ASME Journal of Manufacturing Science and Engineering
Online journal at:
https://asmedigitalcollection.asme.org/manufacturingscience

# 4DYNAMO: Analyzing and Optimizing Process Parameters in 4D Printing for Dynamic 3D Shape Morphing Accuracy

**Michael Biehler**
H. Milton School of Industrial
and Systems Engineering,
Georgia Institute of Technology,
755 Ferst Dr NW,
Atlanta, GA 30332
e-mail: michael.biehler@gatech.edu

**Daniel Lin**
Walton High School,
1590 Bill Murdock Road,
Marietta, GA 30062
e-mail: daniel.tt.lin@gmail.com

**Reinaldo Mock**
Wallace H. Coulter
Department of Biomedical Engineering,
Georgia Institute of Technology,
313 Ferst Dr NW,
Atlanta, GA 30332
e-mail: reinaldo@gatech.edu

**Jianjun Shi**[1]
H. Milton School of Industrial
and Systems Engineering,
Georgia Institute of Technology,
755 Ferst Dr NW,
Atlanta, GA 30332
e-mail: jianjun.shi@isye.gatech.edu

*Additive manufacturing (AM), commonly referred to as 3D printing, has undergone significant advancements, particularly in the realm of stimuli-responsive 3D printable and programmable materials. This progress has led to the emergence of 4D printing, a fabrication technique that integrates AM capabilities with intelligent materials, introducing dynamic functionality as the fourth dimension. Among the stimuli-responsive materials, shape memory polymers have gained prominence, notably for their crucial applications in stress-absorbing components. However, the exact 3D shape morphing of 4D printed products is affected by both the 3D printing conditions as well as the stimuli activation. Hence it has been hard to precisely control the 3D shape morphing accuracy. To model and optimize the dynamic 3D evolution of the 4D printed parts, we conducted both simulation studies and real-world experiments and introduced a novel machine-learning approach extending the concept of normalizing flows. This method not only enables the process optimization of the dynamic 3D profile evolution by optimizing the process conditions during 3D printing and stimuli activation but also provides interpretability for the intermediate shape morphing process. This research contributes to a deeper understanding of the nuanced interplay between process parameters and the dynamic 3D transformation process in 4D printing.* [DOI: 10.1115/1.4066222]

*Keywords: 4D printing, dynamic 3D profile evolution, process optimization and control, 3D point clouds, additive manufacturing, control and automation, inspection and quality control, modeling and simulation, nontraditional manufacturing processes, process engineering, sensing, monitoring, and diagnostics*

## 1 Introduction

In recent years, the development of structured materials capable of dynamic 3D shape changes in response to their environment has opened promising avenues for diverse applications, ranging from foldable robots to foldable electronics. Despite the impressive demonstrations of complex morphing motions, the fabrication and precise control of such devices remain challenging. This paper aims to bridge this gap by introducing a methodology for the optimization of process parameters in 4D printing for dynamic 3D shape morphing accuracy, termed 4DYNAMO. The focus is on controlling the dynamic 3D morphing process. In terms of methodological development, we extend the concept of continuous normalizing flows (CNFs), a machine-learning technique that iteratively transforms a simpler distribution into a more complex one. Conventional normalizing flow models face limitations in incorporating 3D point cloud data and process conditions and lack the ability to control these process conditions for achieving a desired 3D shape morphing behavior. Furthermore, the intermediate transformations of 4DYNAMO serve not only as computational tools but also provide interpretability regarding the speed and behavior of the 3D morphing process. This is crucial for applications like soft robotics, where the 4D-printed component serves as grippers. If the shape transformation occurs too rapidly, there is a risk of the object sliding out of the actuator's grip. Figure 1 shows how the 3D-printed part $X_0$ undergoes a series of dynamic shape transformations during the stimuli activation, induced by different temporal stimuli control $(u_1, \ldots, u_T)$ such as heat or magnetic activation. The intermediate shape transformation is measured via 3D scanning $(X_1, \ldots, X_T)$ during model training and predicted during the model deployment. The shape morphing behavior is modeled via an incremental flow model $f_i$, which is invertible and can be used for process optimization: by specifying a certain target shape $X_T$, the optimal control actions $(u_0^*, \ldots, u_T^*)$ can be derived.

Note that any intermediate shape can also be controlled. Furthermore, we also control the 3D printing process $u_0^*$, since the induced material properties influence the 3D morphing process.

Our model employs unstructured 3D point cloud data to achieve precise control over the entire 3D profile. This is particularly beneficial for applications like soft grippers, where precise control of the

[1] Corresponding author.

Manuscript received April 20, 2024; final manuscript received August 5, 2024; published online August 29, 2024. Assoc. Editor: Qing (Cindy) Chang.

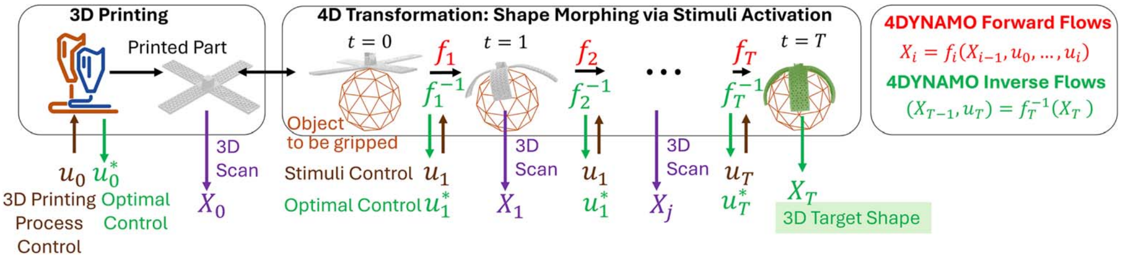

**Fig. 1 Overview of the proposed 4DYNAMO framework demonstrated in the example of a soft gripper**

entire actuator 3D surface is essential. In this paper, we investigate the gripper design shown in Fig. 1 through a finite element simulation study and validate our approach with case study experiments. The process parameters utilized for process optimization include printing speed, printing temperature, and stimuli activation temperature.

The main contributions of this paper are as follows:

- We introduce a novel framework designed to optimize the dynamic 3D shape morphing behavior in 4D printing. Our approach focuses on the direct optimization of process settings during 3D printing and stimuli activation, facilitating the realization of the desired 3D accuracy.
- We leverage and extend the continuous normalizing flow methodology [1], enhancing the efficiency and interpretability of modeling and optimizing 3D morphing behavior. Unlike previous methods that only capture intermediate features such as images of bending angles, our approach directly models the entire 3D shape using unstructured, 3D point clouds.
- We validate our methodology through simulation studies involving soft grippers and real-world case study experiments. These efforts not only validate our approach but also serve as a guiding example for designing and modeling diverse 4D printing processes. Our framework is adaptable to various physical dynamics or stimuli, provided they exhibit similar data characteristics.

The remainder of the paper is organized as follows. Section 2 gives a brief literature review. Then the proposed 4DYNAMO framework for real-time control of time-dependent 3D profiles is introduced in Sec. 3. In Sec. 4, we conduct a simulation study for a soft gripper application. In Sec. 5, we conducted a real-world case study with 4D printed shape memory polymers. Finally, we conclude the paper with a brief discussion and an outline of future work in Sec. 5.

## 2 Literature Review

In this section, we will review two categories of research that are closely related to the 4DYNAMO framework: the impact of process conditions on 3D shape morphing in 4D printing, and the continuous normalizing flow methodology.

**2.1 Impact of 3D Printing Process Conditions on 3D Shape Morphing in 4D Printing.** A limited number of research articles have investigated the impact of process conditions during both 3D printing and stimuli activation. An et al. [2] utilized the bending angle as a simplified 2D indicator of 3D shape and discovered that printing speed and extrusion width during 3D printing have the most significant effects on the bending angle of thermoplastic polyurethane (TPU) and polylactic acid (PLA) filaments. Barletta et al. [3] found that for single-material PLA components, in addition to activation temperature, the shape height plays a significant role in shape recovery. Bodaghi et al. [4] investigated the impact of printing speed on the morphing process of PLA filaments using both experiments and finite element simulation. Tezerjani et al. [5] studied the impact of layer height, print speed, nozzle temperature, and bed temperature on the height of a PLA disk. They subsequently employed linear regression to select process parameters leading to maximum disk height. Akbar et al. [6] conducted thermomechanical testing of rhombus-shaped structures and correlated recovery and fixity ratios with printing parameters layer height, printing temperature, and speed using linear regression. They found that amorphous polymers were highly affected by printing temperature, while semi-crystalline polymers were heavily influenced by the interaction of all three printing parameters considered. Ren et al. [7] discovered that for direct ink writing of polyurethane elastomer materials, the 3D morphing behavior can be influenced by two additional stimuli during the 4D phase of the process: water and heat. Hosseinzadeh et al. [8] utilized response surface methodology to study the effect of fused filament fabrication (FFF) parameters such as layer height, print speed, and nozzle diameter on torsional and flexural deformation of PLA parts. They employed a linear regression model to maximize deformation but did not consider dynamic stimuli activation. Nam and Pei [9] developed some general guidelines for 4D printing after they investigated how the print pattern, infill density, and recovery temperature affect the shape recovery of 16 different filaments to their original shape after a morphing process.

Existing studies on the finite element modeling of this transformation process cannot be directly applied, as numerical models require approximations and simplifications. Several studies have identified discrepancies between the shape morphing behavior simulated by finite element analysis (FEA) and the actual behavior observed [10].

Compared to the existing literature, our proposed method has two unique advantages:

(i) In contrast to methods that rely solely on parameters like bending angles or other reduced measurements [7], our approach directly studies the complete 3D shape and its dynamic deformation. Moreover, we examine both stages of the 4D printing process: the 3D printing phase and the stimuli activation phase, recognizing their profound dependencies.

(ii) In contrast to batch-to-batch compensation strategies [11], our methodology not only allows for the control of a single shape but also offers the capability to control multiple shapes throughout the entire dynamic 3D shape morphing process. This feature enhances interpretability and enables applications like soft robotics.

**2.2 Continuous Normalizing Flow Methodology.** CNFs represent a generative modeling technique utilized to derive complex distributions through successive, invertible transformations from simpler distributions [1]. Flow-based models enable exact likelihood evaluation and efficient sampling. While CNF has been extended to incorporate conditional inputs [12], its predominant application has been in image generation with limited use in inference tasks [13]. Moreover, only a few studies have explored the potential of CNF for modeling 3D point clouds [14,15].

