

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/329714582>

Predicting Large Domain Multi-Physics Fire Behavior Using Artificial Neural Networks

Thesis · December 2018

CITATIONS

0

READS

93

1 author:



[Jonathan Hodges](#)

Virginia Polytechnic Institute and State University

8 PUBLICATIONS 3 CITATIONS

[SEE PROFILE](#)

Predicting Large Domain Multi-Physics Fire Behavior Using Artificial Neural Networks

Jonathan L. Hodges

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

Brian Y. Lattimer, Co-chair

Tomonari Furukawa, Co-chair

Jeffrey T. Borggaard

Scott W. Case

Kevin B. Kochersberger

Danesh K. Tafti

November 16, 2018

Blacksburg, Virginia

Keywords: Wildland, Structure, Fire, Artificial, Neural, Network, Convolutional, CNN

Copyright 2018, Jonathan L. Hodges

Predicting Large Domain Multi-Physics Fire Behavior Using Artificial Neural Networks

Jonathan L. Hodges

(ABSTRACT)

Fire dynamics is a complex process involving multi-mode heat transfer, reacting fluid flow, and the reaction of combustible materials. High-fidelity predictions of fire behavior using computational fluid dynamics (CFD) models come at a significant computational cost where simulation times are often measured in hours, days, or even weeks. A new simulation method is to use a machine learning approach which uses artificial neural networks (ANNs) to represent underlying connections between data to make predictions of new inputs. The field of image analysis has seen significant advancements in ANN performance by using feature based layers in the network architecture. Inspired by these advancements, a generalized procedure to design ANNs to make spatially resolved predictions in multi-physics applications is presented and applied to different fire applications. A deep convolutional inverse graphics network (DCIGN) was developed to predict the two-dimensional spatially resolved spread of a wildland fire. The network uses an image stack corresponding to the spatially resolved landscape, weather, and current fire perimeter (which can be obtained from measurements) to predict the fire perimeter six hours in the future. A transpose convolutional neural network (TCNN) was developed to predict the spatially resolved thermal flow field in a compartment fire from coarse zone fire model predictions. The network uses thirty-five parameters describing the geometry of the room and the ventilation conditions to predict the full-field temperature and velocity throughout the room. The data for use in training and testing both networks was generated using high-fidelity CFD fire simulations. Overall, the ANN predictions in each network agree with simulation predictions for validation scenarios. The computational time to evaluate the ANNs is 10,000x faster than the high-fidelity fire simulations. This work represents a first step in developing super real-time full-field fire predictions for different applications.

Predicting Large Domain Multi-Physics Fire Behavior Using Artificial Neural Networks

Jonathan L. Hodges

(GENERAL AUDIENCE ABSTRACT)

The National Fire Protection Agency estimates the total cost of fire in the United States at \$300 billion annually. In 2017 alone, there were 3,400 civilian fire fatalities, 14,670 civilian fire injuries, and an estimated \$23 billion direct property loss in the United States. Large scale fires in the wildland urban interface (WUI) and in large buildings still represent a significant hazard to life, property, and the environment. Researchers and fire safety engineers often use computer simulations to predict the behavior of a fire to assist in reducing the hazard of fire. Unfortunately, typical simulations of fire scenarios may take hours, days, or even weeks to run which limits their use to small areas or sections of buildings. A new method is to use a machine learning approach which uses artificial neural networks (ANNs) to represent underlying connections between data to make new predictions of fire behavior. Inspired by advancements in the field of image processing, this research developed a procedure to use machine learning to make rapid high resolution predictions of fire behavior. An ANN was developed to predict the perimeter of a wildland fire six hours in the future based on a set of images corresponding to the landscape, weather, and current fire perimeter, all of which can be obtained directly from measurements (US Geological Survey, Automated Surface Observation System, and satellites). In addition, an ANN was developed to predict high-resolution temperature and velocity fields within a floor of a building based on predictions from a coarse model. The data for use in training and testing these networks was generated using high-resolution fire simulations. Overall, the network predictions agree well with simulation predictions for new scenarios. In addition, the time to run the model is 10,000x faster than the typical simulations. The work presented herein represents a first step in developing high resolution computer simulations for different fire scenarios that run very quickly.

Dedication

Dedicated to my family.

Charity

For her willing sacrifice while I pursued this work.

Igou and Linda

For always supporting me and encouraging me to pursue excellence.

Daniel and Rebekah

For opening your home to us when we were in need.

Elizabeth

For your continuous loving support.

The Zimmermans

For treating us as family and opening your home to us.

Jay and Sherry

For helping my wife in her time of need.

Acknowledgments

I would like to express my sincere gratitude to everyone who has helped me in this process. Although there is no way to thank everyone who has helped me over the years, I would like to express gratitude to those who have provided significant support and guidance.

First I would like to thank each of the funding agencies who have helped support me in this research. The sponsorship from JENSEN HUGHES was instrumental to the development of the neural networks to predict wildland fire spread presented in Chapter 3 and to predict spatially resolved thermal flow fields presented in Chapter 4. The sponsorship from the National Institute of Occupational Safety and Health (200-2014-59669) was essential to the development of the neural network to predict spatially resolved thermal flow fields presented in Chapter 4. The support from the Office of Naval Research (N00014-13-1-0590 and N00014-13-0894 DURIP), Department of the Navy (N65540-10-C-0003), and the Naval Engineering Education Consortium (2002542411-UT) were instrumental in the experimental fire testing work discussed in Appendix B and Appendix C.

I would like to thank my advisers for helping me grow into the researcher I am today. Dr. Brian Lattimer thank you for introducing me to the field of fire safety engineering and your guidance throughout my doctoral work. I am very grateful you made sure there was always funding available to support me. Thank you for keeping me focused on finishing even when our research switched directions several times. I look forward to continuing to solve interesting problems. Dr. Tomonari Furukawa thank you for introducing me to the field of Bayesian robotics and opening my eyes to what is possible in autonomous systems. Thank you for letting me be involved in the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) and sponsoring my travel for the competition

in UAE. Even though there was a significant amount of work involved I appreciate the opportunity to see more of the world. Dr. Scott Case thank you for your support and advising during my first year as a doctoral student. Thank you for your continued interest in my work and advice even when our project work diverged. I would like to thank Dr. Alan Lattimer for the great conversations regarding reduced order modeling, neural networks, and life. Thank you for supporting my work while we were at JENSEN HUGHES together. I would like to thank my other committee members, Dr. Jeffrey Borggaard, Dr. Kevin Kochersberger, and Dr. Danesh Tafti for their time and advice on this research.

I would like to thank Robert Mills, Christian Rippe, Stefan Kraft, Joseph Starr, and the other members of the Extreme Laboratory for their help with various sensor technologies, experimental design, and running experiments. Thank you for the camaraderie and great conversations (which were occasionally even work related). I would like to thank Maggie Goetz and Danny Fritsch for their help in preparing for and running experiments. I would like to thank the members of the Mitchell Robotics Laboratory and Computational Multiphysics Laboratory for helping me learn about robotics and autonomous systems.

I would like to thank the members of Heritage PCA in Columbia and Blacksburg Christian Fellowship for their support. Doctoral research is not easy in the best of times, and I appreciate your support through the more difficult times.

Contents

List of Figures	xiii
List of Tables	xx
1 Introduction	1
1.1 Motivation	1
1.2 Research Overview and Contributions	3
1.3 Outline	5
2 Multi-Physics Fire Modeling Using Artificial Neural Networks	6
2.1 Abstract	6
2.2 Introduction	7
2.3 Wall Fire Simulations	8
2.4 Data Model	10
2.4.1 Define Desired Outcomes	10
2.4.2 Define Performance Metrics	14
2.4.3 Identify Impacting Factors	16
2.4.4 List Available Data	16
2.4.5 Limit Scope and Select Input Data	17

2.4.6	Gather, Generate, Combine Data	18
2.4.7	Encoding and Pre-Processing Data	19
2.4.8	Split Data	23
2.5	Predictive Model	24
2.5.1	Neural Network Fundamentals	24
2.5.2	Neural Network Development	34
2.6	Wall Fire ANN Development	43
2.6.1	Data Model Development	43
2.6.2	Predictive Model Development	45
2.6.3	Wall Fire ANN Results	47
2.7	Conclusion	47
3	Wildland Fire Spread Modeling Using Convolutional Neural Networks	49
3.1	Abstract	49
3.2	Introduction	50
3.3	Methods	53
3.3.1	Wildland Fire Prediction	53
3.3.2	Network Architecture	57
3.3.3	Performance Metrics	59
3.3.4	Post Processing	60

3.4	Results	60
3.5	Discussion	63
3.6	Conclusion	75
4	Compartment Fire Predictions Using Transpose Convolutional Neural Networks	77
4.1	Abstract	77
4.2	Introduction	78
4.3	Methods	81
4.3.1	Network Architecture	81
4.3.2	CFD Data Generation	85
4.3.3	Zone Fire Model Data Generation	88
4.3.4	Post Processing	90
4.4	Results	91
4.5	Discussion	100
4.6	Conclusion	107
4.7	Acknowledgements	109
5	Conclusions and Future Work	110
5.1	Summary	110
5.2	Conclusions	113
5.3	Future Work	114

Bibliography 117**Appendix A Fire Modeling Using Computational Fluid Dynamics** 140

A.1 Foreword	140
A.2 Introduction	140
A.3 Summary of Conservation Equations in Low-Mach Number Flows	141
A.4 Summary of Two-Dimensional Rothermel Predictions	144

Appendix B Thermographic Digital Image Correlation (TDIC) Measurements of Mechanically Loaded Structures 148

B.1 Foreword	148
B.2 Abstract	148
B.3 Introduction	149
B.4 TDIC Setup	150
B.5 Fire in Field of View Testing	151
B.6 I-Beam Testing Using Ideal Gray-scale Histogram	155
B.7 Conclusion	158
B.8 Acknowledgements	158

Appendix C Resolving Obstructed Views Using Multiple Thermography Digital Image Correlation Systems 161

C.1 Foreword	161
------------------------	-----

C.2	Abstract	162
C.3	Introduction	162
C.4	Experimental Methods	163
C.4.1	TDIC Calibration	164
C.4.2	Test Articles	164
C.4.3	Data Stitching Process	168
C.4.4	Testing Conditions	170
C.5	Numerical Methods	170
C.6	Results	172
C.7	Conclusion	176
C.8	Acknowledgements	179

Appendix D	Predicting the Structural Response of a Compartment Fire using Full-field Heat Transfer Measurements	180
D.1	Foreword	180
D.2	Abstract	180
D.3	Introduction	181
D.4	Experimental Methods	183
D.4.1	Heat Flux Mapping Experiments	183
D.4.2	Structural Collapse Experiments	188
D.5	Simulations	190

D.5.1	FDS Model	190
D.5.2	Abaqus Thermal Model	191
D.5.3	Abaqus Mechanical Model	192
D.5.4	Simulation Matrix	194
D.6	Results	194
D.7	Discussion	203
D.8	Conclusion	204
D.9	Acknowledgements	205

List of Figures

2.1	Overview of artificial neural network design process.	8
2.2	Wall fire example simulation geometry	9
2.3	Example q_0'' predictions from CFD software	11
2.4	Schematic of data model development process	12
2.5	Example methods to represent the spread of a wildland fire	20
2.6	Schematic of a perception, x 's are inputs, w 's are weights, B is the bias, y is the raw output, and z is the activation function output.	25
2.7	Comparison of popular activation functions. Note the slope on the negative side of leaky ReLU has been exaggerated for illustration purposes.	26
2.8	Sample network architecture with one input layer, two hidden layers, and one output layer.	28
2.9	Example 2-D convolution operation.	29
2.10	Schematic of predictive model development process, Phase 1.	35
2.11	Schematic of predictive model development process, Phase 2.	36
2.12	Example feed-forward network architecture with two trained hidden layers and one dropout layer.	38
2.13	Example convolutional neural network architecture with four trained hidden layers and one dropout layer.	38

2.14 Example transpose convolutional neural network architecture with six trained hidden layers and one dropout layer.	39
2.15 Example deep convolutional inverse graphics network architecture with eight trained hidden layers and one dropout layer.	39
2.16 Example recurrent neural network architecture with two trained hidden layers, one dropout layer, and one recurrent operation.	40
2.17 Example recurrent deep convolutional inverse graphics network architecture with eight trained hidden layers, one dropout layer, and one recurrent operation.	40
2.18 Neural network predicted peak q_0'' compared with peak q_0'' for testing data-set.	47
3.1 Schematic of solution algorithm. The left set of images show the different channels used as inputs to the neural network. The values for each data channel are colorized based on the values shown in Table 3.1 and Table 3.2.	54
3.2 Example Simulated Burn Maps	56
3.3 Deep Convolutional Inverse Graphics Network Architecture.	58
3.4 Mean F-Measure of DCIGN predictions of 9,000 training simulations (27,000 pairs of burn maps) for different threshold values.	61
3.5 Example DCIGN prediction of burn map probability	62
3.6 Example DCIGN prediction results	64
3.7 F-Measure distribution of DCIGN predictions of test cases.	65
3.8 Distribution of input fire size for all cases where $F < 0.8$	67
3.9 Example DCIGN prediction of time resolved burn maps	69

3.10 Example predicted burn map from FARSITE.	71
3.11 Cumulative probability of computational time using FARSITE	71
3.12 Example heterogeneous landscape and vegetation.	73
3.13 Example DCIGN prediction results	74
3.14 F-Measure distribution of DCIGN predictions of heterogeneous test cases	75
4.1 Schematic of solution algorithm.	81
4.2 Example compartment fire simulation	83
4.3 Schematic of the TCNN architecture used in this work.	84
4.4 Schematic of two-room configuration. Predictions of temperature and velocity in Room 1 were used in training the model.	86
4.5 Standard deviation of error from training data with different standard deviation kernels for Gaussian filter.	90
4.6 TCNN predictions of x-axis center-line temperature and velocity compared to CFD simulations for Testing Case 1.	93
4.7 TCNN predictions of y-axis center-line temperature and velocity compared to CFD simulations for Testing Case 1.	94
4.8 TCNN predictions of z-axis ceiling temperature and velocity 0.1m below the ceiling compared to CFD simulations for Testing Case 1.	95
4.9 TCNN predictions of x-axis center-line temperature and velocity compared to CFD simulations for Testing Case 2.	96

4.10 TCNN predictions of y-axis center-line temperature and velocity compared to CFD simulations for Testing Case 2	97
4.11 TCNN predictions of z-axis ceiling temperature and velocity 0.1m below the ceiling compared to CFD simulations for Testing Case 2	98
4.12 Discrete probability density functions of TCNN error from training data set	99
4.13 Discrete probability density functions of TCNN error from testing data set	99
4.14 Geometry from multi-room configuration validation studies. The red square is the source fire position	102
4.15 Comparison of TCNN (left) and CFD (right) predictions in Validation Case 1. The white borders correspond to obstructions to the flow.	103
4.16 Validation Case 1 comparison of change in error with distance from the fire.	104
4.17 Comparison of vertical slices TCNN (left) and CFD (right) predictions in Validation Case 2. The white borders correspond to obstructions to the flow	105
4.18 Comparison TCNN (left) and CFD (right) predictions of temperature 0.1m below the ceiling in Validation Case 2. The white borders correspond to obstructions to the flow	106
4.19 Comparison TCNN (left) and CFD (right) predictions of U-velocity 0.1m below the ceiling in Validation Case 2. The white borders correspond to obstructions to the flow	106
4.20 Comparison TCNN (left) and CFD (right) predictions of V-velocity 0.1m below the ceiling in Validation Case 2. The white borders correspond to obstructions to the flow	107

B.1	Schematic of fire in FOV experiment	151
B.2	Sample images and gray-scale histograms with and without fire exposure	153
B.3	Performance metrics of measurement from sample under various contrast levels with and without fire exposure	154
B.4	Boundary sample gray-scale histograms showing viable contrast levels for TDIC measurements with fire in the FOV from exposure times identified in Fig. B.3 . . .	156
B.5	Percentage of data loss with fire in the FOV using typical TDIC setup and the recommendations presented in this work for various fire sizes	156
B.6	Example images of a structural I-beam test from different cameras	157
B.7	Example of deformation and temperature data fusion from a structural I-beam testing for the red line slice shown in Fig. B.6a at various times throughout testing .	159
B.8	Example of deformation measurements of structural I-beams exposed to 100 kW fire imaged with different exposure settings	160
C.1	Sample stereo pair of calibration images from the primary TDIC system	165
C.2	Sample simultaneous captured image from each camera in a TDIC system used for rigid body transform	166
C.3	Measurement surfaces on T-stiffened aluminum plate for two TDIC systems	167
C.4	Painted surface for TDIC measurements	168
C.5	Same measurement surface viewed from one CCD camera in each TDIC system. The boxed region was used to create the coordinate system transform.	169
C.6	Representative geometries and grid resolution in numerical modeling	171

C.7	Full-field temperature mapped to displaced geometry	173
C.8	Full-field temperature mapped to displaced geometry for a test article exposed to a peak heat flux of 35 kW/m^2 at various times throughout testing	175
C.9	Location of line slices for comparisons shown in Fig. C.10 and Fig. C.11	176
C.10	Experimental measurements of deflection and temperature of a test article exposed to a peak heat flux of 35 kW/m^2 at various times throughout testing	177
C.11	Numerical measurements of deflection and temperature of a test article exposed to a peak heat flux of 35 kW/m^2 at various times throughout testing	178
D.1	Experimental setup for the heat flux mapping experiments	184
D.2	Compartment heat flux testing IR camera perspectives	186
D.3	Experimental setup for the heat flux mapping experiments	187
D.4	Structural collapse experiment compartment setup prior to testing	189
D.5	Non-uniform mesh in Abaqus. The four red circles show the uniform pressure loading points	193
D.6	Inverse heat transfer measurements of net heat flux with $h_{int} = 35 \text{ W/m}^2 - K$, $h_{ext} = 10 \text{ W/m}^2 - K$	196
D.7	Inverse heat transfer measurements of standard heat flux with $h_{int} = 35 \text{ W/m}^2 - K$, $h_{ext} = 10 \text{ W/m}^2 - K$	197
D.8	Comparison of net heat fluxes from IHT method and FDS at various h	198
D.9	Comparison of gas temperatures from heat flux testing with FDS predictions	199

D.10 Comparison of surface temperature predictions with experimental results at two points	200
D.11 Temperature in [K] of back wall after 300s of fire exposure	201
D.12 Comparison of in-plane deflection predictions (y-direction in [mm]) with experimental measurements just prior to compartment collapse	202

List of Tables

2.1	Number of outcomes for an ANN to predict with different combinations of spatial and temporal resolution. The raw data spatial resolution is $0.1m$ and temporal resolution is $1s$	14
2.2	Baseline neural network architectures.	37
3.1	Limits of each parameter in study.	55
3.2	Parameters for Example Simulation	57
3.3	Performance Metrics of DCIGN Predictions of Test Cases	63
3.4	Impact of Noise on Performance Metrics of DCIGN Predictions of Test Cases . . .	68
3.5	Performance Metrics of DCIGN Predictions of heterogeneous Test Cases	72
4.1	Description of points in input vector.	85
4.2	Limits of each parameter in study.	87
4.3	Summary of performance of TCNN predictions on training and test data sets, values correspond to $\mu \pm \sigma$	92
4.4	Summary of performance of TCNN predictions on validation cases, values correspond to $\mu \pm \sigma$	108
4.5	Comparison of computational time for CFD and TCNN predictions of validation cases.	108

Chapter 1

Introduction

1.1 Motivation

The National Fire Protection Agency estimates the total cost of fire in the United States at \$300 billion annually [1]. In 2017 alone there were 3,400 civilian fire fatalities, 14,670 civilian fire injuries, and an estimated \$23 billion direct property loss in the United States [2]. The focus of fire protection engineering is to mitigate the impact of fire on everyday life. Unfortunately, large fires still represent a significant hazard to life, property, and the environment.

Fire dynamics is a complex process involving multi-mode heat transfer, reacting fluid flow, and the reaction of combustible materials. A fire releases heat to the environment which heats the combustion products resulting in a buoyancy driven plume. The released energy is transported to the environment through convection and radiation. Energy which is transferred to fuel in the environment (solid or liquid) may result in the gasification of more fuel which may ignite. Accurately resolving these processes over a large domain is not possible with current computational technology. As a result, large domains are currently solved with simplified models at low resolution. These simplified models are typically developed for specific applications, such as building and wildland fires, but lack the resolution to accurately quantify fire behavior.

The majority of deaths and property damage from fire occurs in structure fires such as buildings or homes [2]. Understanding the transport of energy and combustion products in a compartment fire is

vital in fire hazard analysis. Computational fluid dynamics (CFD) fire models are often used in fire protection engineering to predict complex flow fields for smoke control, estimate smoke detector and sprinkler activation times, and predict ignition and flame spread over combustible surfaces. However, the high spatial-temporal resolution comes at a computational cost, where simulation times are often measured in hours, days, or even weeks depending on the simulation. More rapid predictions over large domains are possible with zone based compartment fire models which apply the transport equations to a single grid point per room in a structure. As a result, there is a trade-off between spatial-temporal resolution and computational cost in structure fire modeling.

The impact of wildland fires on society is expected to continually increase with climate change and increased population of the Wildland Urban Interface (WUI) [3, 4, 5, 6]. Modeling wildland fires presents several unique challenges, mostly related to the large domain of interest (often multiple kilometers) and the fact that fire spread can be dependent on conditions on the meter scale. Ambient parameters such as air temperature, relative humidity, and wind velocity are typically only available on a coarse grid. Quantifying the availability of specific fuel sources and moisture content is typically not possible over a large domain in fine detail. Resolving the thermal flow field using high fidelity approaches such as computational fluid dynamics (CFD) requires a fine grid which is not feasible for large domains. As a result stochastic and phenomenological models are often used by forestry personnel when planning for and responding to emergencies. The applicability of these simplified models to a new fire scenario is directly related to how similar the conditions are to those for which the model was developed.

An alternative method to model fires over large domains that has not been fully explored is to use a machine learning approach, which uses simplified functions to represent underlying connections between data to make estimates of future events, often using artificial neural networks (ANNs). A key advantage of data-driven approaches is the simplified functions can be rapidly evaluated to make new predictions. This type of approach is primarily limited by the amount of data and

computational requirements to train the ANN. Researchers from various fields have successfully applied data-driven approaches to predict multi-physics applications, including wildland fire spread [7, 8, 9, 10, 11, 12, 13], storm surge [14, 15, 16, 17, 18], flood inundation [19, 20, 21, 22, 23, 24, 25], climate modeling [26, 27, 28, 29], remote sensing [30, 31, 32], and power generation [33, 34, 35, 36, 37]. Although ANNs have been used in multi-physics applications, the predictions have primarily been limited to estimates of a single quantity at a single point. The need in fire predictions is full-field conditions (such as temperature, velocity, and pressure) which is the focus of this research.

1.2 Research Overview and Contributions

This research focuses on advancing the state-of-the art in fire modeling using data-driven artificial neural networks to predict full-field fire conditions. The field of image classification and generation has seen significant advancements in network performance by using convolutional and transpose convolutional layers in the network architecture. Inspired by these advancements, a generalized procedure to design neural networks to make spatially resolved predictions in multi-physics applications is presented and applied to different fire applications. The developed networks for each application make spatially resolved predictions using either point data or spatially resolved data as inputs. The data for use in training and testing the networks was generated using CFD fire simulations. The contributions of this research include:

- Design framework for developing an ANN to predict full-field conditions in new scenarios
- ANN for WUI fire spread with full-field input data and full-field output predictions
- ANN for structure fires with point inputs from a coarse zone model to predict full-field outputs

The first aspect of the work focused on a methodology to design artificial neural networks for predicting fire conditions. Although several authors have discussed the benefits of various network architectures for different applications, few discuss the coupled development of the data model and the predictive model for a new application. Often the optimal network architecture for a specific application will depend on the form of the input and output data. This work discusses the fundamentals of the data model and predictive model in detail. The type of data, format, and encoding of data are addressed and the implications on the required network capacity discussed. Recommendations of baseline network architectures to use for different input and output data types are presented for use in physics based predictions. A recommended process for manual tuning of the network hyper-parameters is presented in the context of overall network performance and generalization error. The method is demonstrated and used to predict the standard heat flux on a wall from a fire.

