Summer 2023

Data-driven Porosity Prediction for Directed Energy Deposition

Georgia E. Kaufman

Georgia Kaufman

Candidate

Electrical & Computer Engineering

Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Prof. Meeko Oishi , Chairperson

Prof. Rafael Fierro

Prof. Claus Danielson

Prof. Ganesh Balakrishnan

Data-driven Porosity Prediction for Directed Energy Deposition

by

Georgia Kaufman

B.S., Biology, Humboldt State University, 2015
B.S., Electrical Engineering, University of New Mexico, 2021

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Electrical Engineering

The University of New Mexico

Albuquerque, New Mexico

August, 2023

Dedication

This thesis is dedicated to my grandmother and best friend, Rosemary Wilding. Thank you for being my rock. Your wisdom, kindness, effortless grace, and childlike wonder continue to inspire me.

"I don't think being good at things is the point of doing them. I think you've got all these wonderful experiences with different skills, and that all teaches you things and makes you an interesting person, no matter how well you do them."

– Kurt Vonnegut

Acknowledgments

I would like to thank my advisor, Professor Meeko Oishi, for believing in me when I didn't believe in myself. Her guidance and patience made this thesis possible. I am very grateful to Adam Thorpe for sharing his research, code, and perspectives, and for breaking things down so I could understand them. Thank you to Kendric Ortiz and Rachel DiPirro for their ideas and support.

Thank you to the Sandia National Laboratories Critical Skills Recruiting program for funding my Master's degree and pushing me to pursue a career in research. I would also like to thank Shaun Whetten and Bart Van Bloemen Waanders of Sandia National Labs for their data contributions and suggestions.

I would especially like to thank Harold "Scott" Halliday and his staff and students at Navajo Technical University's Fabrication Laboratory. Scott dropped everything on several occasions to help me collect data and gain a better understanding of the processes involved. The work done by Scott and his team was invaluable, as it is central to this thesis. Thank you also to Anusuya Vellingiri of Navajo Technical University for sharing her data and Matlab code.

Thank you to my friends and family for helping me get through this. You know who you are.

The NASA Additive Manufacturing: Cooperative Agreement #80NNSSC20M0215 provided funds to assist the authors with their research, but this article solely reflects the opinions and conclusions of its authors and not any NASA entity.

Data-driven Porosity Prediction for Directed Energy Deposition

by

Georgia Kaufman

B.S., Biology, Humboldt State University, 2015
B.S., Electrical Engineering, University of New Mexico, 2021
M.S., Electrical Engineering, University of New Mexico, 2023

Abstract

Stochastic flaw formation leading to poor print quality is a major obstacle to the utility of directed energy deposition (DED), a laser and metal powder-based additive manufacturing technology for construction and repair of custom metal parts. While melt pool temperature variability is known to be a major factor in flaw formation, control schemes to decrease flaw formation are limited by a lack of physics-based models that fully and accurately describe DED. In this work, a stochastic reachability analysis with a data-driven model based on thermal images of the melt pool was conducted to determine the likelihood of violating melt pool temperature constraints. As validation, the likelihood maps were compared to the true locations of porosities determined from CT scans of the parts. Additionally, a recurrent neural network (RNN) was trained to predict porosity locations using melt pool thermal images without prior assumptions as to the cause of the porosities.

Contents

List of Figures viii

List of Tables xii

Glossary xiii

1 Introduction 1

1.1 Motivation 1

1.2 Background 2

1.2.1 DED process 2

1.2.2 Learning and control theoretic models of DED 5

1.2.3 Stochastic reachability, kernel methods, and neural networks . 8

1.3 Problem statement 9

1.4 Contributions 9

2 Problem Formulation 10

2.1 Data collection 10

2.2 Modeling description 15

2.3 Data manipulation 17

2.3.1 Establishing inertial reference trajectories 17

2.3.2 Establishing the state space 18

2.3.3 Input trajectories 20

2.3.4 State trajectories 21

2.3.5 CT scans 23

3 Methods **25**

3.1 Stochastic reachability 25

3.2 RNN 28

4 Results and Interpretation **33**

4.1 Reachability 34

4.2 RNN 37

5 Conclusions **40**

References **42**

List of Figures

1.1 Types of DED printing. This work will focus on LENS, a powder- and laser-based form of DED. Adapted from [9] (Open Access). . . . 3

1.2 Illustration of basic LENS printing concept and terms used in this paper. 4

1.3 Representation of (a) DED porosity defect characteristics for lack of fusion and keyholing, and (b) intersection of lack of fusion and keyholing porosity prevalence with respect to energy density. Adapted from [9] (Open Access). 5

2.1 Data collection process. 10

2.2 Sample DED melt pool thermal image. 11

2.3 DED wall printing process. (a) Part design in Mastercam, (b) SS316 powder feedstock, (c) printing walls, and (d) finished parts ready to be excised with wire EDM. 13

2.4 Part layout and orientation. (a) Print layout of 8 printed walls on build plate. The (x,y) origin for the printer is set at the lower left corner of each part before printing. (b) Layout of EDM excised part for CT scanning showing the first layer printing start point and direction. 14

2.5 Screenshot of Volume Graphics Studio visualization and porosity analysis of CT scan for the wall printed at 270 W. Predicted porosity locations are colored based on porosity volume, and large, unlikely porosities are excluded from future analysis. (a) Cross section at z = 0.42 mm, (b) cross section at x = 1.56 mm, (c) cross section at y = 17.67 mm, and (d) 3D view. 15

2.6 Workflow of data processing and analysis for this thesis. Yellow indicates software code modified from existing code written by others, blue indicates entirely custom code. 17

2.7 A selection of layer trajectories generated from Gcode with 30 Hz frame rate. Green circle is the start point, red circle is the end point for each layer. 18

2.8 Representation of discretized state locations as the center points (red) of 0.084 mm square cells. Right is a zoomed-in section of the grid on the left. The state elements are numbered from x_1 on the bottom left to x_{8880} on the top right. 20

2.9 Depiction of a single input trajectory point as a discretization of a laser spot. For each time step, shaded cells are "on" and have the given laser power for the part, the rest of the cells are "off" and have a laser power of zero. 21

2.10 Simplified representation of the procedure for mapping images to state locations. Gold lines are pixel outlines, black are cell outlines. Here the image is represented as 18x18 pixels (3x3 cells), while in actuality the melt pool images are 198x198 pixels (33x33 cells). . . . 23

3.1 An example 3-layer neural network establishing abbreviated notation. 29

3.2 Components of an LSTM cell. In purple are sigmoid (σ) and hyperbolic tangent activation functions (tanh), and in yellow are element-wise operations. 30

3.3 Network architecture overview for (a) LSTM-only, and (b) CNN-LSTM networks used in this analysis. 31

4.1 Porosity likelihood maps determined via stochastic reachability analysis for first printed layer alongside actual porosities from CT scan (far right). Images used as initial conditions for reachability were from the first layer of the 270W part, and CT scan porosities are from the same layer. Lighter colors indicate higher likelihood of reaching an unsafe condition, and therefore higher likelihood of defects, within the specified number of time steps. 35

4.2 Porosity likelihood maps determined via stochastic reachability analysis for fifteenth printed layer alongside actual porosities from CT scan (far right). Images used as initial conditions for reachability were from the fifteenth layer of the 270W part, and CT scan porosities are from the same layer. Red boxes highlight an area of predicted porosities that match with actual porosities. 35

4.3 Depiction of training MSE over training epochs for the LSTM-only and CNN-LSTM algorithms discussed in this work. Sharp valleys in this plot likely correspond to epochs where an outlier layer was left out of the training set. 37

4.4 Predicted porosity likelihoods using both RNNs, tested on part layer 2 of the 340W wall. MSE for this data point went from 0.25 before training to 0.0122 after training using the CNN-LSTM algorithm and 0.0130 after training using the LSTM-only algorithm. 39

4.5 Actual versus CNN-LSTM predicted porosities for layer 8 of the 300W printed wall. 39

List of Tables

2.1 Relevant DED printing parameters. 12

3.1 Neural network parameters. 32

4.1 Scale of NN training. 37

Glossary

AM	Additive manufacturing
CNN	Convolutional neural network
CT	Computed tomography
DED	Directed energy deposition
EDM	Electrical discharge machine
FHT	First hitting time
GED	Global energy density
ILC	Iterative learning control
LENS	Laser engineered net shaping
LMD	Laser metal deposition
LOOCV	Leave-one-out cross-validation
LSTM	Long short-term memory neural network
ML	Machine learning
MSE	Mean squared error

PBF	Powder bed fusion
RKHS	Reproducing kernel Hilbert space
RNN	Recurrent neural network
RPC	Repetitive process control

Chapter 1

Introduction

1.1 Motivation

Directed energy deposition (DED) is an additive manufacturing technology that is used for the creation and repair of small batch, custom metal parts. Like many other additive manufacturing (AM) technologies, DED has the potential to expand design spaces and reduce cost and manufacturing times compared to traditional manufacturing methods for small production runs and prototypes. DED is particularly appealing for aerospace applications because DED parts can be much larger than those made with other metal AM processes. Indeed, the first 3D printed orbital rocket was recently launched [1], made primarily with DED processes. However, DED printed parts have yet to be widely adopted, mainly due to poor and inconsistent print quality.

Problems with DED part quality and consistency are at least partially due to the lack of complete understanding of the underlying process-structure-parameter relationships, contributing to a dearth of control policies [2]. Few physics-based models exist, and those that do are oversimplified or do not take into account all

levels of dynamics: melt pool, track-to-track, and layer-to-layer [3]. In addition, flaw formation in DED printed parts exhibits a high degree of stochasticity [4]. As a result, machine learning techniques are increasingly being used to learn DED models from available data [5], [6]. Successful control of DED printing of complex part geometries will likely only be achieved with improved sensors and a combination of physics-based and data-driven modeling [3].

Control of the size and shape of the melt pool in DED could allow users to avoid common defects like lack-of-fusion voids and keyhole porosities that have been correlated with melt pool variability [7]. In this work I explore two data-driven methods for predicting flaws in DED printed parts, data-driven stochastic reachability analysis and a recurrent neural network, for potential use in DED modeling and control applications.

1.2 Background

1.2.1 DED process

According to ISO/ASTM 52900 standards, DED encompasses all AM processes "in which focused thermal energy is used to fuse materials by melting as they are being deposited" [8]. As shown in Figure 1.1, DED processes can utilize a variety of feedstocks and energy sources. This work will refer mainly to laser engineered net shaping (LENS, a trademark of Optomec), also called laser metal deposition (LMD), a type of DED in which streams of metal powder carried by a gas are melted at the focal point of a laser. As the laser and powder deposition focal point moves laterally across the print bed, the previously molten material cools and solidifies. The metal part is built up from the substrate one layer at a time, with the melt pool at the focal point composed of a mix of melted freshly deposited powder and remelted material

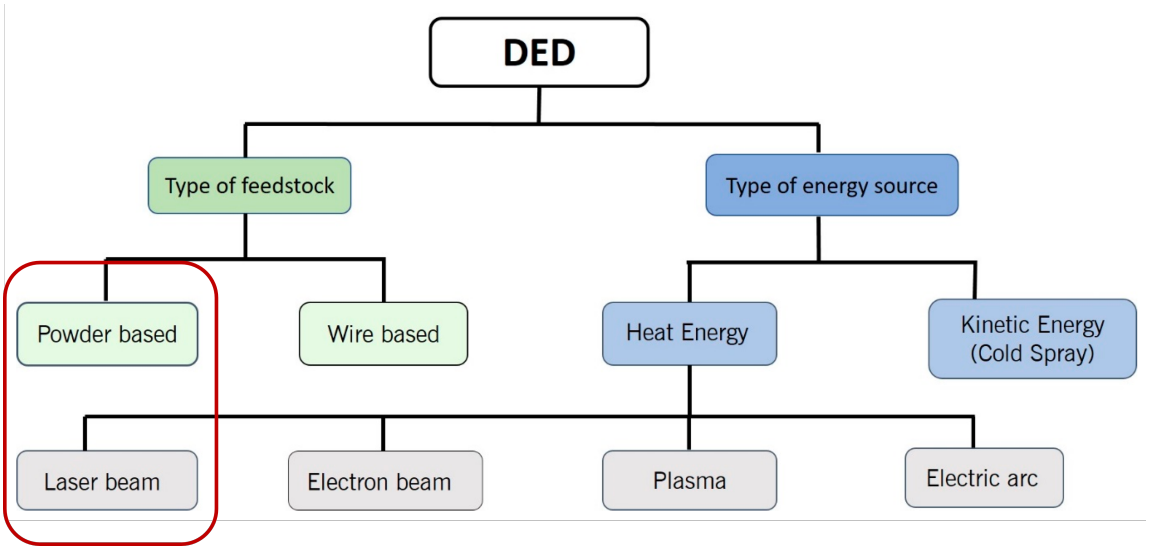

Figure 1.1: Types of DED printing. This work will focus on LENS, a powder- and laser-based form of DED. Adapted from [9] (Open Access).

from the prior layer(s). Figure 1.2 gives a visual overview of the LENS process. In LENS DED, the resulting part is "near net shape", meaning little or no additional machining is required to get the part to its final form.