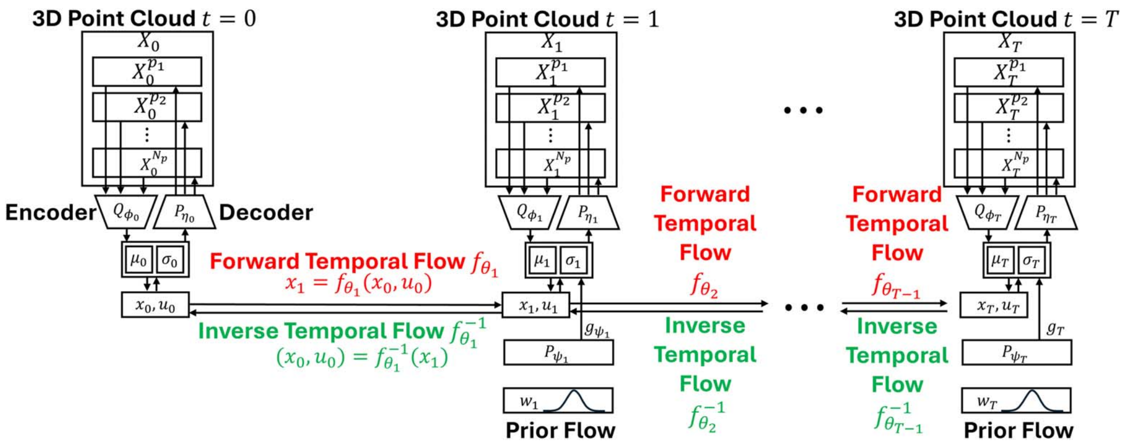

**Fig. 2 Overview of the 4DYNAMO methodology**

In our research, we extend CNF to facilitate inverse design for process parameters and incorporate additional supervision for intermediate transformation steps, a unique data characteristic in 4D printing. By employing this modeling approach, we enhance the interpretability of the 3D morphing process, which holds significance for applications such as soft robotics. Furthermore, we exploit the inverse property of iterative flow models to optimize the printing process. This overcomes the need for a potentially suboptimal two-step procedure involving the derivation of a process model followed by optimization, thus streamlining the process and design optimization workflow.

## 3 4DYNAMO Methodology

In this section, we present 4DYNAMO as a methodology for controlling dynamic 3D morphing behaviors within 4D printed devices. Our model development is grounded in the concept of normalizing flows (NFs) [1].

NFs are generative models with the same goal as diffusion models: to approximate and sample from a probability distribution. NFs have emerged as a competitive generative modeling method by constructing an invertible and efficiently differentiable mapping between a fixed (e.g., standard normal) distribution and the data distribution [1,16]. CNFs express this mapping through a neural ordinary differential equation (ODE) [17]. While CNFs have faced challenges in training and scaling to large datasets [18], our framework demonstrates that incorporating additional supervision signals and conditioning on manufacturing process parameters significantly enhances their effectiveness for 4D printing modeling.

A key advantage of CNFs is their ability to facilitate process optimization by leveraging their invertible properties to obtain optimal process inputs for a desired target shape. This direct optimization is a distinct benefit over other methods, which would require solving an additional optimization problem where performance could be compromised due to the non-convex nature of the prediction function.

Diffusion models, on the other hand, approximate a stochastic differential equation (SDE) that transforms a simple density to the data distribution. Their success partly stems from a straightforward regression training objective that does not necessitate simulating the SDE during training. Recent research has shown that CNFs can also be trained using a regression of the ODE's drift [19], similar to the training of diffusion models, through an objective called flow matching.

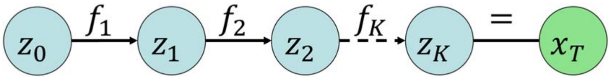

**Fig. 3 Graphical illustration of normalizing flow concept**

Our proposed CNF architecture further stabilizes the training process, specifically tailored to the engineering application of 4D printing. This approach leverages additional supervision from the intermediate shape transformations and the knowledge of the process conditions that affect these transformations. This unique integration ensures that our framework learns meaningful results with fewer training samples and provides a robust solution for 4D printing process optimization.

We begin by introducing the fundamentals of NF, followed by the proposal of a process input-conditional 3D normalizing flow model, and subsequently, the suggested control optimization approach. The novel 4DYNAMO framework is depicted in Fig. 2, and the subsequent paragraphs will introduce the notation and clarify its meanings.

**3.1 Background on Normalizing Flow Methodology.** A normalizing flow comprises a series of invertible mappings that transform an initial distribution into a more complicated one. Formally, let $f_1, \ldots, f_T$ denote a series of diffeomorphic transformations (i.e., invertible and differentiable) to be applied to a latent variable $z_0$ with distribution $P(z_0)$. The output variable is obtained via $x_T = f_T \circ f_{T-1} \circ \cdots \circ f_1(z_0)$, and the probability density of the output variable is expressed by the change of variables in the formula

$$\log P(x_T) = \log P(z_0) - \sum_{k=1}^{n} \log\left|\det \frac{\partial f_k}{\partial z_{0+k}}\right| \tag{1}$$

where $z_0$ can be obtained from $x_T$ using the inverse flow: $z_0 = f_1^{-1} \circ \cdots \circ f_T^{-1}(x_T)$. Graphically, the flow model can be visualized as a chain of successive transformations of $z_0$, with the observed data at the end of the chain, as shown in Fig. 3.

In practice, $f_1, \ldots, f_T$ are implemented as neural networks with an architecture designed to facilitate the efficient computation of the determinant of the Jacobian $\left|\det \frac{\partial f_k}{\partial x_{0+k}}\right|$. The quantity of training samples crucial for highlighting the neural network depends on the complexity of the shapes in 4D printed devices. In instances of relatively simple shapes, such as those in our case study, a few hundred samples appear to be adequate. However, for more intricate shapes, several thousand samples may be required. Additionally, we employ pre-training on the PartNet dataset to initialize the variational autoencoder (VAE) architecture for dimensionality reduction, which is a standard approach in the field [20,21].

**3.2 Problem Setup and Dimensionality Reduction for 3D Point Clouds.** 3D points serve as the primary output from sensing devices like laser scanners, making them a favored representation for their concise and standardized format. Compared to voxel grids, they offer significantly higher resolution, enabling more detailed capture of spatial data. In this paper, the dataset includes a collection of 3D shapes represented by 3D point clouds, along with an accompanying set of process inputs. More formally, $\mathcal{D} = \{ \{X_{i,j}\}_{j=1}^{T}, \{u_{i,j}\}_{j=1}^{T} \}_{i=1}^{N}$, where $i = 1, \ldots, N$ denotes the sample index of $N$ 4D devices and $j = 1, \ldots, T$ denotes the index for the number of shape changes. We assume $X_{i,j} \in \mathbb{R}^{3 \times N_p}$ is a 3D point cloud with three coordinates ($x$, $y$, and $z$) and $N_p$ measurement points. The process inputs $u_{i,j} \in \mathbb{R}^{d}$ consists of a vector of dimension $d$.

Given the high dimensionality of 3D point clouds, typically comprising thousands to millions of 3D measurement points in each sample, dimensionality reduction and obtaining a probabilistic embedding becomes necessary. We adopt a variational approach to model 3D point clouds, addressing their unique characteristics of unstructured organization, permutation- and rotation-invariance. Our approach involves representing 3D point clouds as a two-level hierarchy of distributions: the first level denotes the distribution of shapes, while the second level represents the distribution of points given a shape. This formulation allows for the representation of diverse shapes with varying numbers of points through simply sampling from the distribution.

The core idea lies in parametrizing the distribution of points within a shape as an invertible parameterized transformation of 3D points from a prior distribution, typically a 3D Gaussian, conditioned on the process inputs. Conceptually, this model allows obtaining points for a given shape by initially sampling points from a generic Gaussian prior, inputting the process parameters, and then moving these points according to the parameterized transformation to their new location within the target shape. This parameterization offers several advantages. The invertibility of these transformations facilitates not only sampling but also estimating probability densities. Estimating probability densities enables principled model training using the variational inference framework, maximizing a variational lower bound on the log-likelihood of a training set of point clouds. This probabilistic framework for training circumvents the need for hand-crafting appropriate distance metrics to measure differences between sets of points.

The VAE framework enables the learning of $P(X_{i,j})$ from a dataset of observations by characterizing the data distribution through a latent variable $x_{i,j}$ with a prior distribution $P_{\psi_j}(x_{i,j})$. Additionally, it employs a decoder, $P_{\eta_j}(X_{i,j}|x_{i,j})$, which captures the (ideally simpler) distribution of $X_{i,j}$ given $x_{i,j}$. During training, it also learns an inference model (or encoder) $Q_{\phi_j}(x_{i,j}|X_{i,j})$. The joint training of the encoder and decoder aims to maximize a lower bound on the log-likelihood of the observations, where $D_{KL}(\cdot||\cdot)$ is the Kullback–Leibler (KL) divergence

$$\begin{aligned}\log P_{\eta_j}(X_{i,j}) &\geq \log P_{\eta_j}(X_{i,j}) - D_{\mathrm{KL}}(Q_{\phi}(x_{i,j}|X_{i,j}) \parallel P_{\eta_j}(x_{i,j}|X_{i,j})) \\ &= \mathbb{E}_{Q_{\phi_j}(x_{i,j}|X_{i,j})}[\log P_{\eta,j}(X_{i,j}|x_{i,j})] - D_{\mathrm{KL}}(Q_{\phi_j}(x_{i,j}|X_{i,j}) \parallel P_{\psi_j}(x_{i,j}))\end{aligned} \tag{2}$$

This evidence lower bound (ELBO) can be interpreted as the combination of the negative reconstruction error (the first term) and a regularization term for the latent space (the second term). However, the conventional VAE model proves inadequate for learning a low-dimensional embedding of unstructured 3D point clouds. Typically, there is no prior structure in the set of 3D coordinates for complex shapes. Specifically, their data structure lacks topological awareness, and the spatial relationships among points are unknown, presenting considerable modeling challenges. Because 3D point clouds are represented as sets, they must adhere to permutation invariance. This means that altering the order of points in the point cloud still reflects the same underlying shape. Moreover, the distribution of points is irregular due to widely used devices such as laser scanners, resulting in sparse and dense point cloud regions influenced by acquisition conditions such as illumination. Hence, we proposed to decompose the reconstruction log-likelihood of a point cloud into the sum of the log-likelihoods of each measurement point $X_{i,j}^{p}$ in a point cloud $X_{i,j}$ ($p = 1, \ldots, N_p$).

$$\log P_{\eta,j}(X_{i,j}|x_{i,j}) = \sum_{X_{i,j}^{p} \in X_{i,j}} \log P_{\eta,j}(X_{i,j}^{p}|x_{i,j}) \tag{3}$$

We note that treating points independently may lead to a loss of spatial information, but it significantly increases computational efficiency. Additionally, the probabilistic representation of the 3D point clouds is advantageous. In the context of flow modeling, the initial sampling is less critical, and samples can be drawn independently. Predicting the 3D points for a given shape involves sampling points from a generic Gaussian prior and then transforming them according to a parameterized transformation to their positions in the target shape. In this formulation, a given shape is the variable that parameterizes this transformation. Nonetheless, this is a possible limitation of the model, which could be relaxed in future research.

This allows us to obtain a low-dimensional embedding $x_{i,j}$ of the point cloud $X_{i,j}$. Throughout the remainder of the paper, the lowercase $x_{i,j}$ will denote the embedding obtained from the modified VAE architecture.