An artificial neural network was then created to predict the spread and full-field burn pattern of a wildland fire. The network uses an image stack consisting of 13 full-field images corresponding to the spatially resolved landscape, weather conditions, and current burned area to predict the fire perimeter six hours in the future. Data for use in training and testing the model was generated at six hour time intervals using the phenomenological fire spread model of Rothermel with 10,000 different combinations of input parameters as well as more complex terrain and weather conditions with FARSITE. The ANN was trained and the generalization of the approach examined with a test set of simulations.

An artificial neural network was also created to predict thermal flow fields in a structure fire from coarse zone fire model predictions. Data for use in training and testing the model was generated using the computational fluid dynamics (CFD) software Fire Dynamics Simulator (FDS v6.2) of a simple two-compartment configuration with different fire locations, fire sizes, ventilation configurations, and compartment geometries. Although the model was trained and tested using a

simple two-compartment configuration, the network approach was validated with two more complex multi-compartment FDS simulations by processing each room individually.

This research represents a significant advancement in the field of data-driven fire modeling by using new machine learning approaches adopted from other fields. Prior to the work presented herein, data-driven approaches in fire modeling were limited to predictions of individual points of data. Using the techniques developed in this research, spatially resolved neural network predictions of important fire parameters are possible. Together this represents a first step in creating a framework in developing super real-time fire predictions across multiple applications.

1.3 Outline

This document is organized as a series of papers with each chapter representing one of the research contributions. Chapter 2 presents a general methodology to develop a neural network to predict fire conditions ranging from point quantities to full-field conditions. Chapter 3 includes the research on developing a novel neural network approach to predict the full-field spread of a wildland fire using a convolutional neural network (CNN). Chapter 4 contains a neural network approach to predict spatially resolved temperatures and velocities in a structure fire based on coarse zone fire model conditions using a transpose convolutional neural network (TCNN). Chapter 5 includes conclusions from the research and recommendations on future research directions.

Chapter 2

Multi-Physics Fire Modeling Using Artificial Neural Networks

2.1 Abstract

Advancements in computational resources and machine learning algorithms in recent years have led to an increase in the usage of artificial neural networks (ANNs). Researchers from various fields have successfully applied ANNs to predict single point quantities in multi-physics applications, including wildland fire spread, storm surge, flood inundation, and climate modeling. The field of image classification and generation has seen significant advancements in network performance by using convolutional and transpose convolutional layers; however, these network architectures have not been widely used in the physics community. Although several authors have discussed the benefits of various network architectures for different applications, few discuss the coupled development of the data model and the predictive model for a new application. Often the optimal network architecture for a specific application will depend on the form of the input and output data. This paper presents a general methodology to develop a neural network to predict quantities in new applications. The fundamentals of the data model and predictive model are discussed in detail and applied to develop an ANN to make regression predictions in a supervised learning context. The type of data, format, and encoding of data are addressed and the implications on the required network capacity are discussed. Recommendations of baseline network architectures to

use for different input and output types are presented. A process for manual tuning of the network hyper-parameters is presented in the context of overall network performance and generalization error. As an example, a new neural network to predict the standard heat flux on a wall from an adjacent fire is developed using the methodology developed in this paper. The predictions from the network are within 10% error of the values in the database for 95% of scenarios.

2.2 Introduction

Advancements in computational resources and machine learning algorithms in recent years have led to an increase in the usage of artificial neural networks (ANNs). Researchers from various fields have successfully applied ANNs to predict single point quantities in multi-physics applications, including wildland fire spread [7, 8, 9, 10, 11, 12, 13], storm surge [14, 15, 16, 17, 18], flood inundation [19, 20, 21, 22, 23, 24, 25], climate modeling [26, 27, 28, 29], remote sensing [30, 31, 32], and power generation [33, 34, 35, 36, 37]. The field of image classification and generation has seen significant advancements in network performance by using convolutional and transpose convolutional layers; however, these network architectures have not been widely used in the physics community. Several authors have discussed the benefits of various network architectures for different applications in Refs. [23, 30, 38, 39]; however, few discuss the coupled development of the input and output data model and the predictive model for a new application. Constructing an ANN architecture that considers the problem inputs and outputs is needed to ensure successful results.

At a fundamental level, ANNs are massively parallel equations which have the capability to store observed knowledge about a problem to make predictions of new inputs. Formally, ANNs can be represented by the form

$$Y = f(X, W) + \epsilon \quad (2.1)$$

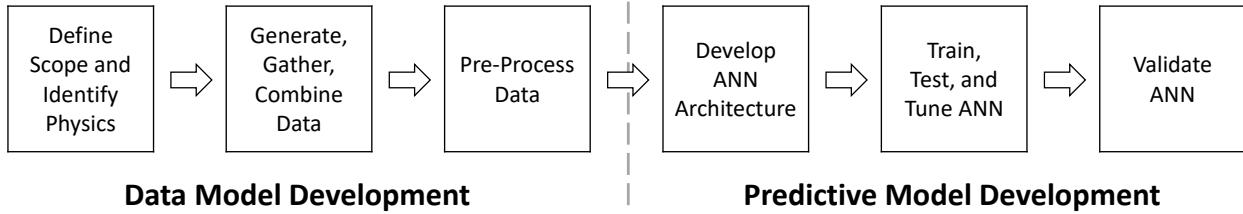


Figure 2.1: Overview of artificial neural network design process.

where Y is the vector of model outputs, X is the vector of model inputs, W is the set of learned parameters, ϵ is model error, and f is the functional form of the network [23]. Significant consideration of each component in Eq. 2.1 is needed to develop an ANN for a new multi-physics application. Herein the type of data, format, and encoding of data in X and Y is referred to as the data model. The type of layers, number of neurons, training, and testing of f is referred to as the predictive model. An overview of the design process is shown in Fig. 2.1.

This paper describes the process to develop an ANN for a new application in detail. The overall design process is split into two phases: Data Model Development and Predictive Model Development. The goal of the Data Model Development phase is to define the overall goal of the ANN, identify the relevant physics which must be captured, and gather and pre-process the data. The goal of the Predictive Model Development phase is to develop, train, test, and validate the ANN architecture. This paper describes each stage of the development process in detail. The process is applied to develop an ANN to predict the standard heat flux entering a vertical wall exposed to a fire. The following sections describe the fire simulations, the data model, the predictive model, and the development of an ANN architecture for the fire example.

2.3 Wall Fire Simulations

The process discussed in this work is applied to develop an ANN to predict the standard heat flux entering a vertical wall exposed to a fire. A schematic of the scenario is shown in Fig. 2.2. A gas

burner is located adjacent to the vertical wall, with the size of the burner and mass flow rate of gas adjusted to change the severity of the fire. The training, testing, and validation data used in this work were generated using Fire Dynamics Simulator (FDS) version 6.2, which is a computational fluid dynamics (CFD) software package often used in fire protection engineering [40].

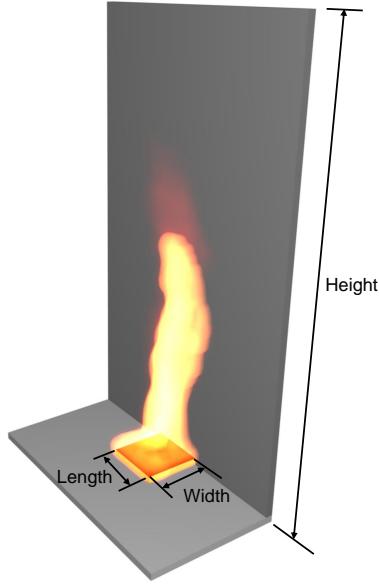


Figure 2.2: Wall fire example simulation geometry

The standard heat flux is defined as the heat flux to a surface at a standard temperature (298 K in this work), which is a useful metric in fire protection engineering for defining thermal exposure since it does not scale with wall surface temperature. The standard heat flux is calculated using the equation

$$q_0'' = \epsilon (q_{inc,rad}'' - \sigma T_0^4) + h (T_g - T_0) \quad (2.2)$$

where $q_{inc,rad}''$ is the incoming radiative heat flux, q_0'' is the standard heat flux, h is the convective heat transfer coefficient, ϵ is the surface emissivity, σ is the Stefan-Boltzmann constant, T_g is the gas temperature, and T_0 is the standard reference temperature (298 K in this work) [40]. Since q_0'' is independent of wall surface temperature, the exposure is a function of the buoyancy driven plume generated by the flame. An example prediction of q_0'' on the vertical wall from the CFD software is

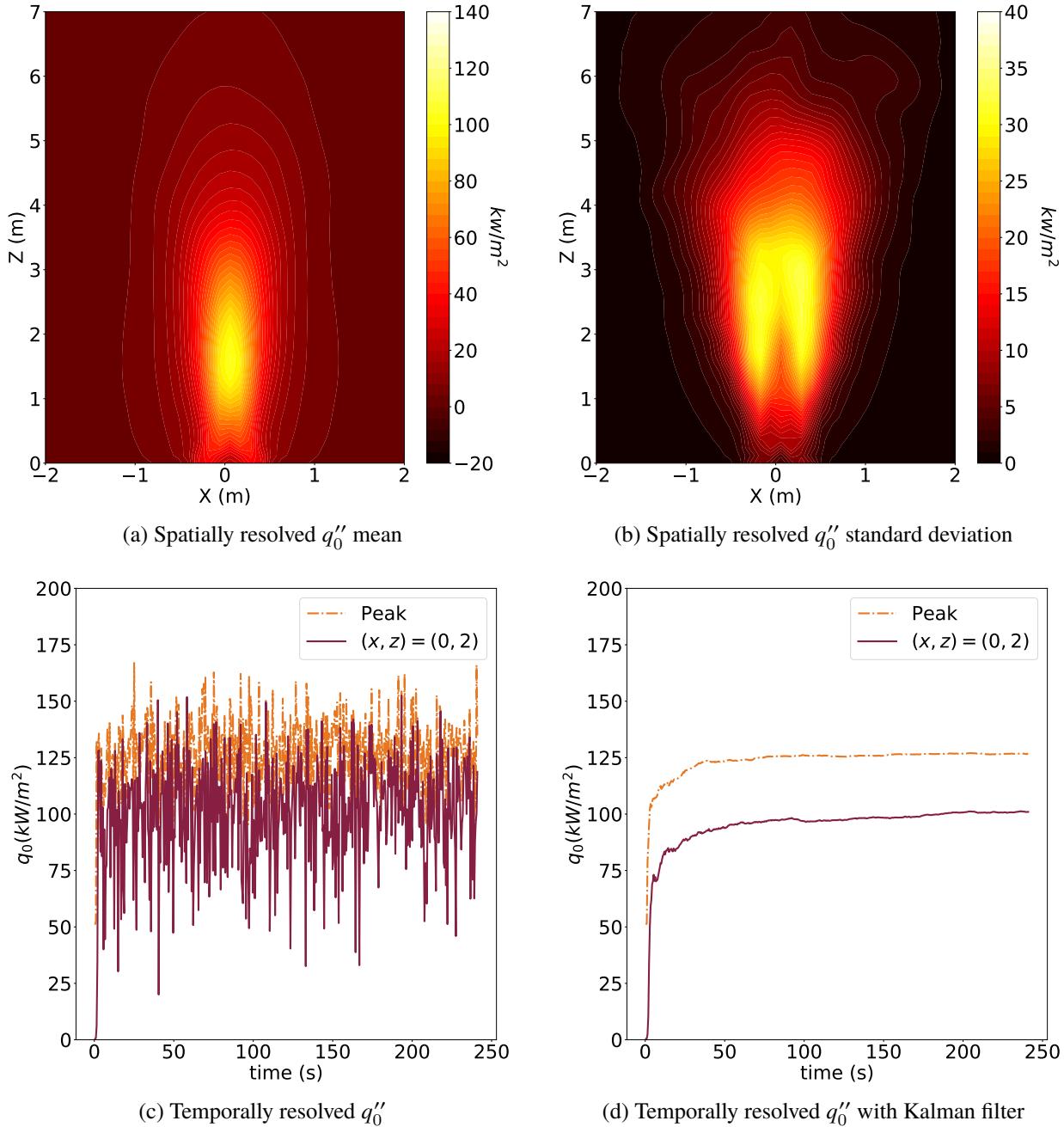
shown in Fig. 2.3. Since the fire plume is turbulent, the gas temperature adjacent to the wall varies significantly which results in large fluctuations in q_0'' with respect to time, as shown in Fig. 2.3b and Fig. 2.3c; however, the q_0'' mean stabilizes after approximately 50 seconds, as shown in Fig. 2.3a and Fig. 2.3d.

2.4 Data Model

The data model is the type of data, format, and encoding of the ANN inputs and outputs. A schematic of the data model development process is shown in Fig. 2.4. The tasks in the development process generally fall into two phases. The planning phase consists of the tasks associated with examining the details of the scenario and identifying the key pieces of data. These tasks include defining the desired outcomes from the ANN, defining how the performance of the ANN will be measured, identifying the factors which impact each quantity to be predicted, listing the available data (can be experimental, simulation, or a combination), and limiting the scope of the ANN and selecting the input data. The preparation phase consists of the tasks associated with obtaining, encoding, and splitting data. The following subsections describe each step of the development process.

2.4.1 Define Desired Outcomes

The first step in this process is to identify the desired outcomes. In the context of machine learning, there are two primary types of outcomes: classification and regression. Classification outcomes identify where the input parameters fall within a pre-defined classification scheme, such as identifying an object in an image or identifying the type of burning fuel based combustion products. The typical question the ANN is trying to answer is to which class do these inputs belong. Regression outcomes are predictions of numeric quantities such as temperature, velocity,

Figure 2.3: Example q''_0 predictions from CFD software

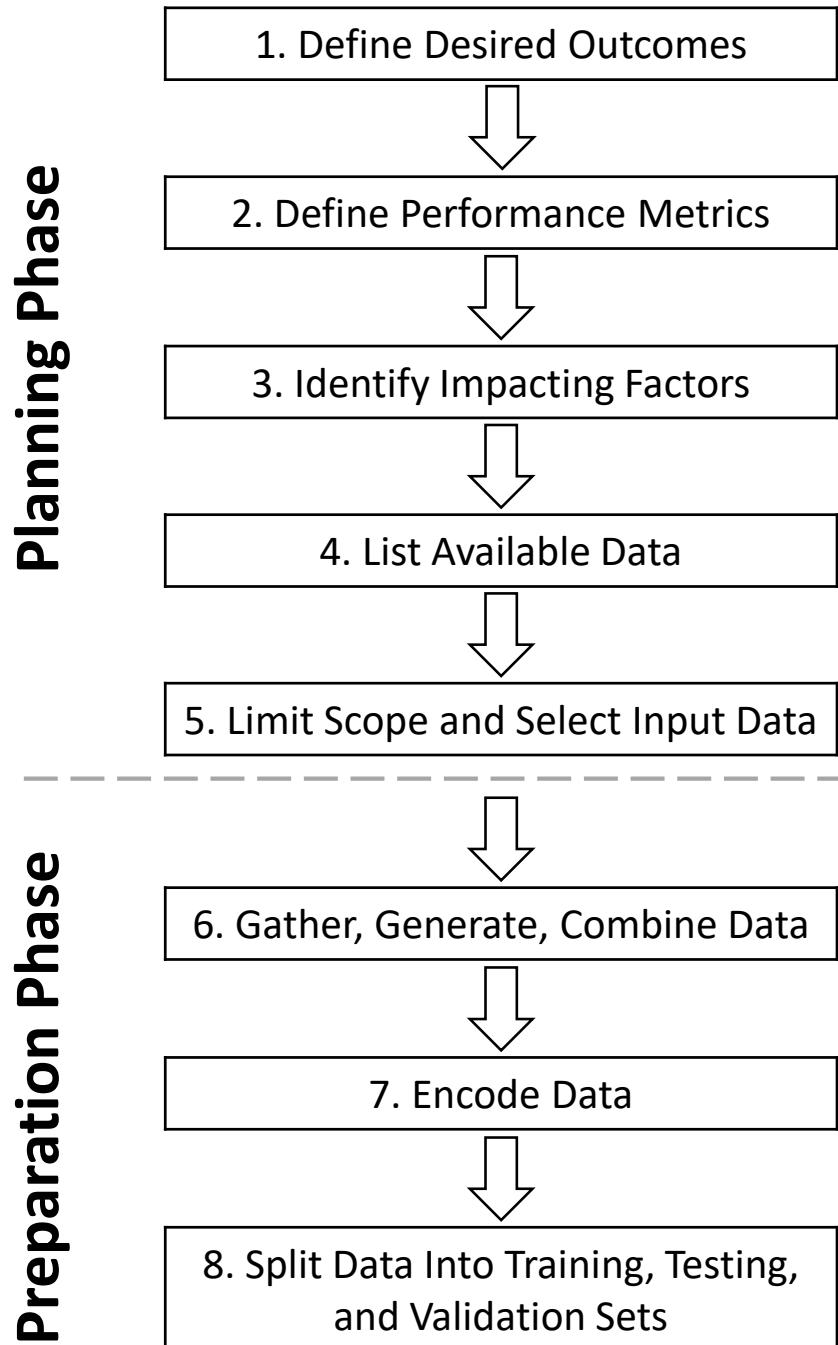


Figure 2.4: Schematic of data model development process

pressure, flow rate, or flame height. The typical question the ANN is trying to answer is based on these inputs, what is the magnitude of the output. As a general rule, it is easier to develop an ANN for classification than regression.

It is sometimes possible to cast a regression problem as a classification problem. Predicting the value of q_0'' in the wall fire example is a regression problem; however, depending on the end goal of the ANN, it may be possible to cast it as a classification task. For example, consider the end goal of the ANN prediction is to determine if a material on the wall will ignite based on the critical heat flux of ignition, q_{cr}'' , (the lowest heat flux at which a material will ignite under piloted ignition). The output data can then be replaced with a binary classifier with the value 0 when $q_0'' < q_{cr}''$ and 1 otherwise. At this stage, each outcome should be identified as either a classification or regression task.

The desired spatial and temporal dimensionality to be predicted must be defined for each outcome. In general, the complexity to setup the ANN and the data required to train will increase as the spatial and temporal resolution is increased. Recall the spatial and temporal variation in q_0'' shown in Fig. 2.3. Depending on the end goal of the ANN, there are many options of spatial and temporal resolutions which may be of interest. For example, three spatial resolutions that may be of interest are: q_0'' at a specific point, q_0'' peak across the surface, and spatially resolved q_0'' . Similarly, three temporal resolutions that may be of interest are: q_0'' at a specific time, the time at which q_0'' first exceeds a threshold, and the temporally resolved q_0'' . A matrix showing the total desired predictions of the ANN for each combination of these three spatial and temporal resolutions is shown in Table 2.1. In Table 2.1 the columns correspond to increasing complexity in the time dimension, and the rows correspond to increasing complexity in the spatial dimension. The total number of predictions to be made increase as the outcome is more spatially and temporally resolved.

Table 2.1: Number of outcomes for an ANN to predict with different combinations of spatial and temporal resolution. The raw data spatial resolution is $0.1m$ and temporal resolution is $1s$.

Spatial Resolution	Time Resolution		
	q_0'' at $t = 50s$	t at $q_0 > 200$	$t = 0 - 120s$
$q_0(0, 1)$	1	1	120
$q_{0, Peak}$	1	1	120
$q_0(x, y)$	3,200	3,200	384,000

2.4.2 Define Performance Metrics

It is important to identify at an early stage how the performance of the ANN will be quantified for each desired outcome. These metrics can be different for each outcome and typically are different than the cost function used to train the ANN [39]. Generally different performance metrics are used for regression and classification tasks. The optimal performance metric will depend on the specific outcomes; however, there are some standard metrics which are a good baseline for each type of task. The desired level of performance for each outcome should be determined after selecting the performance metric for each outcome. The total data needed to train the network will increase for higher levels of performance.

Performance in binary classification tasks is often measured in accuracy. Accuracy is defined as

$$A = \frac{t_p + t_n}{t_p + f_p + t_n + f_n} \quad (2.3)$$

where A is the accuracy, t_p is the correctly identified positive results, t_n is correctly identified negative results, f_p is incorrectly identified positive results, and f_n is incorrectly identified negative results. This metric performs well for tasks where t_p and t_n have similar levels of occurrence. However, accuracy is not useful when predicting the occurrence of rare events (when t_p and t_n have different levels of occurrence) as high accuracy can be achieved by never predicting a rare event to occur. This difficulty can be avoided by using more advanced performance metrics such

as precision, sensitivity, and F-measure [39]. The precision is a measure of commission errors (predicting a positive value when it should have been negative) and is defined as

$$P = \frac{t_p}{t_p + f_p} \quad (2.4)$$

where P is the precision. The sensitivity is a measure of omission errors (predicting a negative value when it should have been positive) and is defined as

$$S = \frac{t_p}{t_p + f_n} \quad (2.5)$$

where S is the sensitivity. F-measure is an overall measure of performance defined as the harmonic mean of P and S ,

$$F = 2 \cdot \frac{P \cdot S}{P + S} \quad (2.6)$$

where F is the F-measure. Together P , S , and F highlight the overall performance of the ANN as well as capture systematic commission and omission errors.

Error in regression tasks is typically calculated in the form

$$E = \left[\frac{100^b}{N} \sum_{i=1}^N \left| \frac{(x_t - x_{ANN})^a}{x_t^b} \right| \right]^{1/a} \quad (2.7)$$

where E is the error, N is the total number of predictions, x_t is the true value, x_{ANN} is the ANN predicted value, and the parameters a and b change the type of error. The most common types of error are the root mean square error, E_{RMSE} ($a = 2$ and $b = 0$), the mean absolute error, E_{MAE} ($a = 1$ and $b = 0$), and the mean absolute percent error, E_{MAPE} ($a = 1$ and $b = 1$). Which type of error is best will depend on the specific goal of a problem. For example, E_{MAE} or E_{MAPE} should be used if the error contribution from all points should be weighed equally; however, E_{RMSE} should be used if the error from outliers should be emphasized over other points. There are occasions where

understanding the tendency of the ANN to under-predict or over-predict can be useful, which can be achieved by removing the absolute value term in Eq. 2.7 such that

$$E = \left[\frac{100^b}{N} \sum_{i=1}^N \frac{(x_t - x_{ANN})^a}{x_t^b} \right]^{1/a} \quad (2.8)$$

which yields the mean error, E_{ME} ($a = 1$ and $b = 0$), and mean percent error, E_{MPE} ($a = 1$ and $b = 1$).

2.4.3 Identify Impacting Factors

Most ANNs do not have any physical laws embedded in the network architecture, (although this is an active field of research [41, 42, 43, 44, 45, 46, 47, 48, 49]). Thus the predictive capability of the ANN will be determined by how well the mechanics of the system are represented by the input data. Impacting factors can be identified from equations based on first principles, prior sensitivity studies, or domain knowledge. The sensitivity of the model and limiting the scope of inputs is discussed in Section 2.4.5.

2.4.4 List Available Data

The next step in the development process is to identify the available data. Available data can be from experiments, simulations, or a combination of both. Experimental data is generally preferable as it contains all physics of the problem. However, it is difficult to fully explore the parameter space (e.g. material properties are difficult to vary). Additionally, spatial resolution is typically fairly limited as the cost to fully resolve the spatial variation can be extensive. Simulation data is generally easier to use as there are less limitations on the parameter space and resolution. One limitation on using data generated from simulations is the computational time required.

