quantiles to be explicitly used as predicted bounds in a multistep robust MPC setting. By combining the computational efficiency of multistep-ahead predictors with the flexibility and accuracy of trajectory-level uncertainty quantification, our framework opens a new direction for robust MPC that is both data-driven and scalable. As MPC is considered a popular model-based decision-making approach that can be built on existing digital twin frameworks, such as Refs. [1,3,4,35], this proposed method pushes the limit of current MPC, providing a new option for decision-making under uncertainty in digital twin applications.

The proposed framework, depicted in Fig. 1, comprises two stages. In stage 1, noisy system data are gathered as the training data. A time-series deep neural network, named time-series dense encoder (TiDE), is employed to perform nonlinear system identification, capturing both the nominal system dynamics and the quantiles of the data uncertainty, encompassing the uncertainty of the nominal prediction. Subsequently, in stage 2, TiDE serves as the predictive model (virtual system) operating with the proposed multistep robust MPC as the virtual-to-physical integration. The nominal prediction (median) is utilized to assess the reference

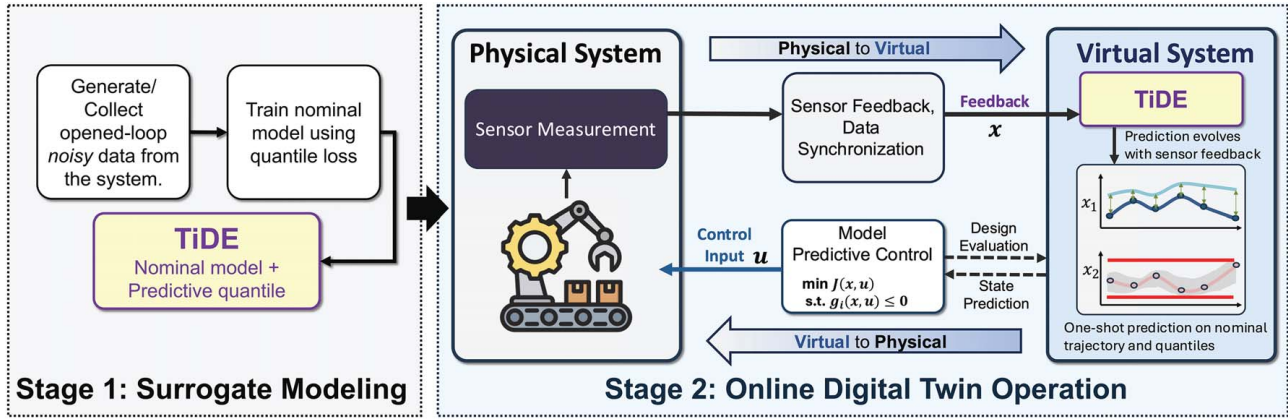


Fig. 1 The proposed multi-step robust MPC framework

tracking performance, while the predictive quantiles are employed to guarantee constraint satisfaction. This proposed method is validated using an illustrative example and an engineering case study in additive manufacturing (AM). The contributions of this work include:

- We propose a robust MPC framework for multistep-ahead prediction models as an embodiment of uncertainty-aware real-time decision-making for digital twins.
- We demonstrate the effectiveness of deep learning quantiles in quantifying data uncertainty.
- We validate the proposed methods using several case studies, showing the generality of this method in the digital twin paradigm.

The rest of the article is structured as follows: In Sec. 2, the technical background of MPC, TiDE, and quantile regression will be introduced. Section 3 details the proposed robust MPC framework, including problem formulation, model preparation, and optimization techniques. In Sec. 4, a numerical model is used as a demonstration to walk through the implementation details, and the result in an engineering case study on AM is revealed in Sec. 5. Lastly, we will conclude this work in Sec. 6.

## 2 Technical Background

**Notation:** The sets of real numbers and non-negative integers are denoted by  $\mathbb{R}$  and  $\mathbb{N}_{\geq 0}$ , respectively. Given  $a, b \in \mathbb{N}_{\geq 0}$  such that  $a < b$ , we denote  $\mathbb{N}_{[a,b]} := \{a, a+1, \dots, b\}$ .  $[\mathbf{A}]_i$  and  $[\mathbf{a}]_i$  denote the  $i$ th row and element of the matrix  $\mathbf{A}$  and vector  $\mathbf{a}$ , respectively.  $\hat{\mathbf{x}}_{k+i}$  denotes the  $i$ -step-ahead predicted value of  $\mathbf{x}$  at time  $k$ . The notation  $\mathbf{I}_{a \times a}$  denotes an  $a$ -by- $a$  identity matrix,  $\mathbf{Q} > 0$  denotes a positive definite matrix, and  $\|\mathbf{x}\|_{\mathbf{Q}}^2 = \mathbf{x}^T \mathbf{Q} \mathbf{x}$  refers to a quadratically weighted norm. Given a random variable  $X$ ,  $\mathbb{E}[X]$  denotes its expected value. A Gaussian distribution with mean vector  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$  is represented as  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ . Given two sets  $A$  and  $B$ , then  $A \oplus B := \{a+b | a \in A, b \in B\}$  (Minkowski sum) and  $A \ominus B := \{a \in A | a+b \in A, \forall b \in B\}$  (Pontryagin difference).

**2.1 Model Predictive Control and Robust Model Predictive Control.** MPC), also known as receding horizon optimal control, is an advanced dynamic model of the system to predict and optimize future control actions within a finite horizon [8]. This ensures that constraints on inputs and outputs are met while minimizing a specified cost function. MPC iteratively solves an optimization problem at each time-step, applies the resulting control action, and repeats the process as the time horizon advances, as depicted in Figs. 2(a) and 2(b).

Assume that a general nonlinear system can be represented as

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k, \mathbf{u}_k) \quad \forall k \in \mathbb{N} \quad (1)$$

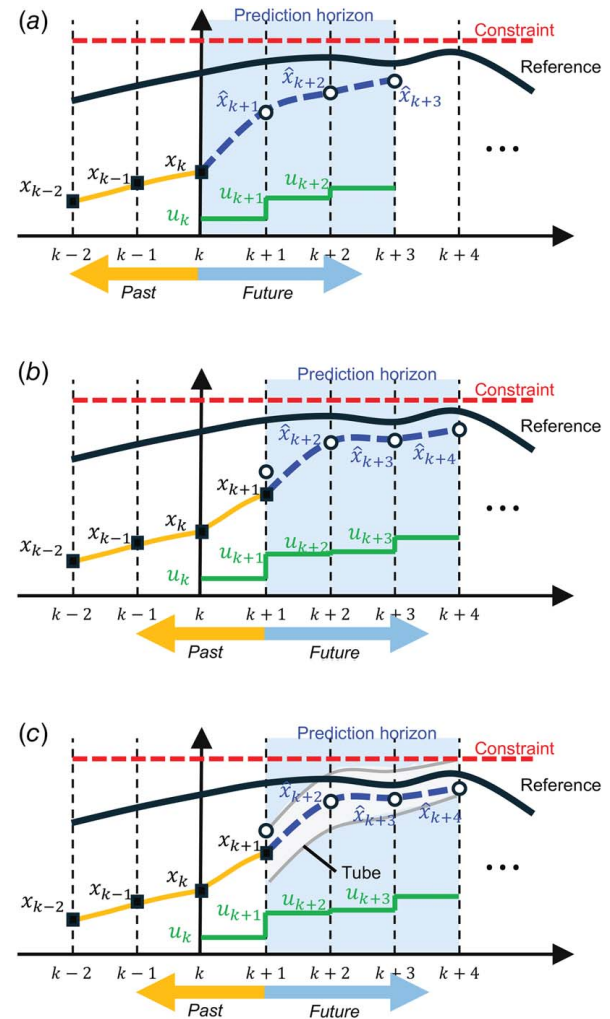


Fig. 2 Illustration of MPC and robust MPC. (a) Illustration of MPC at time  $k$ . (b) Illustration of MPC at time  $k+1$ . (c) Illustration of robust MPC. The green lines/stairs at the bottom is the optimal control input sequence, and the blue dashed lines are the state predictions from the model given the optimal control inputs. The gray tube in (c) surrounding the dashed lines represents the quantified uncertainty.



where  $F$  denotes the dynamic function that maps from the current state and control action to the state at the next step. With the prediction horizon length noted as  $N$  and the specified reference  $\mathbf{r} = [\mathbf{r}_{k+1}, \dots, \mathbf{r}_{k+N}]$ , the MPC can be formulated as an optimization problem for the future control inputs  $\mathbf{u} = [\mathbf{u}_k, \dots, \mathbf{u}_{k+N-1}]$ :

$$\min_{\mathbf{u}} J(\mathbf{u}, \hat{\mathbf{x}}, \mathbf{r}) = \sum_{i=0}^{N-1} \left[ \|\hat{\mathbf{x}}_{k+i+1} - \mathbf{r}_{k+i+1}\|_{\mathbf{Q}}^2 + \|\mathbf{u}_{k+i}\|_{\mathbf{R}}^2 \right] \quad (2a)$$

$$\text{s.t. } \hat{\mathbf{x}}_{k+1} = \hat{F}(\mathbf{x}_k, \mathbf{u}_k) \quad (2b)$$

$$\hat{\mathbf{x}}_{k+i+1} = \hat{F}(\hat{\mathbf{x}}_{k+i}, \mathbf{u}_{k+i}) \quad \forall i \in \mathbb{N}_{[1, N-1]} \quad (2c)$$

$$\hat{\mathbf{x}}_{k+i} \in \mathbb{X} \quad \forall i \in \mathbb{N}_{[1, N]} \quad (2d)$$

$$\mathbf{u}_{k+i} \in \mathbb{U} \quad \forall i \in \mathbb{N}_{[0, N-1]} \quad (2e)$$

$$\mathbf{g}(\hat{\mathbf{x}}_{k+i+1}, \mathbf{u}_{k+i}) \leq \mathbf{0} \quad \forall i \in \mathbb{N}_{[0, N-1]} \quad (2f)$$

$$\mathbf{h}(\hat{\mathbf{x}}_{k+i+1}, \mathbf{u}_{k+i}) = \mathbf{0} \quad \forall i \in \mathbb{N}_{[0, N-1]} \quad (2g)$$

where  $\|\mathbf{x}\|_{\mathbf{Q}}^2 = \mathbf{x}^T \mathbf{Q} \mathbf{x}$  represents the quadratic operation on the state vector  $\mathbf{x}$ , and the weighting matrices  $\mathbf{Q} > 0$  and  $\mathbf{R} > 0$  are symmetric. The MPC objective function  $J$  in Eq. (2a) consists of two types of loss:  $\|\hat{\mathbf{x}}_{k+i+1} - \mathbf{r}_{k+i+1}\|_{\mathbf{Q}}^2$  refers to the reference tracking error throughout the horizon using the predicted states and  $\|\mathbf{u}_{k+i}\|_{\mathbf{R}}^2$  penalizes the control efforts. The two loss terms can be balanced by user-selected  $\mathbf{Q}$  and  $\mathbf{R}$ . Equation (2c) is the general representation of the dynamic equation, where  $\hat{F}$  denotes the predictive model. Equations (2d) and (2e) are the constraints on states and control actions, respectively, while Eqs. (2f) and (2g) explicitly denote all the inequality and equality constraints if any apply.

While traditional MPC effectively optimizes control actions within a finite horizon based on a deterministic system model [36], it does not inherently account for uncertainties or disturbances that can impact the system dynamics, states, or constraints. The general nonlinear system with uncertainties can be represented as

$$\mathbf{x}_{k+1} = F_w(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \quad (3)$$

where  $\mathbf{w}_k$  represents the disturbance vector, often assumed to lie within a known set  $\mathbb{W}$  or be independent and identically normally distributed with zero means and a diagonal covariance matrix  $\Sigma$ :

$$\mathbf{w}_k \in \mathbb{W} \quad \text{or} \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \Sigma_w) \quad (4)$$

where  $\Sigma_w = \text{diag}(\sigma_{w(1)}^2, \dots, \sigma_{w(m)}^2)$ .