### 3.3 Incorporation of Process Inputs and Intermediate 3D Morphing Shapes.

We propose to model $P(x_T)$ using a conditional extension of CNF. A point $x_{i,T}$ in the point cloud $X_{i,\,T}$ is the result of transforming some points $x_{i,0}$ in the prior distribution $P(x_{i,0})$ using a CNF conditioned on the preceding control inputs $\{u_{i,j}\}_{j=1}^{T-1}$ as well as the preceding 3D point cloud embeddings $\{x_{i,j}\}_{j=1}^{T-1}$. To simplify the notation, we denote the collection of control inputs and 3D point cloud embeddings by $y_{i,T} = \{ \{u_{i,j}\}_{j=1}^{T-1}, \{x_{i,j}\}_{j=1}^{T-1} \}$.

$$x_{i,T} = F_{\theta_T}(x_{i,0}; y_{i,T}) \triangleq x_{i,0} + \int_{t_0}^{T} f_{\theta}(x_{i,t}, t, y_{i,t})dt, \; x_{i,0} \sim P(x_{i,0}) \tag{4}$$

where $f_{\theta_T}$ defines the continuous-time dynamics of the flow $F_{\theta_T}$ conditioned on $y_{i,T}$. Note that the inverse of $F_{\theta_T}$ is given by $F_{\theta_T}^{-1}(x_{i,T}, y_{i,T}^{-1}) = x_{i,T} + \int_{T}^{t_0} f_{\theta_T}(x_{i,0}, t, y_{i,T}^{-1})dt$ with $y_{i,T}^{-1}$ denoting all the control inputs and shapes applied after $x_{i,T}$. Here, $y_{i,T}^{-1} = \{ \{u_{i,j}\}_{j=T}^{\tau}, \{x_{i,j}\}_{j=T}^{\tau} \}$, where $\tau$ is the total number of transformation steps. If we are in the last step of the 3D morphing process (i.e., $T = \tau$), then $y_{i,T}^{-1}$ reduces to $y_{i,T}^{-1} = u_{i,T}$. The reconstruction likelihood becomes

$$\log P_{\eta,j}(x_{i,T}|y_{i,T}) = \log P(F_{\theta_T}^{-1}(x_{i,T}, y_{i,T})) - \int_{t_0}^{T} Tr\left(\frac{\partial f_{\theta_T}}{\partial \, x_{i,t}}\right)dt \tag{5}$$

where $Tr(\cdot)$ denotes the trace of a matrix. Note that $\log P(F_{\theta_T}^{-1}(x_{i,T}, y_{i,T}))$ can be computed in closed form for a Gaussian prior. Nevertheless, the Gaussian prior is an uninformative prior that lacks the incorporation of prior knowledge regarding the shape morphing process. In the absence of additional information, the most reasonable assumption would be that the shape resembles the previous shape rather than adhering solely to a Gaussian distribution. In particular, it has shown that a Gaussian prior limits the performance of VAEs and leads to incorrect latent representations [22]. Therefore, we use another CNF to parameterize a learnable prior that considers the preceding shape $x_{i,T-1}$ . We rewrite the KL divergence term in Eq. (2) as

$$\begin{aligned}&D_{\mathrm{KL}}(Q_{\phi_j}(x_{i,T-1}|x_{i,0})) \parallel P_{\psi,j}(x_{i,T-1}) \\ &= -\mathbb{E}_{Q_{\phi_j}(x_{i,T-1},|x_{i,0})}[\log P_{\psi,j}(x_{i,T-1})] - H[Q_{\phi_j}(x_{i,T-1}|x_{i,0})]\end{aligned} \tag{6}$$

where $H$ denotes the entropy, and $P_{\psi_j}(x_{i,T-1})$ is the prior distribution with learnable parameters $\psi_j$, obtained by transforming a standard

Gaussian $P(w_{j,T}) = N(0, 1)$ using the following CNF:

$$(x_{i,T-1}) = G_{\psi_j}(w_{i,T-1}) \triangleq w_{i,T-1} + \int_{t_0}^{T} g_{\psi_j}(w_{i,t}, t)dt, \quad w_{i,T} \sim P(w_{i,T}) \tag{7}$$

where $g_{\psi_j}$ defines the continuous-time dynamics of the flow $G_{\psi_j}$. The inverse of $G_{\psi_j}$ is given by $G_{\psi_j}^{-1}(x_{i,T}) = (x_{i,T-1}) + \int_T^{t_0} g_{\psi_j}^{-1}(w_{i,T-1}(t), t)dt$. The log probability of the prior distribution can be computed by

$$\log_{G_{\psi_j}}(x_{i,T}, y_{i,T}) = \log P(x_{i,T}, y_{i,T}) - \int_{t_0}^{T} Tr\left(\frac{\partial g_{\psi_j}}{\partial w_i(t)}\right)dt \tag{8}$$

Combining the dimensionality reduction, prior flow, and temporal flow, the ELBO of a point $x_{i,T}^p$ in point cloud $X_{i,T}$ can finally be written as

$$\begin{aligned}&\mathcal{L}_{4\text{DYNAMO}}(x_{i,T}^p, x_{i,0}, y_{i,T}, \phi_j, \psi_j, \theta_j, \eta_j)\\&= \mathbb{E}_{Q_{\phi_j}(x_{i,0}, y_{i,T}|x_{i,T})}[\log P_{\eta,j}(x_{i,0}, y_{i,T}) + \log P_{\psi_j}(X_{i,j}|x_{i,0}, y_{i,T})]\\&\quad + H[Q_{\phi_j}(x_{i,0}, y_{i,T}|X_{i,j})]\end{aligned} \tag{9}$$

In comparison to the original flow model shown in Fig. 3, the 4DYNAMO model in Fig. 4 showcases two notable differences. First, it incorporates conditional process inputs rather than initializing the flow with a base distribution like a standard Gaussian. Second, it employs intermediate transformations during the 3D shape morphing stage to guide the successive transformations of the flow model, which leads to a reduction in the required sample size for training.

Our model is trained end-to-end by maximizing the ELBO of all 3D point clouds in the dataset using SGD

$$\phi^*, \psi^*, \theta^*, \eta^* = \arg\max_{\phi,\psi,\theta,\eta} \sum_i \sum_{X_i^p \in X_i} \mathcal{L}_{4\text{DYNAMO}}(x_T^p, x_0, y_T, \phi, \psi, \theta, \eta) \tag{10}$$

The tuning parameters in our model, including the weights of the different terms in the loss function, are selected using the standard procedure of cross-validation (CV) specific to each dataset.

**3.4 Process Optimization for Enhanced 3D Morphing Accuracy.** Process reproducibility and process-aware design optimization present a significant challenge for the widespread adaptation of 4D printing. Our objective is to leverage the invertible property to achieve optimal process settings directly. By doing so, we can utilize the invertible nature of forward flow predictions, eliminating the need for a potentially suboptimal two-step process. Traditionally, this process involves first learning a process model and then optimizing it. However, this approach often necessitates additional constraints or may result in predictions outside the training region of the process models, leading to suboptimal functional approximations.

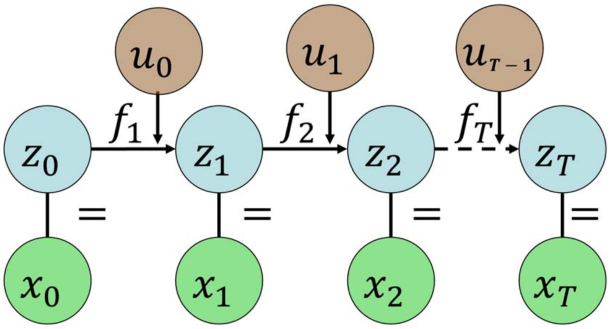

Fig. 4 Graphical illustration of 4DYNAMO flow concept

The ultimate goal of process optimization is to find control parameters $\{u_{i,j}\}_{j=1}^{T}$, which will lead the dynamic 3D morphing to output a given target shape $X_{i,T}$. This is intrinsically a generative problem in the context of machine learning, where the goal is to generate $u_{i,j}$ from a proper distribution $P(u_{i,j}|X_{i,T})$. However, it is important to highlight that we can optimize not only the control inputs but also leverage the intermediate 3D shape transformations to enhance the optimization of 3D morphing toward a desired final shape. For instance, specific applications such as soft robotics may necessitate a smooth, continuous evolution of the 3D shape. In such scenarios, we might prefer to distribute the dynamic 3D changes more evenly throughout the process. Hence, we will then control the intermediate shapes and perform optimization on $y_{i,T} = \{\{u_{i,j}\}_{j=1}^{T-1}, \{X_{i,j}\}_{j=1}^{T-1}\}$. With the forward flow, we can obtain and output $X_{i,T}$ for a given set of control inputs $y_{i,T}$, denotes as $F_\theta(y_{i,T})$. In particular, we seek $y_{i,T} = \{\{u_{i,j}^*\}_{j=1}^{T-1}, \{X_{i,j}^*\}_{j=1}^{T-1}\}$ to best fit for a specific output $X_{i,T}$ by minimizing $y_{i,T}^* = \arg\min_{y_{i,T}} \|F_\theta(y_{i,T}) - X_{i,T}\|_2^2$. However, due to the invertibility and differentiability of the normalizing flows, there is an even more simple and elegant solution. We can directly obtain the desired control inputs by applying the inverse flows

$$y_{i,T}^* = F_\theta^{-1}(X_{i,T}) = f_{\theta_1}^{-1} \circ \cdots \circ f_{\theta_T}^{-1}(X_{i,T}) \tag{11}$$

Figure 5 illustrates how the target 3D shape $X_{i,T}$ is the input to the inverse flow and leads to optimal control and 3D inputs for the shape morphing process.

We note that in general, there may exist multiple optimal solutions due to the non-uniqueness of this inversion process. However, by designing the flow functions $f_{\theta_j}$ as implicit normalizing flows [23] it is possible to obtain a unique solution. The idea is that in general, a normalizing flow is an explicit transformation of the form $z = f(x)$. By defining the mapping implicitly as the roots of the equation $F(z, x) = 0$ the non-uniqueness can be mitigated. For a detailed proof, interested readers are referred to Ref. [23]. We note that, in specific applications, there may be advantages to not restrict the functional form of $f_{\theta_j}$ and explore multiple solutions. Subsequently, one can take additional physical or cost constraints into account to determine the most suitable 3D morphing behavior for a particular case.

**3.5 Theoretical Analysis.** An important consideration in the design of the flow functions $f_{\theta_j}$ pertains to whether process conditions suggested by the inversion process outside of the training distribution will still yield desirable outcomes for process optimization. Generally, it is anticipated that such circumstances may not yield favorable results, with models being valid only within the training region. Generally, quantifying this behavior precisely remains challenging. Through the variational principles underlying our flow model, we can delineate the conditions under which this holds and propose methods to mitigate unfavorable behavior during process optimization.

Deep learning techniques have shown significant promise in addressing ill-posed inverse problems in imaging [24]. Among these techniques, NF-based methods stand out due to their unique ability to generate diverse solutions for a given input. However, artifacts stemming from an exploding inverse in NFs have been observed in the image generation domain [25]. In our initial

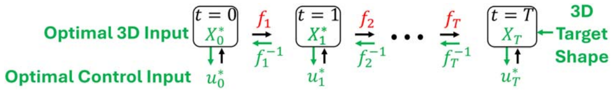

Fig. 5 Process optimization and control procedure

modeling experiments, we observed analogous behavior in certain configurations of the 4DYNAMO model, where process inputs outside the training range resulted in undesirable outcomes. This observation prompted us to investigate this phenomenon deeper and propose potential remedies.