If simulation data is to be used, it is important at this stage to determine the minimum simulation (or the lowest computational time simulation) which will generate the data necessary for a single sample. Some approaches to this may include limiting results to one-dimensional or two-dimensional, simplifying the problem domain, reducing simulation duration, increasing simulation time step, or making a mesh more coarse.

2.4.5 Limit Scope and Select Input Data

The next stage of this process is to reconsider the desired outcomes, parameter space, and available data as a unit to determine realistic limits for each potential input and for the scope of the ANN. The overall number of samples needed to train the ANN will be reduced by limiting the number of parameters and extents of each input parameter. Consider an ANN is to be developed to model the following equation

$$2x + y + z = w. \quad (2.9)$$

It is clear the ANN will be able to predict w if x , y , and z are input to the network. In this hypothetical scenario, x and y are easy to measure; however, z is not. Will an ANN trained with only x and y generate good predictions of w ? The answer is it will depend on the limits of x , y , and z . Consider the case where $z = 1$. During training the constant value of z will be embedded in the network and the ANN will be able to predict w . Consider the case where $z = x^2$. During training the quadratic relationship with x will be learned through the input of x and the ANN will be able to predict w . Consider the case where $z = 0 - 2$, $x = 0 - 10$, $y = 10 - 20$. Assuming no other training error, a maximum E_{MAPE} of 20% will occur when x and y are at their minimum values and z is at its maximum since the ANN has no information regarding the variation in z . If this is less than the target error threshold for the ANN, z can be safely neglected; otherwise it must be included. Unfortunately, it is difficult to know in advance how much each potential input will

impact the desired outcomes.

There are several approaches which have been used by researchers to determine the significance between potential inputs and desired outcomes [23]. In general, the approaches can be categorized as either Model Free or Model Based. Model Free approaches typically use a statistical approach such as linear regression or mutual information to examine the correlation between individual input parameters and desired outcomes. Model Based approaches train several ANNs with different combinations of input parameters to determine the optimal input parameter set. This process can be time consuming as the network architecture and hyper-parameters used in training should be optimized for each ANN. The independent impact of each parameter can also be assessed through a sensitivity study; however, this approach is typically limited to simulation data.

2.4.6 Gather, Generate, Combine Data

The next stage of this process is to collect the data that will be used in training, testing, and validating the ANN. Any new experiment or simulation needs to be run and the results compiled. Data from databases and the literature needs to be downloaded and consistently formatted. There are two methods experimental and simulation data can be combined for use in an ANN: aggregation and augmentation. In aggregation, each simulation and each experiment is considered as an independent sample. This is the preferred method as the total number of samples is higher and the experimental data can reduce the impact of modeling error on the ANN predictions. Unfortunately, measurements in experiments are typically fairly limited which can severely limit the available input parameters. In augmentation, each experiment is simulated and the results of the simulation are used to replace missing data from experimental measurements. This approach bypasses the limitation of available input parameters. However, since the results are coupled into a single sample the total number of samples does not increase. In addition, generating a model to accurately predict each experiment

can be a time consuming process.

Regardless of which method is used it is important to ensure each input and output quantity represents the same physics. As an example, consider the three methods of describing the spread of a wildland fire shown in Fig. 2.5. Figure 2.5a shows a case where the spread of the fire is represented as a list of points along the fire perimeter. Figure 2.5b shows a case where the spread of the fire is represented as all the points which have been burned in the fire. Figure 2.5c shows a case where the spread of the fire is represented as all points which are currently on fire (the points in the center have burned to extinction). Wildland fire simulations typically track the perimeter as in Fig. 2.5a; whereas, satellite imagery of wildland fire spread are typically based on the actively burning areas as in Fig. 2.5c. If both sources of data are to be used in training an ANN, the data should first be pre-processed such that the physics is represented similarly between the two data sets.

2.4.7 Encoding and Pre-Processing Data

The next stage of this process is to encode the collected input and output data to make the ANN easier to train and improve performance. There are many different ways to encode data and the optimal method will depend on the specific application; however, this section describes a few common methodologies used in encoding data.

Standardization is the process of scaling the magnitude and variation in inputs and outputs to be similar. An ANN will learn faster with a standardized data-set since it will not need to learn how to scale each piece of data and can focus on learning the relationships between data. The most simple way to standardize the data is to adjust the mean to zero and variance to one using the equation

$$\hat{x}_i = \frac{x_i - \bar{x}_i}{x_{i,max} - x_{i,min}} \quad (2.10)$$

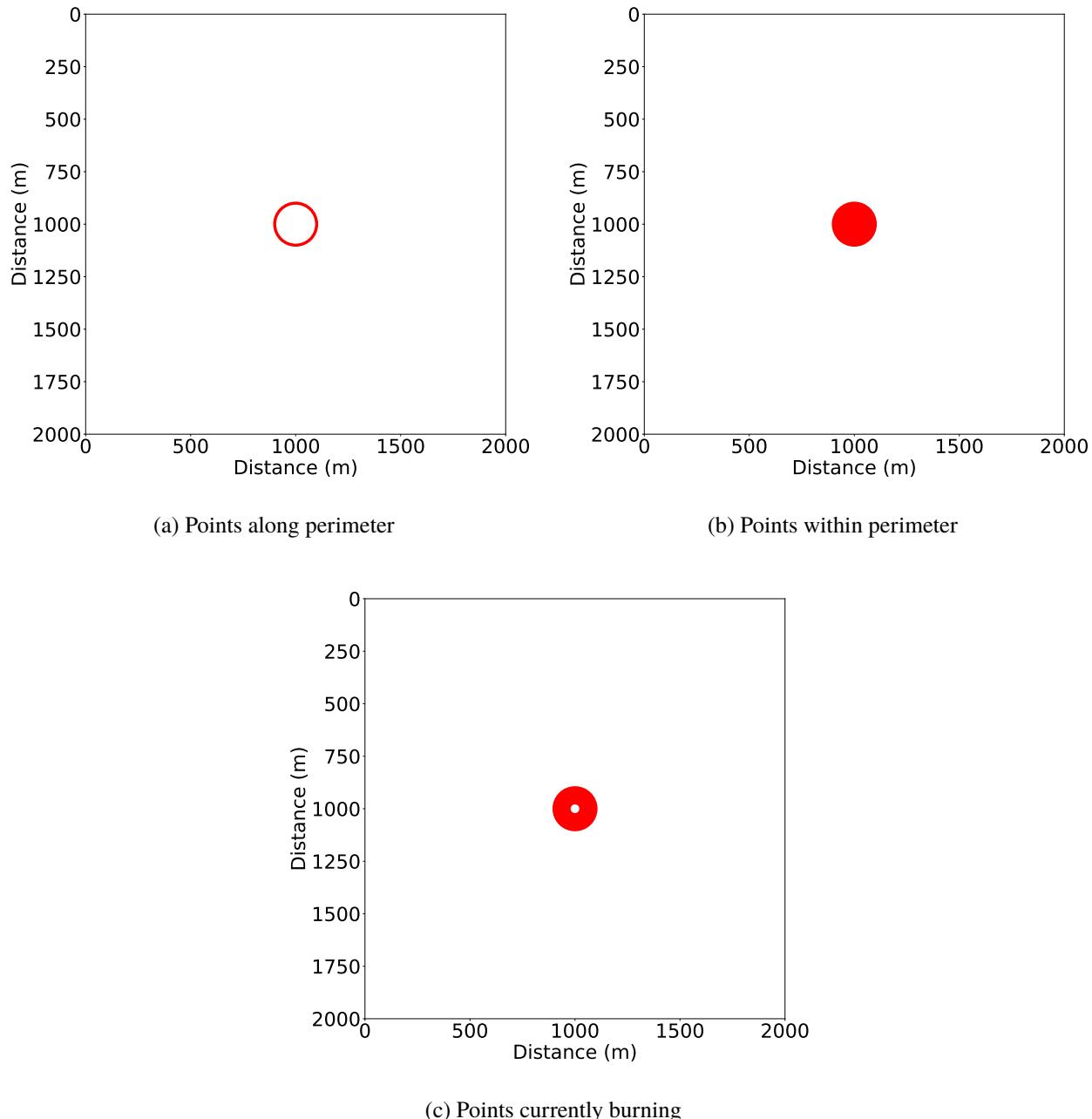


Figure 2.5: Example methods to represent the spread of a wildland fire

where the subscript i corresponds to each input and output, \hat{x}_i is the standardized data, x_i is the raw data, \bar{x}_i is the mean of the data, $x_{i,max}$ is the max of the data, and $x_{i,min}$ is the minimum of the data. The standardization parameters, \bar{x}_i , $x_{i,max}$, and $x_{i,min}$, can be set using only the training data or all of the data.

Organization is the process of sorting input and output parameters such that each input has a monotonic impact on each output. This can improve the performance of the ANN as organized categorical information provides information about its neighbors instead of only itself. For example, fuel model is a commonly used classification scheme in wildland fire modeling. Fuel model range from 90-256, with 90-99 corresponding to non burnable surfaces, 100-119 corresponding to grass models, 120-139 corresponding to grass-shrub models, 140-159 corresponding to shrub models, 160-179 corresponding to timber-understory, 180-199 corresponding to timber-litter, and 200-219 corresponding to slash-blowdown, and the rest unassigned [50]. This organization scheme is helpful for humans to sort like things together; however, in that form the impact of fuel model on the rate of spread of a wildland fire is not monotonic. Sorting fuel model by baseline rate of spread (no wind and no incline) conditions instead will improve the ANN by providing additional information (as the fuel model goes up, the baseline rate of spread goes up).

Substitution is the process of replacing some data values with others. This is useful when working with experimental data with missing data points. Often a missing value is set to negative one or zero. Depending on how much data is missing and how much other data is available providing data with some missing values may help the ANN learn more robust features. If experimental data with missing points is being combined with simulation data, repeating the same simulation data with features randomly removed can help the ANN learn more robust features as well.

Classification encoding (also called one hot encoding) is the process of replacing a value with an array of zeros and single one in the position corresponding to the value. This is typically used to replace a class ID number in a classification problem with a binary array. However, it is also

possible to use this process to cast a regression outcome as a classification problem. For example, if the end goal of the ANN prediction in the wall fire example is to determine if a material on the wall will ignite based on measured critical heat flux of ignition, q''_{cr} , (the lowest heat flux at which a material will ignite under piloted ignition), the output data can be replaced with a binary classifier with the value 0 when $q''_0 < q''_{cr}$ and 1 otherwise.

Periodic encoding is the process of replacing periodic variables such as angles with a different representation that is continuous at the boundaries. For example an angle of 359° and 0° are physically close to each other; however, in the current form the ANN does not know that. For this reason it is common practice to encode the angle with its sine and cosine. The encoding is then

$$\theta : (\theta'_1, \theta'_2) = (\sin(\theta), \cos(\theta)) \quad (2.11)$$

and the decoding is

$$\theta = \tan^{-1} \left(\frac{\sin(\theta'_1)}{\cos(\theta'_2)} \right) \quad (2.12)$$

where θ is the angle, θ'_1 and θ'_2 are the encoded sine and cosine of θ . This same procedure can be applied to other periodic data when the network should recognize the minimum and maximum values are close to one another.

Careful consideration needs to be given how to encode time when it is a factor in an ANN. Depending on the goal of the ANN, time may or may not be considered a periodic variable. Consider the case of wildland fire spread. If the timescale is always increasing such as what is the fire perimeter after 48 hours, time should not be periodic. However, consider if an ANN is developed to predict the surface temperature on a body of water. In that case, it makes more sense to encode time as periodic as the diurnal variation is largely dependent on solar radiation.

When dealing with data with multi-dimensional topology such as images, it is often necessary to re-size all data to be consistently sized. If convolutional layers and fully connected layers are both

being used the dimensions of the data must be consistent. There are a few different strategies which can be used to fix the dimensions across different samples. If the aspect ratio of the data are the same but the size is not the smaller data can be up-sampled using bi-linear interpolation or the larger data can be down-sampled. If the aspect ratio of the data are not consistent, the smaller data can be zero padded by adding zeros to the edges of the images, the smaller image can be up-sampled and then cropped, or the larger images can be cropped. Consider in the wall fire example where the width and height of the wall were varied. Since the size of the wall changes, the total number of points of q_0'' changes as well. The data can be re-sampled to a consistent grid based on the percentage of wall width and height.

2.4.8 Split Data

The last stage in developing the data model is to split the data-set into training, testing, and validation sets. Each of the three data sets has a different purpose. Training data is used directly in the training of the ANN, thus the weights and biases throughout the network are tuned to predict the training set. As a result, the ANN will have a tendency to over-fit the training data. The testing data is used to determine how much over-fitting is occurring during training. The generalization error in an ANN is the difference in error between testing and training data sets. If there is zero over-fitting, the generalization error will be close to zero. If the generalization error is high then the ANN is over-fitting the training data. If the generalization error is negative, either the ANN needs to train longer, the hyper-parameters of the training method need to change, or the network architecture needs to change. Since the network architecture is tuned to try and minimize the generalization error, it is possible that the optimal network architecture developed for the training and testing data will not work for other data. The validation data is used to check the performance of the ANN after training is complete to verify the ANN architecture will work with new data points.

The goal of data division is to have the same statistical properties between the three sets of data [23]. This can be done in either a supervised or unsupervised fashion. In supervised data division, the data is repeatedly shuffled and re-divided through either trial and error or some optimization scheme until the statistics of each set are similar [23]. In practice, unsupervised data division is more common where the raw data is split into three equally sized sets and assumed to be statistically similar. When dealing with randomly sampled simulation data this is a good assumption.

2.5 Predictive Model

The predictive model is the type of layers, number of neurons, training, and testing of the network architecture. A brief overview is provided on the fundamental principles of a neural network. Following this, an overview of the development process of a neural network is given.

2.5.1 Neural Network Fundamentals

This section gives a brief overview of the components in a neural network. At a fundamental level, artificial neural networks are massively parallel equations which have the capability to store observed knowledge about a problem to make predictions of new inputs. Recall the general form from Eq. 2.1. The network architecture is the functional form of f . The network architecture is typically organized into layers. The fundamental building block of the layers are perceptrons (often called neurons). The following subsections describe neurons, network architecture, common layers used in ANNs, training, and tuning of the capacity of the network. Much of this content was aggregated from [39, 51].

2.5.1.1 Perceptrons

A schematic of an example perceptron is shown in Fig. 2.6 [51]. The task of each perceptron is to weigh all the evidence available to it and make a single observation. To accomplish this task each perceptron is made up of multiple inputs with one weight per input, a single bias, and a single output. The mathematical representation of Fig. 2.6 is

$$w_1x_1 + w_2x_2 + w_3x_3 + B = y \quad (2.13)$$

where x 's are inputs, w 's are weights, B is the bias, and y is the raw output. A severe limitation of Eq. 2.13 on its own is the relationship between x 's and y is linear. Activation functions are used to remap the raw output y to a new value

$$z = f(y) \quad (2.14)$$

where z is the output of the perceptron. If the activation function is non-linear, the perceptron will be able to learn non-linear relationships. There are numerous activation functions which can be used, some popular examples are shown in Fig. 2.7. The logistic (or sigmoid) function is defined

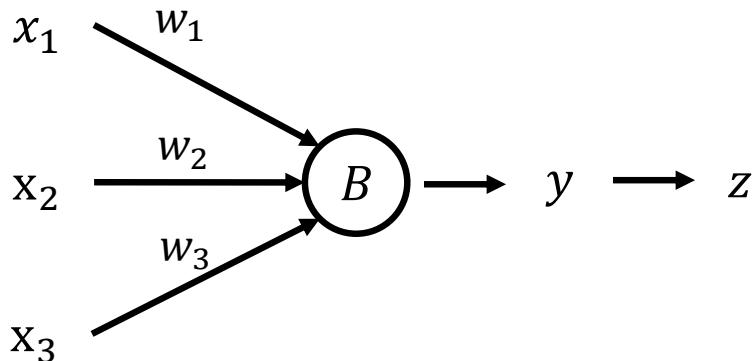


Figure 2.6: Schematic of a perception, x 's are inputs, w 's are weights, B is the bias, y is the raw output, and z is the activation function output.

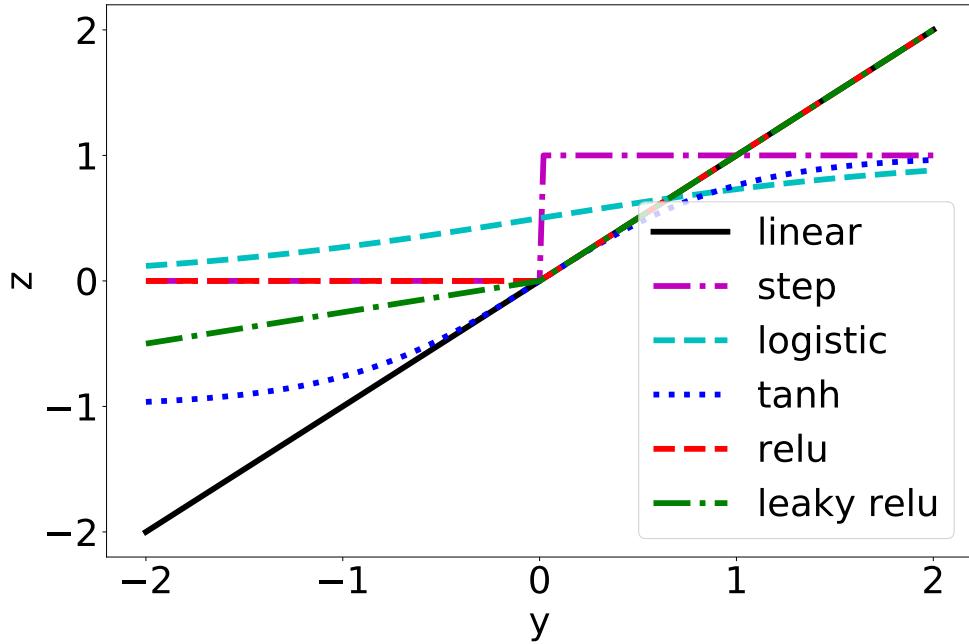


Figure 2.7: Comparison of popular activation functions. Note the slope on the negative side of leaky ReLU has been exaggerated for illustration purposes.

as

$$z = \frac{1}{(1 + e^{-y})}. \quad (2.15)$$

The step function is defined as

$$z = \frac{y}{2|y|} + \frac{1}{2}. \quad (2.16)$$

The hyperbolic tangent function (or TanH) is defined as

$$z = \frac{e^{2y} - 1}{e^{2y} + 1}. \quad (2.17)$$

The rectified linear unit function (or ReLU) is defined as

$$z = \max(0, y). \quad (2.18)$$

Historically, the logistic (or sigmoid) perceptron was the most popular; however, researchers have shown recently ReLU and TanH functions train more quickly for deep networks [52, 53]. One issue with the ReLU perceptron is the zero term. Zeros are generally bad in neural networks since it can lead to neurons no longer learning. One popular modification of the ReLU activation function is the leaky ReLU function defined as

$$z = \max(ay, y). \quad (2.19)$$

where a is typically 0.01 [53]. As a general rule of thumb leaky ReLU activation functions should be used for non-classification layers (hidden layers or regression output layers) and TanH activation functions should be used for classification layers.

2.5.1.2 Network Architecture

A schematic of an example network architecture is shown in Fig. 2.8. The network shown in Fig. 2.8 consists of four layers: one input layer, two hidden layers, and one output layer. The lines connecting individual perceptrons represent weights which will be learned during training. If all the outputs of neurons only go to the next layer, a network is said to be a feed-forward network. A network is said to be deep if it has at least two hidden layers. There are many types of neural networks in the literature, but many times complex networks are simply a combination of many simple layers. Each of the layers shown in Fig. 2.8 are fully-connected (or dense) layers since every perceptron in a layer is connected to every perceptron in the previous and next layer and not connected to any perceptrons in further away layers or its own layer. Some of the common layers in the literature today are convolutional layers, pooling layers, transpose convolutional layers, and recurrent layers.

Convolutional layers were first presented by LeCun in 1989 [54] and recently popularized by Krizhevsky in the 2012 image net competition [55]. The fundamental principle of a convolutional

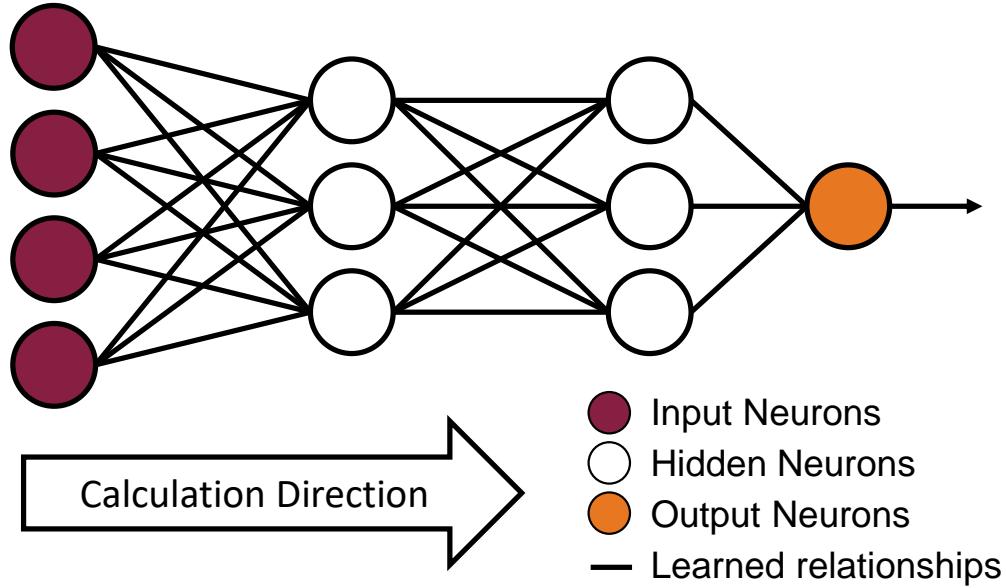


Figure 2.8: Sample network architecture with one input layer, two hidden layers, and one output layer.

layer is the idea that the data has a meaningful grid-like relationship [39]. In a convolutional layer perceptions are arranged into a set of kernels (or filters) which are passed over spatially resolved data (such as images). The output from the layer is a new image per filter which corresponds to the response of the image to that filter. These output images can be thought of as a high level feature map corresponding to a specific filter. An example is shown in Fig. 2.9 where the mean q_0'' from the wall fire example has been convolved with a second order central finite difference x-axis derivative filter. In a convolutional layer, each perceptron in each filter is being trained.