To address this limitation, robust MPC extends the traditional MPC framework by explicitly incorporating uncertainties into the optimization problem, ensuring constraint satisfaction under the effect of uncertainties. The robust MPC optimization problem can be formulated by

$$\min_{\mathbf{u}} J(\mathbf{u}, \hat{\mathbf{x}}, \mathbf{r}) = \sum_{i=0}^{N-1} \left[ \|\hat{\mathbf{x}}_{k+i+1} - \mathbf{r}_{k+i+1}\|_{\mathbf{Q}}^2 + \|\mathbf{u}_{k+i}\|_{\mathbf{R}}^2 \right] \quad (5a)$$

$$\text{s.t. } \hat{\mathbf{x}}_{k+1} = \hat{F}_w(\mathbf{x}_k, \mathbf{u}_k) \quad (5b)$$

$$\hat{\mathbf{x}}_{k+i+1} = \hat{F}_w(\hat{\mathbf{x}}_{k+i}, \mathbf{u}_{k+i}) \quad \forall i \in \mathbb{N}_{[0, N-1]} \quad (5c)$$

$$\hat{\mathbf{x}}_{k+i} \in \mathbb{X} \quad \forall i \in \mathbb{N}_{[0, N-1]} \quad (5d)$$

$$\mathbf{u}_{k+i} \in \mathbb{U} \quad \forall i \in \mathbb{N}_{[0, N-1]} \quad (5e)$$

$$\mathbf{g}(\hat{\mathbf{x}}_{k+i+1}, \mathbf{u}_{k+i}) \leq \mathbf{0} \quad \forall i \in \mathbb{N}_{[0, N-1]} \quad (5f)$$

$$\mathbf{h}(\hat{\mathbf{x}}_{k+i+1}, \mathbf{u}_{k+i}) = \mathbf{0} \quad \forall i \in \mathbb{N}_{[0, N-1]} \quad (5g)$$

Here,  $\hat{F}_w$  is a surrogate model trained using noisy data.  $\hat{\mathbf{x}}_k$  is a general representation of state prediction that can either be deterministic or stochastic. Note that since the disturbance is unknown to  $\hat{F}_w$  when making state prediction, in contrast to Eq. (3),  $\mathbf{w}_k$  is not explicitly treated as the input of  $\hat{F}_w$ .

Among all robust MPC techniques, min-max MPC is easy to implement because the solving procedure does not differ from conventional MPC. The major difference is that min-max MPC handles uncertainties by defining the cost function as the maximum of cost values over all realizations of disturbance sequences by multiple evaluations [13]. It is straightforward but inefficient because simulating all possible disturbances requires considerable cost and computational effort. Another drawback of min-max formulations is that the method results in too conservative solutions that restrict the operation and performance of the system [15,37,38]. Although generating more samples to simulate disturbances can prevent such solutions, the online MPC computation becomes more time-consuming, leading to delayed system actuation.

Tube-based MPC can be used to solve robust MPC problems by explicitly identifying the actual state region surrounding the nominal trajectory (called *tube*), illustrated in Fig. 2(c). A tube accounts for deviations caused by uncertainties and can be included in the robust MPC formulation to satisfy the constraints for all realizations of disturbances [8,10]. However, tube-based MPC approaches typically rely on the assumptions of model representations (e.g., linear dynamic model [16–19] or nonlinear model with Lipschitz functions [39]) and disturbance types. Due to these assumptions, implementing tube-based MPC approaches to complex dynamic models presents a significant challenge.

**2.2 Time-Series Deep Neural Network.** There are two main considerations when selecting a suitable time-series deep neural network (DNN) for MPC: (1) The inference speed should be fast as the solving process of MPC requires several function/model evaluations and (2) the structure of the DNN should accommodate the general format of dynamical systems as denoted in Eq. (1). In this work, we select TiDE [40], illustrated in Fig. 3, as the DNN for surrogating dynamical systems due to its forward speed, model accuracy, and the compatibility of its input structure. TiDE's architecture, with parallelized dense layers and residual connections, ensures both computational efficiency and stable training for dynamic system modeling.

TiDE, designed with a residual network (ResNet) architecture for time-series data, leverages residual connections to enable effective gradient flow during backpropagation, preventing vanishing gradients and capturing long-term dependencies. Its reliance on dense layers allows it to process all time-steps in parallel, making its forward pass faster than popular sequence-to-sequence models like transformers and LSTM. Unlike LSTMs, which process sequences one step at a time recursively, TiDE operates on the entire input sequence as a batch, fully utilizing modern hardware like GPUs. While transformers also parallelize, their self-attention introduces quadratic time complexity with respect to the sequence length, whereas TiDE's complexity grows linearly due to simple matrix multiplications. This linear complexity makes TiDE particularly suitable for real-time applications where low-latency predictions are critical. This efficiency allows TiDE to deliver faster inference while maintaining robust performance for time-series tasks.

The embedding capability of TiDE, realized by the dense encoders and decoders, improves predictions by transforming raw inputs into dense, low-dimensional representations that capture meaningful patterns and relationships. This reduces data dimensionality, encodes complex interactions, and enhances the model's ability to generalize across unseen examples. This dimensionally reduced embedding also plays the role of a noise filter by only identifying and embedding the most important features in its latent space. By effectively compressing input information, the embeddings help

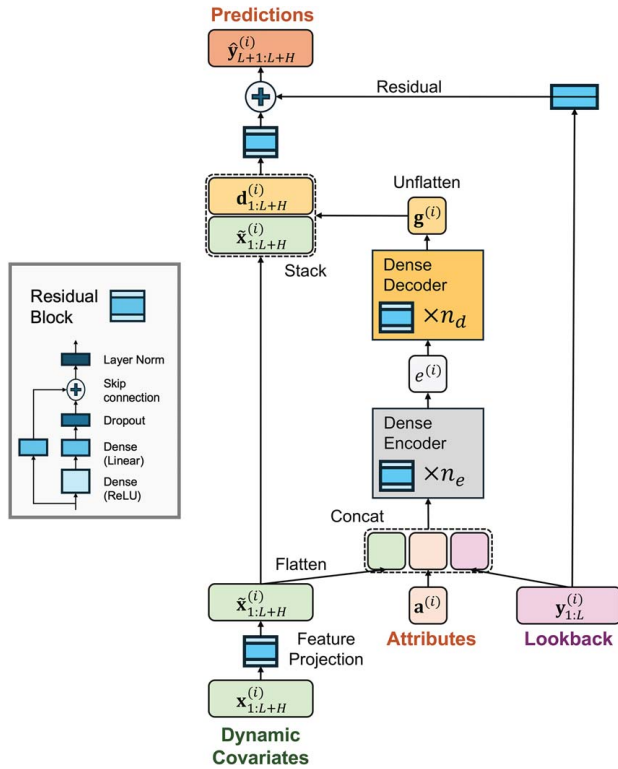


Fig. 3 Network structure of TiDE, modified from Ref. [40]

mitigate the risk of overfitting, especially in high-dimensional datasets. For time-series data, embeddings efficiently represent temporal attributes or categorical features, enabling TiDE to extract richer patterns and improve prediction accuracy.

Different from conventional sequence-to-sequence prediction models, TiDE supports the usage of *covariates* and *targets* as the model input, as illustrated in Fig. 4, making it suitable for surrogating dynamical systems. The target variable is the primary variable of interest in a time-series forecasting model. It represents the value that is aimed to be predicted or forecasted, such as the future states of the system. The covariates are additional variables that provide supplementary information and can aid in predicting the target variable. They can be further categorized as past covariates and future covariates. These variables are often external or supplementary and are not part of the target series but are related to it.

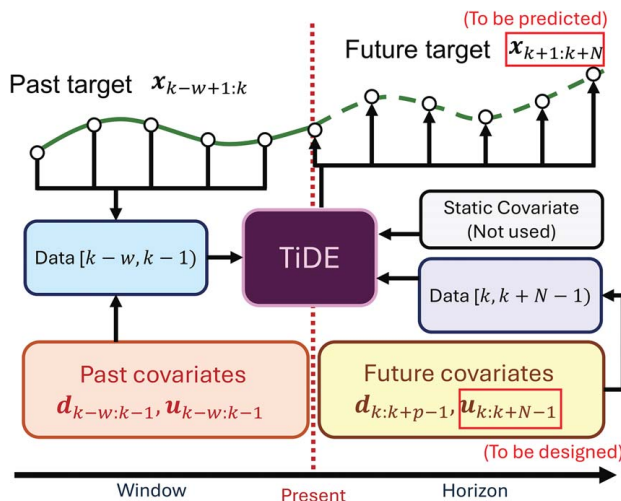


Fig. 4 Data structure of the input and output of TiDE

This structured separation of covariates and targets allows TiDE to capture both short-term dynamics and long-term dependencies more effectively. As shown in Fig. 4, TiDE takes past target (e.g., past states  $\mathbf{x}_{k-w+1:k}$  as an autoregressive system), past covariates (e.g., past input  $\mathbf{u}_{k-w:k-1}$  and other past input conditions  $\mathbf{d}_{k-w:k-1}$  if required), and future covariates (e.g., future input  $\mathbf{u}_{k:k+N-1}$  and other future input conditions  $\mathbf{d}_{k:k+N-1}$ ) as model input to predict the future target  $\mathbf{x}_{k+1:k+N}$  (future states). In particular, we denote  $\mathbf{d}$  as the predefined system variable (e.g., the geometry information of a given part in additive manufacturing) and  $\mathbf{u}$  as the future control input to be optimized in MPC (e.g., the laser power). Therefore, TiDE can be formulated as

$$\hat{\mathbf{x}}_{k+1:k+N} = \text{TiDE}(\mathbf{x}_{k-w+1:k}, \mathbf{d}_{k-w:k-1}, \mathbf{u}_{k-w:k-1}, \mathbf{d}_{k:k+N-1}, \mathbf{u}_{k:k+N-1} | \phi) \quad (6)$$

where  $\phi$  is the trainable NN parameters of TiDE.

As a result, the separation of targets and covariates allows TiDE to resemble the nature of dynamical systems in a multistep-ahead setting. In contrast, some forecasting models, such as transformers [29] and N-BEATS [41], predict future targets solely based on the past target, but do not consider covariates explicitly. This design choice makes TiDE more versatile for applications where external influences, such as control inputs or environmental conditions, significantly affect system behavior.

**2.3 Quantile Regression.** Quantile regression (or quantile loss) [28,42,43] is a versatile statistical technique used to estimate the conditional quantiles of a response variable, such as the median or other given percentiles, based on a set of predictor variables. Unlike ordinary least-squares regression, which focuses on modeling the mean of the response variable, quantile regression captures a broader picture by modeling the entire conditional distribution. Specifically, quantile regression is effective in quantifying aleatoric uncertainty, which arises from inherent variability in the data. One key advantage of quantile regression is its robustness to outliers, as it is less sensitive to extreme values compared to methods such as mean square error loss. This makes it particularly useful for datasets with skewed or irregular distributions. Moreover, quantile regression does not require prior assumptions about the distribution of the data, enabling it to handle heteroscedastic uncertainty situations where the variability of the response changes across levels of the predictors. This flexibility allows the model to adapt to complex, real-world datasets where such variability is common.

The standard loss function for implementing quantile regression in supervised learning is defined as

$$L_{q,j}(y_t, \hat{y}_t) = \begin{cases} q \cdot (y_t - \hat{y}_t) & \text{if } y_t \geq \hat{y}_t \\ (1 - q) \cdot (\hat{y}_t - y_t) & \text{if } y_t < \hat{y}_t \end{cases} \quad (7)$$

where the objective is to minimize the errors of a given quantile level  $q$  (e.g., 0.5 for median) for response  $j$ .  $y_t$  and  $\hat{y}_t$  are the ground truth value and the predicted value of the target at time  $t$ , respectively.