For NFs to be effectively trainable, it is imperative that $f_{\theta_i}$ not only exhibit invertibility but also possess a computationally manageable Jacobian determinant. While various deep learning architectures, including those featuring convolutions, max-pooling, and ReLU layers, have proven successful, NFs cannot utilize them due to their lack of invertibility. Conditional coupling layers, integral to the NF's conditional nature, serve as the primary layer choice in NF architectures. More formally, a conditional coupling transformation is defined as $f\colon \mathcal{S} \to \mathcal{S} \subseteq \mathbb{R}^D$ is defined as

$$f(x_i) = \begin{cases} g(x_i;\, \vartheta_i) & i = d, \ldots, D \\ x_i & i = 1, \ldots, d-1' \end{cases} \tag{12}$$

where $\vartheta_i = NN(x_{1:d-1}, e_\theta(y))$, $NN$ is an arbitrary neural network, $e_\theta$ is an encoder for the conditional input $y$, and $g(x_i;\, \vartheta_i)\colon \mathcal{S}' \to \mathcal{S}' \subseteq \mathbb{R}$ is an invertible function parameterized by a vector $\vartheta_i$. The Jacobian determinant of this transformation is readily obtained from the derivative of $g(\cdot)$, expressed as

$$\det\left(\frac{\partial f}{\partial x}\right) = \prod_{i=d}^{D} \frac{\partial g(x_i;\, \vartheta_i)}{\partial x_i} \tag{13}$$

The inverse of $f(\cdot)$ is given by

$$f^{-1}(x_i) = \begin{cases} g^{-1}(x_i;\, \vartheta_i) & i = d, \ldots, D \\ x_i & i = 1, \ldots, d-1 \end{cases} \tag{14}$$

Affine transformations are commonly utilized due to their computational efficiency for their Jacobian and inverse as well as their sufficient expressive power.

$$g(x_i;\, \vartheta_i(y)) = a_i(y)x_i + b_i(y) \tag{15}$$

where $\vartheta_i = (a_i, b_i)$. Affine coupling transformations are suitable for generation and inference tasks on high-dimensional data. However, in most conditional NFs the affine coupling transformations are the only components that depend on $y$. However, this can lead to a problem known in the image generation literature as an exploding inverse [25,26]. While it is widely understood that out-of-distribution predictions, i.e., inputs beyond the distribution of the training set, may lead to suboptimal performance, theoretically verifying this behavior can be challenging beyond empirical validation alone. However, the probabilistic framework of normalizing flows enables us to quantify the error arising from the inversion process as follows:

**Proposition 1**. If $f_\theta\colon \mathcal{X} \times \mathcal{Y} \to Z \subseteq \mathbb{R}^D$ represents a conditional normalizing flow utilizing the conditional affine coupling transformation and training with a dataset drawn from the distribution $p(x|y)$, then there exist many instances in which $\|x, y\|_\infty \ll \|f_\theta^{-1}(f(x;\, y))\|_\infty$.

This can be seen from an argument using the convexity of the affine coupling layers, which aim to minimize the negative log-likelihood $-\mathbb{E}_{x,y\sim p_{x,y}}[\log q_{x,y}(x|y)]$. For simplicity, we assume that $f_\theta$ consist of one conditional affine coupling layer and $x, y \in \mathbb{R}^2$. Using the definition of the affine coupling layers for $d = D = 2$, we obtain the following negative log-likelihood:

$$\begin{aligned} &-\mathbb{E}_{x,y}\left[\log q_z(f_\theta(x;\, y)) + \log\left|\det \frac{\partial f_\theta}{\partial x}(x;\, y)\right|\right] \\ &= \mathbb{E}_{x,y}\left[\frac{\|f_\theta(x;\, y)\|_2^2}{2\sigma_z^2} + \log\left|\det\begin{bmatrix} 1 & 0 \\ * & a_1 \end{bmatrix}\right|\right] \\ &= \mathbb{E}_{x,y}\left[\frac{x_1^2 + (a_1x_1 + b_1)^2}{2\sigma_z^2} - \log(a_1)\right] \end{aligned} \tag{16}$$

where $*$ denotes a placeholder for the value of the corresponding partial derivative, and $z$ is assumed to be Gaussian. We can see that Eq. (16) is a convex function of $(a_1, b_1)$, which is unbounded from below. Hence there exists a degenerative case in which $a_1 \to \infty$ with $b_1 \to -a_1x_1$, which matches the degenerative behavior observed in our initial experiments as well as in the literature [27]. This analysis allows us to use a simple remedy for this behavior.

**Proposition 2:**. By adding an additional constraint to bound the affine coupling layers in $f_\theta$, the degenerative case can be avoided.

This undesirable unboundedness of the negative log-likelihood loss can be avoided by setting an upper bound on $a_1$ by simply solving

$$\min_{0<a_1\leq 1, b_1} \frac{x_1^2 + (a_1x_1 + b_1)^2}{2\sigma_z^2} - \log(a_1) \tag{17}$$

This approach has been utilized in some recent image-based flow models [28,29] and preserves sufficient expressive power while leading to the elimination of the degenerate case for the inversion process. Hence, we adopt this procedure for our 3D point cloud-based approach based on the theoretical analysis detailed above.

## 4 Retrofit Simulation Study—Soft Gripper

In the realm of 4D printing, inverse design via simulation is a critical research area [30]. However, current simulation methods are computationally expensive [31], highlighting the need for an accurate, differentiable, and fast-to-evaluate surrogate model for 4D printing design, that considers the manufacturing conditions. To address this gap, we conducted finite element simulations with two primary goals:

(1) To showcase the versatility of 4DYNAMO, demonstrating its capabilities not only as a data-driven model but also as a high-fidelity surrogate model for the design process.
(2) To assess the robustness of our 4DYNAMO model under varying levels of 3D acquisition noise. We intentionally introduced noise into our 3D morphing simulations while following the intended 3D surface design (ground truth), allowing us to gauge the model's performance degradation with increasing noise levels. This validation ensures the adaptability of our approach, even with less precise sensing equipment.

For the finite element simulations, we utilized the steady-state thermal and static structural analysis modules within ANSYS. The gripper design (Fig. 6) comprises six fingers anchored to a base for connection to a holding fixture, incorporating TPU (bottom layer in Fig. 6) and solid PLA (top layer in Fig. 6), inspired by prior soft gripper research [32–34]. Two different designs are utilized to show the potential of 4DYNAMO to model diverse designs and utilize it as a surrogate model for inverse design. Design A consists of solid TPU material and PLA material with rectangular grooves to provide bending forces. Design B includes trapezoidal PLA grooves. The interactions between TPU and PLA design are the most important factors for the bending behavior of those structures [2]. Hence, these designs represent important design considerations.

The behavior of 4D printed objects is influenced by various factors, including 3D printing process conditions and stimuli activation. While heat stimuli activation can be directly modeled in FEA software, the impact of 3D printing process conditions necessitates

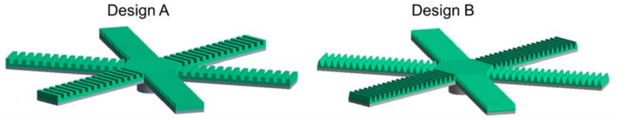

**Fig. 6** Design of the soft gripper used for the simulation studies

**Table 1 Ranges of the process parameters for the simulation studies**

| Process parameter | Material | Range | Literature sources |
|---|---|---|---|
| Density | PLA | 1.20 – 1.25 g/cm$^3$ | [35,36] |
| Coefficient of thermal expansion | PLA | $6.32 \times 10^{-5}$–$2.53 \times 10^{-4}$ K$^{-1}$ | [36] |
| Tensile yield strength | PLA | 27.87–34.63 MPa | [35] |
| Tensile ultimate strength | PLA | 28.41–35.86 MPa | [35] |
| Compressive yield strength | PLA | 26.05–59.94 MPa | [37] |
| Compressive ultimate strength | PLA | 17.17 MPa | [38] |
| Density | TPU | 1.20–1.23 g/cm$^3$ | [39] |
| Coefficient of thermal expansion | TPU | $1.63 \times 10^{-4}$–$2.08 \times 10^{-4}$ K$^{-1}$ | [36] |
| Tensile yield strength | TPU | 21–36 MPa | [40] |
| Tensile ultimate strength | TPU | 20–40 MPa | [41] |
| Compressive yield strength | TPU | 5–10 MPa | [42] |
| Compressive ultimate strength | TPU | 5–10 MPa | [42] |

adjustments in material properties. Thus, a literature review was conducted to find the ranges for essential material properties for PLA and TPU under varying printing conditions. The results are summarized in Table 1.

A space-filling Latin hypercube design of experiments (DOE) with $N = 200$ simulations was conducted based on the identified process parameter ranges. Figure 7 shows four 3D morphing steps in two exemplary simulations.

In each simulation run, we extracted standard triangle language (STL) files representing the initial shape, five intermediate shape transformations during stimuli activation, and the final shape from ANSYS. These files were then converted into 3D point clouds using PYTHON. To assess our framework's robustness under real-world acquisition conditions, we tested various levels of random noise. Figure 8 illustrates this process, showcasing an exemplary STL file exported from ANSYS, its conversion to a 3D point cloud, and the addition of three Gaussian noise levels. The noise levels were selected based on commonly used values reported in previous literature [43–45].

The Gaussian random noise is centered at each coordinate point extracted from the STL model, with variance along each coordinate direction set as a percentage of the range in that direction. We utilize two common point cloud metrics [46] to assess prediction quality: first, the Chamfer (CH) distance, which calculates the distance between point clouds $X_1$ and the predicted point cloud $X_2$ as follows:

$$d_{\mathrm{CH}}(X_1, X_2) = \max\left\{\frac{1}{|X_1|}\sum_{x\in X_1}\min_{y\in X_2}\|x-y\|_2,\ \frac{1}{|X_2|}\sum_{y\in X_2}\min_{x\in X_1}\|y-x\|_2\right\} \tag{18}$$

Second, the Earth mover distance (EMD) or Wasserstein distance is utilized, which is defined as follows:

$$d_{\mathrm{EMD}}(X_1, X_2) = \min\left\{\frac{1}{|X_1|}\sum_{x\in X_1}\min_{y\in X_2}\|x-y\|_2,\ \frac{1}{|X_2|}\sum_{y\in X_2}\min_{x\in X_1}\|y-x\|_2\right\} \tag{19}$$

As a benchmark, we utilize an approach applicable to a 2D abstraction, specifically the 2D bending angle. To the best of our knowledge, no existing methods specifically address the process optimization of the entire 3D shape morphing behavior in 4D printing, measured by a 3D point cloud. To evaluate our process optimization performance, we have chosen a common ad-hoc approach for the 2D bending angle. The target design is assumed to have the same bending angle for each finger of the gripper. We first learn a linear regression model with material conditions ($x_{\text{print}}$) and stimuli activation temperature ($x_{\text{stimuli}}$) as inputs, and the 2D bending angle ($y_b$) as the regression response. The varying material conditions serve as a surrogate for the impact of different printing conditions. This gives us a model of the form $\widehat{y_b} = f(x_{\text{print}}, x_{\text{stimuli}})$.