One of the difficulties of a convolutional layer is it can be sensitive to small translations in the input data. A way to introduce translation invariance is to use a pooling layer after each convolutional layer [39, 56, 57, 58]. In general pooling layers use a rectangular kernel to output a single value. One of the most popular kernels is max pooling which will return the maximum value within a rectangular region around a data point. Since a similar value will be returned for neighboring points, it is often possible to reduce the resolution of the data with a pooling layer without any loss in predictive capability. As a general rule of thumb, a pooling layer should be used after each

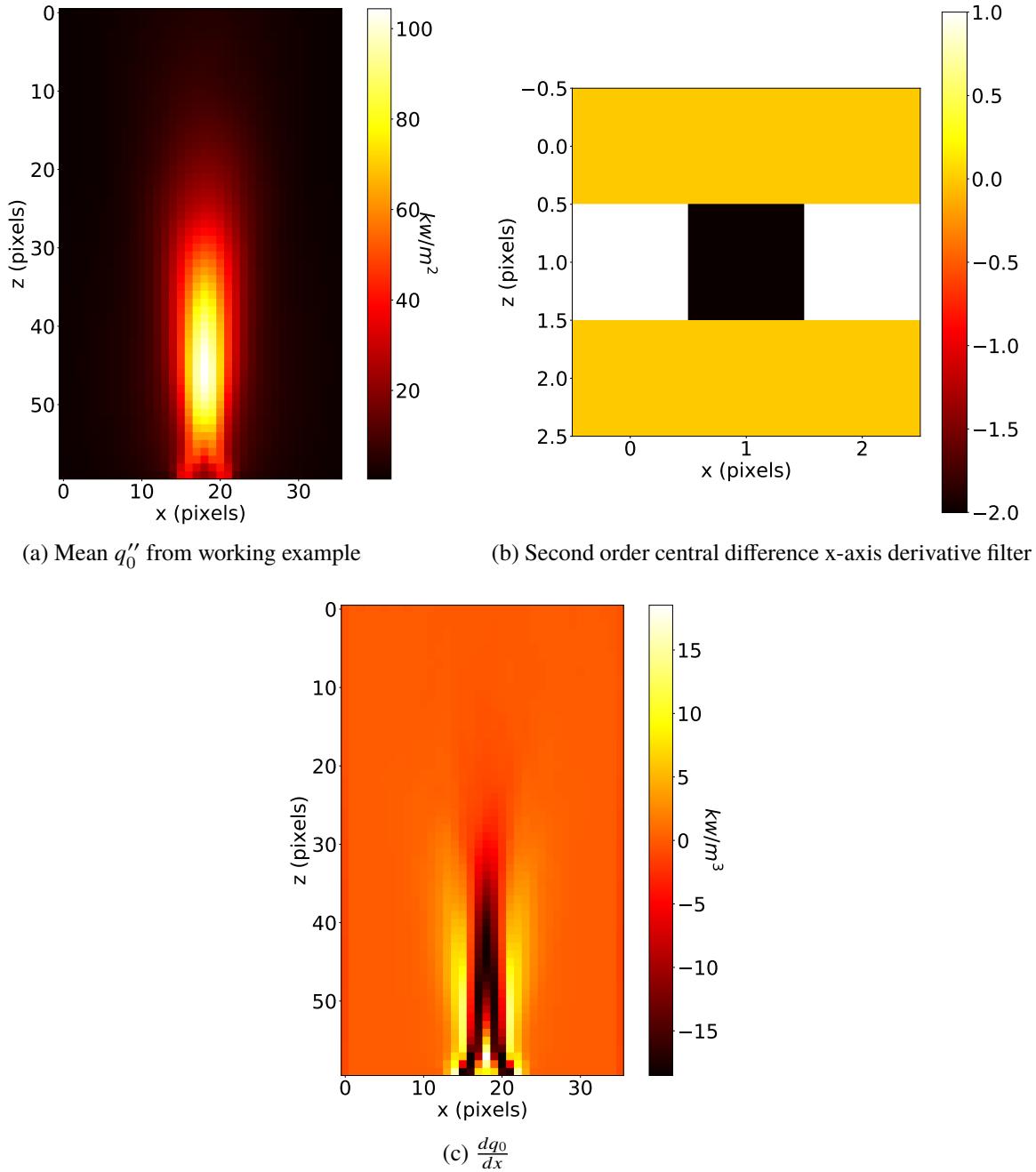


Figure 2.9: Example 2-D convolution operation.

convolutional layer. A good starting point is to use max pooling with a width of three and a stride of two.

Transpose convolutional layers (also called deconvolution, up-convolution, and inverse convolution) reverse the direction of the convolution operation [59, 60]. A transpose convolutional layer is used to fuse the response of an input image to many filters. The dimensionality of the spatial data can be increased by using a stride larger than one. This up-sampling/un-pooling operation can be thought of as a zero-padding between input data prior to applying the transpose convolutional filter. The up-convolutional network presented by Dosovitskiy *et al.* found using a standard convolutional layer directly after each up-convolutional layer produces better results [61, 62].

Recurrent neural networks (RNN) are designed to process sequential data [39, 63, 64]. Unlike most ANN layers, RNN layers contain a feedback mechanism within the hidden layers. The general premise of a RNN is to process a series of sequential inputs one at a time. At some point in the forward propagation, the current state of the data is sent back to an earlier layer in the RNN which is processing the next sequential input. Goodfellow classifies RNNs based on the desired outputs and when the feedback takes place within the network [39].

2.5.1.3 Training

Training is the process of adjusting the weights and biases in the network to improve the model performance. The general procedure is to use the ANN to predict each outcome from a set of inputs. The cost (or error) is calculated between the predicted outcome and the known outcome. The cost is propagated back through the network and used to update the weights and biases. In stochastic gradient descent the weights and biases are updated using the equation

$$w_{i,k} = w_{i,k-1} - \frac{\eta}{M} \sum_j \left(\frac{\partial C_j}{\partial w_{i,k-1}} \right) \quad (2.20)$$

where $w_{i,k}$ is an updated weight or bias, $w_{i,k-1}$ is a prior weight or bias, η is the learning rate, C is the cost function, and M is the number of samples in a batch (batch size) [51]. The gradient of the cost with respect to each weight and bias, $\frac{\partial C_j}{\partial w_{i,k-1}}$, is calculated using the back propagation algorithm. When developing an ANN for a new application, C , M , and η should be considered.

The cost function is the error equation the ANN uses to adjust weights and biases. There are two main requirements for a cost function if stochastic gradient descent is to be used. The cost function must be able to be applied to each sample individually (or expressed as an average quantity) such that

$$C = \frac{1}{N} \sum_x C_x \quad (2.21)$$

where C is the cost, N is the number of training samples, x is a specific training sample, and C_x is the cost for a specific training sample. In addition, C must only depend on the outputs of perceptrons in the final layer of the ANN [51]. The most commonly used cost functions are quadratic and cross-entropy. The quadratic cost is defined as

$$C = \frac{1}{2N} \sum_x (Y_{ANN,x} - Y_x)^2 \quad (2.22)$$

where Y_{ANN} is the ANN prediction and Y is the known output of the training data [51]. The cross-entropy cost function is defined as

$$C = -\frac{1}{N} \sum_x [Y_x \ln(Y_{ANN,x}) + (1 - Y_x) \ln(1 - Y_{ANN,x})]. \quad (2.23)$$

Using a cross-entropy cost function will result in faster learning than a quadratic cost function [51]. Although other cost functions exist [65, 66, 67, 68], cross-entropy cost is generally a good starting point for classification tasks and quadratic cost a good starting point for regression tasks.

The batch size is the number of training samples which are processed simultaneously and aggregated

prior to updating the weights and biases. A larger batch size is a form of regularization which can help avoid over-fitting specific scenarios. However, large batch sizes can also make it difficult for an ANN to learn when all of the training samples are fairly different. As an example, if an ANN is to solve a regression task but is continuously predicting the mean value, increasing the capacity and reducing the batch size to one can force the network to learn a better solution.

The learning rate, η , generally falls in the range of $10^{-1} - 10^{-4}$. If η is too high, the ANN will become unstable and errors will diverge to infinity; however, if η is too low the ANN will take a long time to train and can get stuck at a high error [39]. Researchers have shown an adaptive learning rate can decrease training time and improve results [69, 70, 71]. The idea is based on the premise that high learning rates will get the weights and biases close to an optimal solution quickly but take a long time to converge to the bottom of a bowl. In general, it is good to determine the highest initial η in the rapid prototyping phase and tune the learning rate decay during the model development phase. One way to determine the highest initial η is to start with a high value and steadily decrease it by a factor of 2 until the ANN no longer diverges.

Detailed descriptions of back propagation and stochastic gradient descent can be found in a number of resources [39, 51, 72, 73, 74]. The back propagation algorithm uses four simple matrix equations. A quick overview of these equations is included here based on the derivation presented by Nielsen [51]. The first equation is to calculate the error in the output layer

$$\delta^L = \nabla_a C \odot \sigma'(z^L) \quad (2.24)$$

where δ^L is the error in the output layer L due to error in each neuron z^L , $\nabla_a C$ is the gradient of the cost function with respect to the output activation, $\sigma'(z^L)$ measures how quickly the error in the output changes due to the error in each individual perceptron, and \odot denotes the element-wise product. The $\sigma'(z^L)$ term is equivalent to reversing the direction of the activation function by

feeding the output value. The next equation is to calculate the error in an earlier layer due to the error in the next layer

$$\delta^l = \left((w^{l+1})^T \delta^{l+1} \right) \odot \sigma'(z^l) \quad (2.25)$$

where $(w^{l+1})^T$ is the transpose of the weight matrix for the next layer, l is the current layer. The $\sigma'(z^l)$ term is reversing the activation for the given layer. The error in each bias is calculated using the equation

$$\frac{\partial C}{\partial b} = \delta \quad (2.26)$$

where b is the bias in a single neuron and δ is the error for that neuron. The error in each weight is calculated using the equation

$$\frac{\partial C}{\partial w} = a_{in} \delta_{out} \quad (2.27)$$

where a_{in} is the incoming activation prior to applying the weight and δ is the error of the neuron.

2.5.1.4 Effective Capacity

The total information a neural network can store is often called its effective capacity. The goal when tuning parameters of a neural network is not to have as high an effective capacity as possible, rather the goal is to match the effective capacity of the network to the complexity of the problem to be modeled [39]. Recall the end goal of the ANN is to have the minimum possible error on the test and validation data sets. The error in test data is composed of two components: the training error and the generalization error. The training error is the minimum expected error after training based on the error of the training data set. The generalization error is the observed error in the test set which is higher than the training error. Ideally the generalization error approaches zero and the error from the training, testing, and validation data sets will be similar. However, if the effective capacity is too high, the neural network will start to memorize training data which will result in the training error decreasing but the generalization error increasing.

Regularization is the process of decreasing the effective capacity of the network by adding difficulty to the training process. There are many different approaches to regularization in the literature [75, 76, 77, 78, 79, 80, 81, 82]. One of the most simple and effective regularization methods is dropout [78]. In a dropout layer, the activations of randomly selected neurons are set to zero based on a tunable probability. This helps avoid over-fitting as the neural network is forced to learn more robust relationships between input data.

When tuning the effective capacity, it is generally best to start with a small amount of regularization and steadily increase the number of perceptrons. Once the training error is below the performance goal, steadily increase the regularization until the test data meets the performance goal as well.

2.5.2 Neural Network Development

The development process consists of two phases: rapid prototyping and model development. A schematic of the rapid prototyping phase of the development process is shown in Fig. 2.10. The overall goal of the rapid prototyping phase is to develop an end to end working prototype model for a subset of training data (10-100 data points). The rapid prototyping phase consists of selecting the baseline model, developing the data pipeline, training the network with a subset of data, and tuning hyper parameters. A schematic of the model development phase is shown in Fig. 2.11. The goal of the model development phase is to tune the hyper-parameters of the ANN (such as perceptrons per layer, training rate, regularization amount) until the training, testing, and validation data meet the desired performance criteria. The rapid prototyping phase and the model development phase are discussed in the following subsections. The content of this section was developed using the Practical Methodology chapter of [39, 83] as a baseline and modified based on the author’s experience.

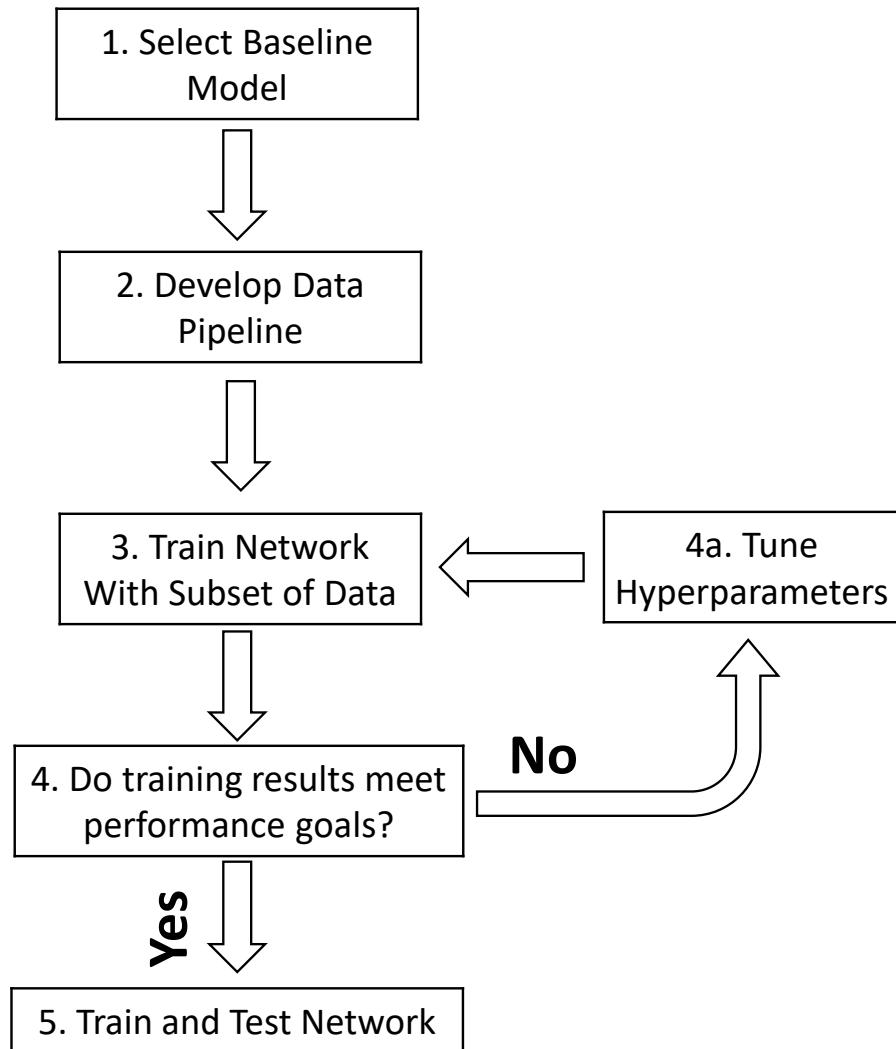


Figure 2.10: Schematic of predictive model development process, Phase 1.

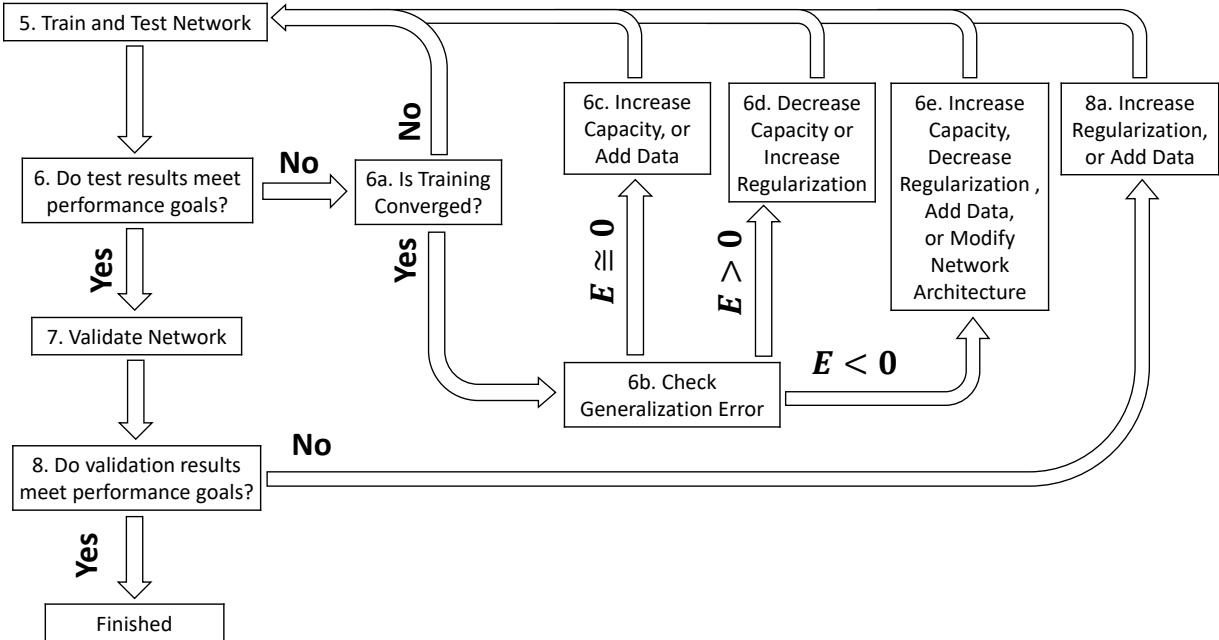


Figure 2.11: Schematic of predictive model development process, Phase 2.

2.5.2.1 Rapid Prototyping

Recommended baseline ANN network architectures based on the input and output data model are shown in Table 2.2. The recommendations are based on the input and output topology (does the data have a grid-like structure such as an image) and frequency (is the network processing a sequence of inputs where the future predictions depend on the previous predictions). If there is no grid-like structure to either the inputs or outputs and the predictions are independent a standard feed-forward network is recommended. A sample feed-forward network architecture is shown in Fig. 2.12. If the input data has a grid-like structure but the output data does not and the predictions are independent, a convolutional network architecture is recommended. A sample convolutional network architecture is shown in Fig. 2.13. If the output data has a grid-like structure but the input data does not and the predictions are independent, a transpose convolutional network architecture is recommended. A sample transpose convolutional network architecture is shown in Fig. 2.14. If

Table 2.2: Baseline neural network architectures.

Topology		Frequency		Network Type	Visual	References
In	Out	In	Out			
No	No	Once	Once	Feed-forward network	Fig. 2.12	[12, 15, 23, 84, 85, 86]
Yes	No	Once	Once	Convolutional neural network (CNN)	Fig. 2.13	[30, 32, 55]
No	Yes	Once	Once	Transpose (inverse, de, up) convolutional neural network (TCNN, ICNN, DCNN, UCNN)	Fig. 2.14	[59, 61, 87, 88]
Yes	Yes	Once	Once	Deep convolutional inverse graphics network (DCIGN)	Fig. 2.15	[89, 90, 91, 92, 93]
No	No	Mult	Mult	Recurrent neural network (RNN)	Fig. 2.16	[63, 64, 94, 95, 96, 97, 98]
Yes	Yes	Mult	Mult	Deep Recurrent Attentive Writer (DRAW)	Fig. 2.17	[99, 100, 101, 102, 103, 104, 105]

both the input and output data has a grid-like structure and the predictions are independent, a deep convolutional inverse graphics network (essentially a convolutional and transpose convolutional network combined) is recommended. A sample deep convolutional inverse graphics network is shown in Fig. 2.15. If the inputs and outputs do not have a grid-like structure and a sequence of coupled predictions are to be made, a recurrent neural network architecture is recommended. A sample recurrent neural network architecture is shown in Fig. 2.16. If the inputs and outputs have a grid-like structure and a sequence of coupled predictions are to be made, a deep recurrent attentive writer (essentially a recurrent network with convolutional and transpose convolutional layers) is recommended. A sample deep recurrent attentive writer network architecture is shown in Fig. 2.17.

If convolutional or transpose convolutional layers are to be used, the padding of the filters and stride length needs to be considered as the resolution of the data will be impacted by these choices. Dumoulin *et al.* presented an overview of the impact of different combinations of filter padding and stride length on the data geometry [60]. If the dimensions of the data should not change in the

convolutional or transpose convolutional layer, half zero padding with a unit stride should be used. In general, increasing the stride length in a convolutional layer will decrease the size of the output data and increase the size of the output data in a transpose convolutional layer.

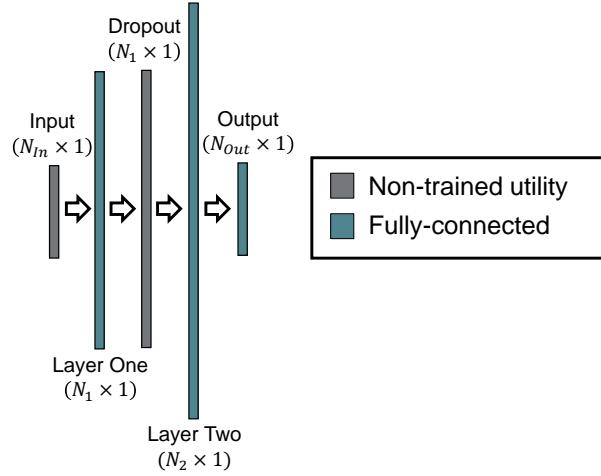


Figure 2.12: Example feed-forward network architecture with two trained hidden layers and one dropout layer.

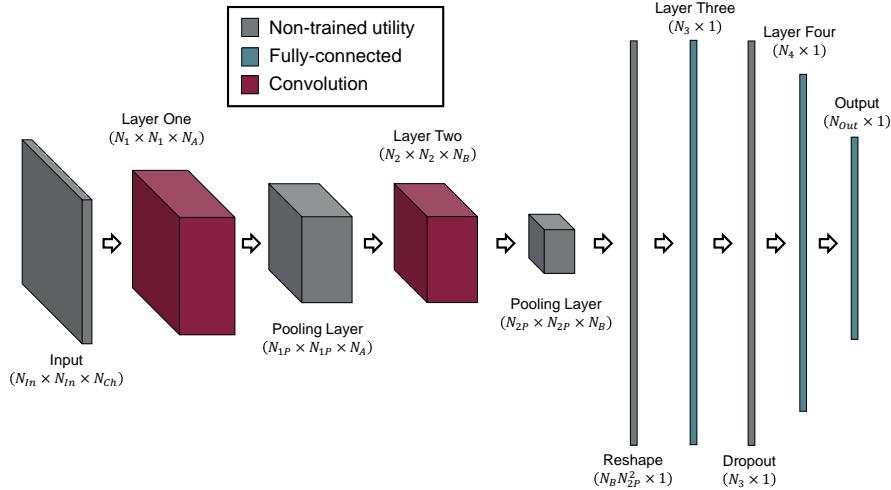


Figure 2.13: Example convolutional neural network architecture with four trained hidden layers and one dropout layer.

After selecting a baseline model, the next step is to develop an end-to-end system [39]. There are many machine learning libraries which can have example implementations of network architectures

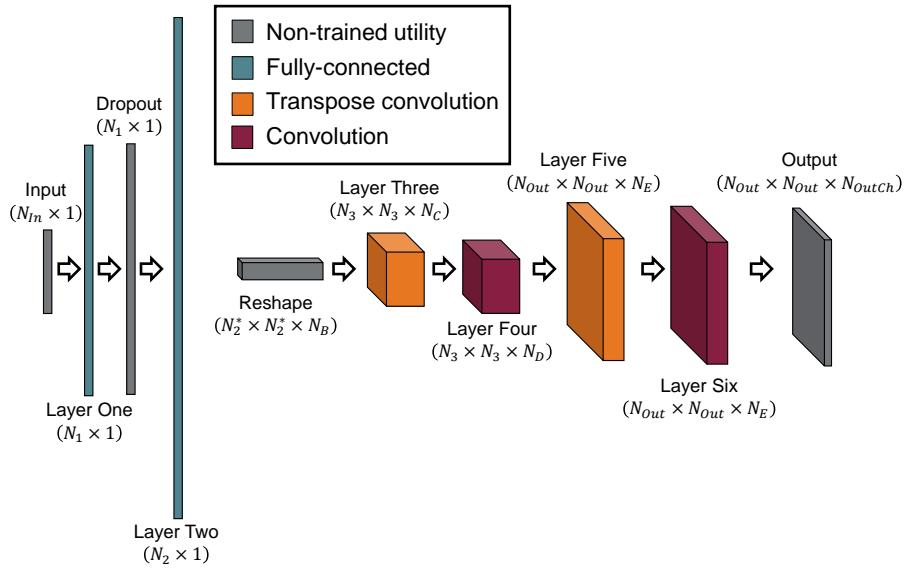


Figure 2.14: Example transposed convolutional neural network architecture with six trained hidden layers and one dropout layer.