The parts built and repaired with DED tend to be larger and lower resolution than parts made with related metal AM technologies like powder bed fusion (PBF). PBF, a metal AM process in which a laser is directed into a container filled with metal powder, is already used extensively in the biomedical and aerospace industries [10]. PBF and LENS DED share many of the same processing parameters, including beam power, scanning speed, layer thickness, beam diameter, and hatch spacing [5]. However, PBF has been studied more thoroughly and has proven easier to model and control than LENS, in part because PBF doesn't involve blown powder and carrier gases.

Besides virtually unlimited size constraints, DED parts have several other advantages over other metal AM processes. DED is much faster and more cost-effective

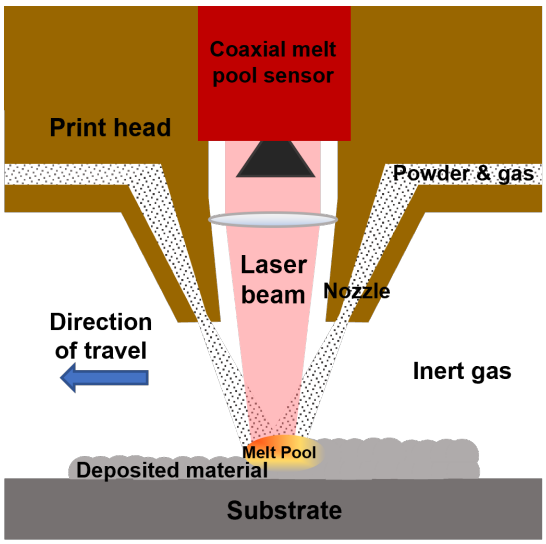

Figure 1.2: Illustration of basic LENS printing concept and terms used in this paper.

than PBF. Unlike metal AM processes like binder jetting and material extrusion, DED parts can be made fully dense, without gaps between metal particles. LENS DED also has the unique ability to manufacture components made up of multiple metal powder feedstocks, the ratios of which can be changed throughout the component. These material combinations, combined with custom geometries, may be lighter and/or stronger than traditional materials [11].

Defects found in DED printed parts include porosities, cracking, poor surface finish, and delamination from the substrate [11], [12]. In this work I focus mainly on porosities, as they are often harder to predict and control than other defects. Porosities can greatly alter material strength and other properties, making the printed parts unreliable or unusable. The types of porosities seen in DED are lack of fusion porosities (intralayer) and keyholing porosities (interlayer) (see Figure 1.3). Lack of fusion porosities are irregular in shape and occur between tracks or layers, or where the part meets the substrate, due to metal particles not fully melting together. Lower energy densities due to faster laser scan speeds and lower laser powers can cause parts to cool more quickly and lack-of-fusion porosities to be more common. In contrast, key-

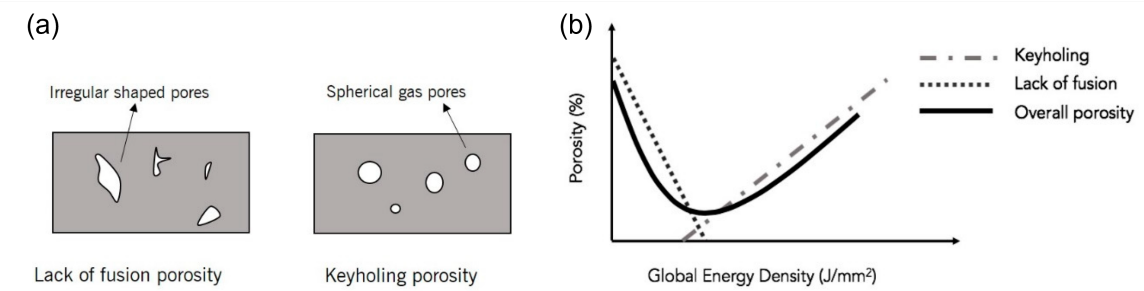

Figure 1.3: Representation of (a) DED porosity defect characteristics for lack of fusion and keyholing, and (b) intersection of lack of fusion and keyholing porosity prevalence with respect to energy density. Adapted from [9] (Open Access).

holing porosities are smaller and more spherical in shape, and occur due to trapped gas in random locations within the part. Higher energy densities create lower cooling rates and can cause metal powder to vaporize and form keyholes. Global energy density (GED) is thus a good predictor of porosity prevalence and type, and can be estimated as

$$GED = \frac{P}{dv}, \tag{1.1}$$

where P is the laser power, d is the laser spot size (diameter), and v is the laser scan speed [9]. Metal particle size, shape, and uniformity also contribute the occurrence of both types of porosities.

1.2.2 Learning and control theoretic models of DED

LENS DED has varied process parameters that are sometimes hard to accurately measure, the major ones being laser power, laser scan speed, laser spot size, powder feed rate, carrier gas flow rate, clad angle, feedstock properties, and layer dimensions. These varied parameters and the complex heat transport phenomena involved make it difficult to model and control the LENS process [9]. Still, physics-based models do exist, though control research on LENS DED has focused largely on maintaining

consistent layer heights in simple, limited cases. For example, Wang et al. (2016) [13] proposed a nonlinear implicit ODE model for a single printed layer derived from momentum, mass, and energy balance in the melt pool. Based on this model, they created a multi-input, multi-output (MIMO) feedback linearization controller with laser power and scan speed as inputs and melt pool height and temperature as outputs.

An approach derived from repetitive process control theory that incorporates dynamics both within a single layer and layer-to-layer was proposed by Sammons et al. (2018, 2019) [14], [15] to control for layer rippling within a single track wall. They used powder flow rate λ and standoff height d_S as control inputs with a 2D model of the current bead height $h(x, j)$ at position x, layer j:

$$\begin{aligned} h(x,j) =& \frac{1}{b\rho} f_\mu(d_S(x,j) - h(x,j-1))\lambda(x,j) * f_s(x) \\ &+ h(x,j-1) * f_r(x) \end{aligned} \tag{1.2}$$

In Equation (1.2) the first line represents the melt pool dynamics and the second line represents the remelt dynamics. The functions f_s, f_μ, and f_r describe melt dynamics, powder catchment, and remelt dynamics, respectively, b is the bead width, ρ is the material density, and $*$ is the spatial convolution operation. The functions f_s, f_μ, and f_r are experimentally derived. Though it incorporates both in-layer and layer-to-layer dynamics, this model does not extend to multi-track, complex build geometries, nor does it take into account thermal effects that contribute to microstructure and defects in the part.

Hoelzle and Barton (2016) [16] proposed a variation on iterative learning control for additive manufacturing. Iterative learning control (ILC), a type of nonlinear repetitive process control, is useful for processes with lots of uncertainty that are repetitive but may lack real-time feedback, as is often the case with AM processes. In standard ILC, measurements are locally weighted in time, with data immediately before and after the current time weighted more heavily than data at distant time

points. Hoelzle and Barton proposed a new form of spatial ILC that was derived from a 2-D spatial convolution instead of the standard 1-D temporal convolution, and which weights points that are near the current data point in 2-D space more heavily than those farther away. This approach makes a lot of sense for AM processes where two data points may be right next to each other in space but very distant in time. So far this approach has only been applied to ink jet and fused deposition modeling (FDM), not DED.

It is more common to use machine learning (ML) approaches when modeling melt pool geometry and predicting defects. Khanzadeh et al. (2017) [6] used self organizing maps to identify and categorize DED melt pool thermal images with anomalous geometries that were more likely to correspond to porosities. Their method was able to accurately predict 96% of porosity locations in the finished part. Akbari et al. (2022) [5] recently introduced a comprehensive melt pool characteristic prediction tool called MeltpoolNet. MeltpoolNet contains a variety of ML models (random forest, Gaussian process regressor and classifier, support vector regressor and classifier, Ridge linear regressor, Lasso linear regressor, logistic regression, gradient boosting, and neural network) that were trained on processing parameters, materials and melt pool characteristics from over 80 different PBF and DED datasets. Their tool also contains data-driven system identification methods that allow for direct interpretation of the relationships between process parameters and melt pool characteristics. Still, the vast majority of the data used to create MeltpoolNet was from PBF, not DED processes, and only about 100 of the 2200 data points used were LENS DED, suggesting that DED remains understudied and not well understood.

1.2.3 Stochastic reachability, kernel methods, and neural networks

Stochastic reachability aims to predict the likelihood that a dynamic, non-deterministic system will get to a desired target while remaining within safety constraints. The reachability analysis performed herein is based on the work of Summers and Lygeros (2010) [17], who proposed a dynamic programming solution to the stochastic reachability problem for discrete time stochastic hybrid systems. They defined two versions of the stochastic reachability problem: first hitting time (FHT), where the system reaches the target set at any point during a finite time horizon while remaining in the safe set, and terminal hitting time (THT), where the system reaches the target set at a pre-defined terminal time while remaining in the safe set. In this work we use the FHT problem to predict the likelihood of upcoming porosities, and validate with the true locations of porosities determined via CT scan.

Kernel methods and neural networks are both ML algorithms that can be used to approximate an unknown function, in this case the unknown DED system dynamics. The stochastic kernel is based on quantifying the similarity between subsequent time steps, and has a closed-form solution. In this thesis, the stochastic kernel is used for reachability analysis in the manner outlined by Thorpe and Oishi (2022) [18]. Similar to kernel methods, neural networks "learn" certain parameters of the function, but unlike the direct approach used in kernel methods, neural networks determine these parameters iteratively. We use recurrent neural networks (RNNs) in our analysis because they maintain a memory of previous time steps, allowing us to capture the system dynamics. Instead of the target and safety constraints used in our reachability analysis, we trained our RNNs in a supervised manner using known

porosity locations. To avoid confusion when using the word "layer", neural network layers will often be referred to as "NN layers", and layers of the DED printed walls as "printed layers".

1.3 Problem statement

The goal of this thesis is to demonstrate the feasibility of two data-driven methods for predicting defects in DED manufactured parts. If we presume that the porosities in DED printed parts are the result of a dynamic stochastic process, then predicting the likelihood of porosities at future times can be formulated as a stochastic reachability problem. If we don't explicitly model the system as a dynamic process, but allow for feedback with a RNN, the problem becomes training of a predictor of porosity formation based on past and present pyrometry data, since hidden states contain information from arbitrary points earlier in the sequence.

1.4 Contributions

This thesis provides a demonstration of two data-driven methods, stochastic reachability analysis and RNNs, for prediction of voids in DED printed parts. The main contribution is in the novel application, as these methods have not previously been used for porosity prediction in DED. The results presented in this thesis will be submitted to an AM conference to be determined.

Chapter 2

Problem Formulation

2.1 Data collection

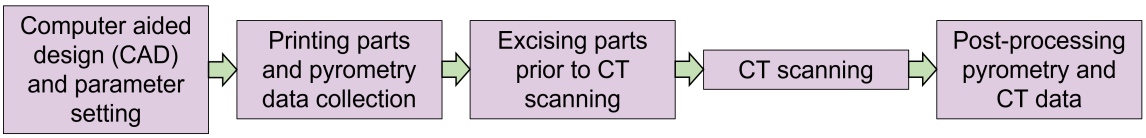

Figure 2.1: Data collection process.