We then formulate the following optimization problem, which can be readily solved using convex solvers due to the linearity of the function. This problem aims to minimize the difference between the predicted bending angle $\widehat{y_b}$ and the desired target bending angle $y_b^T$, while staying within the upper and lower setting values of the process parameters ($\varepsilon^L_{\text{print}}$, $\varepsilon^L_{\text{stimuli}}$, $\varepsilon^U_{\text{print}}$, $\varepsilon^U_{\text{stimuli}}$), respectively.

$$\begin{aligned}&\min_{x_{\text{print}},x_{\text{stimuli}}} \|y_b^T - f(x_{\text{print}}, x_{\text{stimuli}})\|_2^2\\ &\text{s.t. } \varepsilon^L_{\text{print}} \le x_{\text{print}} \le \varepsilon^U_{\text{print}};\ \varepsilon^L_{\text{stimuli}} \le x_{\text{stimuli}} \le \varepsilon^U_{\text{stimuli}}\end{aligned} \tag{20}$$

For the forward prediction of our model, Table 2 displays the mean and standard deviation on the test set of the 4DYNAMO model across both designs and different material conditions according to the DOE derived from Table 1, utilizing tenfold cross-validation.

The results indicate a strong agreement between our predictions and the simulations, even when exposed to higher levels of simulated random noise in the 3D point cloud representation, which mimics low-fidelity 3D scanning acquisition. However, an increase in error and variation of the predictions is evident, as expected with higher noise levels. From the ablation study, we can clearly observe the significant positive impact of adding more supervision signals in the form of intermediate shape morphing scans during the training stage of the model. Therefore, if the experimental efforts are justifiable, it is recommended to 3D scan more intermediate shapes

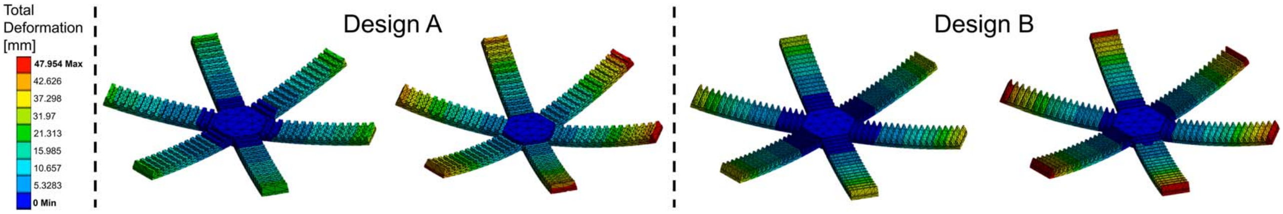

**Fig. 7 Exemplary simulation results**

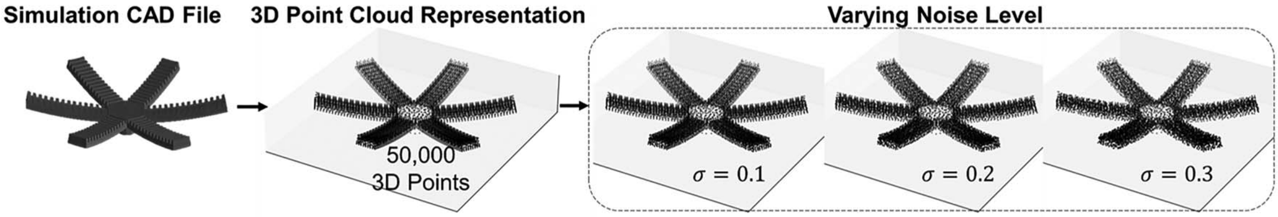

**Fig. 8 Conversion of simulation STL file to 3D point cloud and noise addition to 3D point cloud**

**Table 2 4DYNAMO forward predictions results**

| Model setting | Metric | Noise level | Average | Standard deviation |
|---|---|---|---|---|
| 50 samples and 0 intermediate shapes | CH | 0.1 | 1.868 | 0.441 |
| 50 samples and 0 intermediate shapes | EMD | 0.1 | 0.314 | 0.094 |
| 50 samples and 2 intermediate shapes | CH | 0.1 | 1.687 | 0.349 |
| 50 samples and 2 intermediate shapes | EMD | 0.1 | 0.248 | 0.057 |
| 50 samples and 5 intermediate shapes | CH | 0.1 | 1.147 | 0.224 |
| 50 samples and 5 intermediate shapes | EMD | 0.1 | 0.194 | 0.035 |
| 100 samples and 0 intermediate shapes | CH | 0.1 | 1.341 | 0.342 |
| 100 samples and 0 intermediate shapes | EMD | 0.1 | 0.275 | 0.070 |
| 100 samples and 2 intermediate shapes | CH | 0.1 | 1.193 | 0.373 |
| 100 samples and 2 intermediate shapes | EMD | 0.1 | 0.249 | 0.063 |
| 100 samples and 5 intermediate shapes | CH | 0.1 | 1.095 | 0.132 |
| 100 samples and 5 intermediate shapes | EMD | 0.1 | 0.219 | 0.037 |
| 200 samples and 0 intermediate shapes | CH | 0.1 | 0.937 | 0.254 |
| 200 samples and 0 intermediate shapes | EMD | 0.1 | 0.191 | 0.035 |
| 200 samples and 2 intermediate shapes | CH | 0.1 | 0.892 | 0.128 |
| 200 samples and 2 intermediate shapes | EMD | 0.1 | 0.186 | 0.034 |
| 200 samples and 5 intermediate shapes | CH | 0.1 | 0.779 | 0.105 |
| 200 samples and 5 intermediate shapes | EMD | 0.1 | 0.173 | 0.016 |
| 200 samples and 5 intermediate shapes | CH | 0.2 | 1.054 | 0.138 |
| 200 samples and 5 intermediate shapes | EMD | 0.2 | 0.221 | 0.027 |
| 200 samples and 5 intermediate shapes | CH | 0.3 | 1.256 | 0.241 |
| 200 samples and 5 intermediate shapes | EMD | 0.3 | 0.262 | 0.049 |

rather than fewer. However, future research may be needed to determine when this improvement may saturate. As the sample size increases, the effect of additional supervision signals becomes less significant, indicating a tradeoff between increasing the sample size and adding more intermediate shapes. This tradeoff can be managed based on the experimental efforts required for collecting more samples versus scanning more intermediate shapes. For the sample size, we observe a very clear benefit when increasing the sample size from 50 to 100. Even when increasing the sample size to 200, we still see significant performance gains, and further research could investigate the effect of even larger sample sizes. As a general rule, we recommend assessing the qualitative prediction performance: if the model accurately captures the dynamic 3D shape morphing behavior, it should be sufficient to enable effective process optimization. However, the prediction accuracy directly impacts the optimization results, as the process optimization relies on the inverse flow model. Therefore, higher prediction accuracy leads to better process optimization results. However, it is challenging to provide general guidelines for the number of samples and intermediate shape transformations, as it is highly problem dependent. For qualitative comparison, we present two exemplary visualizations of 4DYNAMO forward predictions in Fig. 9, demonstrating significant overlap with the ground truth simulation shape.

Utilizing 4DYNAMO for process optimization consisted of defining 20 distinct target shapes, which are the desired outcome of the 3D morphing process. Then the inverse flows of 4DYNAMO are utilized to determine the optimal stimuli activation temperature and corresponding material properties. To assess the accuracy of the predictions, we directly compared the computer-aided design (CAD) file representing the target shape with the simulation result. The FEA simulation result is obtained using the optimized process parameters derived from our 4DYNAMO inverse model. As a comparison metric, we use the Euclidean distance, as it is commonly used to compute distances between two CAD files. Table 3 presents the average and standard deviation values between the target shape and the final morphing shape

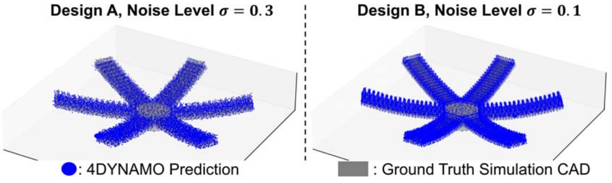
Design A, Noise Level $\sigma = 0.3$
Design B, Noise Level $\sigma = 0.1$
●: 4DYNAMO Prediction
▨ : Ground Truth Simulation CAD

**Fig. 9 Exemplary visualization of 4DYNAMO forward prediction results**

**Table 3 Process optimization results for 20 target shapes**

| Model | Metric | Average | Standard deviation |
|---|---|---|---|
| Benchmark | $L_2$ Euclidian distance | 0.934 | 0.143 |
| **4DYNAMO (Ours)** | $L_2$ Euclidian distance | **0.412** | **0.088** |

Note: Bold indicates best performing model.

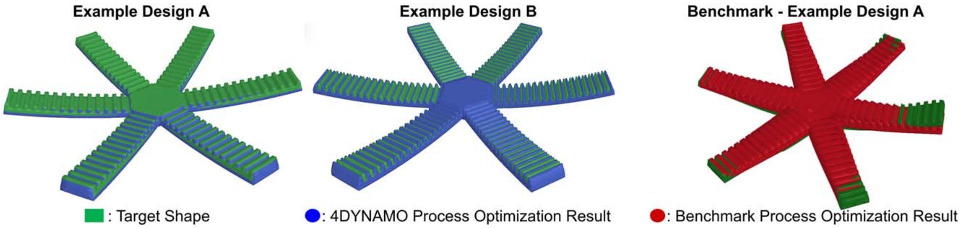
Example Design A
Example Design B
Benchmark - Example Design A
🟩: Target Shape
🔵: 4DYNAMO Process Optimization Result
🔴: Benchmark Process Optimization Result

**Fig. 10 Exemplary visualization of process optimization results for 4DYNAMO model and benchmark comparison**

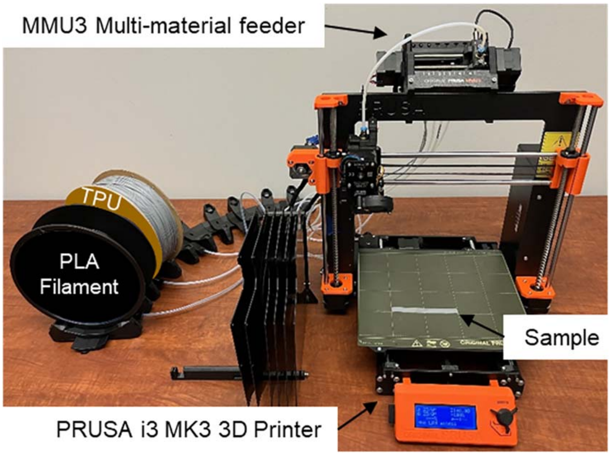

**Fig. 11 Experimental setup**

achieved with the optimal process settings identified through our 4DYNAMO inverse model. We also provide the results for the benchmark approach described earlier in this section.

The benchmark method does not utilize the full 3D information, including the final morphed shape and intermediate shapes. Consequently, the predictions are less accurate, and the optimization is performed in a separate two-stage process. This results in a significantly larger deviation and variance from the intended CAD design. On the other hand, for the 4DYNAMO results there is a substantial overlap between the target and the achieved final morphing shape. Figure 10 presents a qualitative comparison of two exemplary target shapes alongside their corresponding 4DYNAMO-optimized final 3D morphing results as well as an example of the 2D bending angle benchmark result for an example design.