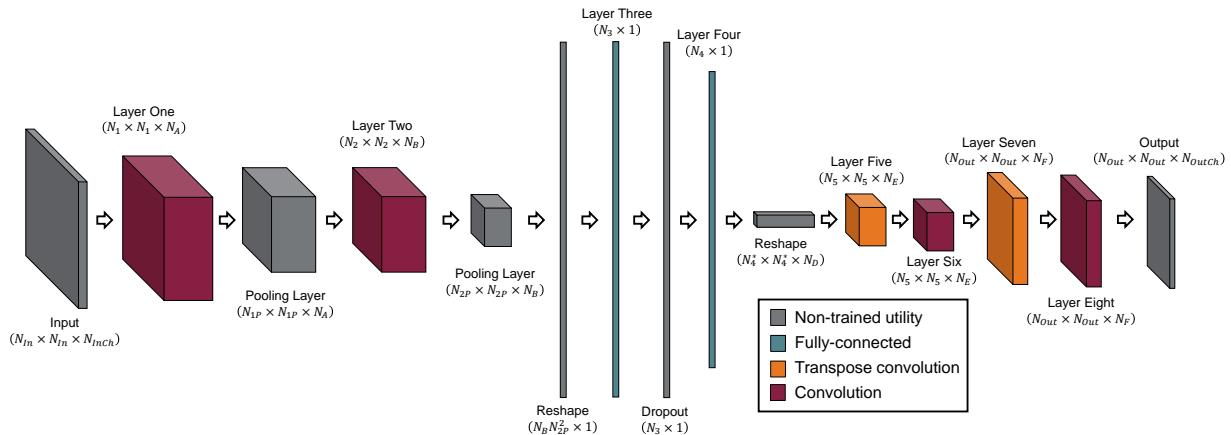


Figure 2.15: Example deep convolutional inverse graphics network architecture with eight trained hidden layers and one dropout layer.

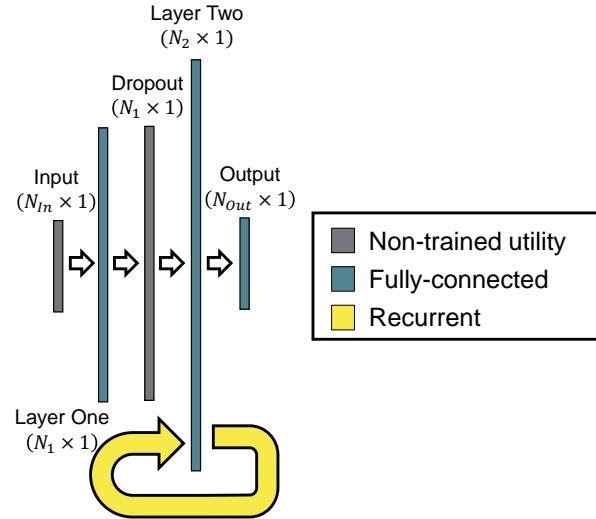


Figure 2.16: Example recurrent neural network architecture with two trained hidden layers, one dropout layer, and one recurrent operation.

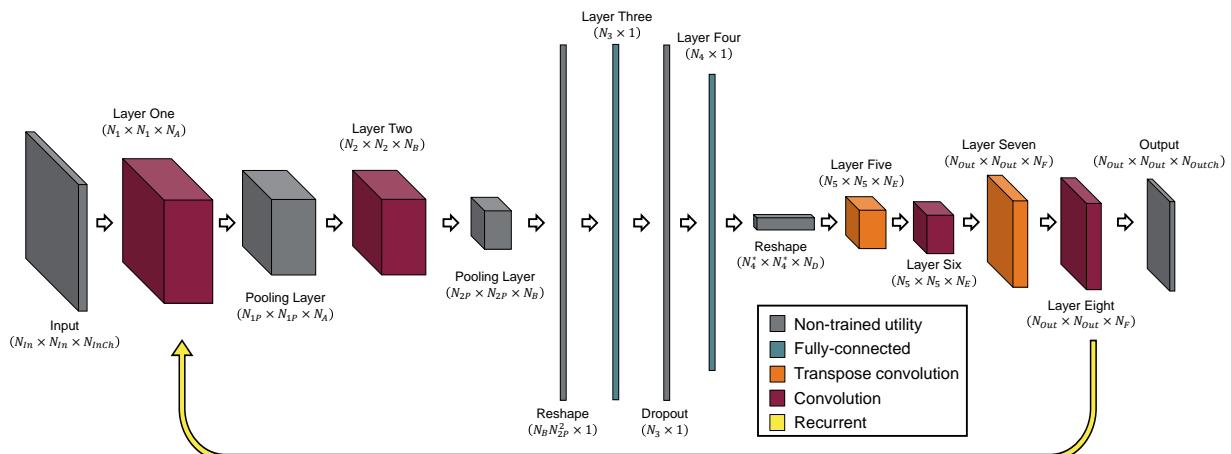


Figure 2.17: Example recurrent deep convolutional inverse graphics network architecture with eight trained hidden layers, one dropout layer, and one recurrent operation.

which can be used to accelerate this process. The ANN for the wall fire example was built using TensorFlow [106] using the bindings in Python 3. Once the end-to-end system is created, the hyper-parameters of the network can be tuned to improve performance.

After developing the end-to-end system the next step is to train a network with a subset of data. The hyper-parameters which need to be decided are the number of perceptrons per layer, learning rate, batch size, and dropout probability. The number of perceptrons per layer will be dependent on the specific problem; however, for the standard feed-forward network shown in Fig. 2.12, 50 neurons per layer is a good starting point as it trains quickly. The learning rate generally falls in the range of $10^{-1} - 10^{-4}$. If the learning rate is too high the ANN will diverge (cost/error increases instead of decreases). One way to set the initial learning rate is to start at a relatively high value of 10^{-1} and steadily decrease it until the ANN no longer diverges. A batch size of 10-100 and a dropout probability of 0.25 is a good starting point. If the ANN is predicting the mean value in a regression task, set the batch size to 1 and the dropout probability to 0.0 then slowly increase the number of neurons in the network until the network begins to learn correctly. Once the network is able to learn, slowly add the regularization parameters back. The next stage of development is to train and test the network with all available data, and is discussed in the next subsection.

2.5.2.2 Model Development

The focus of the model development phase is to tune the hyper-parameters of the ANN (such as perceptrons per layer, training rate, regularization amount) until the training, testing, and validation data meet the desired performance criteria. The general procedure is shown in Fig. 2.11. The total number of cycles through the database needed to train the network will depend on the size of the database and the complexity of the system to be modeled. A good rule of thumb is to train for 10-100 cycles through the database and see if the performance of the network on the training and testing data sets is still improving. At each stage the generalization error is calculated using the

equation

$$E = E_{test} - E_{train} \quad (2.28)$$

where E is the generalization error, E_{test} is the error on the test data set, and E_{train} is the error on the training data set. The scenario where $E > 0$ shows the ANN is over-fitting to the training data set. This can be fixed by either decreasing the capacity of the network or increasing the regularization. There are a few scenarios where E is close to zero but the test performance does not meet the performance goals. In the first scenario the capacity of the network is insufficient to store enough information to make good predictions and should be increased. If this is still an issue after the capacity has been increased, this indicates the ANN has enough capacity but the training data is insufficient to learn the relationships. It is possible this means the network simply needs more samples in the training set to learn the relationships. If this is still an issue after adding more samples, it is possible the data included in X is insufficient to meet the desired performance. Parameters which were neglected as inputs to the model for various reasons should be examined and added to the model if possible to improve the performance of the model, see the discussion in Section 2.4.5. The scenario where $E < 0$ should not occur as that means the model is performing better on the test data set than the training data set. If this occurs remove all regularization (dropout probability to 0.0, batch size to 1) and retrain and test the network. If the issue is fixed, slowly add regularization back in as needed to lower generalization error. If the issue is not fixed, increase the capacity of the network and add more data to the training set. If the issue is still not fixed the network architecture will need to be modified, likely adding more layers to the network.

Once the performance on the test data set meets the performance goals, the performance of the ANN on the validation data set is tested. If the performance on the validation data meets the performance goals, the development process is complete. If the performance of the validation data does not meet the performance goals (but the performance on the test data does meet the performance goals) either increase the regularization of the network or add more data to the training set.

2.6 Wall Fire ANN Development

The development process presented in this work was applied to develop an ANN to predict the standard heat flux entering a vertical wall exposed to a fire. A description of the scenario is given in Section 2.3. The following subsections describe the Data Model Development phase, Predictive Model Development phase, and results for this application.

2.6.1 Data Model Development

The first step is to define the desired outcomes of the ANN. The goal of this example will be to predict the mean peak q_0'' for the last 60s of exposure. Since there is only a single desired outcome without any spatial dependence, there is no grid-like topology of the outputs.

The next step is to define how the performance of the ANN will be measured. The performance of the ANN in this example will be measured using the mean absolute percent error, E_{MAPE} ($a = 1$ and $b = 1$ in Eq. 2.7). The target will be to have $E_{MAPE} < 10\%$ for 95% of test cases.

The next step is to identify each factor which could impact the desired outcome. Recall q_0'' is a function of the buoyancy driven plume generated by a fire. The input set of data may need to contain information regarding fuel and reaction properties (such as heat release rate, chemical formula, heat of combustion, energy per unit mass of oxygen consumed, combustion efficiency, radiative fraction, soot yield, and fire diameter), geometric properties (such as width, height, and thickness of the wall), initial conditions (such as ambient temperature and pressure), surface properties (such as convective heat transfer coefficient, and emissivity), material properties of the wall (such as thermal conductivity, density, and specific heat capacity), and the heat transfer boundary condition of the back-side of the wall (such as adiabatic, isothermal, convection to ambient). A total of 20 parameters have been identified which could affect the predictions which may need to be included

in the input data model.

The next step is to list the available data. Experimental data is fairly limited for this example as many of the factors are difficult to examine independently (limitations on types of fuels and materials for example). For this example data is generated from simulations so the parameter space is not limited. Since simulation data is being used, it is important to identify the most efficient simulation to generate accurate data for this application to minimize the computational time required to generate the data set. A preliminary convergence study showed a three-dimensional simulation using FDS with a simulation duration of 120s and a mesh critical length scale of $D^*/dx = 10$ as recommended in the FDS user guide [107],

$$D^* = \left(\frac{\dot{Q}}{\rho_\infty c_p T_\infty \sqrt{g}} \right)^{2/5} \quad (2.29)$$

where \dot{Q} is the heat release rate, ρ_∞ is the air density at ambient, c_p is the specific heat capacity of air at ambient, T_∞ is the air temperature at ambient, and g is the acceleration due to gravity was required. The total CPU time required for each simulation was around 30 minutes.

The next step is to limit the scope and finalize the input data. The original set of 20 parameters was reduced based on domain knowledge to seven fuel and reaction parameters (fire heat release rate per unit area, radiative fraction, energy per unit mass of oxygen consumed, chemical formula, heat of combustion, burner length, and burner width), and one surface parameter (convective heat transfer coefficient). A set of $(2 \times N + 1)$ simulations (one simulation with each parameter at the nominal center of its range, two simulations with each parameter independently set to its upper and lower limit) was used to determine the relative impact on q_0'' for each parameter. Since there were eight parameters, a total of seventeen simulations were used to select the number of parameters. The size of the gas burner (length and width), fire heat release rate per unit area, energy consumption per unit mass of oxygen, radiative fraction, and convective heat transfer coefficient were found to

have the highest impact. Realistic distributions of each parameter were developed based on data available in the literature. Since each input parameter is a single value with no spatial dependence, there is no grid-like topology for the input parameters.

The next step is to gather, generate, and combine the data. A numerical Monte-Carlo study was performed sampling each of the eight input parameters from the distributions identified in the literature. A total of 3,498 simulations were performed. The predictions of standard heat flux to the wall from each simulation were extracted and organized into a database file.

The next step is to encode the data to improve performance for the ANN. Each of the input parameters are expected to have a monotonic impact on q_0'' so organization encoding does not need to be applied. Since there are no missing values or class IDs, classification encoding does not need to be applied. Since the goal of this network is to predict the steady state q_0'' , there is not a periodic component; therefore, no periodic encoding is required. However, the magnitude and variation of the input parameters do vary significantly, so standardization encoding is appropriate. Each of the inputs and outputs is standardized using Eq. 2.10 with the standardization parameters based on the training data set.

The last stage of Data Model Development is to split the data into training, testing, and validation sets. Since the input parameters were randomly sampled in the Monte-Carlo study, unsupervised splitting of the data is appropriate. One-third of the 3,498 simulations were randomly assigned to the training, testing, and validation data sets.

2.6.2 Predictive Model Development

The first step in the Predictive Model Development phase is to select a baseline network architecture. Recall the input and output data has no grid-like topology. If the spatially resolved heat flux on the wall was desired, the output data would have a grid-like structure which would necessitate a

transpose convolutional neural network. However, predicting the peak q_0'' on the wall removes the spatial component from the output. There are also 120 snapshots in time available (one per second). Rather than predicting the time resolved peak q_0'' , the average peak q_0'' over the last 60 seconds will be predicted. With these inputs and desired outcome, a standard the feed-forward network shown in Fig. 2.12 is appropriate.

The next step is to develop an end-to-end system. The ANN for the wall fire example was built using TensorFlow [106] using the bindings in Python 3. A total of 50 perceptrons were used in each of the fully connected layers in Fig. 2.12 in the initial development.

The next step is to train the ANN with a subset of in input data to determine baseline hyper-parameters for the ANN. Training the ANN with 100 scenarios cycling through each training data 100 times with a learning rate of 10^{-1} yielded a E_{MAPE} of 8.6% and 13.7% for 95% of the training and testing data, respectively. If the training error did not meet the performance goal, the effective capacity of the network would need to be increased by increasing the number of perceptrons in the hidden layers. Since the target performance of $E_{MAPE} \leq 10\%$ for 95% is met by the training data, the hyper-parameters do not need to be tuned at this stage.

The next step is to train and test the network with all of the data. Training the heat flux ANN with the full training set for 50 cycles through the data with a learning rate of 10^{-1} yields a E_{MAPE} of 7.3%, 8.3%, and 8.9% for 95% of the training, testing, and validation data, respectively. If the training, testing, or validation performance did no meet the target performance of $E_{MAPE} \leq 10\%$ for 95% of the data, the hyper-parameters of the network would be tuned using the steps shown in Fig. 2.11. However, since the target performance of $E_{MAPE} \leq 10\%$ for 95% of the data is met by all three data sets, the development process is complete.

2.6.3 Wall Fire ANN Results

The ANN predictions of q_0'' are compared with the database q_0'' for the testing data in Fig. 2.18.

The ANN predictions meet the target performance of $E_{MAPE} \leq 10\%$ for 95% of the data is met by the training, testing, and validation data sets.

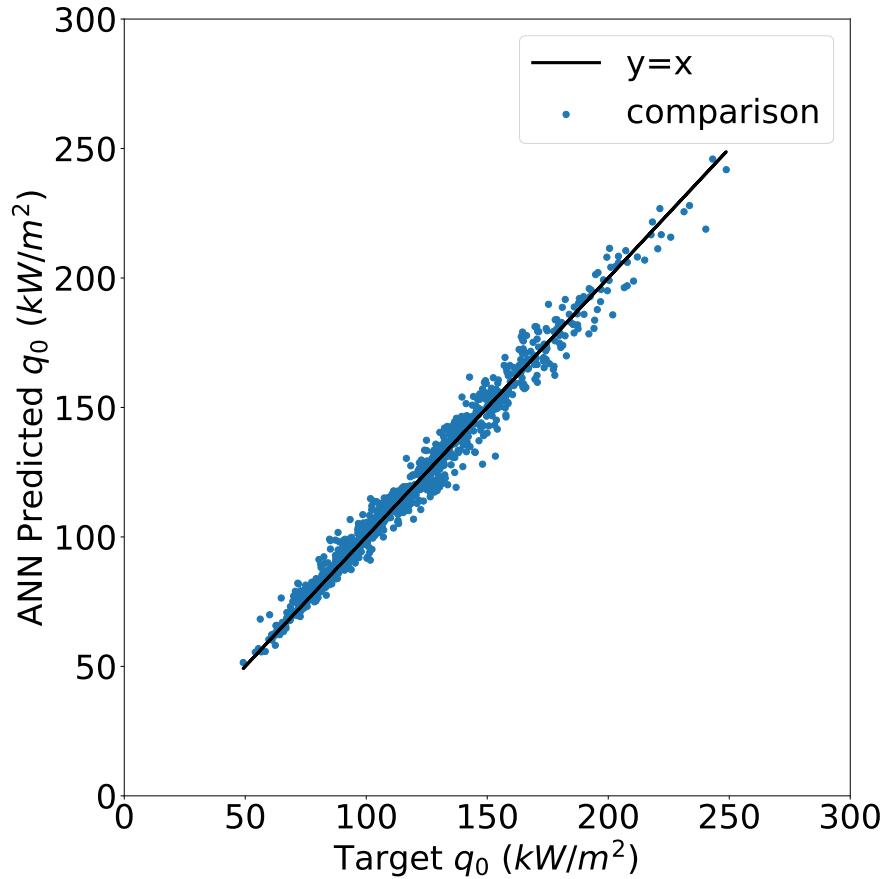


Figure 2.18: Neural network predicted peak q_0'' compared with peak q_0'' for testing data-set.

2.7 Conclusion

This paper presented a general methodology to develop a neural network to predict quantities in new scenarios. The fundamentals of the data model and predictive model were discussed in

detail and applied to develop an ANN to make regression predictions in a supervised learning context. The type of data, format, and encoding of data were addressed and the implications on the required network capacity discussed. Recommendations of baseline network architectures to use for different input and output types were presented. A recommended process for manual tuning of the network hyper-parameters was presented in the context of overall network performance and generalization error. As an example, a new neural network to predict the standard heat flux on a wall from a fire was developed using the methodology developed in this paper. The predictions from the network were within 10% error of the values in the database for 95% of scenarios.

Chapter 3

Wildland Fire Spread Modeling Using Convolutional Neural Networks

3.1 Abstract

This paper presents a novel data-driven approach to estimating the spread of a wildland fire using a deep convolutional inverse graphics network (DCIGN). Simulated burn maps for use in this process were generated at six hour intervals using the phenomenological fire spread model of Rothermel with 10,000 different combinations of input parameters. The robustness of the approach is tested using 1,000 simulations not included when training the DCIGN. Overall the predicted burn maps from the DCIGN-based approach agreed with simulation results, with mean precision, sensitivity, and F-measure of 0.97, 0.92, and 0.93, respectively. Noise in the input parameters was found to not significantly impact the DCIGN-based predictions. The computational cost of the method was found to be comparable to a phenomenological model in homogeneous spatial conditions, and significantly better for heterogeneous spatial conditions. Although trained on predictions six hours apart, the DCIGN-based approach is shown to be capable of predicting burn maps further in the future by recursively using previous predictions as inputs to the model. When the initial fire was small, the model tended to under-predict fire spread; however, predictions generally improved as the fire grew.

3.2 Introduction

Wildland fire propagation is a complex process which involves the interactions of many underlying physical phenomena. Since fully resolving these processes remains a research effort researchers have developed simplified models which describe the growth rate of a wildland fire. Simulating the spatial-temporal flame front across diverse landscapes for large fires may take hours to run depending on the size of the fire perimeter. Considering the impact of varying weather conditions and different intervention strategies on the fire spread requires multiple simulations which may limit the number of scenarios which can be analyzed during an emergency. A new wildland fire spread model which is capable of predicting the two-dimensional spatial-temporal flame front for wildland fires spreading over diverse landscapes, fuel types, and weather conditions which provides rapid predictions regardless of fire perimeter size is needed. Existing wildland fire spread models generally fall into three categories: stochastic, phenomenological, and physical. A number of different fire spread model approaches have been proposed with each class of model having different advantages and limitations [108, 109, 110].

Stochastic fire spread models are based on statistical analysis of historical wildland fires and prescribed burns. Sullivan examined 14 stochastic models developed between 1990-2013 and found each could be summarized by the functional form chosen to describe the impact of wind speed, zero wind fire spread, and fuel moisture content [110]. The impact of wind on wildland fire spread generally follows a power law where the exponent and prefactor are related to the fuel type. The impact of fuel moisture content on wildland fire spread is generally modeled as strongly linear or weakly exponential based on the fuel type. These models are favored by municipalities as they are capable of providing rapid reasonable estimates of the overall fire risk of the region for which they were derived. However, stochastic models fail when predicting conditions outside the historical data, such as new weather conditions, spatial regions, or different fuel types. In addition,

these models generally make no attempt at resolving the spatial-temporal flame front of a wildland fire.

Phenomenological fire spread models are based on conservation of energy, but use experimental measurements to develop functional forms rather than modeling from first principles [108, 110]. The most widely known phenomenological fire spread model is the model of Rothermel [50, 111] which uses empirical correlations for heat source and sink terms [108]. The baseline rate of spread (with no wind or slope) is based on fuel density, type, and moisture content. The impact of wind and slope is modeled as a multiplier of the baseline rate of spread. Although developed to determine the rate of spread from a single ignition source in a single direction, these models have been expanded to work in two-dimensions using Huygens' principle [112, 113] and the level set method [114, 115]. These models are favored by foresters and firefighters in the field as they are capable of providing rapid estimates of the rate of spread [6]. However, phenomenological models inherit similar issues to stochastic models when predicting conditions outside the historical data used in deriving the empirical models. In addition, expanding the model to two-dimensions to predict the spatial-temporal flame front significantly increases the computational requirements. Researchers have used reduced order modeling to decrease the computational cost of evaluating the two-dimensional rate of spread using phenomenological models [116]. However, it is difficult to apply this framework to other models as new derivations are required.

Physical fire spread models are based on the fundamental chemistry and physics of combustion, and fire spread [109]. Physical models can be further subdivided into simple and detailed models. Simple physical models such as that of Weber consider only the transport of energy, neglecting calculation of the full flow field for computational reasons [108]. These models are more general than stochastic or phenomenological models and can be close to real-time [6]. Detailed physical models such as Wildland-Urban Interface Fire Dynamics Simulator (WFDS) consider the transport of mass, momentum, and energy in addition to combustion in a multiphase approach through

computational fluid dynamics (CFD) [117]. These models are favored by researchers interested in the fundamental mechanisms of wildland fire spread as the basis on first principles reduces the impact of user intuition on the results. However, the computational cost to model moderate domains (2.25 km^2) is significant [117].

Data-driven approaches use simplified functions to represent underlying connections between data to make estimates of future events, often using machine learning. Several researchers have presented data-driven approaches to estimate the total burned area of a fire based on meteorological data [7, 8, 9, 10, 11]. Each of these methods is capable of predicting the total burned area; however none attempts to estimate the spatial-temporal fire distribution of a wildland fire front. A data-driven approach to predicting the spatially-resolved final burned region of a wildland fire was presented by McCormick. The model used an artificial neural network to classify the center pixel of a 3×3 neighborhood of pixels as burned or unburned [12, 13]. The initial results are promising; however, the model relies on the assumed ellipsoidal growth profile in the direction of wind [112] and only predicts the final burned region but not the evolution of spread.

A method to predict fully resolved spatial-temporal burn maps is desired. The fundamental principle which makes convolutional neural networks (CNNs) versatile is the capability to learn how to represent complex shapes as combinations of high level feature maps. Krizhevsky showed many of the features learned by the CNN in the ImageNet competition described the inter-relationship of the 3 color channels [55]. Researchers have shown a combination of convolutional and transpose convolutional layers in a network can be used to generate new images [89, 90, 91, 92, 93]. This type of network was coined a deep convolutional inverse graphic network (DCIGN) by Kulkarni *et al.* [89]. As an analogy to image classification, data such as elevation, moisture content, and wind speed can be treated as channels in an image. Given enough data, a DCIGN will be able to learn relationships between these physical parameters which can then be used to predict a future burn map.