Prediction of porosities in DED printed components requires data collected during printing, in this case temperature data from a melt pool sensor called a pyrometer that is commonly installed in LENS machines, as well the true locations of the porosities in the finished part, here determined after printing via computed tomography (CT) scan. Initially, I tried to utilize DED pyrometry and CT scan data collected for prior experiments at Sandia National Laboratories and Navajo Technical University (NTU). However, I soon realized that the existing datasets lacked positional information necessary for aligning melt pool thermal images to porosities. Namely, the pyrometry images, as depicted in Figure 2.2, show only the melt pool and the area immediately surrounding it, and utilize a relative coordinate frame that moves

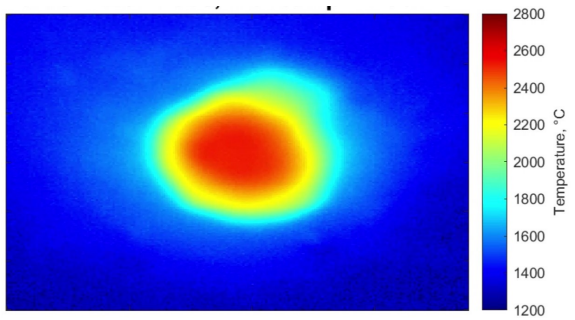

Figure 2.2: Sample DED melt pool thermal image.

with the print head, without a known transformation to "part-level" coordinates to register to CT-determined void locations.

To correct this, I traveled to NTU four times to obtain usable data and to better understand the DED process I was trying to model. All experimental data used in this thesis was collected at NTU in their fabrication lab. The entire procedure for collecting said data is condensed in Figure 2.1. Each of the three main machines involved in the data collection process, the LENS printer, the wire electrical discharge machine (EDM) used to separate the parts from the substrate so they fit into the CT scanner, and the CT scanner itself, are complicated, expensive machines that take a great deal of training and experience to master. I did not have prior experience with any of these machines, so I am very grateful to the NTU fabrication lab staff for their assistance and expertise.

Since the quality of the solutions for both neural networks and kernel methods is largely dependent upon the quality of the data available, informed collection of data using control sequences that are close to optimal is paramount. The DED parts printed for data collection were designed as 3 mm x 20 mm x 10 mm (0.1181" x 0.7874" x 0.3937") rectangular structures called "walls" (see Figure 2.3). I wanted each part to contain around 20 layers, and operator experience dictated that the printed layer height be set to 0.02" (0.508 mm), leading to a 1 cm tall part. This

Layer height	0.02 in
Laser scan speed	20 in/min
Laser power	270W to 340W
Laser spot size	0.6 mm (0.0236 in)
Powder feed rate	7.5 g/min
Powder flow dwell time	0 sec
Curve chord tolerance	0.001 in
Seam avoidance offset	0.04 in
Pass overlap	50% (0.0118 in)
Maximum pass overlap	75% (0.0177 in)
Start/stop overlap	50% (0.0118 in)
Fill/boundary overlap	0%
Lead-in/out length	0.001 in

Table 2.1: Relevant DED printing parameters.

design was selected for its simplicity, large number of layers in a small package, and ease of registration to maintain consistent part-level coordinates for CT scans. Previously, pyrometry and CT data had been obtained for small cylindrical prints ("coins"), but their shape made registration impossible. Thin walls, i.e. multiple stacked single track layers, are commonly used to characterize layer height variation in DED [12], [14], but would not have provided any in-layer track-to-track insights and may have been too thin to easily CT scan. The walls were designed in Mastercam with LENS plugin. The only process parameter that changed between parts was the control input, laser power, which was varied in increments of 10W from 270W to 340W. Within each layer the boundary was printed before the fill, and the fill was done in an identical zigzag pattern for each layer. Other printing parameters are shown in Table 2.1. Parameters were selected based on operator experience.

The walls were printed on NTU's Optomec LENS 3D Hybrid printer equipped with an 1076 nm fiber laser (nLight), controlled atmosphere system, powder feed system, and a melt pool sensor (Stratonics ThermaViz 2-wavelength pyrometer)

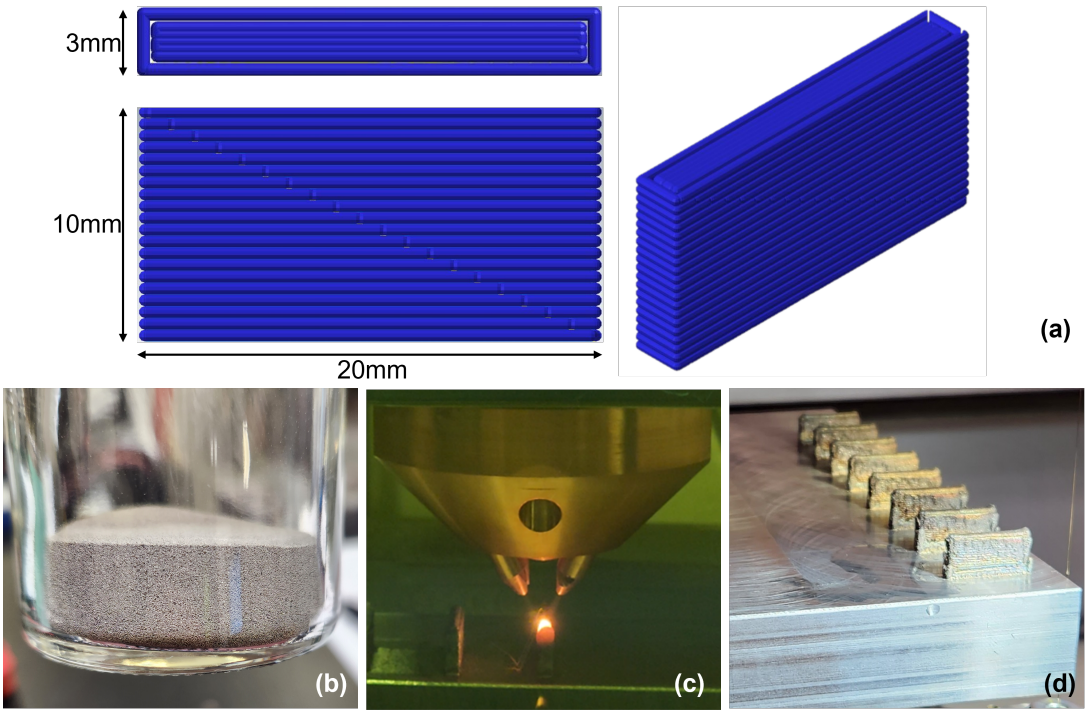

Figure 2.3: DED wall printing process. (a) Part design in Mastercam, (b) SS316 powder feedstock, (c) printing walls, and (d) finished parts ready to be excised with wire EDM.

mounted on the print head to provide thermal images of the melt pool. The parts were constructed from SS316 stainless steel powder (READE), a commonly used material, on a 1" thick stainless steel substrate and oriented as shown in Figure 2.4 (a), with 0.75" between parts and the origin point set at the lower left corner of each part. This same point was later used as the origin point for CT scans of the parts. Each wall took about 6 minutes 12 seconds to print.

While printing, the pyrometer captured thermal images of the melt pool with a 30 Hz frame rate. 30 Hz was determined to be an appropriate minimum sampling rate to capture information relevant to the dynamics of the melt pool and movement of the print head and based on the following: If the print head is traveling at 20 in/min $= \frac{1}{3}$ in/sec and the laser spot size (diameter) is 0.6 mm = 0.02362 in, then

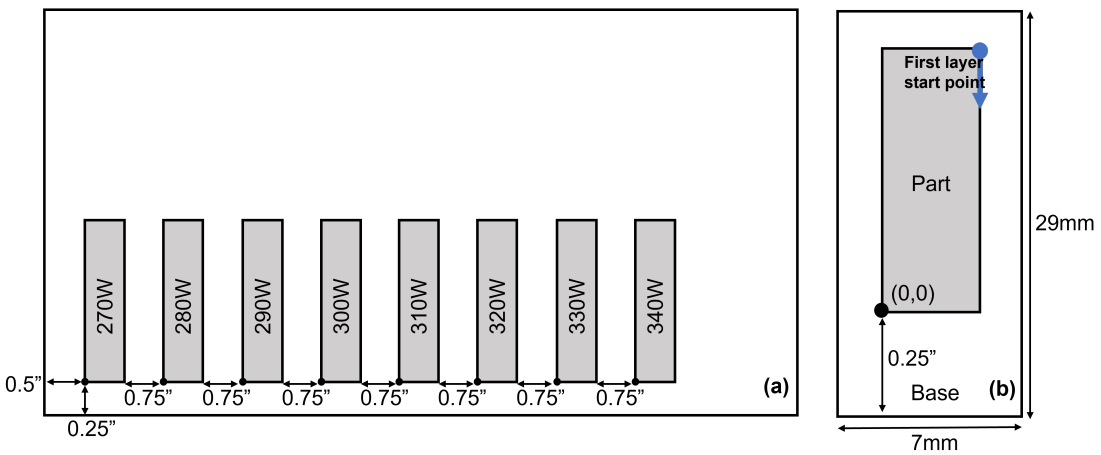

Figure 2.4: Part layout and orientation. (a) Print layout of 8 printed walls on build plate. The (x,y) origin for the printer is set at the lower left corner of each part before printing. (b) Layout of EDM excised part for CT scanning showing the first layer printing start point and direction.

the minimum sampling rate needed to capture subsequent laser spots without any gaps or overlap between them is $\frac{1}{3}$ in/sec $\div$ 0.02362 in = 14.11 Hz. This frequency was doubled to find the Nyquist frequency, 28.22 Hz, which was rounded up to a nice round 30 Hz.

After printing, the parts and substrate were cleaned with isopropanol, and the walls were individually cut out of the substrate using a wire electrical discharge machine (EDM). A portion of the substrate was cut out along with each wall to preserve the substrate-part interface and part orientation, leaving each wall with its own 7 mm x 29 mm x 1 mm substrate, shown in Figure 2.4 (b). Each part was then individually CT scanned to visualize internal flaws, and the radiographs output by the CT scanner were registered to the wall's computer-aided design (CAD) file to maintain consistent part coordinates throughout. This alignment, as well as porosity analyses and CT scan visualization was done using Volume Graphics Studio software. A sampling of the graphics generated by Volume Graphics Studio for a single printed wall are shown in Figure 2.5.