To justify the need for quantile regression, we perform a benchmark testing against popular UQ methods, including GP, MC dropout, ensemble methods, and deep evidential regression [44]. Here, a one-dimensional benchmark problem from Ref. [44] is modified by injecting non-Gaussian noise:

$$y = x^3 + W_{\text{exp}} \quad (8)$$

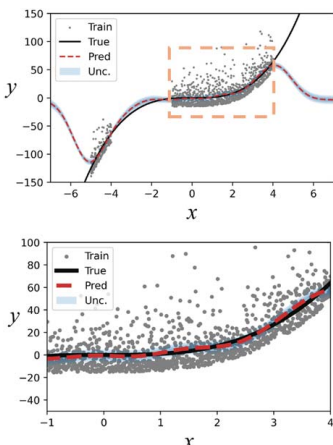
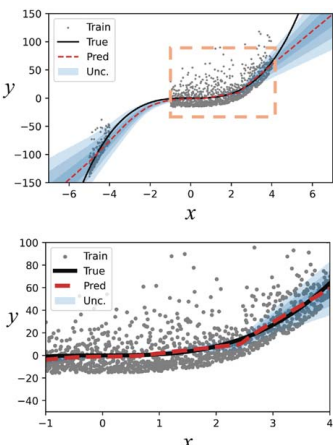
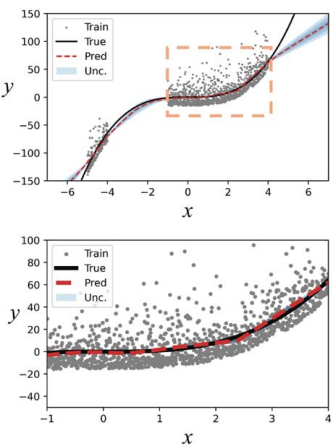
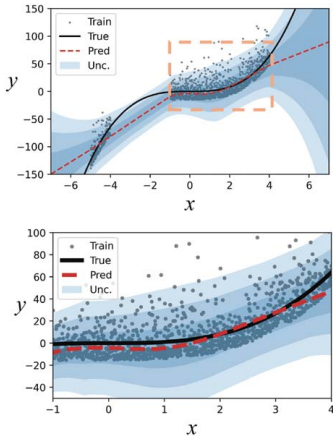
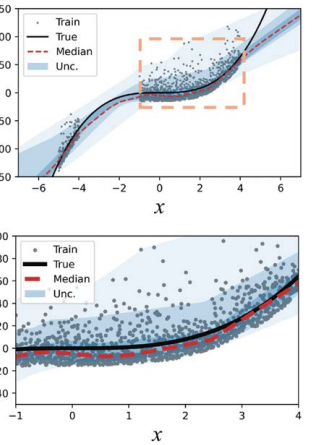
where  $W_{\text{exp}} \sim \text{Exp}(15) - 15$  follows a zero mean exponential distribution with rate  $\lambda = 15$ . The reason for not assuming a normal distribution for the injected noise is that, even if the environmental disturbance is normally distributed, its propagation through a non-linear system does not guarantee a normally distributed state. Therefore, when applying deep learning quantiles to nonlinear systems, assuming the state distribution is normal may not be the most

appropriate. Aside from GP, all the NNs in each method have two hidden layers with size = 64 and use ReLU() as the activation function, following the setting in Ref. [44], while the output layer may be different, depending on each case. For training the model, 1334 training samples are generated within the range of  $[-5, -4] \cup [-1, 4]$ , and all the NN models are trained with Adam for 500 epochs with a learning rate of 0.1. To evaluate the methods, a test set of size 1000 is generated using  $x \in [-7, 7]$  and realized using Eq. (8). Quantitative indices, including residual mean square error (RMSE),  $R^2$ , and coverage rate (number of test samples that fall within probabilistic bounds/total number of test

samples), are provided. Since the input range of the training and test sets is different, we not only evaluate the efficacy of learning noise distribution but also test the generalizability for extrapolation and interpolation at the sparse data region.

As can be seen from the results in Table 1, because GP, MC dropout, ensemble method, and evidential regression all assume the uncertainty distribution to be normal, only quantile regression can effectively capture the skewed distribution, while others failed to provide accurate uncertainty bounds. In fact, since GP, MC dropout, and ensemble method are quantifying epistemic (model) uncertainty and are predicting confidence intervals (CIs)

**Table 1 Benchmarking of UQ methods**

						
Method	Gaussian process		Monte Carlo dropout		Ensemble method	
UQ type	Epistemic (CI)		Epistemic (CI)		Epistemic (CI)	
Visualized bounds	$[\pm 1\sigma, \pm 2\sigma, \pm 3\sigma]$		$[\pm 1\sigma, \pm 2\sigma, \pm 3\sigma]$		$[\pm 1\sigma, \pm 2\sigma, \pm 3\sigma]$	
Coverage rate for each bound	[10.2%, 18.2%, 29.1%]		[10%, 20.8%, 36.4%]		[11.1%, 23.3%, 36.3%]	
RMSE/ $R^2$	113.1452/0.2429		56.2704/0.8127		53.9461/0.8279	
Others	Assume the noise is homeostatic		Dropout rate = 0.2; MC sample = $1 \times 10^5$		Number of NNs: 10	
						
Method	Evidential regression		Quantile regression			
UQ type	Epistemic + aleatoric (PI)		Aleatoric (PI)			
Visualized bounds	$[\pm 1\sigma, \pm 2\sigma, \pm 3\sigma]$		$q = [(0.25, 0.75), (0.05, 0.95), (0.001, 0.999)]$			
Coverage rate for each bound	[40.2%, 68.4%, 85.8%]		[32.7%, 59.5%, 73.2%]			
RMSE/ $R^2$	73.7554/0.6783		53.8638/0.8218			
Others	$\lambda = 1 \times 10^{-4}$		Each quantile is predicted by an individual NN			

Note: The lower figure for each method highlights the region within the pink dashed box.



rather than predictive intervals (PIs), it is reasonable that these methods failed to capture the aleatoric (data) uncertainty, which is more critical in our application. Evidential regression, by learning the uncertainty of the estimated standard deviation, is capable of learning both epistemic and aleatoric uncertainty. However, similarly, it suffers from assuming the distribution to be normal and provides obvious over-conservative CIs. In contrast, by using quantile regression, it can flexibly learn the median response and bounds effectively regardless of the uncertainty distribution. The prediction accuracy of quantile regression also outperforms other methods.

These results support the need for quantile regression as a computationally efficient, flexible, and accurate UQ method for applications in robust MPC for nonlinear systems.

**2.4 Implementation of Quantile Regression on TiDE.** The quantile loss for time-series data is an extension of the standard quantile loss. For time-series data with  $N$  steps ahead to be predicted, the total quantile loss is often calculated as the sum over all time-steps and quantile levels:

$$L_Q = \frac{1}{N \cdot l} \sum_{j=1}^l \sum_{t=1}^N L_{q,j}(y_t, \hat{y}_t) \quad (9)$$

where  $l$  is the level of the assigned quantiles.

Implementing quantile loss in TiDE involves increasing the output dimensions to accommodate state predictions for multiple quantile levels. For a prediction setup with batch size  $B$ , horizon length  $N$ , number of responses  $D$ , and  $l$  quantile levels for each response, the output tensor from TiDE will have dimensions  $[B, N, D, l]$ . This expanded output structure facilitates the direct estimation of uncertainty bounds by providing multiple quantile estimates for each prediction. This design allows TiDE to predict the entire output tensor, including all quantile values for each response, in a single forward pass. The TiDE prediction model with the quantile output can be represented by

$$\begin{aligned} \hat{\mathbf{x}}_{k+1:k+N} &= [\hat{\mathbf{x}}_{k+1:k+N}, \hat{\mathbf{x}}_{k+1:k+N}, \hat{\mathbf{x}}_{k+1:k+N}]^T \\ &= \text{TiDE}(\mathbf{x}_{k-w+1:k}, \mathbf{d}_{k-w:k-1}, \mathbf{u}_{k-w:k-1}, \\ &\quad \mathbf{d}_{k:k+N-1}, \mathbf{u}_{k:k+N-1} | \phi) \end{aligned} \quad (10)$$

In this context,  $\hat{\mathbf{x}}_{k:k+N-1}$ ,  $\hat{\mathbf{x}}_{k:k+N-1}$ , and  $\hat{\mathbf{x}}_{k:k+N-1}$  represent the upper quantile, median, and lower quantile of the predicted states, respectively. To simplify the notation, we utilize superscripts  $p, f$ , and  $p:f$  to denote the past, the future, and the past and future covariates/targets, respectively, at time  $k$ . The equation of the TiDE model becomes

$$[\hat{\mathbf{x}}_{k+1}^f, \hat{\mathbf{x}}_{k+1}^p, \hat{\mathbf{x}}_{k+1}^{p:f}]^T = \text{TiDE}(\mathbf{x}_k^p, \mathbf{u}_k^p, \mathbf{u}_k^f, \mathbf{d}_k^p, \mathbf{d}_k^f | \phi) \quad (11)$$

By employing this one-shot approach, TiDE substantially reduces computation time through efficient parallelization.

### 3 Simultaneous Multistep Robust Model Predictive Control for Digital Twin

The purpose of this work is to integrate the simultaneous multistep-ahead predictive quantile with MPC, demonstrating a learning-based robust MPC that performs decision-making under uncertainty for digital twins. In this section, we first describe the development of the virtual system via TiDE. Then, we focus on the robust MPC formulation incorporating multistep quantile prediction and introduce the optimization tools that can benefit real-time solving. A complete walk through of the process will be detailed in Sec. 4.

**3.1 Building the Virtual System Via TiDE: Data Generation and Model Training.** TiDE serves not only to learn the dynamics of the physical system but also to capture the distribution of system

behaviors under the influence of uncertainty. To achieve this, the training data collected from the physical system must accurately represent its behavior under operational disturbances. This ensures that the model can generalize effectively and provide reliable predictions across a range of operating conditions.

In many existing learning-based robust MPC approaches (when the system model is unknown), the disturbances affecting the system dynamics are assumed to be known beforehand, and training data are typically generated through virtual experiments under predefined disturbance conditions. However, in practical engineering scenarios where the distribution of uncertainties is unknown, the collected data inherently include noise and can be directly utilized as training data for TiDE, enabling it to adapt to real-world conditions. This data-driven approach allows TiDE to model both the nominal system behavior and the variability introduced by stochastic disturbances.

At the model training stage, TiDE directly learns the user-assigned quantile levels from the noisy training data. In this setting, there is no assumption made regarding the boundedness of the response under uncertainty. Consequently, even trained under noisy data, TiDE can provide smooth predictions for the quantile levels and the median, efficiently quantifying the data (aleatoric) uncertainties. This allows the model to balance prediction accuracy with uncertainty estimation, making it suitable for robust decision-making in dynamic systems. This is because the crucial features are extracted and mapped into the dense encoder, which also serves as a noise filter to eliminate the impact of noise and disturbances while preserving the important information. When training the TiDE model using PyTorch, we use the Adam optimizer as default, and add a regularization term to increase the generality of the model. We will exhibit specific details in the examples.

### 3.2 Real-Time Decision-Making Via Model Predictive Control

**3.2.1 Uncertainty-Aware Model Predictive Control Formulation.** In the robust MPC formulation in Eq. (5), the constraints are formulated as hard constraints to account for disturbances that are assumed to be bounded within a predefined set. The tube around the nominal trajectory is specified to ensure that all possible realizations of the bounded disturbances remain within this tube, as shown in Fig. 2(c). However, for those disturbances that are not bounded (e.g., Gaussian noises), it is nearly impossible to guarantee the satisfaction of hard constraints. In this case, the constraints are relaxed into probabilistic (chance) constraints, ensuring that they are satisfied with a specified probability [45]. Therefore, the tubes are derived probabilistically based on the distribution of the disturbances [46]. This approach acknowledges that disturbances may not have strict bounds but instead follow a known or estimated probability distribution. When a confidence level  $\alpha$  (e.g.,  $\alpha = 0.95$ ) is specified, the tube bounds can be explicitly calculated to encapsulate 95% of the disturbance realizations. This probabilistic bounding allows for the construction of stochastic tubes that balance conservatism and feasibility, providing a probabilistic guarantee of constraint satisfaction. Importantly, the probabilistic nature of the tube makes stochastic MPC less conservative than robust MPC while still accounting for uncertainty effectively.

Let us assume that the constraints are only enforced on states and control input. The uncertainty-aware MPC with the single-step nonlinear dynamics and probabilistic constraints can be formulated as

$$\min_{\mathbf{u}} \quad \mathbb{E}_w[J(\mathbf{u}, \mathbf{x}, \mathbf{r}, \mathbf{w})] \quad (12a)$$

$$\text{s.t.} \quad \mathbf{x}_{k+i+1} = F_w(\mathbf{x}_{k+i}, \mathbf{u}_{k+i}, \mathbf{w}_{k+i}) \quad \forall i \in \mathbb{N}_{[0, N-1]} \quad (12b)$$

$$\Pr(\mathbf{x}_{k+i}^{(j)} \in \mathbb{X}) \geq \alpha \quad \forall i \in \mathbb{N}_{[0, N-1]} \quad \forall j \in \mathbb{N}_{[1, n_x]} \quad (12c)$$

$$\mathbf{u}_{k+i} \in \mathbb{U} \quad \forall i \in \mathbb{N}_{[0, N-1]} \quad (12d)$$

where  $\mathbf{w} = [\mathbf{w}_k, \dots, \mathbf{w}_{k+p-1}]$  represents the sequence of random variables for disturbance vectors and  $n_x$  denotes the dimension of the vector  $\mathbf{x}$ .

The optimization problem presented in Eq. (12) poses significant computational challenges due to its probabilistic constraints and the need to consider uncertainty propagation across the prediction horizon. While the formulation elegantly captures the uncertainty-aware nature of the control problem, its direct implementation is computationally intractable, particularly for real-time applications.