The objective of this study is to apply a trained DCIGN framework to predict the spatial-temporal distribution of a wildland fire front in without relying on any other models at run-time. Data for use in training and testing the network was generated using Rothermel's phenomenological model. The quantitative performance of the model on 1,000 simulations not included during training is presented, and the sensitivity of the network to noise is examined. The work presented herein demonstrates the concept on a simple configuration with future work to expand the method to use higher resolution input data and incorporate experimental measurements in the training process.

3.3 Methods

A schematic showing a high level view of the solution algorithm is shown in Fig. 3.1. Each primary driver of wildland fire spread is included as a channel in an image which is input to the DCIGN. The DCIGN then uses its prior training to predict a new image with two channels corresponding to the probability a pixel has burned or not burned in the future. The two probability masks are then post-processed to output a single future burn map. The temporal resolution of six hours and spatial resolution of one pixel/km physically correspond to available global satellite measurements. The following subsections describe the simulation conditions, network architecture, post-processing, and performance metrics used in this work.

3.3.1 Wildland Fire Prediction

Data for this study was generated using the surface fire spread model presented by Rothermel/Albini [50, 111, 118]. In the Rothermel/Albini phenomenological fire spread model, the peak surface fire

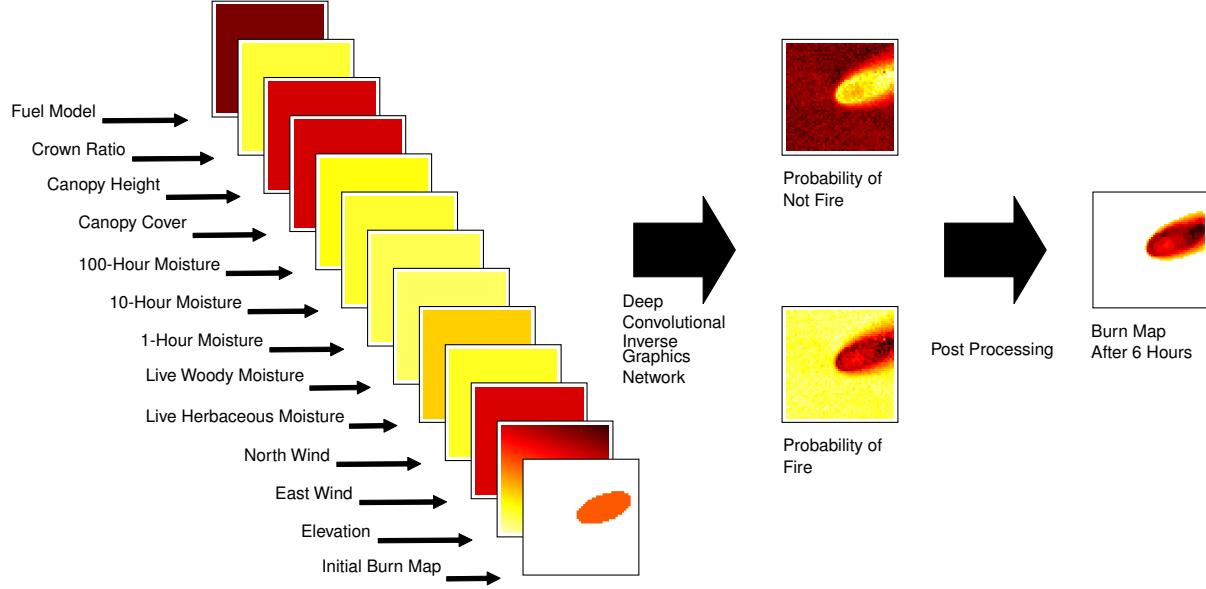


Figure 3.1: Schematic of solution algorithm. The left set of images show the different channels used as inputs to the neural network. The values for each data channel are colorized based on the values shown in Table 3.1 and Table 3.2.

spread rate, $V_{s,peak}$ is calculated using the equation

$$V_{s,peak} = \frac{Q''\zeta}{\rho\epsilon Q_{ig}} (1 + \phi_s + \phi_w) \quad (3.1)$$

where Q'' is the heat release rate per unit area, ρ is the fuel density, Q_{ig} is the heat of pre-ignition, ζ is the propagating flux ratio (percentage of heat released which pre-ignites fuel), ϵ is the effective heating number (percentage of fuel which is involved in ignition), ϕ_w is the wind coefficient, and ϕ_s is the slope coefficient.

Various researchers have developed empirical relationships for the different parameters in Eq. 3.1. A commonly used approach in the literature is to specify Q'' , ρ , ζ , ϵ based on classifying the primary fuel in a region into a fuel model. A total of 53 fuel models were considered in this work including 13 developed by Rothermel/Albini [111, 118], and 40 developed by Scott [50]. Rothermel presented an empirical relationship for Q_{ig} based on the fuel model and moisture content, and Scott extended

the relationship to handle dynamic fuel models. Rothermel presented empirical relationships for ϕ_w and ϕ_s based on fuel model, midflame wind speed, and slope. Andrews presented an algorithm to adjust typical atmospheric wind measurements (10m or 20ft) to midflame wind speed based on three additional parameters describing the upper story vegetation (canopy cover, canopy height, and crown ratio) [119]. Researchers have shown wildland fires grow in a generally ellipsoidal shape for homogeneous spatial conditions based on $V_{s,peak}$ and wind speed [112, 120, 121]. The primary drivers in the fire spread model were identified as landscape (slope, aspect, and fuel model type), moisture content (1-hour, 10-hour, 100-hour, live woody, and live herbaceous), canopy type (height, ratio, and percent coverage), and 10m wind (intensity and direction). The allowable bounds used for each parameter in this work are shown in Table 3.1. The fuel model types were assigned indexes based on the peak rate of spread under the same spread conditions (low moisture, 10 mph wind up a 0.5 slope). Since slope and aspect can be summarized as a two-dimensional difference in elevation, a single channel for elevation was used instead of two channels for slope and aspect in the neural network.

Table 3.1: Limits of each parameter in study.

Parameter	Unit	Min	Max
Aspect	Degrees	0.0	360
Fuel Model	Index	0.0	53
Slope	Fraction	0.0	1.0
1-Hr Moisture	Percent	1.0	40
10-Hr Moisture	Percent	1.0	40
100-Hr Moisture	Percent	1.0	40
Live Herbaceous Moisture	Percent	30	100
Live Woody Moisture	Percent	30	100
Canopy Cover	Percent	0.0	1.0
Canopy Height	Feet	1.0	20
Crown Ratio	Fraction	0.1	1.0
Wind Direction	Degrees	0.0	360
Wind Velocity	Mi/Hr	0.0	30

A custom implementation of Rothermel’s fire spread model was developed in Python to streamline

the integration with the DCIGN software. The implementation of Rothermel's model in BehavePlus was used to validate the simulation framework [122]. A total of 10,000 different combinations of the parameters shown in Table 3.1 were simulated. The total time to run 10,000 simulations was 30.6 seconds. Raster images of the burned map were generated at a resolution of 1 pixel/km every six hours for a total of 24 hours. Three cases for use in the DCIGN were generated from each simulation by creating pairs of burned maps six hours apart. The zero hour to six hour pair was not considered as a case for the DCIGN as the zero hour burned image always contains no fire. Example burned maps from one simulation are shown in Fig. 3.2 for the parameter values shown in Table 3.2.

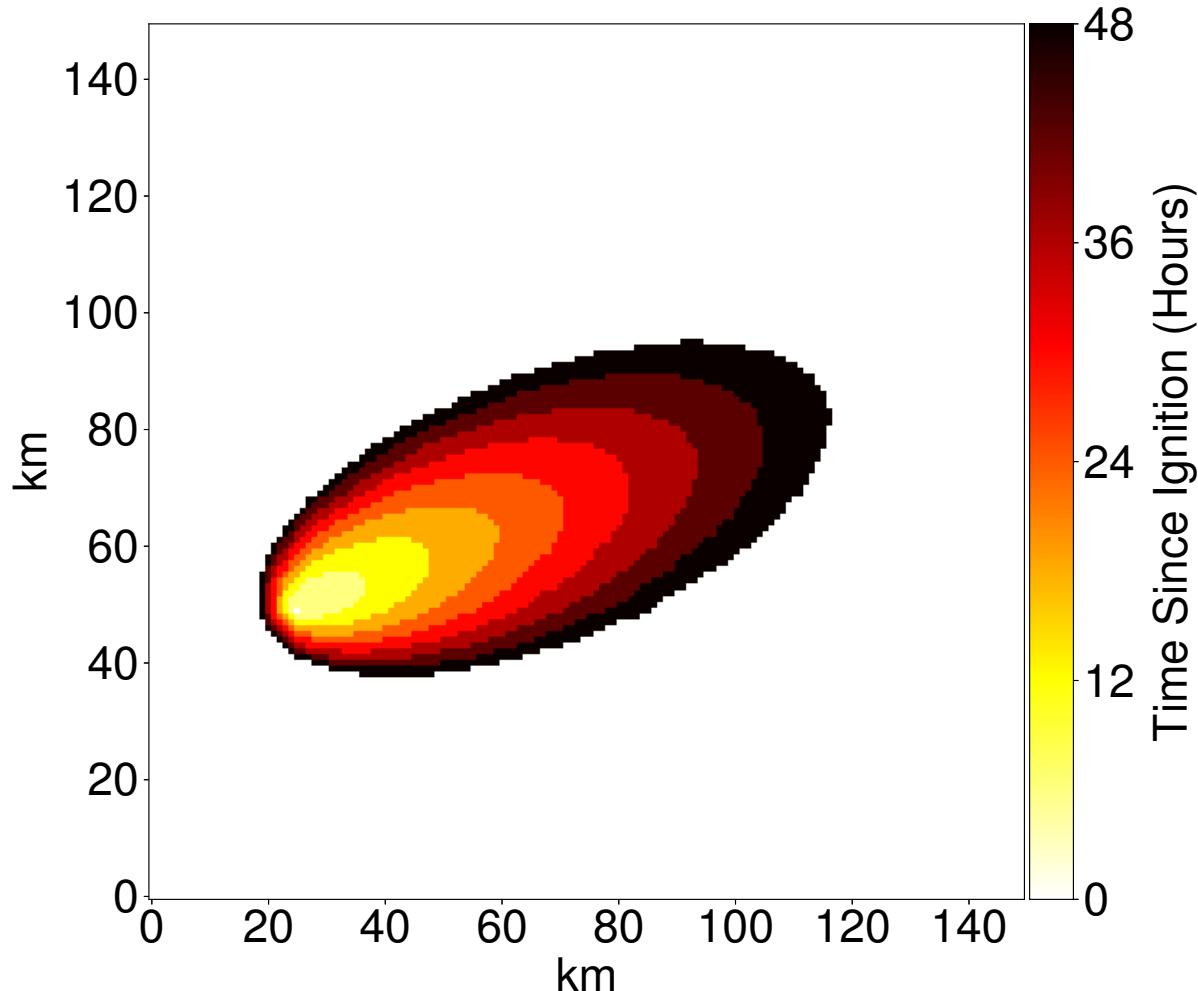


Figure 3.2: Example Simulated Burn Maps

Table 3.2: Parameters for Example Simulation

Parameter	Unit	Value
Aspect	Degrees	130
Fuel Model	Index	FM1 (44)
Slope	Fraction	0.8
1-Hr Moisture	Percent	5.3
10-Hr Moisture	Percent	6.3
100-Hr Moisture	Percent	7.3
Live Herbaceous Moisture	Percent	69
Live Woody Moisture	Percent	49
Canopy Cover	Percent	0.7
Canopy Height	Feet	14
Crown Ratio	Fraction	0.2
Wind Direction	Degrees	34
Wind Velocity	Miles per Hour	13.5

3.3.2 Network Architecture

At a fundamental level, artificial neural networks are massively parallel equations which have the capability to store observed knowledge about a problem to make predictions of new inputs. A convolutional neural network (DCIGN) assumes the input data has distinct spatial dependence within the input parameters. Since the network assumes spatial dependence, less connections need to be made to inputs which are far from each other. This allows a DCIGN to contain much fewer connections and parameters than a similarly sized standard feed forward network with minimal loss in optimal performance for appropriate problems. In addition, this makes it possible to use deeper and more broad hidden layers without increasing computational requirements beyond what is feasible on current technology [55]. Representing the spread of a wildland fire front with a DCIGN is reasonable as wildland fire spread is a local phenomena [112].

The DCIGN architecture used in this work is shown in Fig. 3.3. The input images were 50x50 pixels with 13 image channels corresponding to the image stack shown in Fig. 3.1. The output image contains 50x50 pixels with two image channels corresponding to the probability the burn

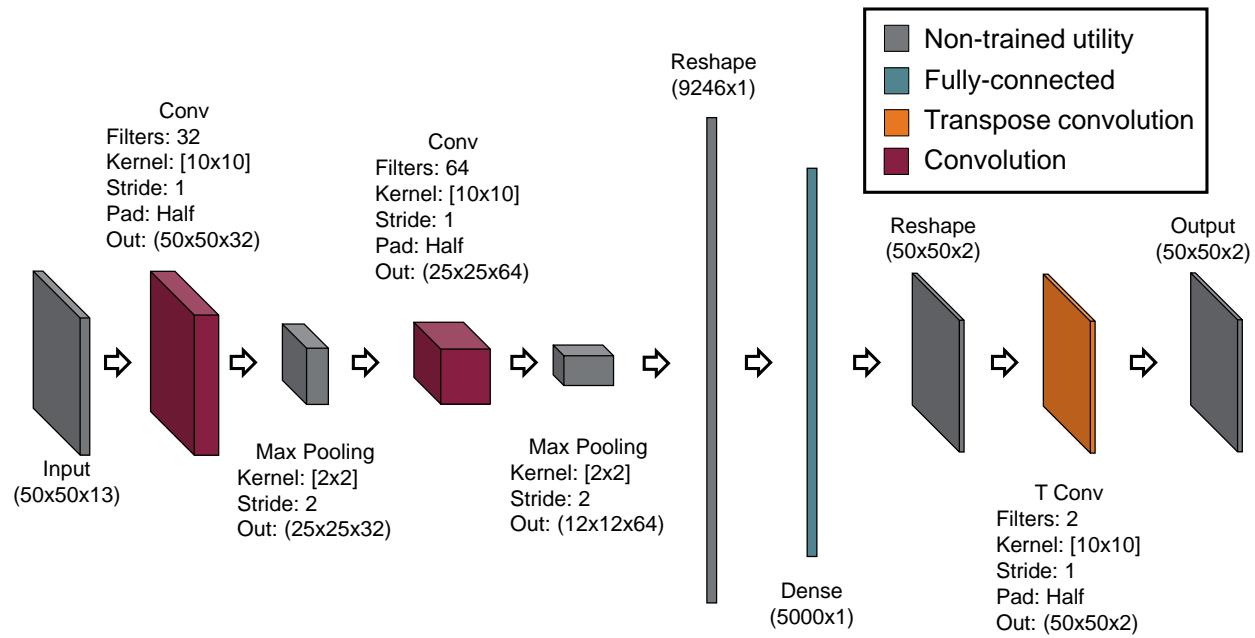


Figure 3.3: Deep Convolutional Inverse Graphics Network Architecture.

map has reached a pixel and the probability the burn map has not reached a pixel. A total of six hidden layers are included in the network including two convolutional, two max pooling, one fully connected classification, and one transpose convolutional layers. The number of filters and step size in each convolutional and transpose convolutional layers and number of neurons in fully connected layers were specified to steadily decrease the degrees of freedom from the 32,500 in the input layer ($50 \times 50 \times 13$ inputs) to the desired degrees of freedom of 5,000 in the output layer ($50 \times 50 \times 2$ probability of fire and not fire).

All layers in the network presented in this work used leaky rectified linear unit (ReLU) activation functions except the fully connected layer which used hyperbolic tangent (TanH) activation functions and the output layer which used a softmax activation function. Researchers have shown deep neural networks learn more quickly and more accurately using leaky or parametric ReLU than logistic or TanH for activation [52, 53]. However, since the activation is unbounded, ReLU is not appropriate for classification layers. Since the intent of the fully connected layer is to classify the feature maps

into high level descriptions of the fire and environment, a TanH activation function is appropriate. Since the desired output is a probabilistic estimate of whether or not each pixel would be burned, a soft max activation function is appropriate to scale the activations of the fire and non-fire probability masks.

The network architecture was built using the Python 3 bindings for TensorFlow [106]. The models were trained using stochastic gradient descent with a batch size of 100 samples. All weights and biases were initialized from a uniform distribution between -1 and 1. The learning rate was fixed for all layers throughout training at 0.0001. The cost function used in training was based on sum square error. Over-fitting was reduced by using 50% dropout on the input layer and shuffling the order of the samples during training. The network was trained using 9,000 simulations (27,000 pairs of burn maps) for 50,000 cycles using a single NVIDIA Quadro K620. The total time to train the network was 18 hours 7 minutes. The total time to evaluate the network with all 10,000 simulations (30,000 pairs of burn maps) was 38.1 seconds.

3.3.3 Performance Metrics

The metrics used to quantify the performance of the DCIGN in this study were precision, sensitivity, and F-measure. For each metric, the range of possible values is zero to one, with a perfect score being one. The precision, P , is a measure of commission errors (predicting a fire where there was not fire) and is defined as

$$P = \frac{t_p}{t_p + f_p} \quad (3.2)$$

where t_p is the number of correctly identified fire pixels, and f_p is the number of falsely identified fire pixels. The sensitivity, S , is a measure of omission errors (predicting no fire where there was a fire) and is defined as

$$S = \frac{t_p}{t_p + f_n} \quad (3.3)$$

where f_n is the number of fire pixels which were identified as non-fire pixels. F-measure, F , is an overall measure of performance defined as the harmonic mean of P and S ,

$$F = 2 \cdot \frac{P \cdot S}{P + S}. \quad (3.4)$$

3.3.4 Post Processing

The output layer of the DCIGN contains two normalized probability masks, one for fire and one for not fire. The normalized probability mask for fire is post-processed to convert the probabilistic estimate of burned or unburned to a single contour. A 3x3 median filter is applied to smooth the image. A threshold value on probability of fire is used to determine whether or not each pixel has been burned.

The optimal threshold to use in post-processing was determined by calculating the mean F-measure for DCIGN predictions of the 9,000 training simulations (27,000 pairs of burn maps) with thresholds ranging from 0.01 to 0.99. The mean F-measure was found to be mostly independent of the post-processing threshold in the range of 0.2-0.6 as shown in Fig. 3.4. The maximum mean F-measure of the training data was calculated with a threshold of 0.41. This threshold was fixed and used for post-processing throughout this work. An example neural network prediction before and after post-processing is shown in Fig. 3.5.

3.4 Results

The robustness of the neural network to predict new fires was examined by considering 1,000 simulations (3,000 pairs of burn maps) which were not included when training the network. Sample DCIGN predictions from five of these test cases are compared with simulation predictions

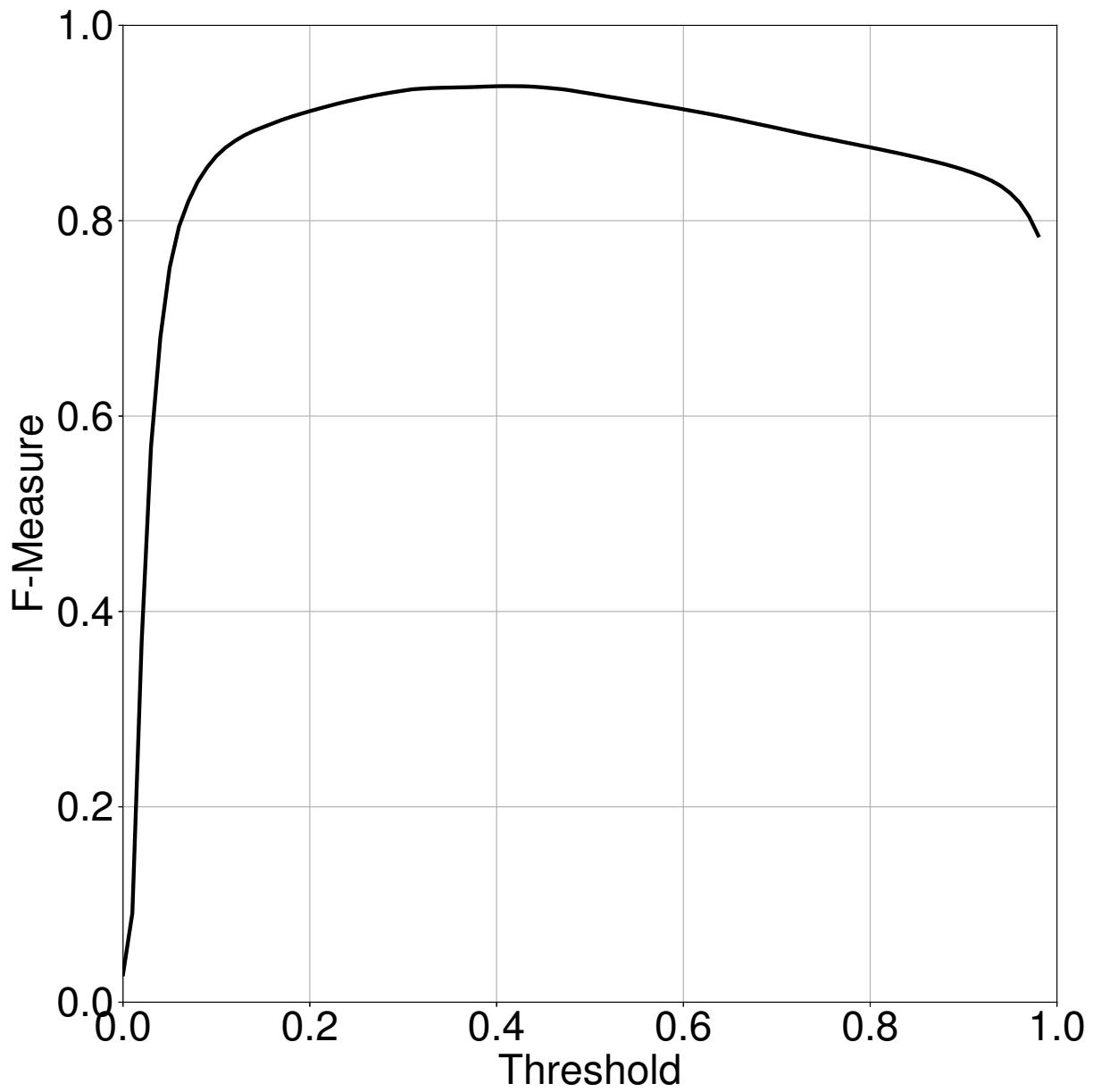
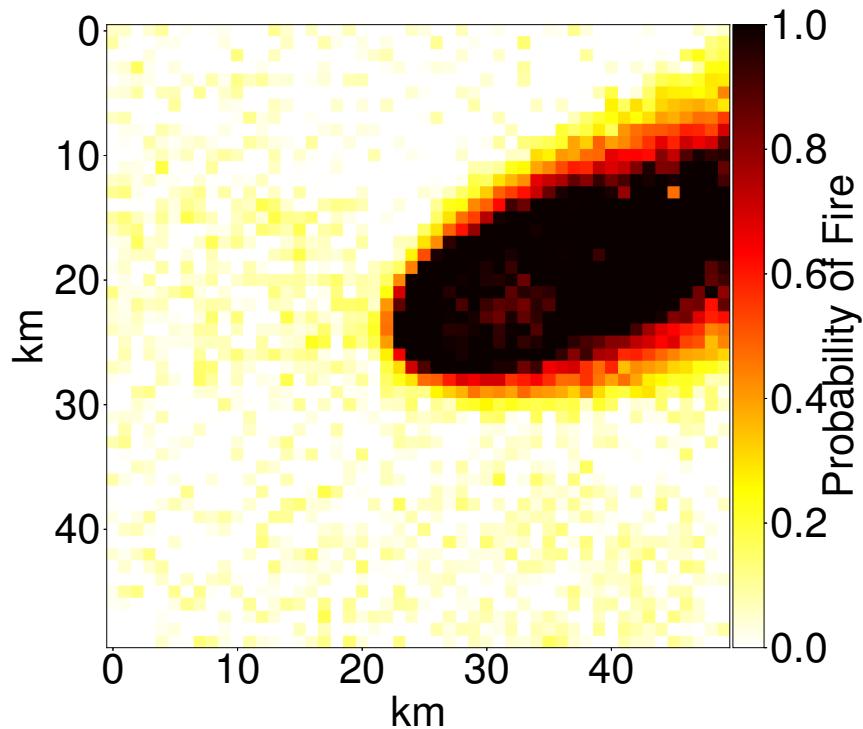
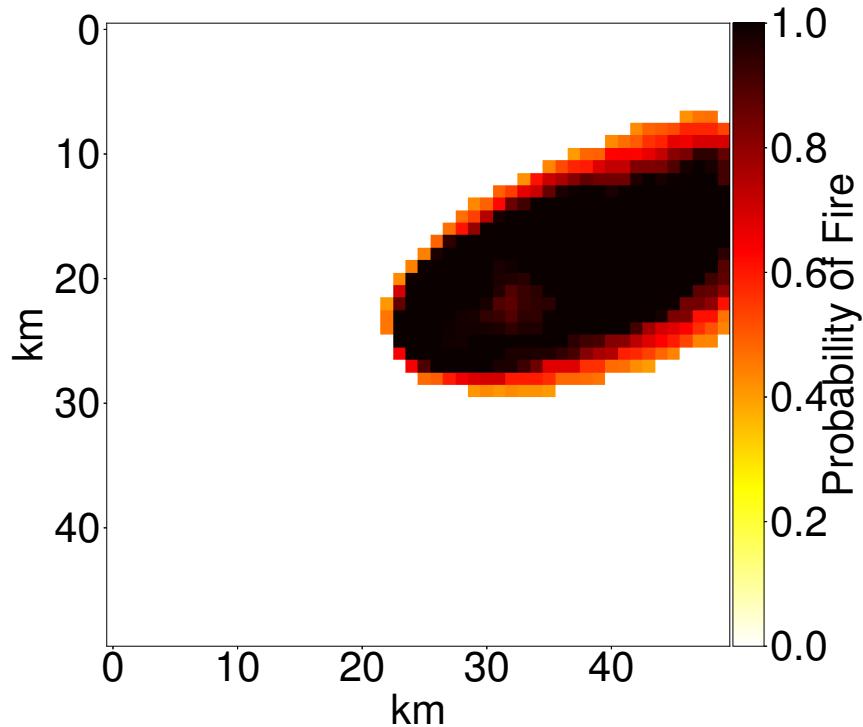


Figure 3.4: Mean F-Measure of DCIGN predictions of 9,000 training simulations (27,000 pairs of burn maps) for different threshold values.