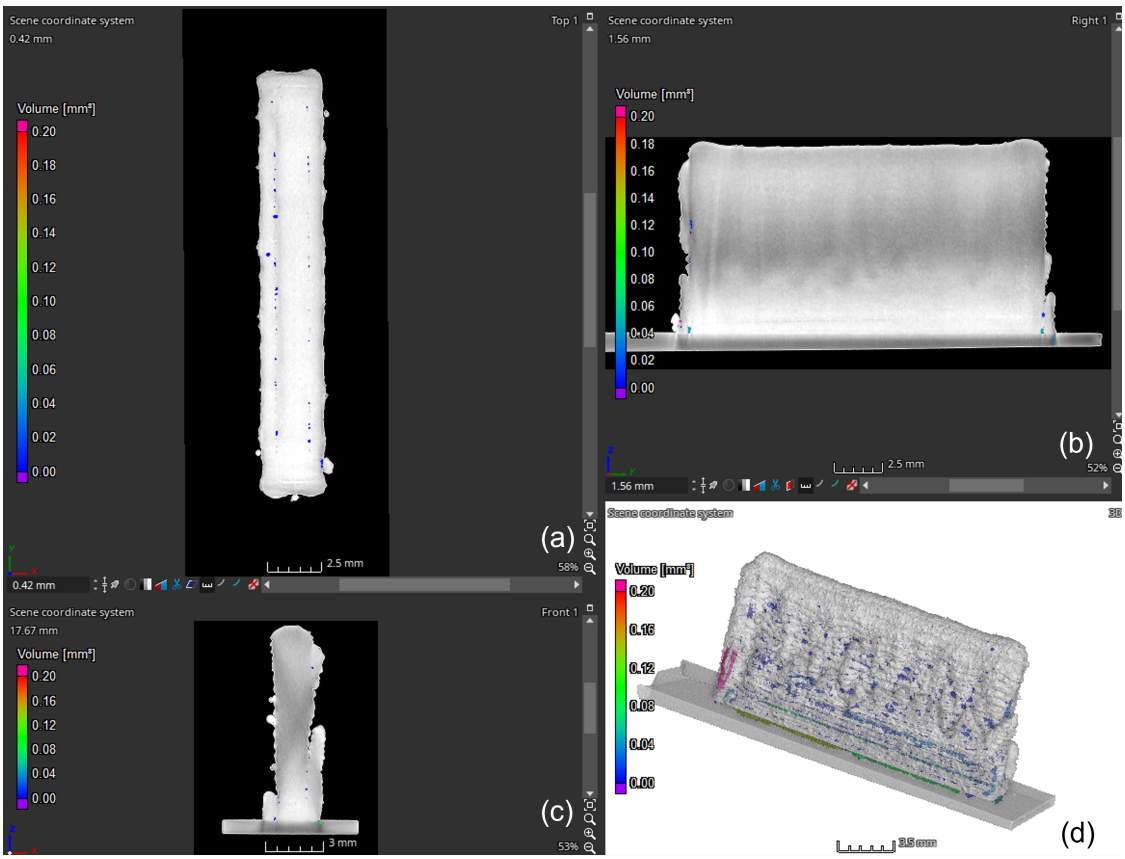

Figure 2.5: Screenshot of Volume Graphics Studio visualization and porosity analysis of CT scan for the wall printed at 270 W. Predicted porosity locations are colored based on porosity volume, and large, unlikely porosities are excluded from future analysis. (a) Cross section at z = 0.42 mm, (b) cross section at x = 1.56 mm, (c) cross section at y = 17.67 mm, and (d) 3D view.

2.2 Modeling description

We conceptualized the DED printing process as a stochastic dynamical system, described for each layer of the part by the difference equation

$$\mathbf{x_{k+1}} = f(\mathbf{x_k}, \mathbf{u_k}, \mathbf{w_k}), \tag{2.1}$$

where the state $\mathbf{x_k} \in \mathcal{X}$ is the temperature at time k, the control input $\mathbf{u_k} \in \mathcal{U}$ is the laser power at time k, and $\mathbf{w_k}$ is the external disturbance process. We interpret the state as the temperature associated with a discretized section of the printed part, with a value that is averaged over all pixels within the section. Hence state trajectories of this dynamical system describe the discrete time evolution of temperature as the print head moves across a part. We interpret the input as similarly quantized, and that it takes on a nonzero value when the laser spot impacts a given section of the part. We have no explicit knowledge of the disturbance process.

One of the major challenges for control, from a dynamical systems point of view, is that sensing and actuation are co-located. The pyrometer travels with the print head. This means the measurements can be corrupted by stochastic variability in laser power and spot shape. Further, pyrometry data is only gathered within a certain region around the print head, meaning that our measurements are only local to the actuation. Because measurements for the state and input are described only locally, we keep track of two coordinate frames: a global, inertial frame that is fixed to the part, and a local, relative frame that is fixed to the laser head. The relationship between these two is described by the reference trajectories of the print head.

2.3 Data manipulation

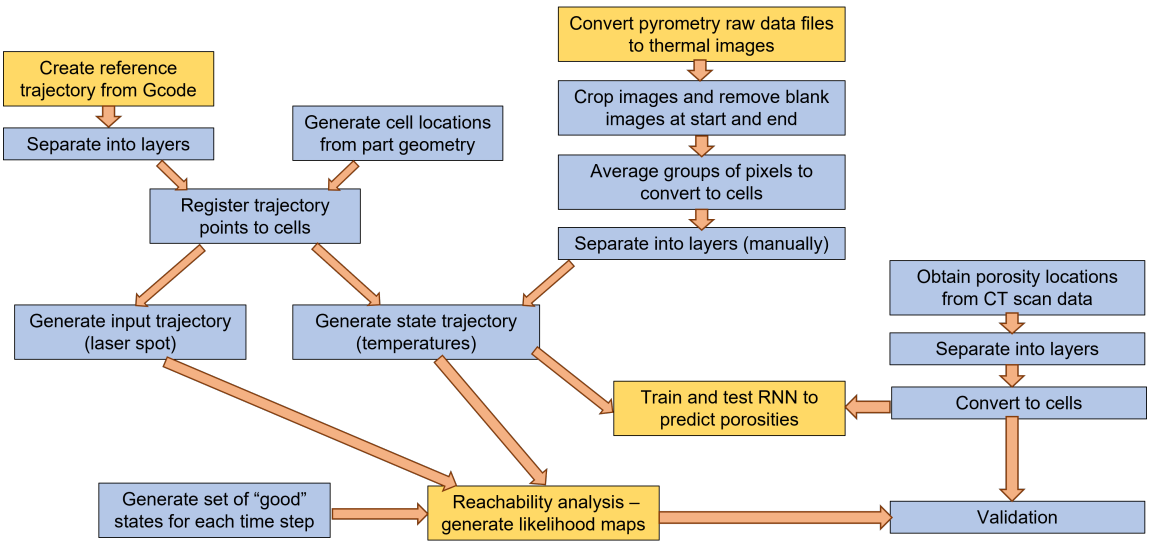

Figure 2.6: Workflow of data processing and analysis for this thesis. Yellow indicates software code modified from existing code written by others, blue indicates entirely custom code.

Figure 2.6 gives an overview of the tasks performed for this thesis after the raw data collection. A custom Matlab or Python script was written for nearly every item in this figure, so it should be apparent that the bulk of the work was done in preprocessing the data.

2.3.1 Establishing inertial reference trajectories

Before the melt pool thermal data could be used in reachability or neural network analyses, the relative coordinate frame used to capture the data, with the camera looking down on the melt pool and moving with the print head, had to be converted to an inertial coordinate frame of the entire printed part. This conversion was complicated by the fact that the DED machine at NTU has no sensors to record the position of the print head at any given time.

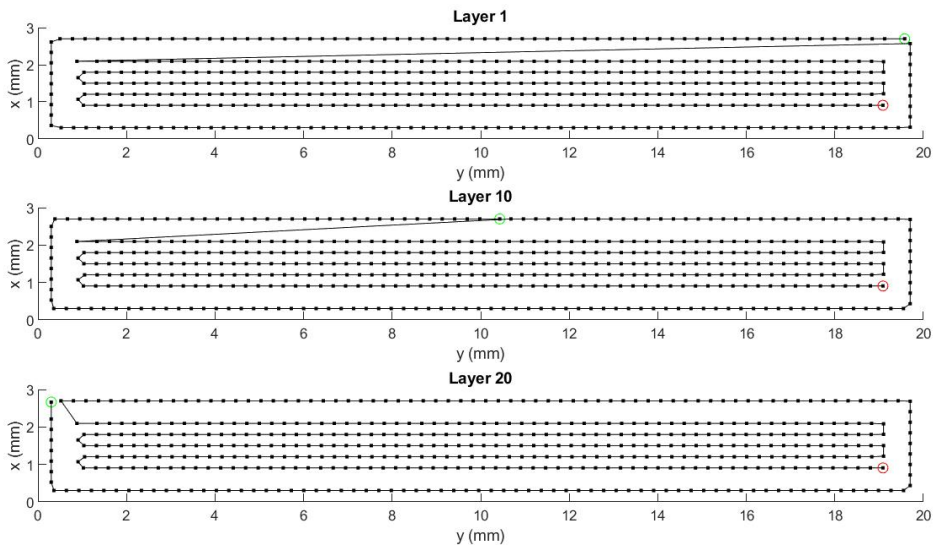

Figure 2.7: A selection of layer trajectories generated from Gcode with 30 Hz frame rate. Green circle is the start point, red circle is the end point for each layer.

To solve this issue, the position of the print head at the time each thermal image was taken had to be inferred from the frame rate and the Gcode commands given to the printer. Gcode is a common programming language used to direct computer numerical control (CNC) machines, which in this case contains basic instructions like inertial coordinates for the print head to travel to at a set speed. A Matlab code was modified from [19] to convert Gcode generated by Mastercam with LENS plugin to sets of inertial coordinate points that correspond to the location of the print head at each time step. This reference trajectory containing the inertial position of the print head over time was separated into 2D layers, a sampling of which are shown in Figure 2.7.

2.3.2 Establishing the state space

The discretized state elements x_n were defined as the time-dependent temperatures at a set of given locations within each layer of a printed wall. These locations

were chosen by first dividing the horizontal (xy-plane) cross-sectional area of the wall, $20mm \times 3mm = 60mm^2$, into 10,000 square cells of width 0.07746 mm. The melt pool thermal images contain 0.014 mm square pixels, making each cell 5.53 pixels wide. The cell width was rounded up to 6 pixels (0.084 mm) so that images could be easily translated into cells. Thus each state element was defined as the temperature average of the 36 pixels in a given cell. The total number of state elements is equal to the number of cells in one layer of the printed part, 8880. The cells in the 37×240 grid representing one layer of a wall are shown in Figure 2.8. The state elements x_n are numbered to create a state vector $\mathbf{x} \in \mathbb{R}^{8880}$, starting with x_1 at the origin of the part (bottom left) and increasing n first in the +y direction (up each column) and then in the +x direction (subsequent columns left to right). The 2D states $\mathbf{x}$ were parameterized by layer to represent the temperature at each discretized 3D location.

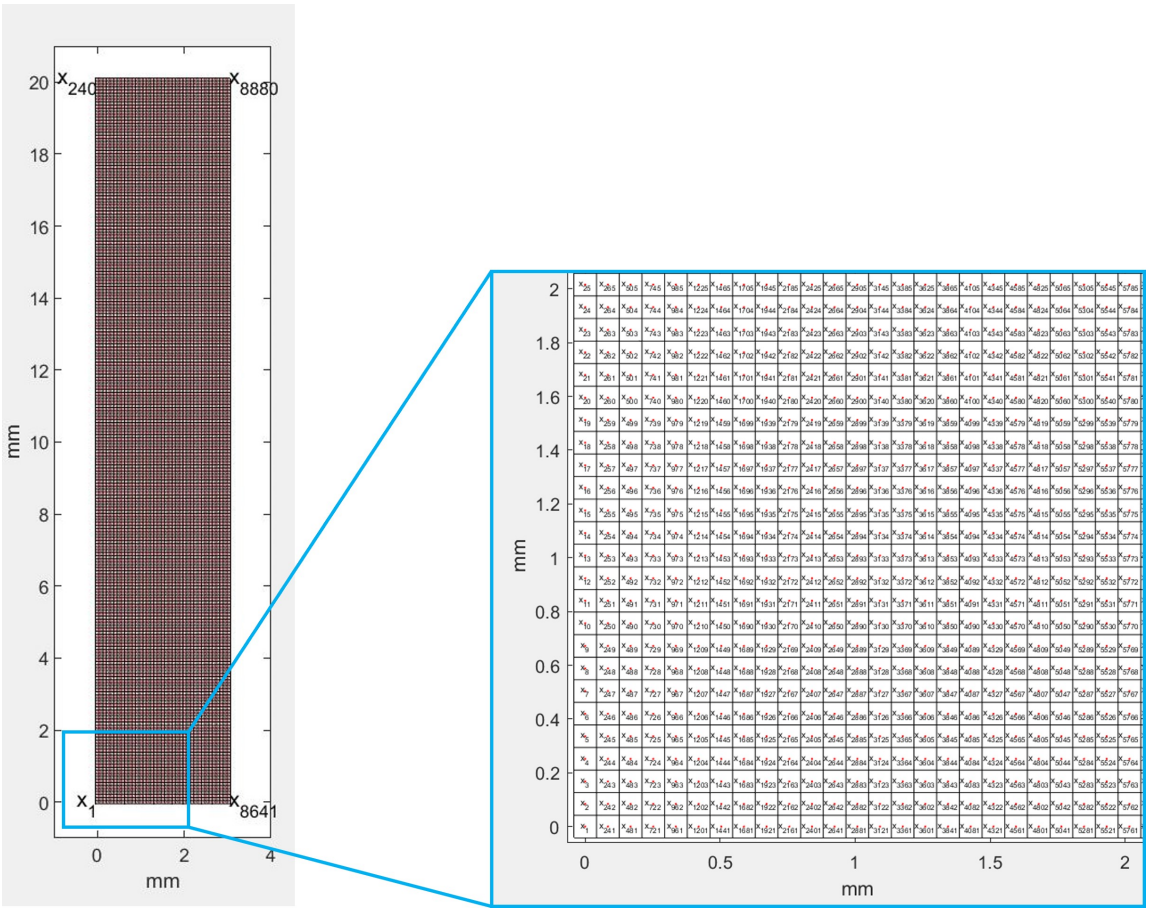

Figure 2.8: Representation of discretized state locations as the center points (red) of 0.084 mm square cells. Right is a zoomed-in section of the grid on the left. The state elements are numbered from x_1 on the bottom left to x_{8880} on the top right.