**3.2.2 Reformulation With Quantiles and Constraint Tightening.** To address this computational challenge, the uncertainty-aware MPC problem is reformulated using the quantile information and constraint-tightening techniques with an ancillary controller design. The predictive model in Eq. (11) provides information about the upper and lower bounds of future states, which can be directly utilized to ensure constraint satisfaction under uncertainty. The problem from Eq. (12) can be reformulated as

$$\min_{\mathbf{v}} J(\mathbf{v}, \hat{\mathbf{x}}, \mathbf{r}) \quad (13a)$$

$$\text{s.t. } [\tilde{\mathbf{x}}_k^f, \tilde{\mathbf{x}}_k^f, \tilde{\mathbf{x}}_k^f]^T = \text{TiDE}(\mathbf{x}_k^p, \mathbf{u}_k^p, \mathbf{u}_k^f, \mathbf{d}_k^p, \mathbf{d}_k^f | \phi) \quad (13b)$$

$$\tilde{\mathbf{x}}_{j,k+i} \leq x_{j,\text{ub}} \quad \forall i \in \mathbb{N}_{[0, N-1]} \quad \forall j \in \mathbb{N}_{[1, n_x]} \quad (13c)$$

$$\tilde{\mathbf{x}}_{j,k+i} \geq x_{j,\text{lb}} \quad \forall i \in \mathbb{N}_{[0, N-1]} \quad \forall j \in \mathbb{N}_{[1, n_x]} \quad (13d)$$

$$\mathbf{v}_{k+i} \in \mathbb{U} \ominus \mathbf{K}\mathbb{Z}_{k+i} \quad \forall i \in \mathbb{N}_{[0, N-1]} \quad (13e)$$

$$\mathbf{u}_k = \mathbf{v}_k + \mathbf{K}\mathbf{e}_k \quad (13f)$$

where the subscripts “ub” and “lb” denote the upper and lower bounds, respectively, the subscript  $j$  represents the index of the state vector  $\mathbf{x}$ ,  $\mathbf{v} = [\mathbf{v}_k, \mathbf{v}_{k+1}, \dots, \mathbf{v}_{k+N-1}]$  denotes the sequence of the nominal control inputs (which are the decision variables),  $\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k$  represents the deviation between the actual and predicted states, and  $\mathbb{Z}_{k+i}$  is the set of the quantile bound at time  $k+i$ .

Theoretically, the optimal value of  $\mathbf{K}$  can be determined using the LQR for linear tube-based robust MPC [8,16]. This value plays a crucial role in stabilizing the system and preventing the estimation error bound from diverging during multistep-ahead prediction. It also influences the level of conservativeness in estimating error bounds when propagating noise through recursive rollout. However, for nonlinear (and even unknown) systems, LQR cannot be used to obtain  $\mathbf{K}$ , and it must be fine-tuned or optimized using black-box methods such as Bayesian optimization [47]. It is important to note that optimizing  $\mathbf{K}$  is beyond the scope of this work. In contrast, TiDE directly captures the actual distribution from the open-loop data, unlike tube-based robust MPC. Therefore, the value of  $\mathbf{K}$  does not affect the conservativeness of the predicted error bounds. In online solving, the value of  $\mathbf{K}$  tightens the design space of  $\mathbf{u}_k$  to account for system disturbances. This is demonstrated in the example provided, which can also be derived from the Riccati equation [16]. A more rigorous implementation would involve continuous linear approximations [48] on the TiDE model and deriving  $\mathbf{K}$ . However, this is also beyond the scope of our work.

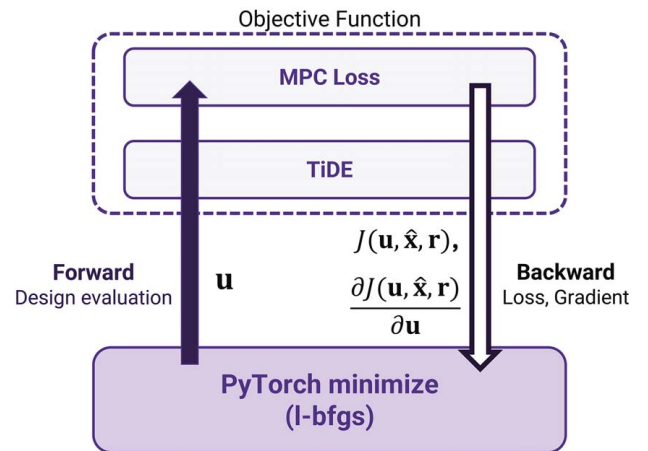
The state constraints from Eqs. (13c) and (13d) are managed through the quantile information derived from the predictive model. By utilizing these bounds, we can guarantee that the system states remain within their feasible region with the specified probability level  $\alpha$ . This approach effectively transforms the probabilistic state constraints into deterministic bounds based on the predicted quantiles of the state distribution.

For the control input constraints in Eq. (13e), a more careful strategy is necessary to ensure that the actual applied control actions remain within the physical limitations of the controller or actuator. The control input constraints are tightened [18,39,49] to accommodate the additional control effort that may be required by the ancillary controller in Eq. (13f). The ancillary controller is used for rejecting the real-time disturbance while maintaining the satisfaction of the original constraints [50]. In this study, a linear representation is chosen for computational efficiency. The constraint-tightening technique creates a safety margin that can avoid the actual state  $\mathbf{x}_k$  and the total control input  $\mathbf{u}_k$  violating the original constraints  $\mathbb{X}$  and  $\mathbb{U}$ , respectively.

**3.3 Optimization Setup.** Although one motivation of simultaneous multistep MPC is to accelerate the solving process of MPC by parallelizing the state prediction in one shot, in this work, we implement other techniques to further accelerate the optimization process, making the MPC solvable in actionable time. Here we detail the methods and algorithms used to accelerate the solving process of MPC.

**3.3.1 Gradient-Based Optimization With Automatic Differentiation.** One way to accelerate the solving process of MPC using a numerical optimization solver is to apply gradient-based optimization with automatic differentiation [22]. The key idea is to acquire an analytical evaluation of the first-order derivative of the loss function with respect to the design variables (control input  $\mathbf{u}$ ) and use the gradient information to perform gradient-based optimization. Since the evaluation of the MPC loss  $J(\mathbf{u}, \hat{\mathbf{x}}, \mathbf{r})$  as well as TiDE are both computed using PyTorch [51], the gradient of MPC loss  $\partial J(\mathbf{u}, \hat{\mathbf{x}}, \mathbf{r}) / \partial \mathbf{u}$  can be obtained analytically using backpropagation instead of numerical approximations such as finite difference, as shown in Fig. 5. Lastly, in this work, we choose l-bfgs [52] with a PyTorch wrapper developed by Feinman [53] as MPC’s numerical optimizer. The l-bfgs is a light memory-used algorithm that approximates the Hessian (second-order derivative) using the first-order derivative of the loss function. Since the gradient information can be obtained automatically, the evaluation of the Hessian can also be done in only one function evaluation. As a result, the integration of l-bfgs and PyTorch enables efficient gradient-based optimization by utilizing cheap but accurate gradient evaluation.

**3.3.2 Penalty Method: Augmented Lagrangian Method.** However, even when the gradient of the objective function can be automatically computed, evaluating the optimality conditions in l-bfgs when constraints are enforced still requires additional numerical approximations of second-derivative terms, resulting in



**Fig. 5 Illustration of gradient-based optimization using autodifferentiation**



a significant increase in function evaluation time and computational speed. Therefore, we employ the penalty method to transform the constrained optimization problem into an unconstrained optimization problem. By directly incorporating the penalty terms into the objective function, we convert the hard constraints into soft constraints, thereby bypassing the evaluation of the optimality conditions associated with constraint optimization.

In particular, we use the augmented Lagrangian method [52] for the penalty method. Assume the objective function for the constrained MPC/robust MPC is  $J(\mathbf{u}, \hat{\mathbf{x}}, \mathbf{r})$  with constraint  $c_i(\mathbf{u}, \mathbf{x})$ ,  $i \in \mathcal{E}$  generalized for both equality and inequality constraints, where  $\mathcal{E}$  is the number of constraints. The augmented Lagrangian method then solves the following unconstrained optimization problem by adding the constraints as a penalty:

$$\min_{\mathbf{u}} \Phi_s(\mathbf{u}, \hat{\mathbf{x}}, \mathbf{r}) = J(\mathbf{u}, \hat{\mathbf{x}}, \mathbf{r}) + \frac{\mu_i}{2} \sum_{i \in \mathcal{E}} [\text{ReLU}(c_i(\mathbf{u}, \hat{\mathbf{x}}))]^2 + \sum_{i \in \mathcal{E}} \lambda_i \text{ReLU}(c_i(\mathbf{u}, \hat{\mathbf{x}})) \quad (14)$$

where  $s$  indicates the  $s$ th iteration when solving the optimization problem. Both  $\mu_i$  and  $\lambda_i$  are the penalty parameter and the estimated Lagrange multiplier corresponding to the  $i$ th constraint and follow the updating rules:

$$\mu_i \leftarrow \alpha \mu_i \quad (15)$$

$$\lambda_i \leftarrow \lambda_i + \text{ReLU}(c_i(\mathbf{u}_s, \hat{\mathbf{x}})) \quad (16)$$

where  $\alpha > 1$  is the increasing rate of  $\mu_i$ , and  $\mathbf{u}_s$  is the solution for solving unconstrained optimization (Eq. (14)) at iteration  $s$ . In this work, we select  $\lambda_0 = 10$ ,  $\mu_0 = 1$ , and  $\alpha = 3$  from trial and error that balance constraint satisfaction and convergence rate. In the next iteration, the solver will resolve the problem using  $\mathbf{u}_s$  as the initial guess for a warm start. Here, we apply the  $\text{ReLU}()$  function in Eq. (14) because it is a continuous and differentiable function that only penalizes  $\Phi_s$  when the constraints  $c_i$  are violated, enabling the smooth computation of the gradient of  $\Phi_s$ .

While the augmented Lagrangian method introduces some deviation from strict Karush–Kuhn–Tucker (KKT) conditions to improve numerical stability and feasibility, it still provides robust handling of both equality and inequality constraints. Furthermore, it avoids the ill-conditioning issues of pure penalty methods using Lagrange multipliers, reducing sensitivity to the penalty parameter. Lastly, it converges more efficiently to feasible solutions, even for problems with nonlinear constraints [52].

Lastly, the warm start is used to provide a potential starting point near the optimal solution, i.e., the optimal solution from the previous step is used as the initial guess for the current step. Also, if warm starting MPC is unable to identify a feasible solution, the MPC will terminate the optimization and use the predefined control input to achieve feasibility of online operation.

## 4 Illustrative Example

We first verify the proposed method using a linear invariant system so that the result can be compared with the tube-based

method, which is one of the most widely adopted robust MPC methods. This section provides a complete walk through by introducing the physical system, developing the virtual system, building the virtual-to-physical connection via robust MPC, and result validation and comparison.

**4.1 Physical System.** A discrete linear system is selected for demonstration. The system with exogenous noise on input  $\epsilon_k \sim \mathcal{N}(0, 0.1^2)$  is formulated as follows:

$$\mathbf{x}_{k+1} = \begin{bmatrix} 0.3 & 0.1 \\ 0.1 & 0.2 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} (u_k + \epsilon_k) \quad (17)$$

$$= F_w(\mathbf{x}_k, u_k, \mathbf{w}_k) \quad (18)$$

where the disturbance vector is the multiplication of matrix  $\mathbf{B}$  and the noise vector  $\epsilon_k$ , i.e.,  $\mathbf{w}_k = \mathbf{B}\epsilon_k$ , which also follows a Gaussian.

## 4.2 Virtual System

**4.2.1 Data Generation and Model Training.** The development of the virtual system in our work follows the steps of stage 1 in Fig. 1. To generate state trajectories for system identification using TiDE, a sequence of input  $\mathbf{D}_u = \{u_0, \dots, u_{n-1}\}$ , where  $\mathbf{u} \in [-5, 5]$  is uniformly sampled with a size of  $n = 422,000$ . By setting the initial state  $\mathbf{x}_0 = [0, 0]^T$ , the trajectory of  $\mathbf{D}_x = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  can be simulated using  $\mathbf{u}$  as the input. Furthermore, both  $\mathbf{D}_u$  and  $\mathbf{D}_x$  are divided into fractions using the moving window approach, with each fraction having a length of  $w + N$ . In this case, the window size is  $w = 10$ , and the horizon length is  $N = 10$ . By denoting the  $l$ th fraction as  $\mathbf{D}_u^l = \{u_l, \dots, u_{l+w+N-1}\}$  and  $\mathbf{D}_x^l = \{\mathbf{x}_l, \dots, \mathbf{x}_{l+w+N-1}\}$ , respectively, we can assign  $\mathbf{x}_p = [\mathbf{x}_{l+1}, \dots, \mathbf{x}_{l+w}]$  as the past target,  $\mathbf{x}_f = [\mathbf{x}_{l+w+1}, \dots, \mathbf{x}_{l+w+N}]$  as the future target, with  $\mathbf{u}_p = [u_l, \dots, u_{l+w-1}]$  and  $\mathbf{u}_f = [u_{l+w}, \dots, u_{l+w+N-1}]$  as the past and future covariates (inputs), respectively. Then,  $\mathbf{D}_x^l$  and  $\mathbf{D}_u^l$  are further split into training, validation, and test sets with a 8:1:1 ratio. Following Eq. (6), the TiDE model for identifying this system can be trained via supervised learning using quantile loss, which is formulated as

$$\min_{\phi} L_Q(\mathbf{x}^f, \hat{\mathbf{x}}^f) \quad (19a)$$

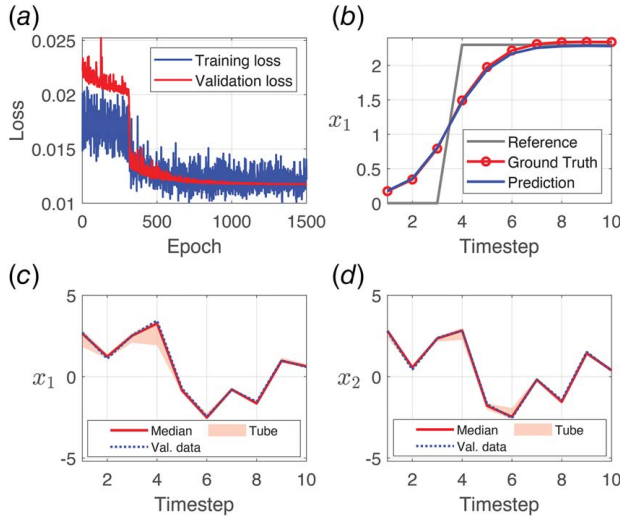
$$\text{s.t. } \hat{\mathbf{x}}^f = \text{TiDE}(\mathbf{x}^p, \mathbf{u}^p, \mathbf{u}^f | \phi) \quad (19b)$$

The details of the training and model setup are described in Table 2. The training and validation loss is shown in Fig. 6(a).