This comprehensive simulation study confirms the effectiveness of our methodology for two advanced designs, spanning various process conditions and noise levels. The benchmark, which only models the 2D bending behavior, fails to accurately capture the 3D shape morphing dynamics. Consequently, it cannot perform effective process optimization, resulting in high variance and a less accurate match to the desired final morphed shape. These results emphasize the significant potential of the 4DYNAMO model. Not only can it serve as a data-driven model rooted in real experiments, but it also is suitable as an efficient surrogate model for computationally intensive FEA simulations. This has the potential to streamline the design optimization process for 4D printed devices.

## 5 Retrofit Case Study—Shape Memory Polymers

To investigate the 4DYNAMO framework outlined above, we conducted real-world case study experiments utilizing FFF to produce 4D printed devices made from PLA and TPU filaments. These materials are commonly employed in 4D printing for self-folding materials via immersion in a hot water bath [7]. Rectangular shapes were printed using a multi-material switching upgrade kit (MMU3) on a PRUSA i3 MK3 Printer to ensure smooth transition between the two materials and ensure layer adherence. The experimental setup is shown in Fig. 11 with one printed layer of TPU filament for a specific sample.

These shapes represent a single finger of the soft robotics gripper from the simulation study, excluding the groove design. We varied the process parameters crucial for 3D shaping morphing behavior, as identified in the literature review in Sec. 2.1. The ranges for these parameters are provided in Table 4. For the experiments, we utilized a space-filling Latin hypercube design. The parts have the same number of PLA and TPU layers, so the total number of layers is $2 \cdot u_{0,4}$.

Here we model $T = 5$ shape transformation steps to achieve the final shape. The 4D shape morphing process was executed in five-second intervals (repeated five times) within a scientific water bath offering precise temperature regulation. After each five-second increment, the parts were subjected to 3D scanning using a high-precision laser scanner, capturing unstructured, 3D point clouds, which were subsequently subsampled to a fixed size of $N_p = 50, 000$ measurement points. A total of 200 samples were collected. In our qualitative analysis, we find that the width of the component has a negligible impact on morphing behavior. However, an increased actuator length, water temperature, and printing speed are positively correlated with increased bending of the parts. Conversely, an increase in layer height, the number of PLA layers, and nozzle temperature tends to reduce bending. Additionally, increasing the number of layers, layer height, and nozzle temperature tends to reduce bending behavior. Conversely,

**Table 4 Process parameter settings for case study experiments**

| Process parameter | Settings | Value used for process optimization |
|---|---|---|
| Length of the part $u_{0,1}$ | 20, 50, 75, 110 mm | Discrete setting value |
| Width of the part $u_{0,2}$ | 5, 15, 30, 50 mm | Discrete setting value |
| Layer height $u_{0,3}$ | 0.12, 0.20, 0.28, 0.36 mm | Discrete setting value |
| Number of material layers $u_{0,4}$ | 2, 3, 4, 5 | Discrete setting value |
| Printing speed $u_{0,5}$ | 20, 40, 60, 80 mm/s | Discrete setting value |
| Printing nozzle temperature $u_{0,6}$ | 200, 210, 220, 230 °C | Discrete setting value |
| Activation temperature (hot water) $u_T$, $T = 1, \ldots, 5$ | 65, 70, 75, 80 °C | Five setting values—one for each 5 s increment |

**Table 5 Impact of process parameters on 2D bending angle and comparison to literature**

| Model parameter | Estimate | Standard error | Literature validation |
|---|---|---|---|
| Intercept | −197.888 | 127.301 | — |
| Length of the part $u_{0,1}$ | 4.741 | 0.296 | — |
| Width of the part $u_{0,2}$ | 2.259 | 0.775 | — |
| Layer height $u_{0,3}$ | 42.899 | 1.730 | Small positive effect [47] |
| Number of material layers $u_{0,4}$ | 0.854 | 0.189 | Small positive effect [47] |
| Printing speed $u_{0,5}$ | 370.875 | 36.887 | Positive effect [47] |
| Printing nozzle temperature $u_{0,6}$ | −4.040 | 0.497 | Negative effect [3] |
| Activation temperature (hot water) $u_T$ | 8.222 | 0.725 | Positive effect [3] |

elevating the activation temperature and print speed tends to enhance bending behavior.

To validate the experimental behavior of our procedure, we utilize a 2D abstraction: the 2D bending angle, a parameter frequently discussed in the literature concerning the impact of process conditions on 4D printed shape memory polymers. We conducted a linear regression on the 200 space-filling factorial design experiments to determine the impacts of process parameters as well as design parameters on the 2D bending behavior, as reported in Table 5. When applicable, we reference related literature that validates this behavior.

For the process parameters, we observe that the activation temperature has the most significant positive impact on the bending angle. In contrast, the printing speed exerts the most substantial negative effect, resulting in a decreased bending angle, followed by the printing temperature. For the design parameters, we find that increasing the material thickness significantly reduces the ability to achieve a large bending behavior. To further validate this behavior, we conduct repeatability experiments, varying one factor at a time, and print another 100 samples. For this preliminary analysis, we only measure the 2D bending angle to reduce data collection efforts, rather than measuring the 3D point clouds. The results, depicting the impact of different process conditions on the bending angle, are presented in Fig. 12. These results demonstrate that the effects of varying certain process conditions are repeatable, highlighting the importance of the process conditions for the final 3D shape accuracy. However, we also observe that only controlling the bending angle is not enough as the interaction of the process conditions, design choices, and environmental uncertainties may lead to 3D twisting and bending behavior that is not captured solely by the 2D bending angle.

The 200 samples, for which we conducted the Latin hypercube DOE and collected 3D measurements, form the basis for training our 4DYNAMO model. In Fig. 13, we present different intermediate shape morphing results of two exemplary samples to illustrate

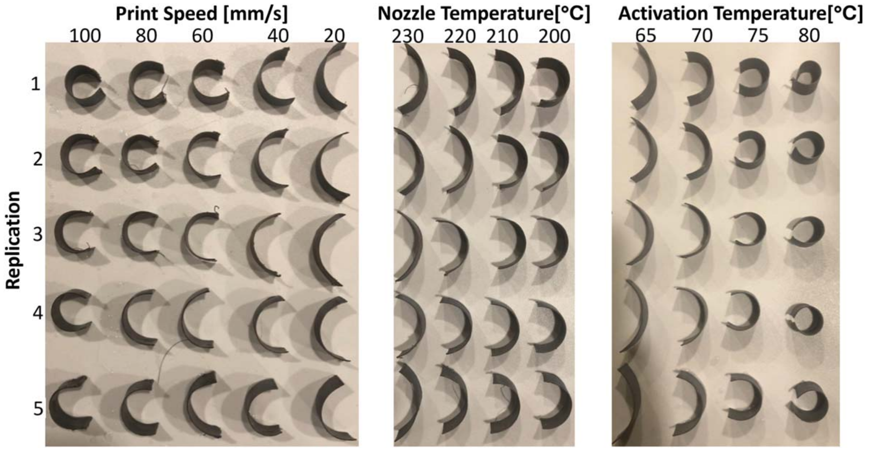

**Fig. 12 Impact of process conditions on 2D bending angle and process repeatability**

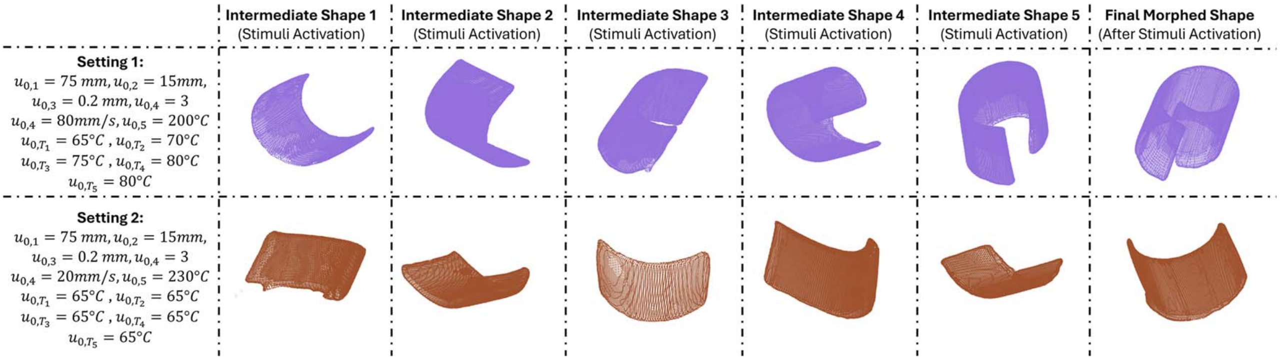
Intermediate Shape 1 (Stimuli Activation)
Intermediate Shape 2 (Stimuli Activation)
Intermediate Shape 3 (Stimuli Activation)
Intermediate Shape 4 (Stimuli Activation)
Intermediate Shape 5 (Stimuli Activation)
Final Morphed Shape (After Stimuli Activation)
Setting 1:
$u_{0,1} = 75\ mm, u_{0,2} = 15mm,$
$u_{0,3} = 0.2\ mm, u_{0,4} = 3$
$u_{0,4} = 80mm/s, u_{0,5} = 200°C$
$u_{0,T_1} = 65°C\ , u_{0,T_2} = 70°C$
$u_{0,T_3} = 75°C\ , u_{0,T_4} = 80°C$
$u_{0,T_5} = 80°C$
Setting 2:
$u_{0,1} = 75\ mm, u_{0,2} = 15mm,$
$u_{0,3} = 0.2\ mm, u_{0,4} = 3$
$u_{0,4} = 20mm/s, u_{0,5} = 230°C$
$u_{0,T_1} = 65°C\ , u_{0,T_2} = 65°C$
$u_{0,T_3} = 65°C\ , u_{0,T_4} = 65°C$
$u_{0,T_5} = 65°C$

**Fig. 13 Exemplary 3D shape morphing behavior with intermediate 3D scanned shapes**

**Table 6 4DYNAMO tenfold cross-validation testing results for different model settings**

| Model setting | Metric | Average | Standard deviation |
|---|---|---|---|
| 50 samples and 0 intermediate shapes | CD | 1.965 | 0.58 |
| 50 samples and 0 intermediate shapes | EMD | 1.617 | 0.43 |
| 50 samples and 2 intermediate shapes | CD | 1.744 | 0.45 |
| 50 samples and 2 intermediate shapes | EMD | 0.147 | 0.03 |
| 50 samples and 5 intermediate shapes | CD | 1.566 | 0.30 |
| 50 samples and 5 intermediate shapes | EMD | 0.122 | 0.02 |
| 100 samples and 0 intermediate shapes | CD | 1.682 | 0.36 |
| 100 samples and 0 intermediate shapes | EMD | 0.146 | 0.03 |
| 100 samples and 2 intermediate shapes | CD | 1.378 | 0.25 |
| 100 samples and 2 intermediate shapes | EMD | 0.128 | 0.02 |
| 100 samples and 5 intermediate shapes | CD | 0.938 | 0.13 |
| 100 samples and 5 intermediate shapes | EMD | 0.079 | 0.01 |
| 200 samples and 0 intermediate shapes | CD | 1.086 | 0.19 |
| 200 samples and 0 intermediate shapes | EMD | 0.084 | 0.01 |
| 200 samples and 2 intermediate shapes | CD | 0.919 | 0.12 |
| 200 samples and 2 intermediate shapes | EMD | 0.075 | 0.01 |
| 200 samples and 5 intermediate shapes | CD | 0.896 | 0.08 |
| 200 samples and 5 intermediate shapes | EMD | 0.0706 | 0.01 |

the supervision signal the 4DYNAMO model utilizes for training and the dynamics of the process.