2.3.3 Input trajectories

The reference trajectory points generated from Gcode were then registered to the nearest discretized state locations. Input trajectories containing the laser power seen by each cell over time were generated by presuming that each registered reference trajectory point is the center point of the laser spot at a particular time step. There are 37 cells centered around each reference trajectory point that are within the laser spot diameter (0.6 mm) and turned "on" with a given laser power at any given time

step. All other cells are "off" and set to zero. A representation of a single input trajectory point is shown in Figure 2.9.

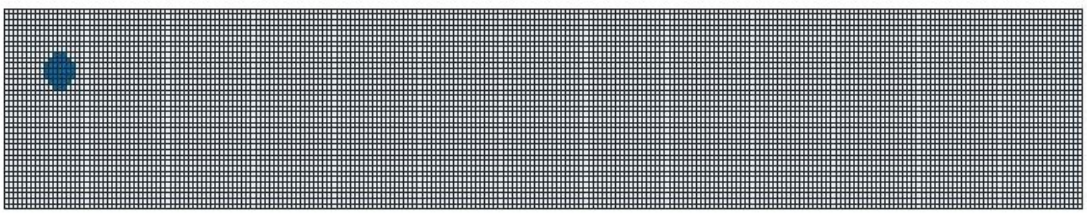

Figure 2.9: Depiction of a single input trajectory point as a discretization of a laser spot. For each time step, shaded cells are "on" and have the given laser power for the part, the rest of the cells are "off" and have a laser power of zero.

2.3.4 State trajectories

The thermal "images" taken by the pyrometer are actually sets of temperature values, one value for each 0.014 mm square pixel, saved in a proprietary ThermaViz file format. These were converted to Matlab matrices using code provided by Anusuya Vellingiri of NTU. The thermal images were cropped to 198x198 pixels around the melt pool to reduce extraneous data. Although Gcode commands exist to trigger the melt pool sensor to turn on and off, for example simultaneous with the laser turning on and off, the setup at NTU did not have such capabilities, which meant that melt pool sensor recording had to be manually triggered before and after the printing of each wall. Excess minimum temperature ("blank") frames were trimmed from the start and end of each of the 8 image sets ("videos"). Separation of layers, and removal of frames between layers and between the printing of boundary and fill where the laser is turned off, had to be done manually because these images were not simply blanks. Many still showed residual heat, or would "flash" and cover a single frame with a high temperature due to sensor issues at low temperatures. Manual frame removal was facilitated by a Matlab code that displayed the frame

numbers of blank frames along with a set of frames before and after the blank frame that may require removal, but the ultimate decision of where to slice and what to excise was up to me. After this time-consuming process was completed, the number of remaining frames was often not exactly equal to the number of frames in a layer predicted by the reference trajectory (481 laser-on frames/layer), so frames had to be added to or removed from the end of the set. I emphasize, this process would have been much more accurate and efficient if I had been able automatically trigger the melt pool sensor to record, and especially if the inertial coordinates at which each image was taken were automatically recorded.

For transformation from relative melt pool image coordinates into inertial layer coordinates, the center of each melt pool image was presumed to coincide with a reference trajectory point, which is also the location of a state x_c. The 198x198 images with centers thus described could be divided evenly into 6x6 pixel squares with the center point of the image at the center of one of these squares. The pixels within each square were averaged to obtain a single temperature value for each cell. If the center point of the image is the center of the cell corresponding to state x_c, then the cells that contain the image are cells that correspond to states x_i, where $i \in \{c + 240(j - 17) + (k - 17)\}$ and $j, k \in \{1, 2, \ldots, 33\}$, for the portion of the image that lies within the part boundaries. This mapping procedure is illustrated by Figure 2.10. States outside the image bounds were set to 1000°C, an approximation of the common temperature found at the edges of the melt pool images. A single state trajectory $\{\mathbf{x_k}\}_{k=0}^{481}$ represents a single printed layer.

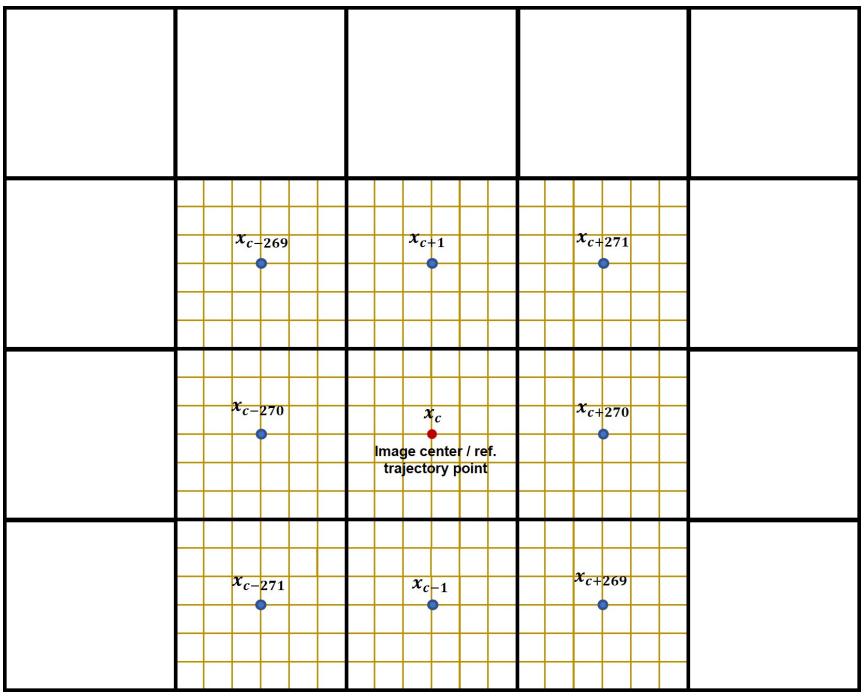

Figure 2.10: Simplified representation of the procedure for mapping images to state locations. Gold lines are pixel outlines, black are cell outlines. Here the image is represented as 18x18 pixels (3x3 cells), while in actuality the melt pool images are 198x198 pixels (33x33 cells).

2.3.5 CT scans

The porosity analyses generated by Volume Graphics Studio from the CT scans can be exported as .csv files containing the size and 3D location of every suspected porosity in the part, as well as the likelihood of actually being a porosity based on contrast with the surrounding voxels. Porosities with low likelihoods and porosities with large volumes that were probably not actually porosities were eliminated from the list. Porosities were sorted into layers based on the z-position of their center point. It should be noted that while layer height was assumed to always be the set value of 0.5 mm, in actuality layer height varies throughout the part due to melt pool variability and other stochastic processes. The porosity (x,y) center points were then

registered to the nearest state locations, and all cells within a circle specified by the porosity radius around these locations were set to 1. All other discretized locations within each layer were set to 0.

Chapter 3

Methods

3.1 Stochastic reachability

For sets $\mathcal{A}$ and $\mathcal{B}$, the set of all elements in $\mathcal{A}$ which are not in $\mathcal{B}$ is denoted $\mathcal{A}\backslash\mathcal{B}$. For some nonempty set $\mathcal{A} \subseteq \mathcal{B}$, denote the indicator function as $\mathbf{1}_{\mathcal{A}} : \mathcal{B} \to \{0, 1\}$ where $\mathbf{1}_{\mathcal{A}}(x) = \begin{cases} 1 & \text{if } x \in \mathcal{A} \\ 0 & \text{if } x \notin \mathcal{A} \end{cases}$

We consider the system modeled as a Markov control process.

Definition 3.1.1 (Markov Control Process, [20])**.** *A Markov control process is a 3-tuple* $(\mathcal{X}, \mathcal{U}, Q)$*, consisting of:*

- *A state space* $\mathcal{X}$*;*
- *A control space* $\mathcal{U}$*; and*
- *A stochastic kernel* $Q : \mathscr{B}_{\mathcal{X}} \times \mathcal{X} \times \mathcal{U} \to [0, 1]$ *that assigns a probability measure* $Q(\cdot \mid x, u)$ *to every* $x \in \mathcal{X}$ *and* $u \in \mathcal{U}$ *on the state space* $\mathcal{X}$*.*

The system evolves from an initial condition $x_0 \in \mathcal{X}$ over a finite time horizon $k = 0, 1, \ldots, N$, $N \in \mathbb{N}$, with inputs taken from a stochastic control policy π.

Definition 3.1.2 (Stochastic Policy)**.** *A stochastic policy is a collection* $\pi = \{\pi_0, \pi_1, \ldots, \pi_{N-1}\}$*, such that for every* $k = 0, 1, \ldots, N-1$*,* $\pi_k : \mathscr{B}_{\mathcal{U}} \times \mathcal{X} \to [0, 1]$ *is a stochastic kernel from* $\mathcal{X}$ *to* $\mathcal{U}$*. We denote the set of all stochastic policies as* Π*.*

We model the DED printing process via a Markov control process, where the pyrometry images in the inertial reference frame are the states, and the laser spot intensities in the same reference frame are the control inputs. Our goal is to compute the likelihood that the system will remain within the correct temperature bounds when the laser is melting a given area. This is because if the system goes outside the desired bounds, we may create voids. Thus, we seek to compute the likelihood that the system will remain safe using stochastic reachability, as in [17].

Here, we focus on the first-hitting time problem, which seeks to compute the likelihood that a system will reach a target set $\mathcal{T}$ at some point $j \leq N$ and remain within a "safe" set $\mathcal{K}$ for all time $t < j$. In other words, we seek to compute the probability,

$$r_{x_0}(\mathcal{K}, \mathcal{T}) = \mathbb{P}^{\pi}_{x_0}[(\exists j \in [0, N] : x_j \in \mathcal{T}) \wedge (x_i \in \mathcal{K} \setminus \mathcal{T}, \forall i \in [0, j-1])] \tag{3.1}$$

As shown in [17], we can write the above probability as the expected product of indicator functions,

$$r^{\pi}_{x_0}(\mathcal{K}, \mathcal{T}) = \mathbb{E}^{\pi}_{x_0}\left[\sum_{j=0}^{N}\left(\prod_{i=0}^{j-1} \mathbf{1}_{\mathcal{K}\setminus\mathcal{T}}(x_i)\right)\mathbf{1}_{\mathcal{T}}(x_j)\right] \tag{3.2}$$

In our case, we define the safe set $\mathcal{K}$ to be the entire state space,

$$\mathcal{K} = \mathcal{X}. \tag{3.3}$$

Then, we define the target set $\mathcal{T}$ to be a set of undesirable states, in this case time-varying because the melt pool is moving across the part. When the laser is shining

on a particular area, we do not want temperature of that area to be less than the melting temperature of stainless steel 316, which is just under 1400°C.

$$\mathcal{T} = \begin{cases} \{x \in \mathcal{X} : x < 1400\} & \text{if within the laser spot at time } t \\ \mathcal{X} & \text{if outside the laser spot at time } t \end{cases} \tag{3.4}$$

Thus, we seek to determine the likelihood that the system will reach the "bad" set $\mathcal{T}$ at some point $j \leq N$.

However, (3.2) is difficult to compute, especially if the stochastic kernel Q in Definition 3.1.1 (which captures the dynamics and uncertainty of the process) is unknown. As shown in [17], we can solve (3.2) as a dynamic program.