**4.2.2 Model Evaluation.** We first evaluate the accuracy of TiDE using the test set. Quantitatively, the mean absolute percentage error (MAPE) and the relative residual mean square error (RRMSE) for the predicted  $x_1$  and  $x_2$  achieve [5.96%, 5.05%] and [0.0419, 0.0414], respectively, showing the high predictive capability of TiDE. Qualitatively, as shown in Figs. 6(c) and 6(d), the tube (i.e., the error interval bounded by the 0.95 and 0.05 quantiles), as well as the median of the predicted values of  $x_1$  and  $x_2$  by TiDE, is compared with the validation data (with

**Table 2 Hyperparameters of training the TiDE model**

Details for TiDE model setup						
# Encoder layers 1	# Decoder layers 1	Decoder output dim. 16	Hidden size 128	Decoder hidden size 32	Dropout rate 0.2	Layer normalization True
Details for TiDE model training						
Learning rate 0.001	Regularization 0.002	Step size 10	Rate decay 0.95	# Epoch 1500	Batch size 64	Shuffle data True



**Fig. 6 Evaluation of TiDE. (a) The training and validation loss of TiDE. (b) Comparison of the state prediction and the ground truth in a single MPC step. (c) One-shot prediction of the median and quantiles of  $x_1$ . (d) One-shot prediction of the median and quantiles of  $x_2$ .**

noise injected) using a randomly selected fraction in the test set. As can be seen from the figure, the predicted median matches the validation data well with slight deviation, and the validation data are mostly bounded by the upper and lower quantiles. This shows the capability of TiDE to capture the dynamics of the system and to quantify the uncertainty of the prediction even when the training data are noisy.

In Fig. 6(b), we compare the predicted median of  $x_1$   $\hat{x}_{1,k+1:k+N}$  and the ground truth (obtained from the system model without injected noise) using the optimal control input  $\mathbf{u}_k^*$  obtained by solving Eq. (20) with a step function as the reference. Here, we demonstrate that the optimizer successfully solves the MPC problem that minimizes the reference tracking error using the multistep-ahead prediction by TiDE. By examining the results, we observe that the predicted response shares a similar trend with the true response. This test validates the optimization capability of TiDE as the multistep-ahead predictor in MPC.

**4.3 Virtual-to-Physical Integration.** The bidirectional interaction between the physical and virtual systems is constructed with robust MPC, where the current state of the physical system is fed back to the virtual system as its prediction input. Furthermore, an optimal control input is solved by an online robust MPC, using the predicted quantiles as the safety buffer to explicitly handle constraints under uncertainty. Then, the optimal control input is applied to the physical system.

The goal of this example is to use robust MPC to perform a reference tracking task on  $x_1$ , while maintaining  $x_1$ ,  $x_2$ , and  $u$  within the feasible regions subjected to unbounded disturbance. The multistep robust MPC using the quantile prediction from TiDE can be formulated as

$$\min_{\mathbf{v}} J(\mathbf{v}, \tilde{\mathbf{x}}, \mathbf{r}) = \sum_{i=0}^{N-1} \|\tilde{x}_{1,k+i+1} - r_{k+i+1}\|_{\mathbf{Q}}^2 + \|v_{k+i}\|_{\mathbf{R}}^2 \quad (20a)$$

$$\text{s.t. } \hat{\mathbf{x}}_{k+1}^f = [\tilde{x}_{k+1}^f, \tilde{x}_{k+1}^f, \hat{\mathbf{x}}_{k+1}^f]^T = \text{TiDE}(\mathbf{x}_k^p, \mathbf{u}_k^p, \mathbf{v}) \quad (20b)$$

$$\mathbf{v} = [v_k, \dots, v_{k+N-1}] \quad (20c)$$

$$\Pr(\hat{x}_{1,k+i} \geq -2) \geq 0.95 \quad \forall i \in \mathbb{N}_{[1,N]} \quad (20d)$$

$$\Pr(\hat{x}_{1,k+i} \leq 2.5) \geq 0.95 \quad \forall i \in \mathbb{N}_{[1,N]} \quad (20e)$$

$$\Pr(\hat{x}_{2,k+i} \geq -3.5) \geq 0.95 \quad \forall i \in \mathbb{N}_{[1,N]} \quad (20f)$$

$$\Pr(\hat{x}_{2,k+i} \leq 3.5) \geq 0.95 \quad \forall i \in \mathbb{N}_{[1,N]} \quad (20g)$$

$$v_{k+i} \in \mathbb{U} \ominus \mathbf{K}\mathbb{Z}_{k+i} \quad \forall i \in \mathbb{N}_{[0,N-1]} \quad (20h)$$

$$u_k = \pi(v_k) = v_k + \mathbf{K}e_k \quad (20i)$$

$$\mathbf{u}_k^p = [u_{k-w}, \dots, u_{k-1}] \quad (20j)$$

Both  $\mathbf{Q}$  and  $\mathbf{R}$  are set to  $\mathbf{I}$ . The original input bound is  $\mathbb{U} \in [-5, 5]$ , while it is dynamically tightened based on the prediction error bound  $\mathbb{Z}$ . The linear regulator  $\mathbf{K} = [-0.0621, -0.2027]$  is assigned in this case. Furthermore, since the predicted upper and lower quantiles have already taken into account the probabilistic bounds on  $\hat{x}_{1,k}$ , the constraints in Eqs. (20d)–(20g) can be rewritten as

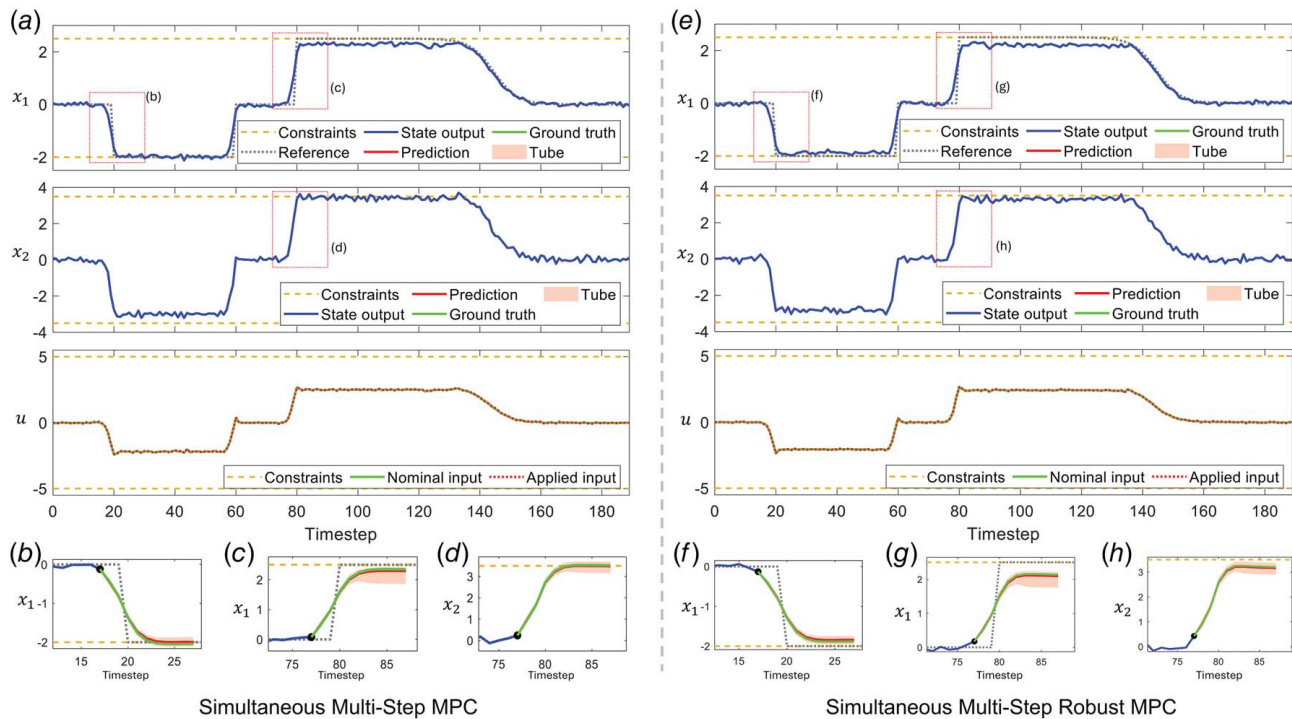
$$\bar{\hat{x}}_{1,k+i} \leq 2.5, \quad \underline{\hat{x}}_{1,k+i} \geq -2 \quad (21a)$$

$$\bar{\hat{x}}_{2,k+i} \leq 3.5, \quad \underline{\hat{x}}_{2,k+i} \geq -3.5 \quad (21b)$$

The performance of the proposed robust MPC method is compared with the multistep MPC without uncertainty awareness (so-called nominal MPC) in Fig. 7. In this example, the reference trajectory is designed to overlap with the bounds of  $x_1$  and will violate the bounds of  $x_2$  during this reference tracking task, aiming to test the constraint handling capability of robust MPC at extreme scenarios. As can be seen from Fig. 7(a), where the nominal multistep MPC is solved without considering the uncertainty of the prediction, the result yields significant constraint violations due to the disturbance. Since the gain of the ancillary controller is small in this case, the applied input is almost identical to the nominal input. Figures 7(b)–7(d) compare the nominal TiDE prediction and the ground truth at different instances when the optimal control input sequence is solved. Here, the ground truth (green line) is verified by simulating Eq. (18) without disturbance. As a result, even though the discrepancy between the nominal prediction and the ground truth is not significant, indicating the accuracy of the prediction, the optimal control input only provides ideal solutions that minimize the MPC loss but ignore the impact of uncertainty and do not provide a safety buffer to accommodate disturbance.

In contrast, the proposed robust MPC method explicitly uses the learned quantile as the tube to provide an optimal control input that compromises the tracking performance in exchange for safety buffers to increase the chance of constraint satisfaction, as shown in Fig. 7(e). By taking a closer look at Figs. 7(f) and 7(g), we can see that the error bound (tube) is explicitly used when solving the constrained MPC, i.e., the robust MPC provides the solution where the predicted quantiles satisfy the constraints. Since the learned quantiles have already captured the possible state distribution under disturbance, the proposed robust MPC consequently allows future states to deviate from nominal values while still satisfying constraints.

Finally, we compare the distribution of the output trajectories with tube-based MPC as the benchmark using 1000 replicates. Tube-based MPC [16] is a well-established, computationally efficient, and widely understood and applied approach that provides robust constraint satisfaction, particularly effective when the system is linear. However, since the error bound in tube-based MPC is approximated using the worst-case scenario, it may be too conservative in practical applications. Figure 8 illustrates the distribution of the trajectories, where the thick lines represent the median of the trajectories at each time instance, and the color



**Fig. 7 Comparison of simultaneous multi-step MPC with and without robust consideration. (a) Trajectories of the states and input under multi-step (nominal) MPC. (b), (c), and (d) show the selected highlights of  $x_1$  and  $x_2$  from (a) where the constraints are active. (e) Trajectories of the states and input under multi-step robust MPC. (f), (g), and (h) show the selected highlights of  $x_1$  and  $x_2$  from (e) where the constraints are active**

shades indicate the interval between 0.05 and 0.95 quantiles. In this study, we examine the 1000 replicates at each time-step to calculate the constraint violation rate. We then use the maximum rate among the entire trajectory to represent the failure rate of each method. Figure 8(a) shows that without robust consideration, MPC easily violates the constraint, resulting in a 56.3% failure rate. In Fig. 8(b), although tube-based MPC yields a 6.2% failure rate and exhibits reliable performance, the margin between the lower bound of  $x_1$  and the reference/constraint is significant, echoing the over-conservative feature. On the other hand, in Fig. 8(c), the proposed robust MPC achieves a 5.8% failure rate and exhibits a smaller margin compared to that from the tube-based MPC. These results suggest that the learned quantile can be utilized as UQ while performing robust MPC. Furthermore, since TiDE directly learns the multistep-ahead response distribution from the data rather than approximating the error bound through uncertainty propagation, the simultaneous multistep robust MPC exhibits less conservative uncertainty estimation, leading to improved performance.