(a) Directly predicted from DCIGN



(b) After post-processing

Figure 3.5: Example DCIGN prediction of burn map probability

in Fig. 3.6. Figure 3.6a shows the initial and next burn map from the simulation. Figure 3.6b shows the final burn map predicted by the DCIGN. Figure 3.6c highlights pixels which the DCIGN prediction did not match the simulation predictions. Pixels shown as black represent commission errors (false positive of fire), and pixels shown as orange represent omission errors (false negative of fire).

The mean precision, sensitivity, and F-measure of the 1,000 test simulations (3,000 burn map pairs) are shown in Table 3.3. Each metric ranges from zero to one with a perfect score being one. The distribution of F-measure for the 3,000 burn map pairs are shown in Fig. 3.7. Percentiles of each metric are shown in Table 3.3 to quantify the spread of each score. Here a percentile is defined as

$$X = \frac{1}{N_{total}} \int_{Y_{min}}^1 N(Y) dy \quad (3.5)$$

where X is the percentile, $N(Y)$ is the number of burn maps achieving a specific score, N_{total} is the total burn maps Y_{min} is the minimum score. For example, the F-measure of 0.86 for $X = 80\%$ shown in Table 3.3 means 80% (2400/3000) of the burn map pairs had an F-measure of 0.86 or higher.

Table 3.3: Performance Metrics of DCIGN Predictions of Test Cases

Parameter	Mean	$X = 80\%$	$X = 90\%$	$X = 95\%$
Precision	0.97	0.95	0.85	0.79
Sensitivity	0.92	0.80	0.67	0.59
F-Measure	0.93	0.86	0.80	0.73

3.5 Discussion

The overall shape of the burn maps predicted by the DCIGN are consistent with the simulations for the 1,000 test simulations (3,000 burn map pairs) examined in this work, as shown in the examples

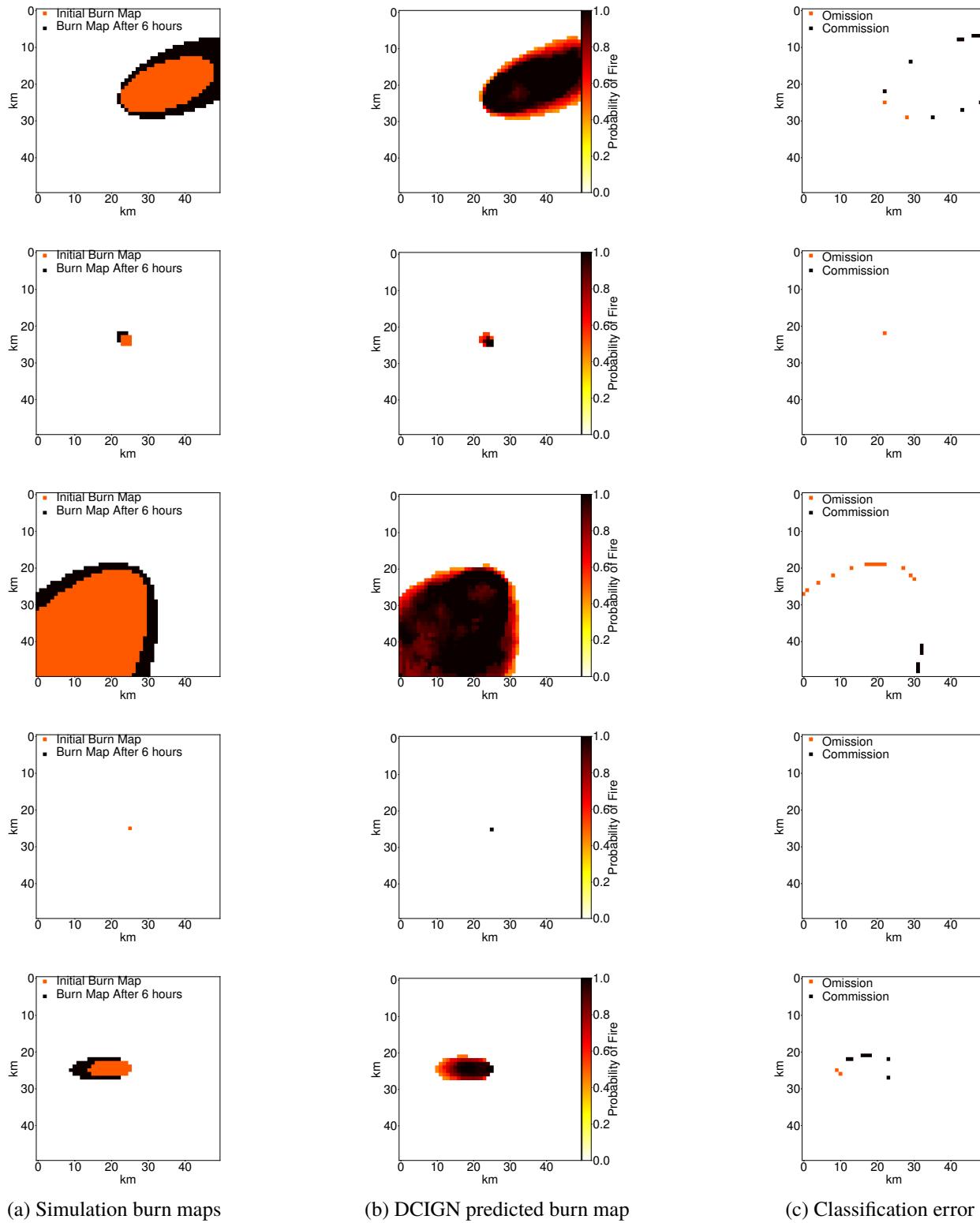


Figure 3.6: Example DCIGN prediction results

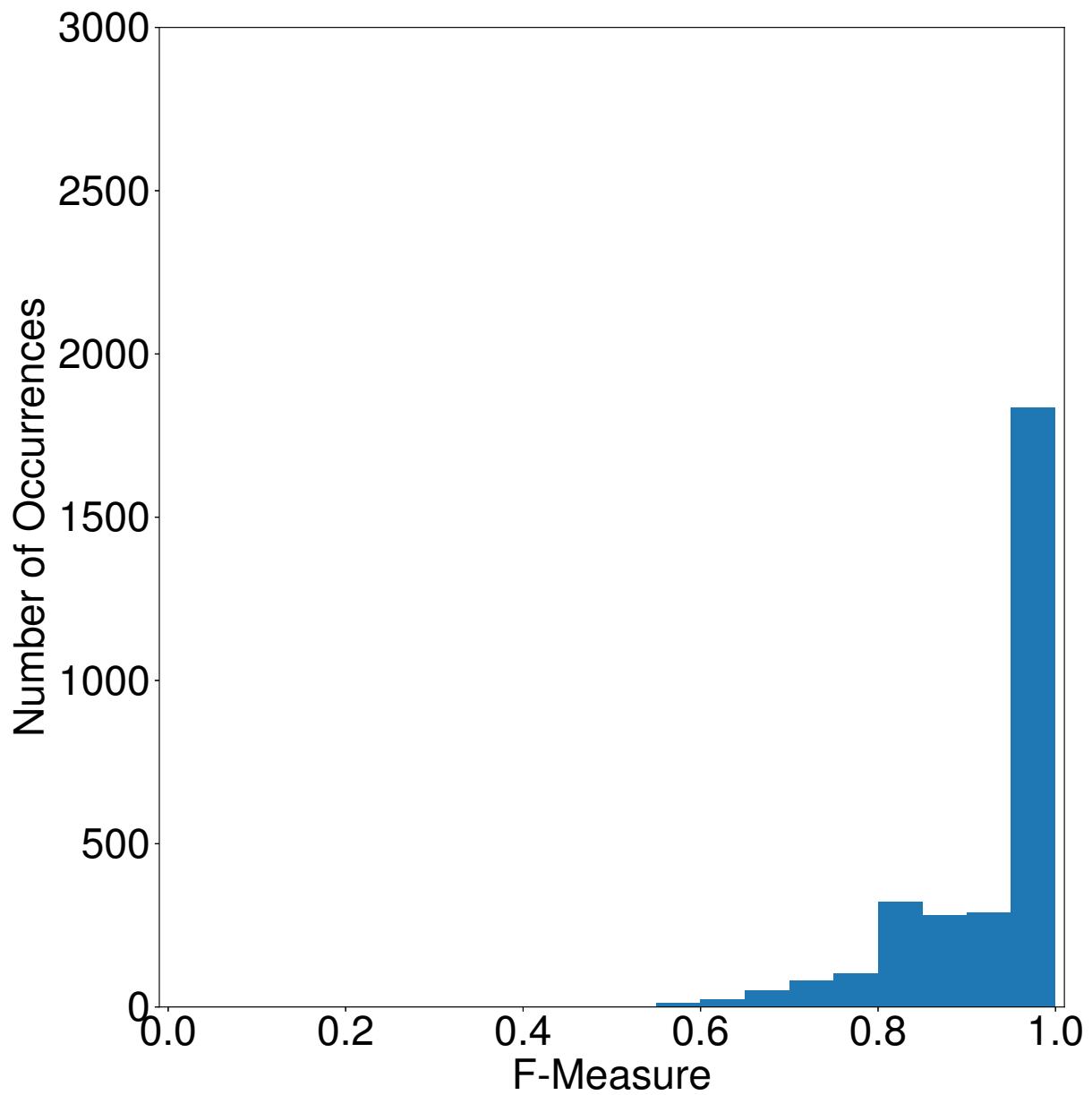


Figure 3.7: F-Measure distribution of DCIGN predictions of test cases.

in Fig. 3.6. The burn maps predicted by the DCIGN do not contain non-physical holes or excessive noise. The direction of maximum growth is captured well. Figure 3.6 shows the DCIGN is able to predict the growth of small (1km^2 , rows 2 and 4), intermediate (10km^2 , rows 1 and 5), and large (100km^2 row 3) fires.

Across all test cases there was not significant bias observed in over-predicting or under-predicting fires, as is shown by the comparable mean sensitivity and precision shown in Table 3.3. However, the spread in sensitivity is higher than the spread in precision as is shown by $X = 80\%$, $X = 90\%$, and $X = 95\%$ shown in Table 3.3. Examining the histogram of F shown in Fig. 3.7, there is a sharp drop in number of occurrences at $F < 0.8$. The initial fire size of 82% of the cases where $F < 0.8$ was found to be nine pixels or less, as shown in Fig. 3.8. Combined with the percentiles shown in Table 3.3, this shows the sensitivity is impacted to a greater extent than precision for small fire sizes. Since a DCIGN relies on feature recognition, low feature density in the inputs (such as fires less than 9 pixels in size) can lead to a decrease in the accuracy of the model. If smaller fires are of primary interest, this effect could be reduced by increasing the spatial resolution of the model.

The sensitivity of the model to noise in the input parameters was examined by adding noise to the 5 examples shown in Fig. 3.6 and examining the impact on S , P , and F . Noise was added by multiplying each input channel by a random value from a log-normal distribution with a mean of 0 and a standard deviation of 1. Note the input and output burn maps were not changed. A total of 3,000 iterations for each of the five test cases were predicted by the DCIGN. The zero noise, noise mean, and noise 80% scores for each case for each metric are shown in Table 3.4. The largest impact was observed in the last test case where the mean and 80% sensitivity dropped from the baseline score by 0.07 and 0.14, respectively. This shows the addition of noise resulted in an increase in the under prediction of the burn map. However, the minimal overall impact on the mean and 80% scores shows the general shape of the predicted burn maps are still in agreement with simulation results.

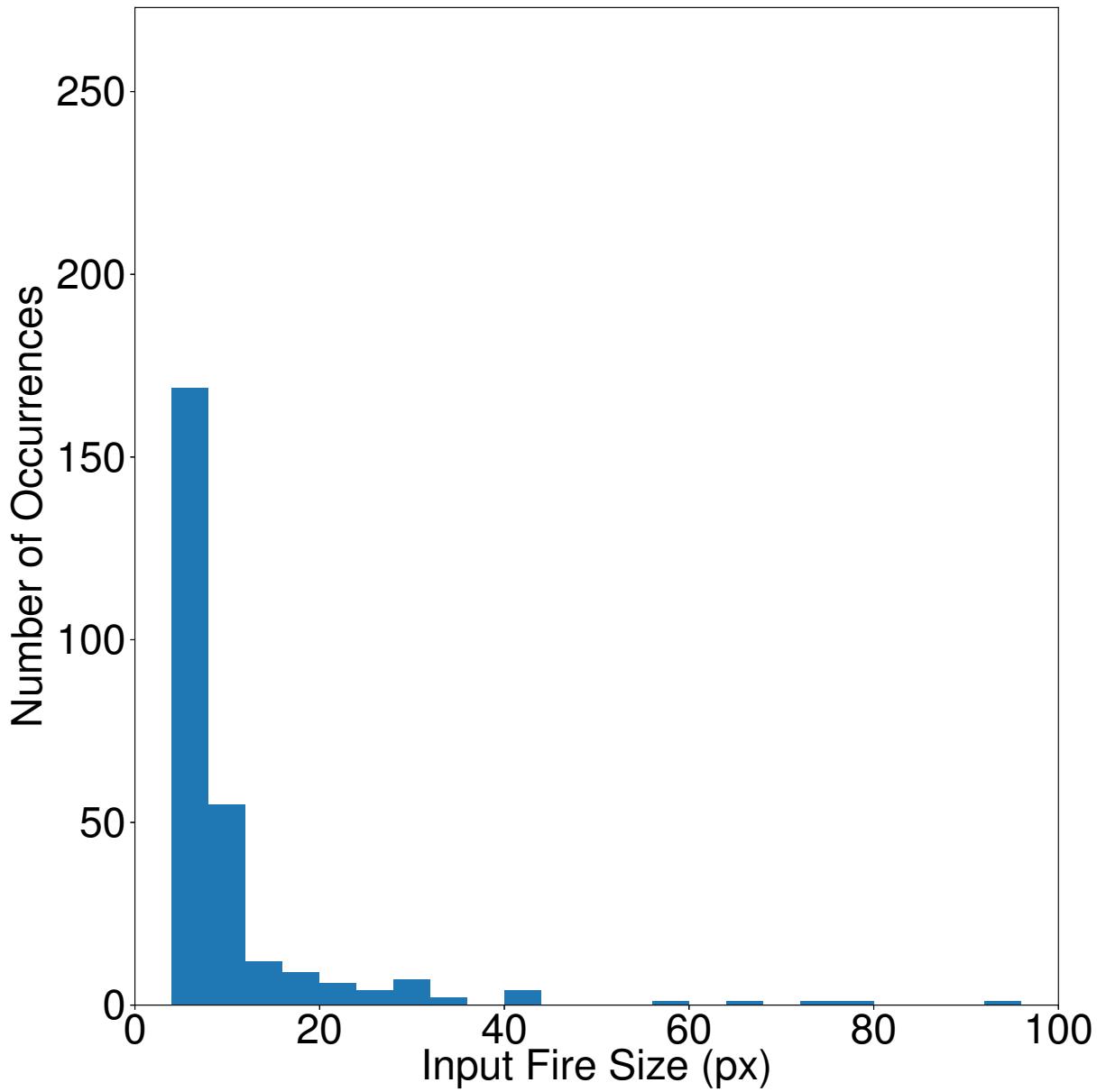


Figure 3.8: Distribution of input fire size for all cases where $F < 0.8$.

Table 3.4: Impact of Noise on Performance Metrics of DCIGN Predictions of Test Cases

Sensitivity			Precision			F-Measure		
Base	Noise	80%	Base	Noise	80%	Base	Noise	80%
0.99	0.98	0.99	0.98	0.98	0.98	0.98	0.98	0.98
0.93	0.91	0.93	1.00	1.00	1.00	0.96	0.94	0.96
0.98	0.97	0.95	0.99	0.99	0.99	0.98	0.98	0.97
1.00	1.00	1.00	1.00	0.93	1.00	1.00	0.95	1.00
0.98	0.91	0.84	0.92	0.90	0.85	0.95	0.90	0.88

One of the key benefits of the DCIGN architecture presented in this work is the scalability in heterogeneous spatial conditions. Recall the total time to run 10,000 simulations using the model of Rothermel was 30.6s compared to the 38.1s required for the DCIGN. The phenomenological model will likely perform faster whenever evaluating the spread of a fire from a single point in homogeneous conditions. However, when using Huygens' principle to evaluate Rothermel's model in heterogeneous spatial conditions, the fire perimeter must be discretized and evaluated for each point along the perimeter. This leads to a significant increase in computational time as the size of the fire increases. Since the DCIGN is already analyzing all the input channels as two-dimensional image channels, there is no increase in computational cost to add heterogeneous conditions. As an example, a 10km fire perimeter with a 30m grid will require 333 evaluations of the phenomenological model, with a total computational time of 1.0s. The DCIGN approach will require a single evaluation, with a total computational time of 0.004s.

Although the model was trained using a six hour time interval between the input and output burn maps, it is possible to obtain predictions at points further in the future at six hour intervals by recursively using the previous prediction as an input to the DCIGN. Figure 3.9 shows five example cases where this process was used to predict burn maps up to 24 hours from ignition based on an input burn map six hours after ignition. The results for each case show the general direction of spread is captured well, with $F > 0.8$ in all cases except the fourth case where the input and

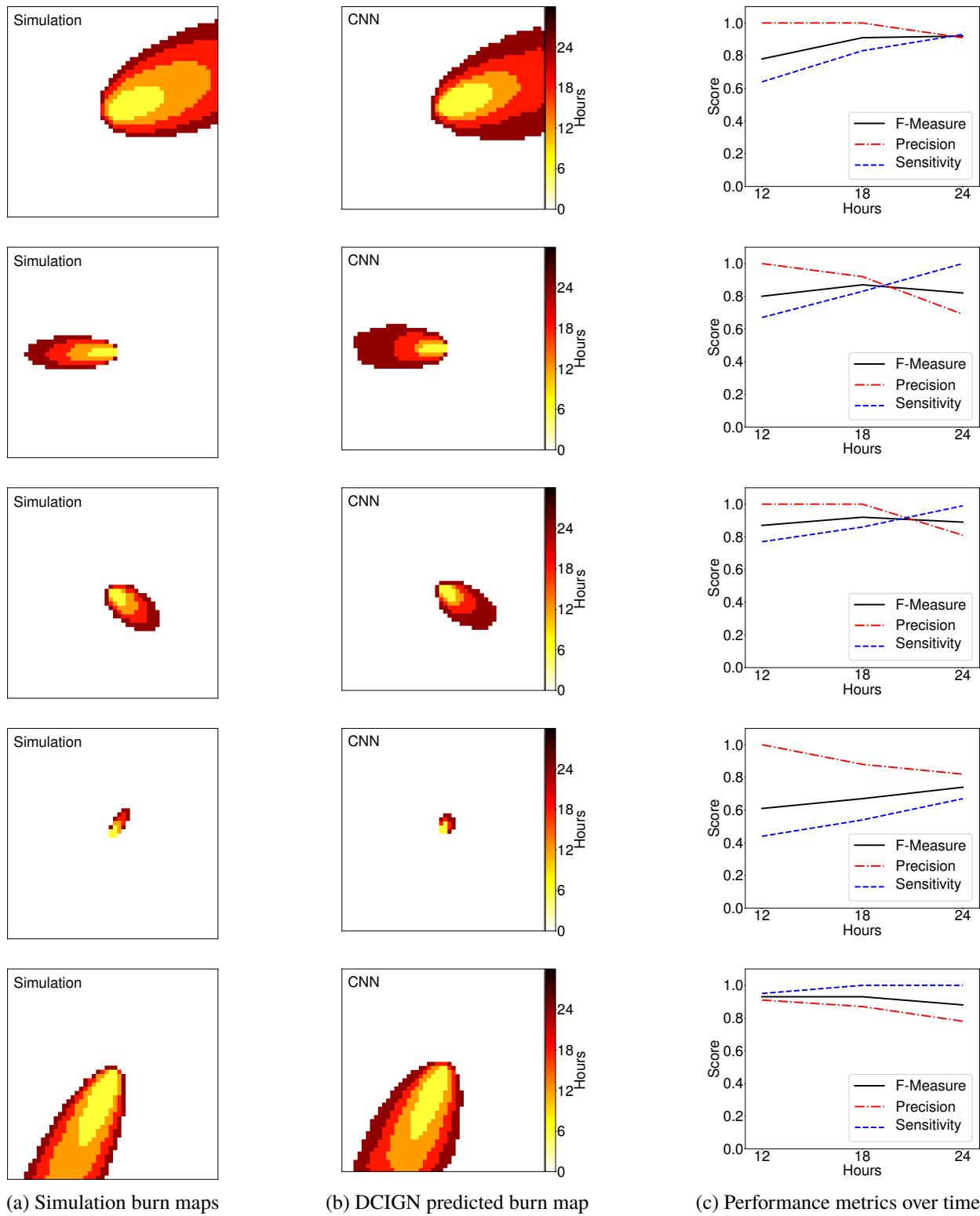


Figure 3.9: Example DCIGN prediction of time resolved burn maps

output fires are small. The sensitivity of the predictions generally increases with time, whereas the precision generally decreases with time. This shows early in the progression of the fire the model under-predicts the rate of spread, which highlights the difficulty the DCIGN can have when dealing with low feature density. Later in the progression of the fire, the model begins to over-predict the rate of spread. Since the network was trained on constant rates of spread for each configuration, the over-prediction is likely a result of the network over-correcting for the initial under prediction of rate of spread.