Following [17], for a fixed policy π, define the value functions $V_k^\pi : \mathcal{X} \to [0, 1]$, $k = 0, 1, \ldots, N$, as

$$V_N^\pi(x) = \mathbf{1}_\mathcal{T}(x) \tag{3.5}$$

$$V_k^\pi(x) = \mathbf{1}_\mathcal{T}(x) + \mathbf{1}_{\mathcal{K}\setminus\mathcal{T}}(x)\mathbb{E}[V_{k+1}] \tag{3.6}$$

Then, the safety probabilities $r_{x_0}^\pi(\mathcal{K}, \mathcal{T}) = V_0^\pi(x_0)$.

In practical terms, this means we are computing the likelihood that the system will reach a set of "bad" states $\mathcal{T}$ at some time $j \leq N$ while remaining outside $\mathcal{T}$ for all time $t < j$,

$$V_k^\pi(x) = \mathbf{1}_\mathcal{T}(x) + \mathbf{1}_{\mathcal{T}^c}(x)\mathbb{E}[V_{k+1}] \tag{3.7}$$

where $\mathcal{T}^c$ is the complement of $\mathcal{T}$ in $\mathcal{X}$.

However, because the stochastic kernel Q is unknown, we cannot compute the expectations in (3.6). Instead, we use a Python toolbox of kernel-based methods for stochastic optimal control developed by Thorpe and Oishi (2022) [18] to approximate the expectations using data. We used data from a single layer of all 8 printed parts to compute the kernel for this analysis, since using all of the collected data proved too

computationally intensive. We chose the initial conditions x_0 as the actual state of the system every time step in the first layer of the 270W wall to obtain predictions spatially distributed over the entire layer. Safety probabilities for each location were calculated for several short time horizons to visualize the likelihood of future porosities.

3.2 RNN

Each layer in a neural network (NN) consists of a set of input features and a set of neurons that transform the inputs with multiplicative weights and additive biases. Layers will also often have activation functions which impart nonlinearity to the learned solution, making it possible to simulate nonlinear functions. It is the weights and biases, also called the network parameters, which are adjusted during the training of the network. In this case we used backpropagation, a common way to train multilayer NNs to minimize the mean squared error (MSE) between calculated and expected outputs. The backpropagation algorithm is described by Hagan (2003) [21] as follows.

Assume we have a neural network with M layers where the output of layer m is $\mathbf{h}^m$, the external input features are $\mathbf{h}^0 = \mathbf{x}$, and the output of the network is $\mathbf{h}^M = \mathbf{y}$. $\mathbf{W}^m$ and $\mathbf{b}^m$ are the weight matrix and bias vector for layer m, respectively, and $\mathbf{f}^m(\mathbf{n}^m)$ is the activation function for layer m, where the inputs to the activation function, $\mathbf{n}^m$, are called the net input. This notation is illustrated in Figure 3.1. The gradient of the activation function with respect to the net input for layer m is represented by the diagonal matrix $\dot{\mathbf{F}}^m(\mathbf{n}^m)$. The true MSE of the network, E, is approximated as the squared error between the network output $\mathbf{y}$ and the predefined target output $\mathbf{t}$ at iteration k,

$$\hat{E} = (\mathbf{t}(k) - \mathbf{y}(k))^T(\mathbf{t}(k) - \mathbf{y}(k)), \tag{3.8}$$

and the sensitivity of $\hat{E}$ to changes in the net input at layer m is defined as

$$\mathbf{s}^m := \frac{\partial \hat{E}}{\partial \mathbf{n}^m}. \tag{3.9}$$

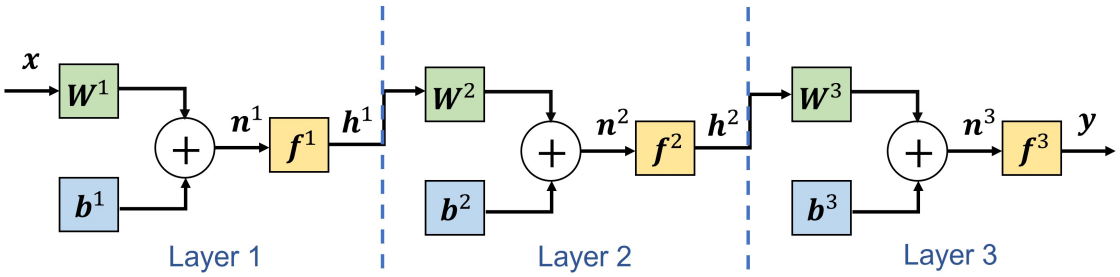

Figure 3.1: An example 3-layer neural network establishing abbreviated notation.

The first step in each backpropagation iteration k is to propagate the inputs forward through the network:

$$\begin{aligned} &\mathbf{h}^0 = \mathbf{x} \\ &\mathbf{h}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{h}^m + \mathbf{b}^{m+1}) \quad \text{for} \quad m = 0, 1, \cdots, M-1 \\ &\mathbf{h}^M = \mathbf{y}. \end{aligned} \tag{3.10}$$

Once the output $\mathbf{y}$ is obtained, the sensitivities $\mathbf{s}^m$ are propagated backward through the network:

$$\begin{aligned} &\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{y}) \\ &\mathbf{s}^m = \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T\mathbf{s}^{m+1} \quad \text{for} \quad m = M-1, \cdots, 2, 1. \end{aligned} \tag{3.11}$$

Finally, the network parameters are updated for iteration $k+1$ using steepest gradient descent:

$$\begin{aligned} &\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha\mathbf{s}^m(\mathbf{h}^{m-1})^T \\ &\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha\mathbf{s}^m \end{aligned} \tag{3.12}$$

where the predetermined parameter α is called the learning rate.

Recurrent neural networks (RNNs) are NNs with feedback layers, where some outputs are connected back to prior layers as inputs. RNNs can be more powerful than the traditional feedforward NNs of the type depicted in Figure 3.1 because they can model temporal behavior. There are many types of RNNs, but one of the most popular and the one used in this thesis is the long short-term memory network (LSTM), first introduced by Hochreiter and Schmidhuber in 1997 [22]. LSTMs were developed to address the vanishing gradient problem present in earlier RNNs, since their "memory" of past inputs can persist for thousands of time steps. LSTM layers take the form of an LSTM cell, depicted in Figure 3.2. The input to the cell at time t is $\mathbf{x}(t)$, the cell output is $\mathbf{h}(t)$, and the cell state $\mathbf{c}(t)$ holds the long-term memory from past time steps. Within each LSTM cell are 3 "gates" that alter the cell state: the forget gate determines which information should be removed from the previous cell state, the input gate combines the current input and past output to determine what to add to the cell state, and the output gate computes the current output from the past output and the current cell state and input [23].

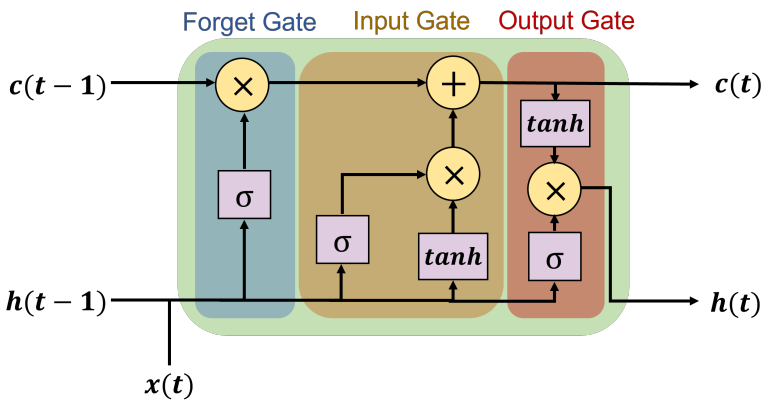

Figure 3.2: Components of an LSTM cell. In purple are sigmoid (σ) and hyperbolic tangent activation functions (tanh), and in yellow are element-wise operations.

We trained two LSTMs whose architecture is depicted in Figure 3.3. The first

consists of an LSTM layer followed by a fully-connected (linear) layer. The second has an added convolutional neural network (CNN) layer before the LSTM layer to help with identifying relevant features in the input pyrometry images. CNNs are commonly used for image classification, and CNN-RNN combinations similar to the one implemented here are prevalent in video classification applications. The input features for the LSTM without a CNN layer were the state trajectories $\{\mathbf{x_k}\}_{k=0}^{481}$ for a single printed layer, as described in Section 2.3.4. If the first layer of the network was a convolutional layer, then the state vectors were converted back into 2D images of the entire printed layer prior to input into the network, and the 2D outputs of the CNN layer were flattened before being fed into the LSTM layer. After the CNN layer, a rectified linear unit (ReLU) activation function was implemented to accommodate nonlinearities, and a max pooling layer was used to downsample and make training and using the network computationally tractable. In both networks, the final fully connected layer was followed by a sigmoid activation function to scale the network outputs between 0 and 1 and to accommodate nonlinearity. The network outputs were predicted porosity likelihoods between 0 and 1, with target outputs derived from from CT scans as described in Section 2.3.5.

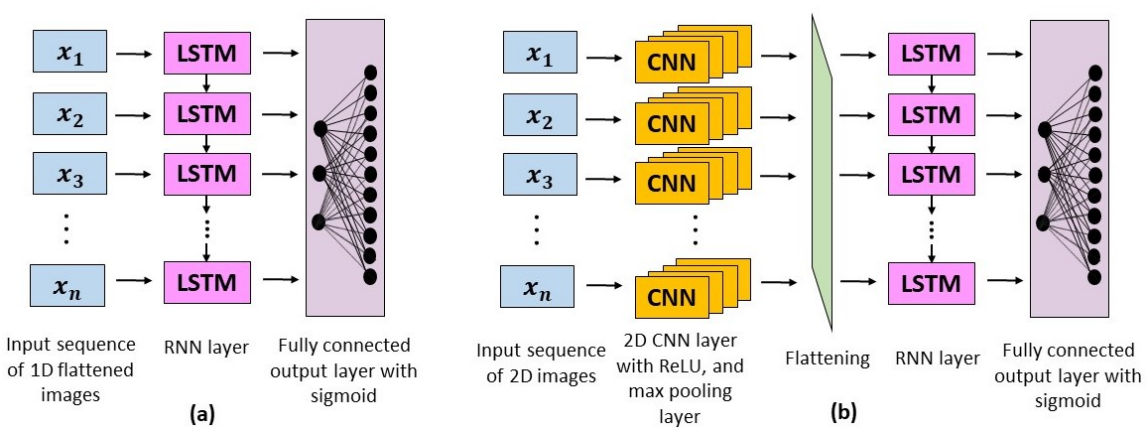

Figure 3.3: Network architecture overview for (a) LSTM-only, and (b) CNN-LSTM networks used in this analysis.

LSTM-only	
Learning rate (α)	100
Training epochs	160
Input dimension	8880
LSTM output features	100
Output dimension	8880
CNN-LSTM	
Learning rate (α)	100
Training epochs	160
Input dimension	37x240
Convolution kernel	3x3
CNN output features	50
Max pooling	10x10
LSTM output features	100
Output dimension	8880

Table 3.1: Neural network parameters.

The parameters used in the architecture and training of the RNNs are shown in Table 3.1. The PyTorch library was employed to implement the RNNs. The training set for each training iteration (epoch) was all 160 printed layers except for one layer, and the left-out layer was different for each of the 160 epochs. This was done as an attempt at leave-one-out cross-validation (LOOCV), though it was not actually LOOCV since it was done during the initial training phase and not as validation. True LOOCV is done as a validation step after training a ML model by further splitting a test data set not used to train the model into a new training set and a single test data point. The model is then trained on the new training set and tested on the test point, the process is repeated so that all points in the test data set have a chance to be the test data point, and the average test MSE becomes the test MSE for the pre-validation ML model. Due to this misunderstanding of LOOCV, validation could not be performed on the final RNN models because all data was used at some point during their initial training, leaving no test set.