## 5 Engineering Case Study: Directed Energy Deposition Additive Manufacturing

In this section, we implement the proposed multistep robust MPC as the online decision-making process for the digital twin of the directed energy deposition (DED) AM system. Given the inherent uncertainty associated with material variability and environmental factors in the DED process, proactive control strategies, such as MPC, become crucial to achieve desired material properties while minimizing defects [54]. Furthermore, the intricate dynamics of the melt pool make it challenging to develop a physics-based model capable of providing accurate predictions in real-time. Therefore, data-driven methods have become promising tools for addressing this challenge.

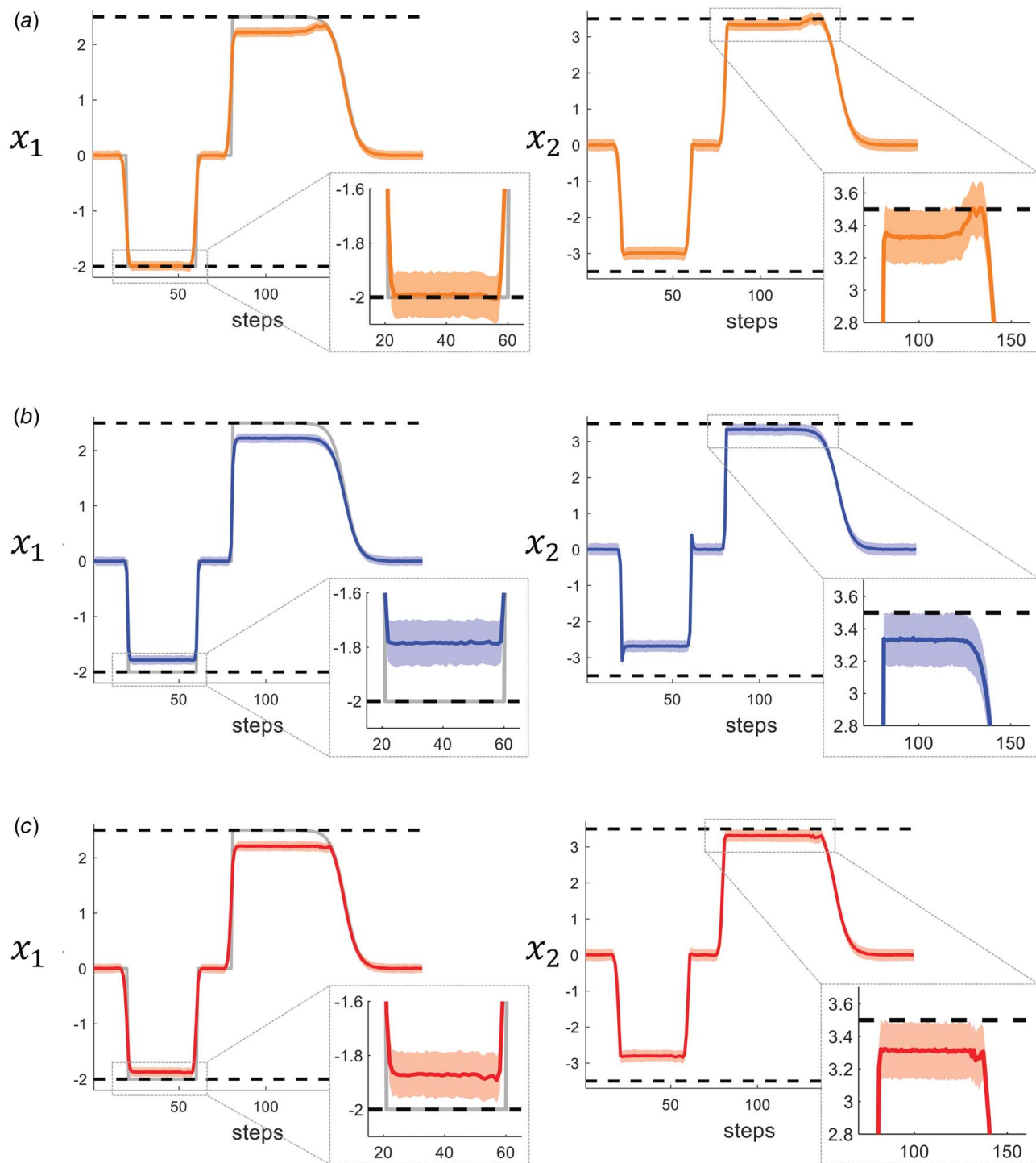
In this case study, our digital twin focuses exclusively on the melt pool rather than the entire part to enable real-time decision-making for process control. The melt pool is the most dynamic and sensitive

region of the process, where critical quality indicators such as porosity, microstructure, and residual stress originate. Modeling the entire part with sufficient fidelity for in situ control would be computationally prohibitive and incompatible with the actionable time requirements of online decision-making. In contrast, a melt pool-level digital twin allows for high-resolution, low-latency predictions that align with the frequency and spatial resolution of available in situ sensing data, such as pyrometer readings. Moreover, control inputs like laser power directly influence melt pool behavior, making it the most actionable domain for decision-making via MPC. By concentrating modeling efforts on the melt pool, the digital twin remains both computationally tractable and operationally relevant, enabling accurate state prediction and MPC throughout the build process.

The digital twin of the DED process in this work comprises three key modules: (a) the physical system, represented by a high-fidelity finite element analysis (FEA) simulation, (b) the virtual system, implemented as a TiDE model that predicts melt pool temperature and depth, and (c) the virtual-to-physical integration, achieved through the proposed robust MPC framework. At each MPC time-step, the current melt pool temperature and depth are extracted from the physical system and used as *feedback* to the virtual system. The virtual system then predicts future melt pool behavior conditioned on both the feedback and the candidate control input, which is subsequently optimized by the robust MPC. Once the optimal control input is determined, it is *applied* to the physical system, completing the bidirectional interaction between the physical and virtual domains.

**5.1 Physical System: Directed Energy Deposition Setup.** In this study, the physical DED is replaced by an in-house developed explicit FEA code developed by Liao et al. [55]. The code is accelerated by GPU computation using CuPy. It is employed for part-scale transient heat transfer simulation of the DED process. We select a single-track square as the target geometry, as shown in Fig. 9, and its specifications are listed in Table 3. This numerical





**Fig. 8** Distribution of trajectories under 1000 replicates. (a) Trajectories of multistep (nominal) MPC (b) Trajectories of tube-based robust MPC and (c) Trajectories of the proposed robust MPC.

setup allows efficient simulation of complex thermal dynamics while maintaining high accuracy in capturing melt pool behavior. Here, we highlight that since the layer height is 0.75 mm, we can set  $x_{\text{depth,lb}}$  and  $x_{\text{depth,ub}}$  to 0.225 and 0.075, corresponding to 10% and 30% dilution. For the details of the setup of DED, feature extraction, and data processing, please refer to our previous work [56] for the technical details.

**5.2 Virtual System: Surrogate Modeling for the Melt Pool of Directed Energy Deposition.** To effectively generate a

variety of laser power profiles, the design of experiment of the time series of the laser power profile is implemented. This method, proposed by Karkaria et al. [3], represents each time series with a 10-dimensional space using the Fourier transform and generates laser power trajectories using Latin hypercube sampling. For details of this approach, please refer to our previous work [3,56]. A total of 100 simulations with varying laser power profiles are conducted. The melt pool temperature and depth are extracted from the FEA model at each time-step. This diverse dataset ensures that TiDE can learn the relationship between process parameters and thermal responses across a wide operational

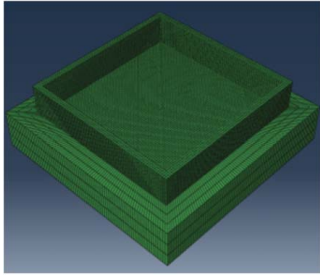


Fig. 9 Single-track square

Table 3 Specification of the printed square

Item	Quantity
Side length	40 mm
Track width	1.5 mm
Layer height	0.75 mm
Num. of layers	10 layers
Element size	0.375 mm
Num. of elements	40,540
Substrate height	10 mm
Scanning speed	7 mm/s

range. As the data are generated, we train a TiDE model as a multivariate multistep-ahead predictor with window size  $w = 50$  and horizon size  $N = 50$ , and the 0.95 and 0.05 quantiles are assigned as the upper and lower bounds. The specifics of training details, as well as model evaluation, are provided in our previous work [56]. As a result, the MAPE and the RRMSE for melt pool temperature predictions are 1.29% and 0.054, respectively, and those for the melt pool depth are 4.25% and 0.0441.

In our simulation, the primary sources of data uncertainty are the numerical errors in the FEA simulation. Specifically, since the laser's travel distance at each simulation time-step does not correspond to the element size, the heat treatment time of each element will vary, resulting in substantial fluctuations in both melt pool temperature and depth. Consequently, these fluctuations will overshadow the effects of injected uncertainties on material variability or disturbances of our quantities of interest. This variability introduces aleatoric uncertainty, which TiDE effectively captures through its quantile-based predictions. Although this type of noise is repeatable, it appears to be irreducible by increasing data collection, as evidenced by the TiDE prediction. In fact, TiDE only extracts the pertinent features and smooths the nominal (median) response using the dense encoder. It then allows the learning quantile to handle the fluctuations within the training set. Given the objective of this work to demonstrate how the predicted model quantifies uncertainty and leverages it to enhance decision-making in digital twins, we contend that this source of uncertainty presents a more extreme scenario to evaluate the efficacy of the proposed method.

**5.3 Virtual-to-Physical Integration: Robust Model Predictive Control for Directed Energy Deposition.** The objective of implementing MPC in DED is to establish a proactive control strategy that effectively mitigates defects when an arbitrary reference trajectory for melt pool temperature is provided. In DED, porosity emerges as the most prevalent and critical defect, directly impacting the mechanical properties of printed components. Therefore, to mitigate defects, it is suggested to maintain the melt pool depth within a dilution range of 10–30% to avoid interlayer and intralayer porosity [57]. Here, we assume that the melt pool depth is observable. In our previous work [56], the simultaneous multistep MPC has been

successfully implemented in melt pool depth constraint handling, using only nominal MPC. However, due to the intrinsic aleatoric uncertainty in the collected data and the processing environment, we aim to extend our previous work to perform robust MPC to enhance the constraint satisfaction rate.