We train and test the 4DYNAMO model using a tenfold CV procedure. The evaluation of prediction accuracy involves computing CD and EMD as reported in Table 6, which demonstrates the strong predictive performance of the 4DYNAMO predictions on the real experimental samples.

From the ablation study, we observe that incorporating additional supervision signals in the form of intermediate shapes significantly enhances prediction performance. However, this effect diminishes as the sample size increases, highlighting a tradeoff between collecting more samples and generating more intermediate shape activations. This tradeoff can be exploited based on the experimental efforts required. Figure 14 presents a qualitative assessment showcasing the strong alignment between three exemplary predictions and the ground truth 3D profiles. The green ground truth measurements closely match the blue 4DYNAMO predictions.

Following this, we used the best model, which was trained with the full sample size and all five intermediate shapes as supervision signals, to optimize the process for ten different target shapes. These target shapes are characterized by bending a rectangular shape without any twisting. The target shapes represent different desired final morphed forms. Figure 14 provides three examples of these target shapes in green. We note that the process inputs are normalized before being used in the model, as they have different units and magnitudes. In Table 7, we report the Euclidean distance between the ground truth CAD design and the 3D scanned parts after printing and activating them with the optimized process settings prescribed by the inverse 4DYNAMO model. We observe a strong agreement between the 4D printed devices with the target design. However, the variance reflects the challenging alignment process between an actual printed part, which may exhibit some level of twisting, and the CAD model. The performance of the benchmark, which can only control the 2D bending angle and therefore misses some important dynamics of the shape morphing process, results in a significantly larger deviation from the CAD model and a higher variance. These inaccuracies in predictions propagate through the process optimization, exacerbating the overall deviation.

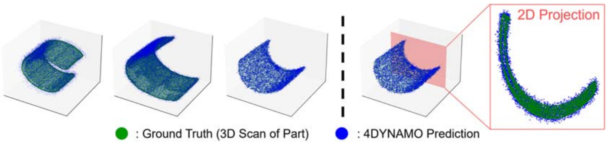

**Fig. 14 Qualitative prediction results for three exemplary test shapes**

**Table 7 4DYNAMO process optimization results for 10 target shapes**

| Model | Metric | Average | Standard deviation |
|---|---|---|---|
| Benchmark | $L_2$ Euclidian distance | 1.924 | 0.632 |
| **4DYNAMO (ours)** | $L_2$ Euclidian distance | **1.119** | **0.456** |

Note: Bold indicates best performing model.

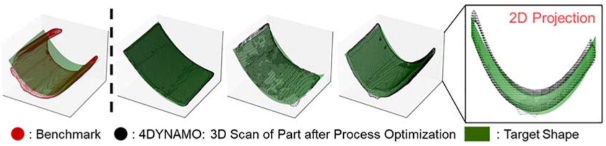

**Fig. 15 Qualitative process optimization results for benchmark and 4DYNAMO approach**

Figure 15 presents qualitative results from three exemplary optimization outcomes, underscoring the remarkable precision and potential of 4DYNAMO compared to the benchmark in controlling the dynamic 3D shape morphing behavior. By adjusting the control variables during both 3D printing and 4D shape morphing (i.e., $u_{0,1}, .., u_{0,6}, u_2, \ldots, u_{10}$), we achieve an impressive alignment with the target shape, while the benchmark misses some of the complex, nonlinear dynamics of the process since it uses a linear model to predict a 2D abstraction, specifically the bending angle.

This extensive case study validates our proposed 4DYNAMO framework and demonstrates its immense potential as a tool for optimizing the dynamic 3D shape accuracy of 4D printed devices, considering the process conditions during both 3D printing and 4D stimuli activation.

## 6 Conclusion

In this paper, we introduced a novel machine-learning framework tailored for precise control of 3D shape morphing in 4D printed devices. Through extensive simulation studies and real-world experiments, we showcased the framework's significant practical potential. Moreover, our approach offers interpretability regarding intermediate transformations and enables precise control over these steps, including associated process conditions. We foresee exciting opportunities for diverse manufacturing processes and the development of impactful new products.

Future research directions should explore more intricate shapes and incorporate uncertainty quantification while considering additional design and engineering constraints. For example, those encountered in soft robotics gripping of deformable objects necessitate specific speed and force distributions at each dynamic 3D morphing step. Furthermore, exploring transfer learning methodologies could be crucial for quickly extending the method to new designs. Additionally, creating a diverse training set of shapes would enhance the method's generalization capabilities.

Another crucial future direction could involve extending the proposed framework from process optimization to real-time dynamic control, compensating for process variability encountered during the 3D printing process and during the stimuli activation phase. Currently, cycle time is not a concern for offline process optimization, but it becomes critical for online control.

Furthermore, the calibration of simulation studies to match real 4D printed parts requires an in-depth examination of material properties under various printing conditions. Presently, material properties are sourced from multiple studies, potentially leading to inconsistencies, and parts are assumed to possess solid material, overlooking the line-by-line, layer-by-layer deposition characteristic of 3D printing techniques like fused filament fabrication.

An important area for future research is the uncertainty quantification of the proposed framework. Our method naturally enables uncertainty quantification through the probabilistic representation of the 3D point clouds. However, assessing and reducing uncertainty involves multiple factors such as controlling the 3D printing process and stimuli activation, as well as improving the accuracy of sensing devices. The interactions among these factors and their underlying causes warrant detailed investigation. For instance, uncertainties may originate from measurement inaccuracies alone, or process uncertainties might be compounded by measurement errors. Another important factor is the uncertainty about the input materials to the process. Future research should investigate how this could be mitigated by enhancing the consistency of the raw materials, quantifying the range of material properties' uncertainty and their interactions with 3D printing, and determining to what extent 3D printing and stimuli activation can compensate for material uncertainty.

Despite these areas of improvement, we view this work as an essential initial step in demonstrating the influence of process settings on dynamic 3D shape morphing and the potential of advanced machine-learning models to optimize these processes toward a desired target shape with high 3D accuracy.

## Funding Data

- National Science Foundation Award ID 2019378.

## 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.

## Nomenclature

| | | |
|---|---|---|
| $i$ | = | sample index |
| $j$ | = | temporal index (i.e., 3D morphing steps) |
| $\mathcal{D}$ | = | dataset |
| $H$ | = | entropy |
| $N$ | = | total number of samples |
| $\mathcal{S}$ | = | domain of flow model |
| $T$ | = | number of 3D morphing steps |
| $e_{\theta}$ | = | encoder for conditional input $y$ |
| $f_{\theta_i}$ | = | forward temporal flow function parameterized by $\theta_i$ |
| $z_{i,j}$ | = | latent variable |
| $x_T$ | = | low-dimensional embedding of the 3D point cloud measurement at shape morphing step $T$ |
| $x_{i,j}$ | = | low-dimensional embedding of the 3D point cloud of sample $i$ in shape morphing step $j$ |
| $x_{i,0}$ | = | low-dimensional embedding of the prior distribution drawn from prior distribution $P(x_{i,0})$ |
| $u_0$ | = | process inputs during 3D printing |
| $u_0^*$ | = | optimal process inputs during 3D printing to achieve final target shape |
| $u_1$ | = | process inputs during 4D stimuli activation |
| $u_1^*$ | = | optimal process inputs during 4D stimuli activation to achieve final target shape |
| $u_{i,j}$ | = | process input of sample $i$ in shape morphing step $j$ |
| $y_{i,T}$ | = | concatenation of preceding 3D shapes and process inputs up to shape morphing step $T$ for sample $i$ |
| $z_0$ | = | initial embedding of the normalizing flow model, i.e., initialized with random Gaussian distribution |
| $F_{\theta_i}$ | = | discrete time temporal flow function parameterized by $\theta_i$ |
| $N_p$ | = | number of 3D measurement points in the 3D point clouds |
| $P_{\psi_j}$ | = | prior distribution at shape morphing step $j$ |
| $P_{\eta_j}$ | = | decoder of the 3D Point cloud at shape morphing step $j$ |
| $Q_{\phi_j}$ | = | encoder of the 3D Point cloud at shape morphing step $j$ |
| $D_{KL}$ | = | Kullback–Leibler divergence |
| $X_0$ | = | part after 3D printing, before stimuli activation |
| $X_{i,j}$ | = | 3D point cloud of sample $i$ in shape morphing step $j$ |
| $g(\cdot)$ | = | affine transformation function |
| $\mathcal{L}_{\mathrm{4DYNAMO}}$ | = | 4DYNAMO loss function |
| $\vartheta_i$ | = | parameter of the flow model |
| AM | = | additive manufacturing |
| CAD | = | computer-aided design |
| CH | = | chamfer distance |
| CNF | = | continuous normalizing flow |
| DOE | = | design of experiments |
| ELBO | = | evidence lower bound |
| EMD | = | Earth mover distance |
| FEA | = | finite element analysis |
| FFF | = | fused filament fabrication |
| KL | = | Kullback–Leibler divergence |
| NF | = | normalizing flow |
| NN | = | neural network |
| PLA | = | polylactic acid |
| ReLU | = | rectified linear unit |
| SGD | = | stochastic gradient descent |
| SMP | = | shape memory polymer |
| STL | = | standard triangle language |
| TPU | = | thermoplastic polyurethane |
| VAE | = | variational autoencoder |
| 4DYNAMO | = | analyzing and optimizing process parameters in 4D printing for dynamic 3D shape morphing Accuracy |

## References

[1] Rezende, D., and Mohamed, S., 2015, "Variational Inference With Normalizing Flows," International Conference on Machine Learning, Lille, France, July 6–11, PMLR, pp. 1530–1538.

[2] An, B., Tao, Y., Gu, J., Cheng, T., Chen, X. A., Zhang, X., Zhao, W., et al., 2018, "Thermorph: Democratizing 4D Printing of Self-Folding Materials and

Interfaces," Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Montreal, Canada, Apr. 21–26, pp. 1–12.

[3] Barletta, M., Gisario, A., and Mehrpouya, M., 2021, "4D Printing of Shape Memory Polylactic Acid (PLA) Components: Investigating the Role of the Operational Parameters in Fused Deposition Modelling (FDM)," J. Manuf. Processes, **61**, pp. 473–480.

[4] Bodaghi, M., Noroozi, R., Zolfagharian, A., Fotouhi, M., and Norouzi, S., 2019, "4D Printing Self-Morphing Structures," Materials, **12**(8), p. 1353.