Chapter 4

Results and Interpretation

The hardware used for this analysis was a laptop with an 11th Gen Intel(R) Core(TM) i7-1165G7 2.80GHz processor. Programs were written in Visual Studio Code using Python 3.10.2.

Overall, there were more porosities in the lower layers of the printed parts compared to the upper layers, which seems to mainly be due to the boundary falling away in the upper layers, giving the walls a tapered shape (see Figure 2.5). There was not much variability in number, location, and size of porosities across printed parts for the same layer, which was unexpected, as we were assuming the melt pool temperature distribution was the main driver of porosity formation in this experiment. However, we also found that the largest number of porosities in the CT scans were located between the boundary and fill, especially in the first several layers. This finding suggests that the greatest factor in porosity formation in these parts was actually the fill/boundary overlap. The lack of overlap between melt pools on the edge and interior of the printed parts likely created spaces of low energy density, causing lack-of-fusion porosities [9]. The lack of fill/boundary overlap can be seen in the printed part design itself in Figure 2.3 (a). Fill/boundary overlap should probably

be changed from 0% to something closer to 50% in future print iterations. Once this is fixed, voids caused by stochastic variability of the melt pool should dominate, and differences in the number of porosities between parts printed with different laser powers should become more apparent.

4.1 Reachability

The safety probabilities $r_{x_0}(\mathcal{K}, \mathcal{T})$ for each location in a printed layer were determined using the 270W state trajectory as initial states x_0 so that each safety probability could be associated with a point in the reference trajectory. Linear interpolation was used to fill in the rest of the locations in the layer that lie within the convex hull of the reference trajectory. The safety probabilities at the end of a layer trajectory cannot be reliably established as the they are computed based on likely subsequent states that lie within the next printed layer. In this case, the end of the trajectory is always near the (x,y) position (1mm,19mm) (see Figure 2.7), so predictions near this location should be ignored. The bandwidth of the Gaussian kernel, σ, dictates how far apart two data points can be and still have significant overlap in the tails of their kernel functions. Values of σ less than 50 resulted in undefined likelihoods because our data points, the state vectors, were so different from each other. Thus, a large σ of 1000 was used in this analysis.

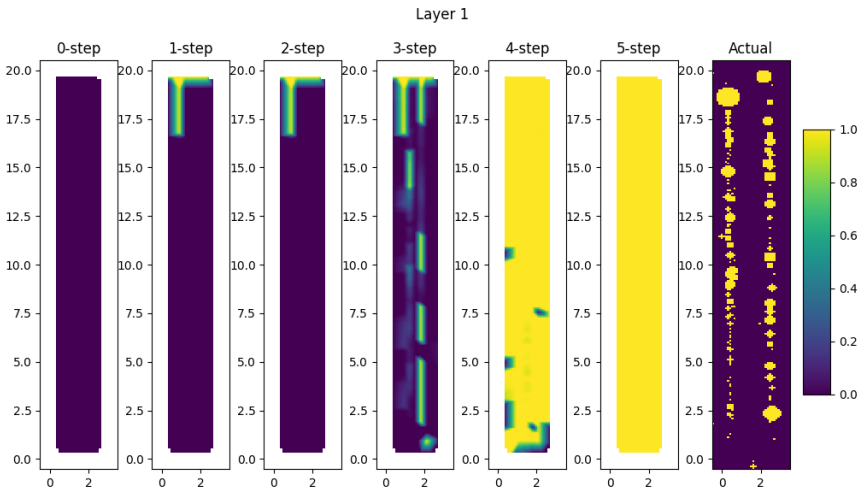

Figure 4.1: Porosity likelihood maps determined via stochastic reachability analysis for first printed layer alongside actual porosities from CT scan (far right). Images used as initial conditions for reachability were from the first layer of the 270W part, and CT scan porosities are from the same layer. Lighter colors indicate higher likelihood of reaching an unsafe condition, and therefore higher likelihood of defects, within the specified number of time steps.

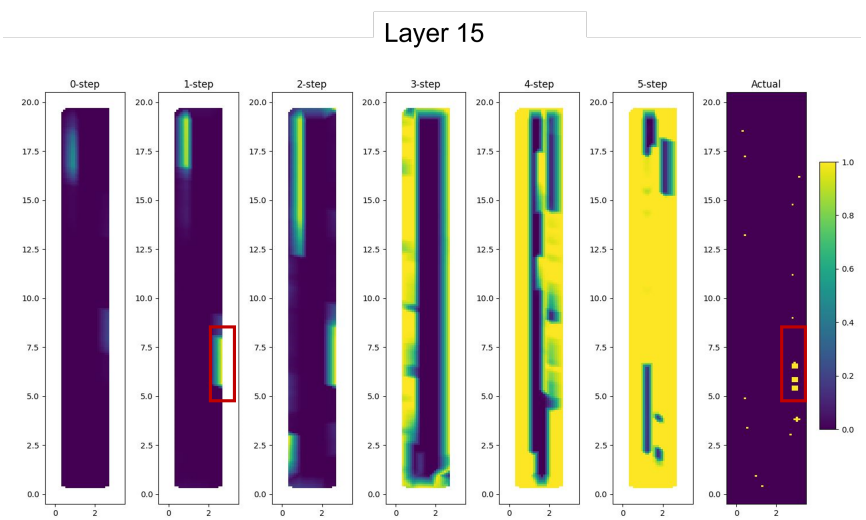

Figure 4.2: Porosity likelihood maps determined via stochastic reachability analysis for fifteenth printed layer alongside actual porosities from CT scan (far right). Images used as initial conditions for reachability were from the fifteenth layer of the 270W part, and CT scan porosities are from the same layer. Red boxes highlight an area of predicted porosities that match with actual porosities.

As seen in Figures 4.1 and 4.2, the safety probabilities start out at zero for almost all locations, and increase monotonically with increasing prediction horizon. The prediction horizon for the FHT stochastic reachability analysis was set to 10 time steps ($\frac{1}{3}$ sec), but after 5 time steps the likelihood of reaching the target set was almost always 1 for all locations, and so predictions past 5 time steps are not shown. We presume that porosities are likely to form in areas with higher likelihood of reaching the unsafe target set in the shortest number of time steps. Between steps 3 and 4, almost all locations go from low porosity likelihood to high porosity likelihood. We hypothesize that this is due to the large σ value, which could dictate that any initial state is close enough to the target set to reach it within 3 or 4 time steps. We also see a phenomena in Figures 4.1 and 4.2 of likelihoods streaking backward in time, which is expected for the first hitting time problem.

In Figure 4.1, none of the predicted porosity locations, shown in yellow at prediction time steps 1 and 2, match up with the actual porosity locations from CT scans of printed layer 1 of the 270W wall. We can see that most of the true porosities, at least for this layer, are occurring along the gap between fill and boundary tracks (see Figure 2.7), but that predicted porosities are all internal. We hypothesize that this is due to the absence of data points within the fill/boundary gap, and that the dynamics of observed temperatures are not relevant to the formation of porosities occurring along a gap between tracks. Figure 4.2 shows the reachability analysis and true porosity locations at printed layer 15, higher up in the part where the boundary has mostly fallen away and only internal porosities remain, and as expected the reachability analysis is better at predicting the locations of likely porosities as early as the 1-step prediction.

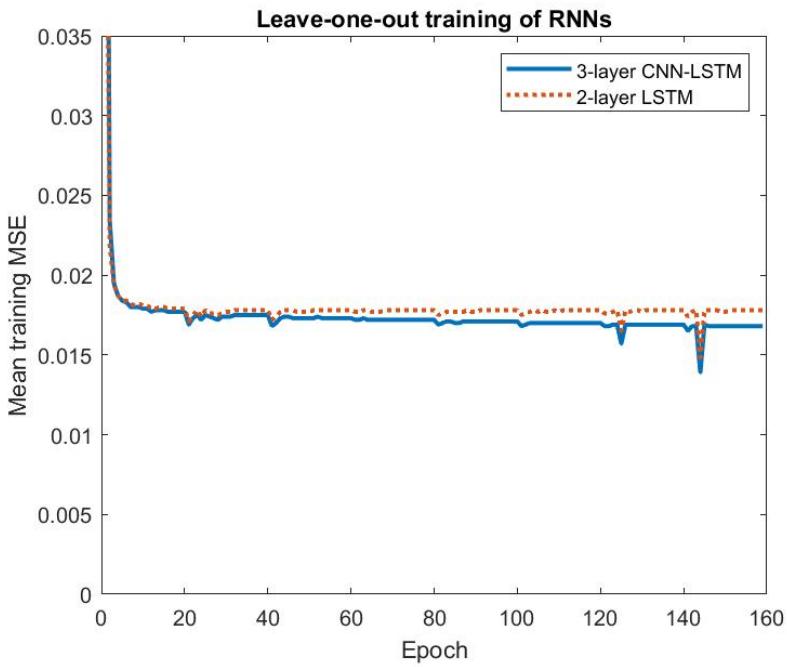

Figure 4.3: Depiction of training MSE over training epochs for the LSTM-only and CNN-LSTM algorithms discussed in this work. Sharp valleys in this plot likely correspond to epochs where an outlier layer was left out of the training set.

4.2 RNN

NN	Accuracy (training MSE)	Training time	Training data
LSTM-only	0.0178	hours	160 printed layers, 481 images/layer
CNN-LSTM	0.0168	days	160 printed layers, 481 images/layer

Table 4.1: Scale of NN training.

Overall, the RNNs trained in this experiment were able to predict the true locations of porosities fairly well, and certainly much better than the reachability analysis did. The LSTM-only model trained as described in Section 3.2 for 160 epochs had a

final average training mean squared error (MSE) of 0.0178. The CNN-LSTM model similarly trained got down to an average training MSE of 0.0168. A comparison of training errors over all training iterations is shown graphically in Figure 4.3, and a summary of the training scale and relevant results in Table 4.1. The CNN-LSTM model took significantly longer to train (days) than the LSTM-only model (hours) for limited benefit in terms of MSE. The MSE of the CNN-LSTM model appears to still be slowly decreasing in MSE at epoch 160, indicating that the learning rate was set correctly, and that further training may slightly improve these results. The MSE for the LSTM-only model stagnates quickly after the first few epochs, indicating that the learning rate might be too high. However, decreasing the learning rate from 100 to 10 after the first 160 epochs and training for another 160 epochs did not cause any decrease in MSE for the LSTM-only model.

As an example, predicted results for layer 2 of the 340W wall are shown in Figure 4.4. This figure provides visual confirmation that the probable porosity locations given by both RNNs are indeed well aligned with the true porosity locations. Compared to the CNN-LSTM model, the LSTM-only model seemed to have larger uncertainty in size and location of porosities, as evidenced by the larger lower-likelihood areas in its prediction. In Figure 4.5, the additional example of CNN-LSTM predicted results for layer 8 of the 300W wall serves to demonstrate the accuracy and variety of prediction outputs for different printed layer melt pool sequence inputs.

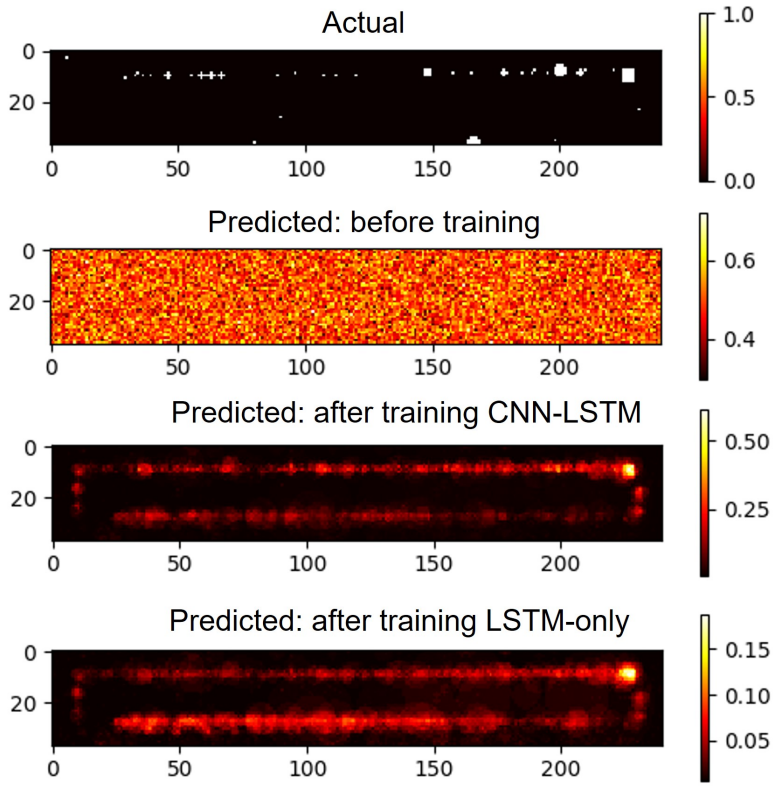

Figure 4.4: Predicted porosity likelihoods using both RNNs, tested on part layer 2 of the 340W wall. MSE for this data point went from 0.25 before training to 0.0122 after training using the CNN-LSTM algorithm and 0.0130 after training using the LSTM-only algorithm.