The robust MPC for melt pool temperature tracking can be formulated as

$$\min_{\mathbf{u}} \sum_{i=0}^{N-1} \left[ \|\hat{\mathbf{x}}_{\text{temp},k+i+1} - r_{\text{temp},k+i+1}\|_{\mathbf{Q}}^2 + \|\Delta \mathbf{u}_{k+i}\|_{\mathbf{R}}^2 \right] \quad (22a)$$

$$\text{s.t. } \Pr(\hat{x}_{\text{depth},k+i} \geq x_{\text{depth,lb}}) \geq 0.95 \quad \forall i \in \mathbb{N}_{[1,N]} \quad (22b)$$

$$\Pr(\hat{x}_{\text{depth},k+i} \leq x_{\text{depth,ub}}) \geq 0.95 \quad \forall i \in \mathbb{N}_{[1,N]} \quad (22c)$$

$$[\hat{\mathbf{x}}_{\text{temp},k+1}^f, \hat{\mathbf{x}}_{\text{depth},k+1}^f]^T = \text{TiDE}(\mathbf{x}_{\text{temp},k}^p, \mathbf{x}_{\text{depth},k}^p, \mathbf{d}_{x,k}^{p:f}, \mathbf{d}_{y,k}^{p:f}, \mathbf{z}_k^{p:f}, \mathbf{u}_k^{p:f}) \quad (22d)$$

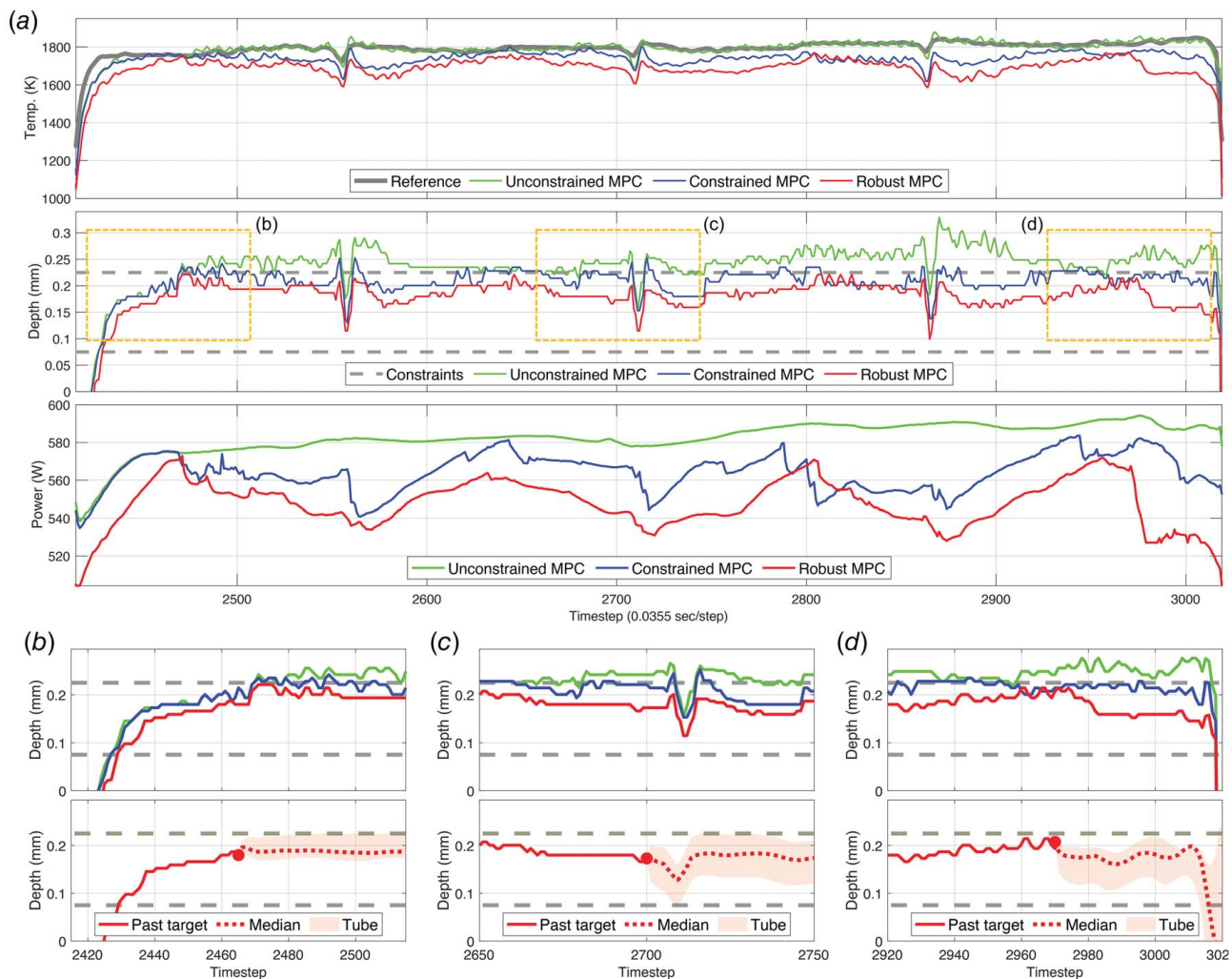
$$\mathbf{u}_{k+i} \in \mathbb{U} := \{\mathbf{u} \in \mathbb{R} \mid 504 \text{ W} \leq u_i \leq 750 \text{ W}\} \quad (22e)$$

where  $\Delta \mathbf{u}_{k+i-1}$  represents the differences between two consecutive terms in the designed future laser power, and the distance between the laser nozzle and the closest edge on the  $x$ - and  $y$ -directions are denoted as  $\mathbf{d}_x$  and  $\mathbf{d}_y$ , respectively.  $\mathbf{z}$  represents the nozzle location on the  $z$ -direction. These three quantities are determined based on the geometry and are treated as additional covariates to enhance the prediction capabilities of TiDE. In this case, we do not tighten the input bound since the bounds will not be active throughout the process.

**5.4 Model Predictive Control Results.** The outcomes of implementing a robust MPC algorithm in DED are presented in Fig. 10, where they are compared with the results obtained using unconstrained MPC and constrained MPC (which employs nominal prediction only), all employing the identical TiDE model as the multistep-ahead predictor. For a more detailed analysis, we have selected the trajectory from layer 5 as the representative. Figure 10 illustrates the trajectory across the entire layer, encompassing the start of a new layer, the abrupt temperature fluctuations at three corners due to the sharp change in scanning velocity, and the termination of the layer where the laser power is turned off. Given that the temperature/depth jumps and drops at the corners are unavoidable, we disregard the constraint violation penalty within a radius of 1 mm centered on the turning point.

Figure 10(a) presents a comparison of the trajectory of melt pool temperature and depth, along with the corresponding laser power input. The unconstrained MPC demonstrates exceptional reference tracking performance, with its trajectory yielding  $r^2 = 0.9730$  (for the entire trajectory) compared to the reference. This affirms the effectiveness of employing TiDE as a surrogate for multistep MPC. However, as constraints are enforced, the constrained MPC compromises its reference tracking performance in favor of constraint satisfaction, where  $r^2$  drops to 0.8261. As depicted in the figure, the resulting melt pool depth precisely adheres to the upper bound of the depth constraint. Nevertheless, since only nominal predictions are utilized in MPC and a safety buffer is not established, the constrained MPC occasionally violates the depth constraint. In contrast, robust MPC takes into account the potential state distribution, thereby generating a larger safety buffer from the melt pool depth constraint. Consequently, it effectively mitigates the constraint violation rate by compromising more on reference tracking, which results in  $r^2 = 0.6920$ .

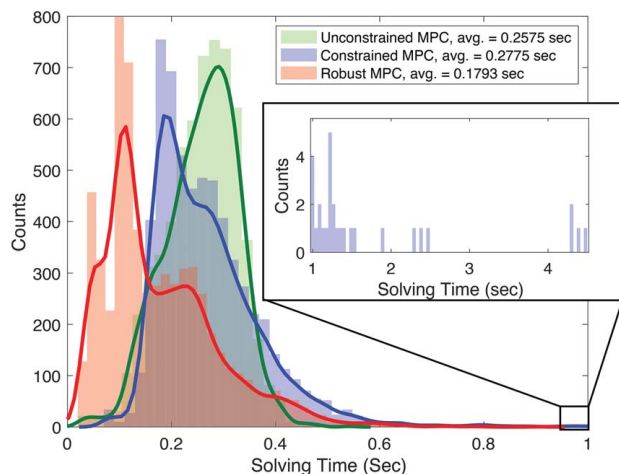
Figures 10(b)–10(d) highlight the critical regions on melt pool depth that are worth close examination. The upper subfigures



**Fig. 10 DED result comparison. (a) Trajectories of melt pool temperature, melt pool depth, and the corresponding laser power input. (b), (c), and (d) are the selected highlights comparing the details of the trajectories from different MPC methods, as well as the predicted medians (dashed lines) and tubes (shaded area) at particular time-steps.**

zoom in on the comparison of the trajectories, while the lower sub-figures present the predicted median and tube resulting from the optimal control inputs solved by robust MPC. Figure 10(b) illustrates the rise at the beginning of the layer, where the trajectory of the constrained MPC violates the constraint. It is evident that

the upper bound of the tube is utilized in robust MPC to adjust to the constraint, accommodating the disturbance during the process. Figure 10(c) exhibits that TiDE, along with its predicted tube, captures the distribution of depth variation at the corner. Figure 10(d) illustrates that since the laser power will be deactivated, and the feasible solution is not attainable from time-step 2960 until the end of the layer because the lower bound cannot be satisfied, the lower bound constraint is relaxed in this region to ensure feasibility.



**Fig. 11 Histogram of MPC solving time**

**5.5 Computational Time.** The histogram presented in Fig. 11 illustrates the computational time required to solve the unconstrained, constrained, and robust MPC problems at each step, computed using an AMD Ryzen Threadripper PRO 3975WX 32-Cores CPU. The results demonstrate that the average computational time for robust MPC is 0.1793 s, with a maximum of 0.903 s. These findings indicate that the proposed method can be effectively applied in various real-world scenarios, enabling real-time decision-making for digital twins. The relatively faster solving time of robust MPC compared to unconstrained MPC can be attributed to the creation of a safety buffer by the tube constraint. This buffer limits the feasible solution space, potentially leading to a shorter optimization process. Furthermore, robust MPC also results in a shorter solution time compared to constrained MPC. The primary reason for this is that constrained MPC encounters more constraint violations, requiring more iterations to converge. This is because employing an



infeasible solution as the initial guess for the penalty method hinders its constraint handling capability.

This case study verifies the suitability of our method for complex engineering applications. It highlights TiDE's capability as a surrogate model for multistep predictions with UQ and demonstrates the effectiveness of multistep robust MPC in efficiently handling constraints and solving MPC problems. The integration of these approaches presents a robust and practical framework for decision-making processes of the digital twin of engineering systems.

## 6 Closure

This work proposed a simultaneous multistep robust MPC framework that integrates TiDE with quantile regression to enable real-time decision-making for digital twins with uncertainty awareness. By leveraging TiDE's capacity for multistep predictions and efficient UQ from quantile regression, the proposed framework demonstrated an effective approach to quantify aleatoric uncertainty, and further benefit solving robust MPC with a series of acceleration techniques using automatic differentiation. In contrast to conventional single-step MPC approaches that necessitate recursive rollout for state prediction and conservative uncertainty approximation, the simultaneous multistep predictions reduced the number of function calls associated with recursive state propagation. Furthermore, the quantile-based uncertainty representation improved constraint satisfaction in the presence of stochastic disturbances. Through the validation of numerical simulations and engineering case studies employing DED, we demonstrate the exceptional surrogate modeling capabilities of TiDE for complex system dynamics with multistep-ahead prediction. Furthermore, we highlight the potential of this learning-based MPC framework to provide precise and proactive control strategies for intricate, nonlinear systems. This establishes a foundation for future advancements in uncertainty-aware digital twin applications.

While this work demonstrates substantial advancements, several limitations remain. To begin with, the effectiveness of the proposed framework relies heavily on the quality and diversity of the training data, which may limit its generalizability to scenarios involving unseen disturbances or operating conditions not captured during model training. Additionally, while TiDE reduces computational overhead compared to traditional methods, the computational demands may still pose challenges for real-time applications with a high-dimensional design space. Furthermore, the current implementation lacks mechanisms for dynamic adaptation to evolving system dynamics or disturbances beyond pretrained models, which could limit its robustness in highly variable environments. Moreover, we only demonstrate the effectiveness of the proposed method on stable systems, while it might be challenging for the implementation of unstable systems. Lastly, this study does not carry out comprehensive proofs of stability, recursive feasibility, performance guarantee, convergence, etc. These limitations highlight opportunities for future work to enhance the framework's adaptability, efficiency, and applicability across more complex and unpredictable systems.

In the future, we will develop a framework that enables the dynamic adaptation of the surrogate model through effective parameter fine-tuning methods. This will enhance the resilience, trustworthiness, and flexibility of the surrogate model as well as the decision-making process, thereby fulfilling the full potential of digital twin systems.

## Acknowledgment

We are grateful for the grant support from the National Science Foundation's Engineering Research Center for Hybrid Autonomous Manufacturing: Moving from Evolution to Revolution (ERC-HAMMER), under the Award Number EEC-2133630, the NSF MADE-PUBLIC Future Manufacturing Research Grant Program under Award Number CMMI-2037026, and the NASA Research Grant under the Award Number 80NSSC24M0176. Yi-Ping Chen also acknowledges the Taiwan-Northwestern

Doctoral Scholarship and the fellowship support from Northwestern University for the Predictive Science & Engineering Design (PS&ED) Cluster project. We acknowledge the technical support from Faith Rolark and Daniel Quispe on the DED simulation, and the insightful discussion with Dr. Aaditya Chandrasekhar, Dr. Jian Cao, and Dr. Robert X. Gao.

## Conflict of Interest

There are no conflicts of interest.

## Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

## References

- [1] National Academies of Sciences, Engineering, and Medicine, 2023, "Foundational Research Gaps and Future Directions for Digital Twins."
- [2] van Beek, A., Neville Karkaria, V., and Chen, W., 2023, "Digital Twins for the Designs of Systems: A Perspective," *Struct. Multidiscipl. Optim.*, **66**(3), p. 49.
- [3] Karkaria, V., Goeckner, A., Zha, R., Chen, J., Zhang, J., Zhu, Q., Cao, J., Gao, R. X., and Chen, W., 2024, "Towards a Digital Twin Framework in Additive Manufacturing: Machine Learning and Bayesian Optimization for Time Series Process Optimization," *J. Manuf. Syst.*, **75**, pp. 322–332.
- [4] Karkaria, V., Tsai, Y.-K., Chen, Y.-P., and Chen, W., 2025, "An Optimization-Centric Review for Integrating Artificial Intelligence and Digital Twin Technologies in Manufacturing," *Eng. Optim.*, **57**(Issue 1: 50th Anniversary Special Issue of Engineering Optimization), pp. 161–207.
- [5] Zenskov, A. D., Fu, Y., Li, R., Wang, X., Karkaria, V., Tsai, Y.-K., Chen, W., Zhang, J., Gao, R., and Cao, J., 2024, "Security and Privacy of Digital Twins for Advanced Manufacturing: A Survey," preprint arXiv:2412.13939.
- [6] Zhong, D., Xia, Z., Zhu, Y., and Duan, J., 2023, "Overview of Predictive Maintenance Based on Digital Twin Technology," *Heliyon*, **9**(4), p. e14534.
- [7] Grieves, M., and Vickers, J., 2017, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, F. J. Kahlen, S. Flumerfelt, and A. Alves, eds., Springer, AG Switzerland, pp. 85–113.
- [8] Rawlings, J. B., Mayne, D. Q., and Diehl, M., 2017, *Model Predictive Control: Theory, Computation, and Design*, Vol. 2, Nob Hill Publishing, Madison, WI.
- [9] Mesbah, A., 2016, "Stochastic Model Predictive Control: An Overview and Perspectives for Future Research," *IEEE Control Syst. Mag.*, **36**(6), pp. 30–44.
- [10] Mayne, D. Q., Seron, M. M., and Raković, S. V., 2005, "Robust Model Predictive Control of Constrained Linear Systems With Bounded Disturbances," *Automatica*, **41**(2), pp. 219–224.
- [11] Manzano, J. M., Limon, D., and Calliess, J.-P., 2020, "Robust Learning-Based MPC for Nonlinear Constrained Systems," *Automatica*, **117**, p. 108948.
- [12] Hewing, L., Wabersich, K. P., Menner, M., and Zeilinger, M. N., 2020, "Learning-Based Model Predictive Control: Toward Safe Learning in Control," *Annu. Rev. Control Rob. Auton. Syst.*, **3**(1), pp. 269–296.
- [13] Löfberg, J., 2003, *Minimax Approaches to Robust Model Predictive Control*, Linköping University Electronic Press, Linköping.
- [14] Bemporad, A., and Morari, M., 2007, "Robust Model Predictive Control: A Survey," *Robustness in Identification and Control*, Springer, New York, pp. 207–226.
- [15] Gonzalez, R., Fiacchini, M., Alamo, T., Guzmán, J. L., and Rodríguez, F., 2011, "Online Robust Tube-Based MPC for Time-Varying Systems: A Practical Approach," *Int. J. Control*, **84**(6), pp. 1157–1170.
- [16] Tsai, Y.-K., and Malak, R. J., 2023, "Robust Control Co-design Using Tube-Based Model Predictive Control," 2023 American Control Conference (ACC), San Diego, CA, May 31–June 2, IEEE, pp. 769–775.
- [17] Langson, W., Chrysoschoos, I., Raković, S., and Mayne, D. Q., 2004, "Robust Model Predictive Control Using Tubes," *Automatica*, **40**(1), pp. 125–133.
- [18] Lorenzen, M., Dabbene, F., Tempo, R., and Allgöwer, F., 2016, "Constraint-Tightening and Stability in Stochastic Model Predictive Control," *IEEE Trans. Autom. Control*, **62**(7), pp. 3165–3177.
- [19] Tsai, Y.-K., and Malak, R., 2025, "Control Co-design With Performance-Robustness Trade-Off Using Tube-Based Stochastic Model Predictive Control," *ASME Lett. Dyn. Syst. Control*, **5**(3), p. 030903.
- [20] Thelen, A., Zhang, X., Fink, O., Lu, Y., Ghosh, S., Youn, B. D., and Todd, M. D., 2022, *A Comprehensive Review of Digital Twin—Part 1: Modeling and Twinning Enabling Technologies*, Vol. 65, Springer, Berlin.
- [21] Drgha, J., Kiš, K., Tuor, A., Vrabie, D., and Klaučo, M., 2022, "Differentiable Predictive Control: Deep Learning Alternative to Explicit Model Predictive Control for Unknown Nonlinear Systems," *J. Process Control*, **116**, pp. 80–92.
- [22] Jung, M., da Costa Mendes, P. R., Önnheim, M., and Gustavsson, E., 2023, "Model Predictive Control When Utilizing LSTM as Dynamic Models," *Eng. Appl. Artif. Intell.*, **123**, p. 106226.

- [23] Wu, Z., Rincon, D., and Christofides, P. D., 2019, "Real-Time Adaptive Machine-Learning-Based Predictive Control of Nonlinear Processes," *Ind. Eng. Chem. Res.*, **59**(6), pp. 2275–2290.
- [24] Huang, K., Wei, K., Li, F., Yang, C., and Gui, W., 2022, "LSTM-MPC: A Deep Learning Based Predictive Control Method for Multimode Process Control," *IEEE Trans. Ind. Electron.*, **70**(11), pp. 11544–11554.
- [25] Kabir, H. D., Khosravi, A., Hosen, M. A., and Nahavandi, S., 2018, "Neural Network-Based Uncertainty Quantification: A Survey of Methodologies and Applications," *IEEE Access*, **6**, pp. 36218–36234.
- [26] Ren, L., and Xi, Z., 2024, "Advancing Vehicle Trajectory Prediction: A Probabilistic Approach Using Combined Sequential Models," *Journal of Autonomous Vehicles and Systems*, **5**(2), p. 021001.
- [27] Hewing, L., Kabzan, J., and Zeilinger, M. N., 2019, "Cautious Model Predictive Control Using Gaussian Process Regression," *IEEE Trans. Control Syst. Technol.*, **28**(6), pp. 2736–2743.
- [28] Fan, D. D., Agha-mohammadi, A.-A., and Theodorou, E. A., 2020, "Deep Learning Tubes for Tube MPC," preprint arXiv:2002.01587.
- [29] Park, J., Babaei, M. R., Munoz, S. A., Venkat, A. N., and Hedengren, J. D., 2023, "Simultaneous Multi-Step Transformer Architecture for Model Predictive Control," *Comput. Chem. Eng.*, **178**, p. 108396.
- [30] Köhler, J., Wabersich, K. P., Berberich, J., and Zeilinger, M. N., 2022, "State Space Models vs. Multi-Step Predictors in Predictive Control: Are State Space Models Complicating Safe Data-Driven Designs?", 2022 IEEE 61st Conference on Decision and Control (CDC), Cancun, Mexico, Dec. 6–9, IEEE, pp. 491–498.
- [31] Terzi, E., Fagiano, L., Farina, M., and Scattolini, R., 2019, "Learning-Based Predictive Control for Linear Systems: A Unitary Approach," *Automatica*, **108**, p. 108473.
- [32] Terzi, E., Farina, M., Fagiano, L., and Scattolini, R., 2022, "Robust Multi-Rate Predictive Control Using Multi-Step Prediction Models Learned From Data," *Automatica*, **136**, p. 109852.
- [33] Cheung, J., Rangarajan, S., Maddocks, A., Chen, X., and Chandra, R., 2024, "Quantile Deep Learning Models for Multi-Step Ahead Time Series Prediction," preprint arXiv:2411.15674.
- [34] Hu, J., Tang, J., and Liu, Z., 2024, "A Novel Time Series Probabilistic Prediction Approach Based on the Monotone Quantile Regression Neural Network," *Inf. Sci.*, **654**, p. 119844.
- [35] Thelen, A., Zhang, X., Fink, O., Lu, Y., Ghosh, S., Youn, B. D., Todd, M. D., Mahadevan, S., Hu, C., and Hu, Z., 2022, "A Comprehensive Review of Digital Twin—Part 1: Modeling and Twinning Enabling Technologies," *Struct. Multidiscipl. Optim.*, **65**(12), p. 354.
- [36] Tsai, Y.-K., and Malak, R. J., 2022, "Design of Approximate Explicit Model Predictive Controller Using Parametric Optimization," *ASME J. Mech. Des.*, **144**(12), p. 124501.
- [37] Zhang, X., Margellos, K., Goulart, P., and Lygeros, J., 2013, "Stochastic Model Predictive Control Using a Combination of Randomized and Robust Optimization," 52nd IEEE Conference on Decision and Control, Firenze, Italy, Dec. 10–13, IEEE, pp. 7740–7745.
- [38] Zhang, J., and Ohtsuka, T., 2021, "Stochastic Model Predictive Control Using Simplified Affine Disturbance Feedback for Chance-Constrained Systems," *IEEE Control Syst. Lett.*, **5**(5), pp. 1633–1638.
- [39] Gao, Y., Gray, A., Tseng, H. E., and Borrelli, F., 2014, "A Tube-Based Robust Nonlinear Predictive Control Approach to Semiautonomous Ground Vehicles," *Veh. Syst. Dyn.*, **52**(6), pp. 802–823.
- [40] Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., and Yu, R., 2023, "Long-Term Forecasting With Tide: Time-Series Dense Encoder," preprint arXiv:2304.08424.
- [41] Oreshkin, B. N., Carpio, D., Chapados, N., and Bengio, Y., 2019, "N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting," preprint arXiv:1905.10437.
- [42] Koenker, R., and Hallock, K. F., 2001, "Quantile Regression," *J. Econ. Perspect.*, **15**(4), pp. 143–156.
- [43] Yu, K., Lu, Z., and Stander, J., 2003, "Quantile Regression: Applications and Current Research Areas," *J. R. Stat. Soc. Ser. D: Stat.*, **52**(3), pp. 331–350.
- [44] Amini, A., Schwarting, W., Soleimany, A., and Rus, D., 2020, "Deep Evidential Regression," *Adv. Neural Inform. Process. Syst.*, **33**, pp. 14927–14937.
- [45] Heirung, T. A. N., Paulson, J. A., O'Leary, J., and Mesbah, A., 2018, "Stochastic Model Predictive Control—How Does It Work?" *Comput. Chem. Eng.*, **114**, pp. 158–170.
- [46] Cannon, M., Kouvaritakis, B., and Ng, D., 2009, "Probabilistic Tubes in Linear Stochastic Model Predictive Control," *Syst. Control Lett.*, **58**(10–11), pp. 747–753.
- [47] Chen, Y.-P., Wang, L., Comlek, Y., and Chen, W., 2024, "A Latent Variable Approach for Non-Hierarchical Multi-Fidelity Adaptive Sampling," *Comput. Methods Appl. Mech. Eng.*, **421**(1), p. 116773.
- [48] Schwedersky, B. B., and Flesch, R. C., 2022, "Nonlinear Model Predictive Control Algorithm With Iterative Nonlinear Prediction and Linearization for Long Short-Term Memory Network Models," *Eng. Appl. Artif. Intell.*, **115**, p. 105247.
- [49] González, E., Sanchis, J., García-Nieto, S., and Salcedo, J., 2020, "A Comparative Study of Stochastic Model Predictive Controllers," *Electronics*, **9**(12), p. 2078.
- [50] Lopez, B. T., Slotine, J.-J. E., and How, J. P., 2019, "Dynamic Tube MPC for Nonlinear Systems," 2019 American Control Conference (ACC), Philadelphia, PA, July 10–12, IEEE, pp. 1655–1662.
- [51] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., and Antiga, L., 2019, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *Adv. Neural Inform. Process. Syst.*, **32**.
- [52] Nocedal, J., and Wright, S. J., 1999, *Numerical Optimization*, Springer.
- [53] Feinman, R., 2021, "PyTorch-Minimize: A Library for Numerical Optimization With Autograd," <https://github.com/rfeinman/pytorch-minimize>
- [54] Gunasegaram, D., Barnard, A., Matthews, M., Jared, B., Andreaco, A., Bartsch, K., and Murphy, A., 2024, "Machine Learning-Assisted In-Situ Adaptive Strategies for the Control of Defects and Anomalies in Metal Additive Manufacturing," *Addit. Manuf.*, **81**(5), p. 104013.
- [55] Liao, S., Golgoon, A., Mozaffar, M., and Cao, J., 2023, "Efficient GPU-Accelerated Thermomechanical Solver for Residual Stress Prediction in Additive Manufacturing," *Comput. Mech.*, **71**(5), pp. 879–893.
- [56] Chen, Y.-P., Karkaria, V., Tsai, Y.-K., Rolark, F., Quispe, D., Gao, R. X., Cao, J., and Chen, W., 2025, "Real-Time Decision-Making for Digital Twin in Additive Manufacturing With Model Predictive Control Using Time-Series Deep Neural Networks," *J. Manuf. Syst.*, **80**, pp. 412–424.
- [57] Dass, A., and Moridi, A., 2019, "State of the Art in Directed Energy Deposition: From Additive Manufacturing to Materials Design," *Coatings*, **9**(7), p. 418.