[5] Tezerjani, S. M. D., Yazdi, M. S., and Hosseinzadeh, M. H., 2022, "The Effect of 3D Printing Parameters on the Shape Memory Properties of 4D Printed Polylactic Acid Circular Disks: An Experimental Investigation and Parameters Optimization," Mater. Today Commun., **33**, p. 104262.

[6] Akbar, I., El Hadrouz, M., El Mansori, M., and Tarfaoui, M., 2023, "Thermomechanical Shape Memory Testing of 4D Printed Novel Material Rhombus-Shape Structure," Appl. Mater. Today, **33**, p. 101876.

[7] Ren, L., Li, B., Liu, Q., Ren, L., Song, Z., Zhou, X., and Gao, P., 2021, "4D Printing Dual Stimuli-Responsive Bilayer Structure Toward Multiple Shape-Shifting," Front. Mater., **8**, p. 655160.

[8] Hosseinzadeh, M., Ghoreishi, M., and Narooei, K., 2023, "4D Printing of Shape Memory Polylactic Acid Beams: An Experimental Investigation Into FDM Additive Manufacturing Process Parameters, Mathematical Modeling, and Optimization," J. Manuf. Processes, **85**, pp. 774–782.

[9] Nam, S., and Pei, E., 2020, "The Influence of Shape Changing Behaviors From 4D Printing Through Material Extrusion Print Patterns and Infill Densities," Materials, **13**(17), p. 3754.

[10] Sun, X., Yue, L., Yu, L., Shao, H., Peng, X., Zhou, K., Demoly, F., et al., 2022, "Machine Learning-Evolutionary Algorithm Enabled Design for 4D-Printed Active Composite Structures," Adv. Funct. Mater., **32**(10), p. 2109805.

[11] Huang, Q., Wang, Y., Lyu, M., and Lin, W., 2020, "Shape Deviation Generator – A Convolution Framework for Learning and Predicting 3-D Printing Shape Accuracy," IEEE Trans. Autom. Sci. Eng., **17**(3), pp. 1486–1500.

[12] Abdelhamed, A., Brubaker, M. A., and Brown, M. S., 2019, "Noise Flow: Noise Modeling With Conditional Normalizing Flows," Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, South Korea, Oct. 27–Nov. 2, pp. 3165–3173.

[13] Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B., 2021, "Normalizing Flows for Probabilistic Modeling and Inference," J. Mach. Learn. Res., **22**(57), pp. 1–64.

[14] Yang, G., Huang, X., Hao, Z., Liu, M.-Y., Belongie, S., and Hariharan, B., 2019, "Pointflow: 3D Point Cloud Generation With Continuous Normalizing Flows," Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, South Korea, Oct. 27–Nov. 2, pp. 4541–4550.

[15] Biehler, M., Lin, D., and Shi, J., 2023, "DETONATE: Nonlinear Dynamic Evolution Modeling of Time-Dependent 3-Dimensional Point Cloud Profiles," IISE Trans., **56**(5), pp. 541–558.

[16] Biehler, M., Li, J., and Shi, J., 2024, "FUSION3D: Multimodal Data Fusion for 3D Shape Reconstruction – A Soft Sensing Approach," IISE Trans.

[17] Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K., 2018, "Neural Ordinary Differential Equations," Advances in Neural Information Processing Systems, Montreal, Canada, Dec. 3–8.

[18] Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., and Duvenaud, D., 2019, "FFJORD: Free-Form Continuous Dynamics for Scalable Reversible Generative Models," International Conference on Learning Representations, New Orleans, LA, May 6–9.

[19] Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M., 2023, "Flow Matching for Generative Modeling," International Conference on Learning Representations, Kigali, Rwanda, May 1–5.

[20] Biehler, M., Mock, R., Kode, S., Mehmood, M., Bhardwaj, P., and Shi, J., 2024, "AUDIT: Functional Qualification in Additive Manufacturing Via Physical and Digital Twins," ASME J. Manuf. Sci. Eng., **146**(2), p. 021001.

[21] Biehler, M., and Shi, J., 2024, "RETROFIT: Real-Time Control of Time-Dependent 3D Point Cloud Profiles," ASME J. Manuf. Sci. Eng., **146**(6), p. 061006.

[22] Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O., 2019, "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations," International Conference on Machine Learning, Long Beach, CA, June 9–15, PMLR, pp. 4114–4124.

[23] Lu, C., Chen, J., Li, C., Wang, Q., and Zhu, J., 2021, "Implicit Normalizing Flows," International Conference on Machine Learning, Virtual, May 3–7, PMLR, pp. 1–24.

[24] Lucas, A., Iliadis, M., Molina, R., and Katsaggelos, A. K., 2018, "Using Deep Neural Networks for Inverse Problems in Imaging: Beyond Analytical Methods," IEEE Signal Process. Mag., **35**(1), pp. 20–36.

[25] Behrmann, J., Vicol, P., Wang, K.-C., Grosse, R., and Jacobsen, J.-H., 2021, "Understanding and Mitigating Exploding Inverses in Invertible Neural Networks," International Conference on Artificial Intelligence and Statistics, Virtual Conference, Apr. 13–15, PMLR, pp. 1792–1800.

[26] Hong, S., Park, I., and Chun, S. Y., 2023, "On the Robustness of Normalizing Flows for Inverse Problems in Imaging," Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, Oct. 1–6, pp. 10745–10755.

[27] Kirichenko, P., Izmailov, P., and Wilson, A. G., 2020, "Why Normalizing Flows Fail to Detect Out-of-Distribution Data," Adv. Neural Inf. Process. Syst., **33**, pp. 20578–20589.

[28] Lugmayr, A., Danelljan, M., Van Gool, L., and Timofte, R., 2020, "Srflow: Learning the Super-Resolution Space With Normalizing Flow," Computer Vision–ECCV 2020: 16th European Conference, Proceedings, Part V 16, Glasgow, UK, Aug. 23–28, Springer, pp. 715–732.

[29] Wang, Y., Wan, R., Yang, W., Li, H., Chau, L.-P., and Kot, A., 2022, "Low-Light Image Enhancement With Normalizing Flow," Proc. AAAI Conf. Artif. Intell., **36**(3), pp. 2604–2612.

[30] Athinarayanarao, D., Prod'hon, R., Chamoret, D., Qi, H. J., Bodaghi, M., André, J. C., and Demoly, F., 2023, "Computational Design for 4D Printing of Topology Optimized Multi-material Active Composites," npj Comput. Mater., **9**(1), p. 1.

[31] Yu, Y., Liu, H., Qian, K., Yang, H., McGehee, M., Gu, J., Luo, D., et al., 2020, "Material Characterization and Precise Finite Element Analysis of Fiber Reinforced Thermoplastic Composites for 4D Printing," Comput.-Aided Des., **122**, p. 102817.

[32] Godaba, H., Sajad, A., Patel, N., Althoefer, K., and Zhang, K., 2020, "A Two-Fingered Robot Gripper With Variable Stiffness Flexure Hinges Based on Shape Morphing," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, Oct. 25–29, IEEE, pp. 8716–8721.

[33] Gutierrez, R. B., Garcia, M., McDuffie, J., Long, C., and Tekes, A., 2020, "Development of Wire Actuated Monolithic Soft Gripper Positioned by Robot Manipulator," Dynamic Systems and Control Conference, Vol. 84270, American Society of Mechanical Engineers, p. V001T18A001.

[34] Lee, L. Y., Syadiqeen, O. A., Tan, C. P., and Nurzaman, S. G., 2021, "Closed-Structure Compliant Gripper With Morphologically Optimized Multi-material Fingertips for Aerial Grasping," IEEE Robot. Autom. Lett., **6**(2), pp. 887–894.

[35] Rismalia, M., Hidajat, S., Permana, I., Hadisujoto, B., Muslimin, M., and Triawan, F., 2019, "Infill Pattern and Density Effects on the Tensile Properties of 3D Printed PLA Material," J. Phys. Conf. Ser., **1402**(4), p. 044041.

[36] Bute, I., Tarasovs, S., Vidinejevs, S., Vevere, L., Sevcenko, J., and Aniskevich, A., 2023, "Thermal Properties of 3D Printed Products From the Most Common Polymers," Int. J. Adv. Manuf. Technol., **124**(7), pp. 2739–2753.

[37] Subeshan, B., Alonayni, A., Rahman, M. M., and Asmatulu, E., 2018, "Investigating Compression Strengths of 3D Printed Polymeric Infill Specimens of Various Geometries," Nano-, Bio-, Info-Tech Sensors, and 3D Systems II, Denver, CO, Mar. 5–7, Vol. 10597, SPIE, pp. 89–94.

[38] Tang, C., Liu, J., Yang, Y., Liu, Y., Jiang, S., and Hao, W., 2020, "Effect of Process Parameters on Mechanical Properties of 3D Printed PLA Lattice Structures," Compos., Part C: Open Access, **3**, p. 100076.

[39] Bates, S. R., Farrow, I. R., and Trask, R. S., 2019, "Compressive Behaviour of 3D Printed Thermoplastic Polyurethane Honeycombs With Graded Densities," Mater. Des., **162**, pp. 130–142.

[40] Lee, H., Eom, R.-I., and Lee, Y., 2019, "Evaluation of the Mechanical Properties of Porous Thermoplastic Polyurethane Obtained by 3D Printing for Protective Gear," Adv. Mater. Sci. Eng., **2019**, pp. 1–10.

[41] Arifvianto, B., Iman, T. N., Prayoga, B. T., Dharmastiti, R., Salim, U. A., Mahardika, M., and Suyitno, 2021, "Tensile Properties of the FFF-Processed Thermoplastic Polyurethane (TPU) Elastomer," Int. J. Adv. Manuf. Technol., **117**(5), pp. 1709–1719.

[42] Hasan, H., 2021, "The Effect of Process Parameters on Mechanical Properties of TPU Parts Fabricated by Fused Deposition Modelling," Master's thesis, School of Mechanical and Mining Engineering, The University of Queensland, St Lucia, Australia.

[43] Li, R., Li, X., Heng, P.-A., and Fu, C.-W., 2021, "Point Cloud Upsampling Via Disentangled Refinement," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, June 20–25, pp. 344–353.

[44] Potamias, R. A., Bouritsas, G., and Zafeiriou, S., 2022, "Revisiting Point Cloud Simplification: A Learnable Feature Preserving Approach," European Conference on Computer Vision, Tel Aviv, Israel, Oct. 23–27, Springer, pp. 586–603.

[45] Biehler, M., Yan, H., and Shi, J., 2023, "ANTLER: Bayesian Nonlinear Tensor Learning and Modeler for Unstructured, Varying-Size Point Cloud Data," IEEE Trans. Autom. Sci. Eng., **21**(1), pp. 402–415.

[46] Nguyen, T., Pham, Q.-H., Le, T., Pham, T., Ho, N., and Hua, B.-S., 2021, "Point-Set Distances for Learning Representations of 3D Point Clouds," Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, Canada, Oct. 11–17, pp. 10478–10487.

[47] Kačergis, L., Mitkus, R., and Sinapius, M., 2019, "Influence of Fused Deposition Modeling Process Parameters on the Transformation of 4D Printed Morphing Structures," Smart Mater. Struct., **28**(10), p. 105042.