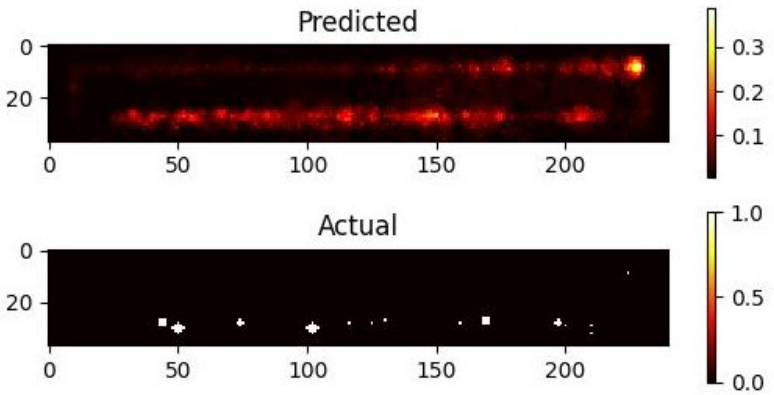

Figure 4.5: Actual versus CNN-LSTM predicted porosities for layer 8 of the 300W printed wall.

Chapter 5

Conclusions

Additively manufactured parts tend to have more consistency and materials reliability issues than traditionally manufactured parts, and this is especially true for DED. This work introduced two possible data-driven, dynamics-focused methods for flaw prediction in DED printed components; stochastic reachability and RNNs. As it stands, the reachability analysis based on melt pool observations did not provide as many accurate insights into porosity formation as the RNN, but this method should not be ruled out yet. Tweaking the reachability problem formulation, for example establishing a better target set of unsafe conditions, changing the value of the kernel bandwidth σ, or changing to the terminal hitting time problem may drastically change these results. LSTM RNNs with and without convolutional layers, trained on melt pool images with CT scan porosity locations as labels, showed promise in anticipating the regions of porosity present in individual printed layers. This work could help inform future additive manufacturing solutions by providing a metric of part quality without the use of long, expensive CT scans or destructive characterization methods. Importantly, since melt pool data is collected real-time while the part is printing, it could allow future DED control strategies to predict upcoming porosities and correct printing parameters such as laser power before the porosity

occurs.

There are several areas where this research could be improved upon and extended. Accuracy and efficiency of data collection could be greatly improved by the additions of a position sensor on the print head, or another way to export the print head's current position in real-time, and a trigger for the melt pool sensor to start or stop recording based on the laser status. Placing a small registration mark at the origin of each part that is visible after CT scanning, for example with a contrast ink, could help facilitate accurate alignment for a consistent spatial reference frame. Modeling the laser spot based on a measured laser power intensity distribution instead of a 3D step function could provide a more accurate input for reachability analysis. Lack of standardization in data collection and a shortage of publicly available datasets is perhaps largest obstacle to effective data-driven process control for many types of AM [24], but especially for DED. Robust ML algorithms will need large, standardized datasets for training and validation if they are ever going to generalize beyond the narrow, specific cases of the type presented in this thesis. The National Institute of Science and Technology's (NIST) additive manufacturing material database (AMMD), launched in 2017, attempts to remedy this by providing a reference data structure and repository of datasets [25], however it remains quite sparse, and a quick search for DED yields no results. Continued work on this project may involve incorporating publicly available data, if amenable data can be found, and uploading experimental datasets to databases like AMMD.

Future work should include refinement of the neural network structures by optimizing parameters such as the learning rate, number of neurons in the LSTM hidden layer, and number of CNN output features. Adding a second set of convolutional and max pooling layers to the CNN-LSTM network could help with recognizing finer melt pool features, or adding a dropout layer could help the network to generalize better when faced with data it's never seen before. Additional image processing to

eliminate extraneous information outside of the melt pool may also improve ML algorithms. Similarly, full exploration of the reachability analysis parameters should be performed before ruling out this approach. Modifying the safety and target datasets, adding more data used to calculate the kernel, and changing the kernel bandwidth σ are among the unexplored possibilities. As for control, spatial iterative learning control (SILC) of the type proposed in [16] seems promising for AM applications and has yet to be applied to DED processes. Ultimately, finding a way to integrate physics-based models of the dynamics at various temporal and spatial scales with data-based models of uncertainty seems like the most promising route for potential breakthroughs in control of DED.

References

[1] O. Whang, "The first 3-d printed rocket fails shortly after launch," *The New York Times*, Mar. 23, 2023. [Online]. Available: https://www.nytimes.com/2023/03/23/science/relativity-space-launch-terran.html.

[2] Y. Chen, S. Clark, A. C. L. Leung, *et al.*, "Melt pool morphology in directed energy deposition additive manufacturing process," *IOP Conference Series: Materials Science and Engineering*, vol. 861, no. 1, p. 012 012, May 2020. DOI: 10.1088/1757-899X/861/1/012012. [Online]. Available: https://dx.doi.org/10.1088/1757-899X/861/1/012012.

[3] S. Mishra, "Control of additive manufacturing," in *Encyclopedia of Systems and Control*, J. Baillieul and T. Samad, Eds. London: Springer London, 2019, pp. 1–7, ISBN: 978-1-4471-5102-9. DOI: 10.1007/978-1-4471-5102-9_100146-1. [Online]. Available: https://doi.org/10.1007/978-1-4471-5102-9_100146-1.

[4] Z. Smoqi, B. D. Bevans, A. Gaikwad, *et al.*, "Closed-loop control of meltpool temperature in directed energy deposition," *Materials & Design*, vol. 215, p. 110 508, 2022, ISSN: 0264-1275. DOI: https://doi.org/10.1016/j.matdes.2022.110508. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0264127522001290.

[5] P. Akbari, F. Ogoke, N.-Y. Kao, *et al.*, "Meltpoolnet: Melt pool characteristic prediction in metal additive manufacturing using machine learning," *Additive Manufacturing*, vol. 55, p. 102 817, 2022, ISSN: 2214-8604. DOI: https://doi.org/10.1016/j.addma.2022.102817. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214860422002172.

[6] M. Khanzadeh, S. Chowdhury, M. Tschopp, H. Doude, M. Marufuzzaman, and L. Bian, "In-situ monitoring of melt pool images for porosity prediction in directed energy deposition processes," *IISE Transactions*, vol. 51, Dec. 2017. DOI: 10.1080/24725854.2017.1417656.

[7] J. Ning, D. E. Sievers, H. Garmestani, and S. Y. Liang, "Analytical modeling of part porosity in metal additive manufacturing," *International Journal of Mechanical Sciences*, vol. 172, p. 105 428, 2020, ISSN: 0020-7403. DOI: https://doi.org/10.1016/j.ijmecsci.2020.105428. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002074031932689X.

[8] International Organization for Standardization, "Additive manufacturing — General principles — Terminology," Vernier, Geneva, Switzerland, Tech. Rep., 2018. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-astm:52900:dis:ed-2:v1:e.

[9] A. Dass and A. Moridi, "State of the art in directed energy deposition: From additive manufacturing to materials design," *Coatings*, vol. 9, no. 7, 2019, ISSN: 2079-6412. DOI: 10.3390/coatings9070418. [Online]. Available: https://www.mdpi.com/2079-6412/9/7/418.

[10] L. Ladani and M. Sadeghilaridjani, "Review of powder bed fusion additive manufacturing for metals," *Metals*, vol. 11, no. 9, 2021, ISSN: 2075-4701. DOI: 10.3390/met11091391. [Online]. Available: https://www.mdpi.com/2075-4701/11/9/1391.

[11] M. Armstrong, H. Mehrabi, and N. Naveed, "An overview of modern metal additive manufacturing technology," *Journal of Manufacturing Processes*, vol. 84, pp. 1001–1029, 2022, ISSN: 1526-6125. DOI: https://doi.org/10.1016/j.jmapro.2022.10.060. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1526612522007459.

[12] M. Liu, A. Kumar, S. Bukkapatnam, and M. Kuttolamadom, "A review of the anomalies in directed energy deposition (ded) processes and potential solutions - part quality and defects," *Procedia Manufacturing*, vol. 53, pp. 507–518, 2021, 49th SME North American Manufacturing Research Conference (NAMRC 49, 2021), ISSN: 2351-9789. DOI: https://doi.org/10.1016/j.promfg.2021.06.093. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978921001189.

[13] Q. Wang, J. Li, M. Gouge, A. R. Nassar, P. (Pan) Michaleris, and E. W. Reutzel, "Physics-Based Multivariable Modeling and Feedback Linearization Control of Melt-Pool Geometry and Temperature in Directed Energy Deposition," *Journal of Manufacturing Science and Engineering*, vol. 139, no. 2, Sep. 2016, 021013, ISSN: 1087-1357. DOI: 10.1115/1.4034304. eprint: https://asmedigitalcollection.asme.org/manufacturingscience/article-pdf/139/2/021013/6405073/manu_139_02_021013.pdf. [Online]. Available: https://doi.org/10.1115/1.4034304.

[14] P. M. Sammons, D. A. Bristow, and R. G. Landers, "Two-Dimensional Modeling and System Identification of the Laser Metal Deposition Process," *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 2, Oct. 2018, 021012, ISSN: 0022-0434. DOI: 10.1115/1.4041444. eprint: https://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/141/2/021012/6129245/ds_141_02_021012.pdf. [Online]. Available: https://doi.org/10.1115/1.4041444.

[15] P. M. Sammons, M. L. Gegel, D. A. Bristow, and R. G. Landers, "Repetitive process control of additive manufacturing with application to laser metal deposition," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 566–575, 2019. DOI: 10.1109/TCST.2017.2781653.

[16] D. J. Hoelzle and K. L. Barton, "On spatial iterative learning control via 2-d convolution: Stability analysis and computational efficiency," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1504–1512, 2016. DOI: 10.1109/TCST.2015.2501344.

[17] S. Summers and J. Lygeros, "Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem," *Automatica*, vol. 46, no. 12, pp. 1951–1961, 2010, ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2010.08.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109810003547.

[18] "SOCKS: A kernel-based stochastic optimal control and reachability toolbox," in *Proceedings of the 25th ACM International Conference on Hybrid Systems: Computation and Control*, 2022.

[19] T. Williamson, *G-code reader*, 2023. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/67767-g-code-reader.

[20] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st. USA: John Wiley & Sons, Inc., 1994, ISBN: 0471619779.

[21] M. Hagan, H. Demuth, and M. Beale, *Neural Network Design*. Vikas Publishing House, 2003, ISBN: 9789812403766. [Online]. Available: https://hagan.okstate.edu/NNDesign.pdf.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artificial Intelligence Review*, vol. 53, Dec. 2020. DOI: 10.1007/s10462-020-09838-1.

[24] S. Di Cataldo, S. Vinco, G. Urgese, *et al.*, "Optimizing quality inspection and control in powder bed metal additive manufacturing: Challenges and research directions," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 326–346, 2021. DOI: 10.1109/JPROC.2021.3054628.

[25] Y. Lu, P. Witherell, and M. Donmez, "A collaborative data management system for additive manufacturing," en, 37th Computers and Information in Engineering Conference (CIE), Cleveland, OH, Aug. 2017. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=923075.