The robustness of the developed approach was examined by re-training and testing the network using wildland fire simulations with heterogeneous landscape and vegetation conditions. The fire spread simulations were completed using FARSITE which has a two-dimensional implementation of Rothermel's fire spread model [113]. The computational domain for each simulation consisted of a 50x50 km domain with a grid resolution of 0.03km. Each simulation was run for 48 hours of fire spread with fire perimeters output every 1.0 hours of simulation time. Realistic landscape and vegetation maps from the LANDFIRE project were used in this study [123], with 1,000 randomly selected regions from the state of California. For each simulation landscape and vegetation was randomly selected from the set of 1,000 regions, moisture content and wind speeds were randomized. A total of 2,500 FARSITE simulations were used in this study. Burn map pairs were extracted every 6 hours, yielding approximately 17,500 scenarios. An example simulated burn map using FARSITE is shown in Fig. 3.10. The computational time to run each simulation in FARSITE varied depending on the size of the fire perimeter. The cumulative probability density of computational time for the 2,500 simulations used in this study is shown in Fig. 3.11. Approximately 80% of FARSITE simulations took less than hour to run; however, many of the larger fires required 1-6 hours of run time.

The predicted burn-maps and spatial inputs were down-sampled to a 1km spatial grid to match the resolution of the previous study using a median filter. An example set of down-sampled landscape,

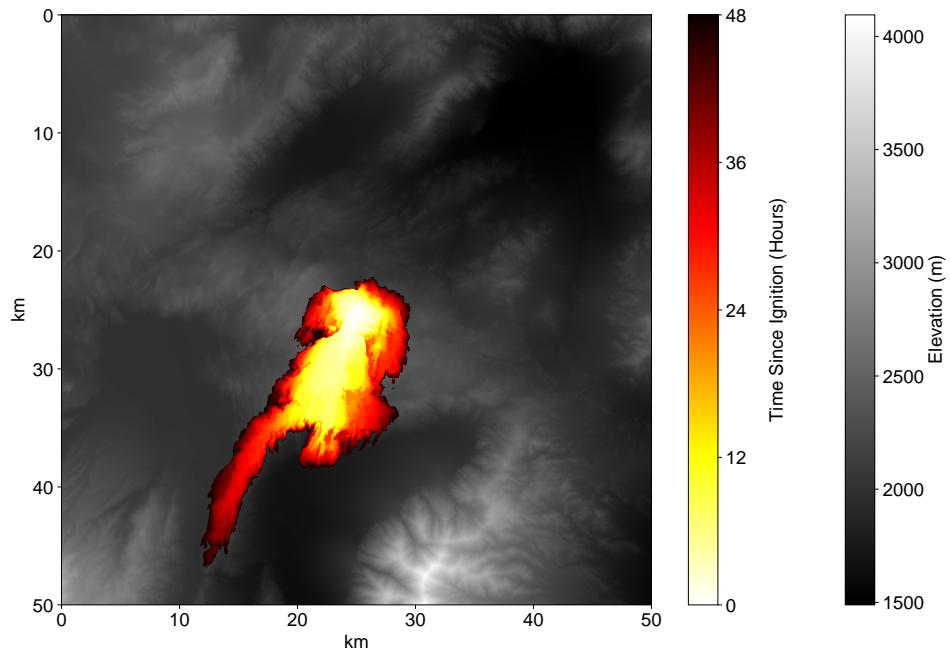


Figure 3.10: Example predicted burn map from FARSITE.

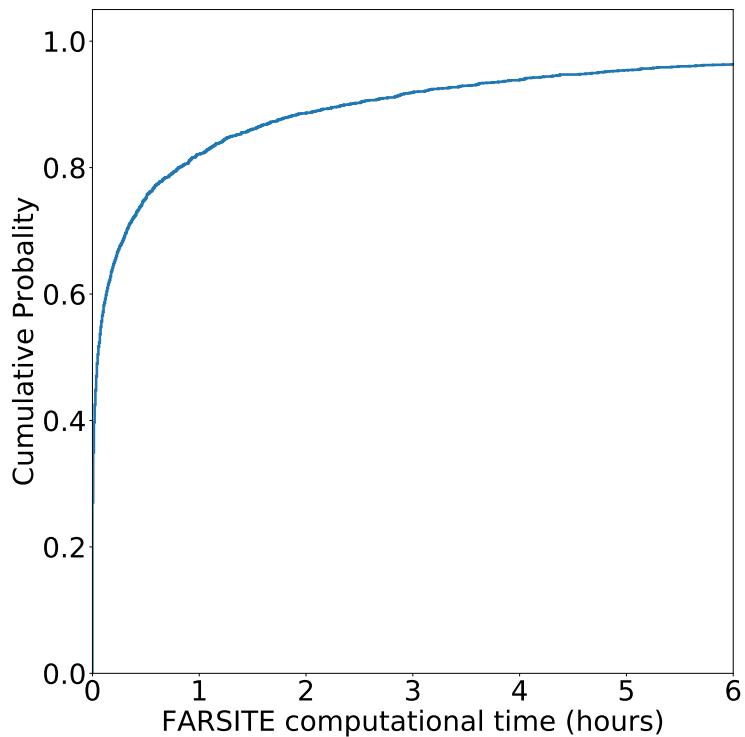


Figure 3.11: Cumulative probability of computational time using FARSITE

vegetation, and ambient parameters which were used as inputs and outputs to the DCIGN are shown in Fig. 3.12. The DCIGN was trained using 15,500 scenarios and tested with 2,000 scenarios. The total computational time to evaluate all 17,500 scenarios using the DCIGN method was 185 seconds, or 0.01 seconds per scenario. This corresponds to a factor of $10^2 - 10^5$ faster than the phenomenological model.

Sample DCIGN predictions from five of the heterogeneous landscape test cases are compared with simulation predictions in Fig. 3.13. Figure 3.13a shows the initial and next burn map from the simulation. Figure 3.13b shows the final burn map predicted by the DCIGN. Figure 3.13c highlights pixels which the DCIGN prediction did not match the simulation predictions. Pixels shown as black represent commission errors (false positive of fire), and pixels shown as orange represent omission errors (false negative of fire).

Figure 3.14a shows the distribution of F-Measure for the DCIGN predictions for all of the heterogeneous test cases. Similar to the DCIGN predictions in the homogeneous case, many scenarios with an initial fire with less than 9 pixels resulted in poor predictions. Figure 3.14b shows the distribution of F-Measure for all cases where the initial fire was greater than 9 pixels in size. Overall the DCIGN predictions for the intermediate to large size fires align well with the simulations. The precision, sensitivity, and F-Measure statistics for the test cases with a initial fire greater than 9 pixels in size are shown in Table 3.5.

Table 3.5: Performance Metrics of DCIGN Predictions of heterogeneous Test Cases

Parameter	Mean	$X = 80\%$	$X = 90\%$	$X = 95\%$
Precision	0.89	1.00	0.92	0.80
Sensitivity	0.80	0.94	0.79	0.68
F-Measure	0.82	0.91	0.82	0.75

The first and fifth example in Fig. 3.13 show regions where the DCIGN predicted fire spread where the simulation did not. One difference in the heterogeneous simulations from the homogeneous

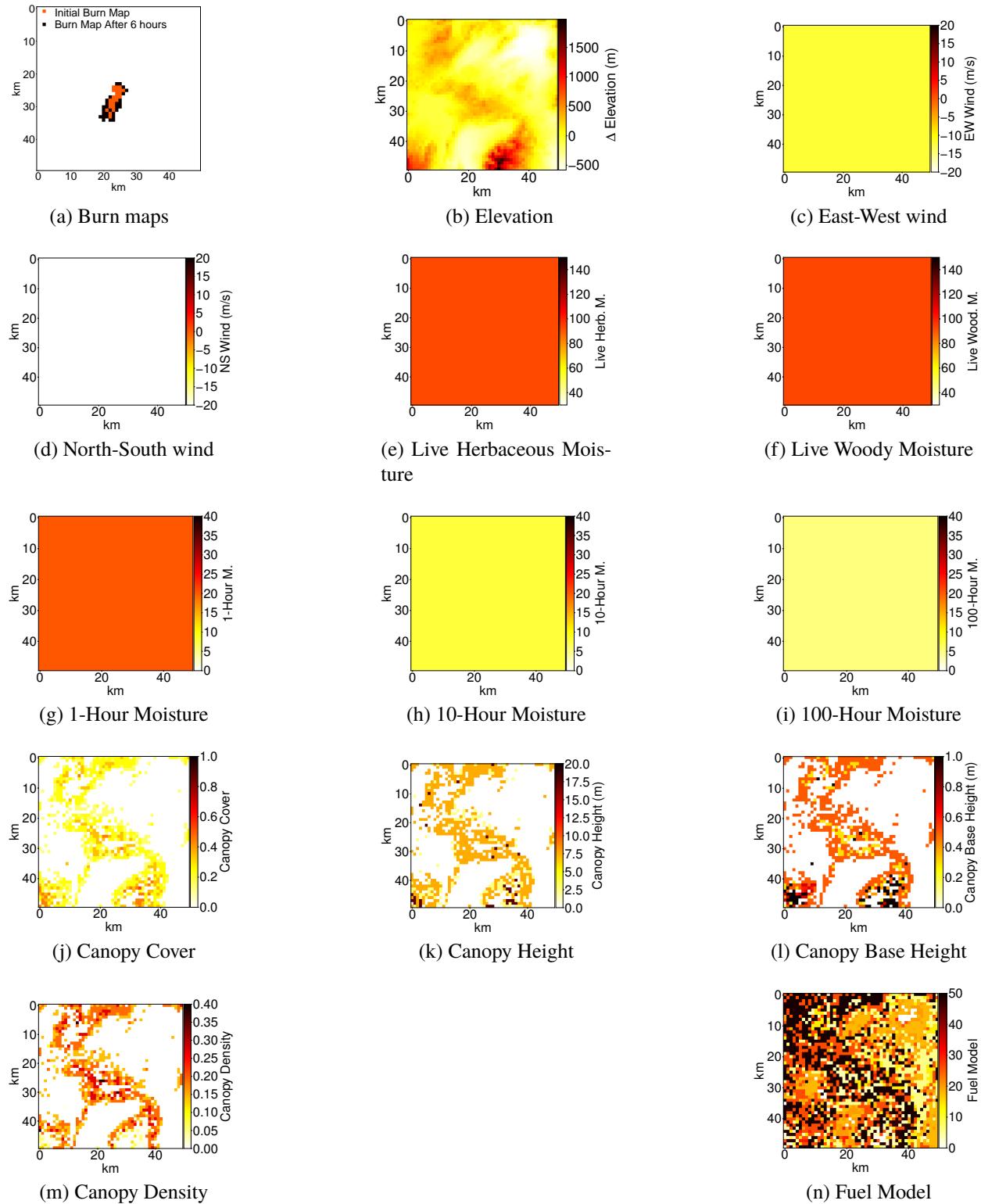


Figure 3.12: Example heterogeneous landscape and vegetation.

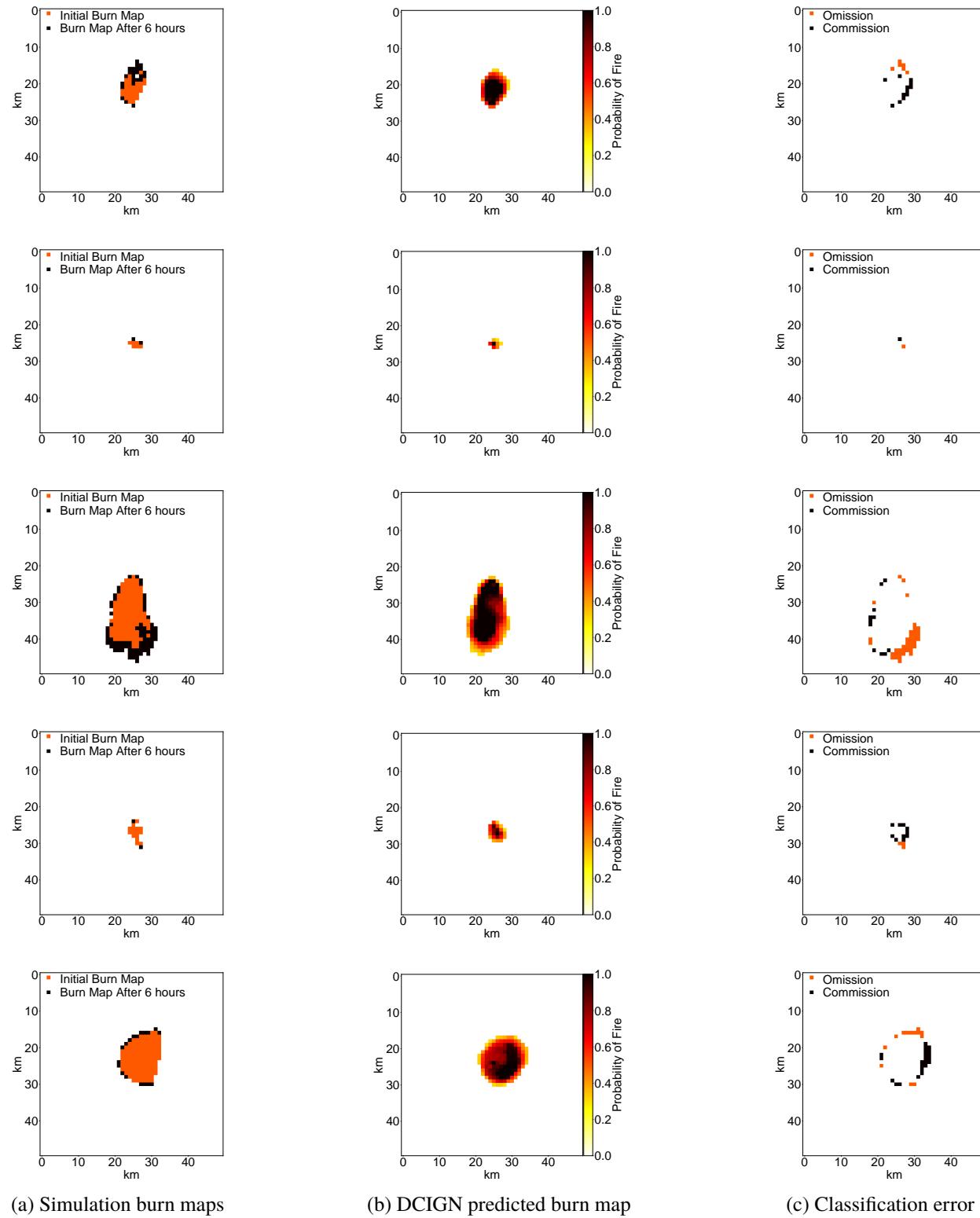


Figure 3.13: Example DCIGN prediction results

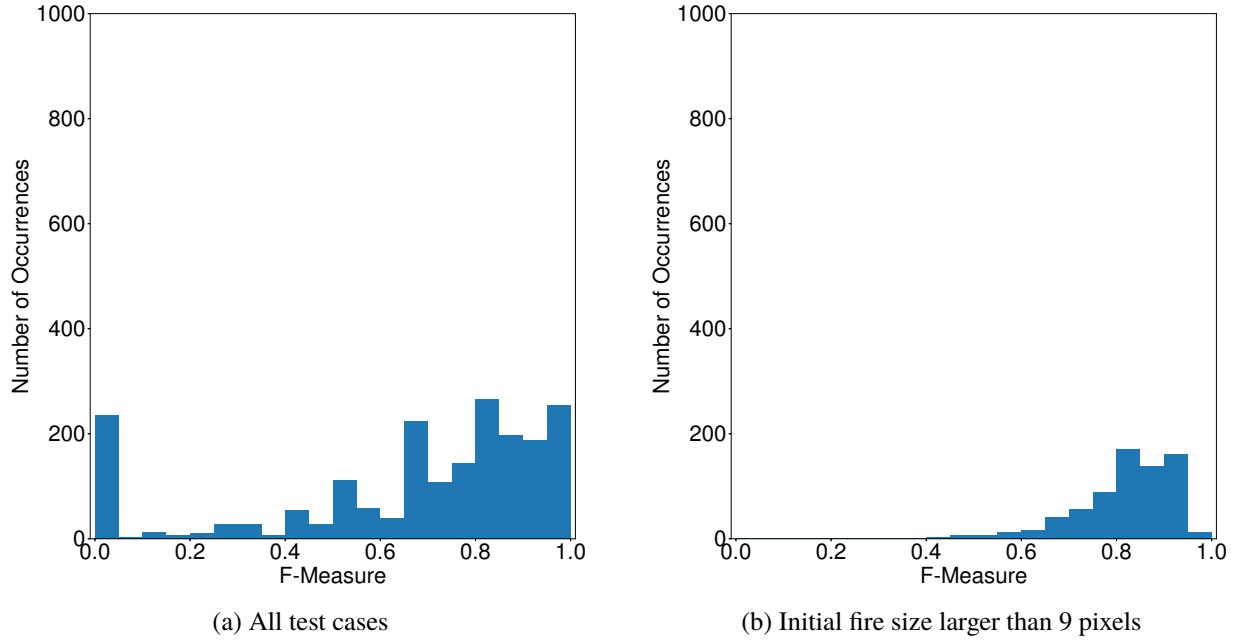


Figure 3.14: F-Measure distribution of DCIGN predictions of heterogeneous test cases

simulations was the presence of non-burnable surfaces (typically corresponding to roads, rivers, etc). It was observed that the non-burnable surfaces often corresponded to only one or two pixels in the landscape file. During the down-sampling process these pixels were often removed which resulted in the DCIGN being given no knowledge of the non-burnable surface. These results show the fire spread predictions over heterogeneous landscapes should be made at a higher resolution to better capture the impact of non-burnable surfaces.

3.6 Conclusion

A novel data-driven approach to predicting the spread of a wildland fire using a deep convolutional inverse graphics neural network (DCIGN) was presented. The robustness of the approach was tested using 1,000 simulations (3,000 burn map pairs) not included when training the network. The predictions of burn maps from the DCIGN-based approach agreed with simulation results, with

mean precision, sensitivity, and F-measure of 0.97, 0.92, and 0.93, respectively across a diverse set of input parameters. Noise in the input parameters was found to have a minimal affect on the DCIGN predictions. The computational cost of the method was found to be comparable to simulating a phenomenological model for homogeneous spatial conditions, and a factor of $10^2 - 10^5$ faster than the phenomenological model under heterogeneous spatial conditions. Although trained on predictions six hours apart, the DCIGN-based approach is capable of predicting burn maps further in the future by recursively using previous predictions as inputs to the model. The cases where F-measure was observed to be less than 0.80 were found to have input burn maps of less than nine pixels. This indicates as the fire continues to grow, the predictions will continue to improve. A preliminary study predicting spread over heterogeneous landscapes showed the method has the potential to provide rapid estimates of fire spread; however, fire spread over heterogeneous landscapes should be made at a higher resolution to better capture the spatial variation.

This work represents a first step in creating a framework to predict wildland fire spread based on physics based models and historic data. Although the data used to train the DCIGN in this work was generated using a phenomenological model, the model does not have any information about whether the data is from a computational fluid dynamics model, phenomenological model, or even experimental measurements of burn maps. Additionally, the preliminary study with heterogeneous vegetation and landscapes shows the feature learning aspect of the DCIGN-based method is well posed to predict realistic fire spread conditions. The next step in this process is to increase the working resolution and optimize the network architecture for heterogeneous spatial conditions.

Chapter 4

Compartment Fire Predictions Using Transpose Convolutional Neural Networks

4.1 Abstract

This paper presents a novel data-driven approach to predict spatially resolved temperatures and velocities within a compartment based on coarse zone fire modeling using a transpose convolutional neural network (TCNN). A total of 1,000 Fire Dynamics Simulator simulations (FDS) of a simple two-compartment configuration with different fire locations, fire sizes, ventilation configurations, and compartment geometries were used in training and testing the model. Although the model was trained and tested using a simple two-compartment configuration, the TCNN approach was validated with two more complex multi-compartment FDS simulations by processing each room individually. Overall the flow fields in the multi-compartment validation tests agreed well with FDS predictions with 95% of TCNN predicted temperatures within $\pm 11\%$ of FDS predictions, and 95% of TCNN predicted velocities within ± 0.25 m/s of FDS predictions. The computational time for the TCNN approach is less than 1 second per evaluation (not including the zone model simulation), whereas the computational time required for FDS is 1,200-3,600 seconds for the two-room configuration and 9,300-22,800 seconds for the multi-room configuration. This is a 1,200-22,800x decrease in computational time. This work represents a first step in creating a framework to predict spatially and temporally resolved flow fields in compartment fires at the same

speed as coarse zone fire models.

4.2 Introduction

Understanding the transport of energy and combustion products in a compartment fire is vital in fire hazard analysis. Although advancements in computing technology have made computational fluid dynamics (CFD) of compartment fires possible, the computational cost can be prohibitive. Parametric studies and predictions of large structures often rely on more coarse predictions such as those from zone fire models. Researchers have presented multi-scale modeling approaches to fuse three-dimensional CFD with one-dimensional zone models; however, each approach inherits the computational cost of CFD. Recent advancements in the field of generative modeling using machine learning have made it possible to develop rapid data-driven predictions from a small vector of high level descriptors. A new compartment fire model which uses the predictions from a zone fire model as inputs to a generative machine learning model may be able to provide high resolution predictions of intensive properties at low computational costs.

CFD fire models solve conservation equations of mass, momentum, and energy for each grid cell, often numbering $10^5 - 10^7$ [124]. These models are often used in fire protection engineering to predict complex flow fields for smoke control, to estimate smoke detector and sprinkler activation times, and to estimate ignition times of different surfaces. However, the high spatial-temporal resolution comes at a computational cost, where simulation times are often measured in hours, days, or even weeks depending on the simulation. In addition the cost associated with CFD predictions of large domains or parametric studies can be prohibitive.

More rapid predictions over large domains are possible with zone models. Zone based compartment fire models take advantage of the thermal stratification typical in enclosure fires to represent each room as either one or two well-mixed gas regions. Since the number of grid points using a one-

dimensional zone model is significantly less, rapid predictions of the mean transport of energy and species through compartments can be calculated. Predictions of mean properties have been shown to correlate well with experimental measurements [125]. It is not possible to determine local variations in temperature and flow velocity using a zone fire model since the gas regions are assumed to be well-mixed. Although zone fire models provide valuable insight, they are not capable of making predictions for complex configurations and lack the resolution that is sometimes needed in hazard analysis.

Multi-scale modeling approaches couple a CFD model in regions with high spatial variation (near to fire) with a more coarse model further away. This type of approach has been used in many structural applications including contaminant dispersion [126, 127], energy efficiency [128, 129, 130, 131, 132, 133], and fire response [134, 135, 136, 137, 138, 139, 140, 141]. Researchers in the area of tunnel fire modeling have shown this type of approach can reduce computational times with relatively low loss in accuracy [140, 141]. Although the computational time reduction can be significant, simulations of large domains are on the order of hours.

Generative modeling is a new field of research in the machine learning community which has focused primarily on developing techniques to generate RGB images. Researchers have started to apply generative modeling techniques to CFD applications such as studying wake flow dynamics [41, 42, 43, 44], modeling turbulence [45, 46], and accelerating the computations of time-dependent ODEs and PDEs [47, 48, 49]. Chang presented an overview of current neural network based CFD modeling techniques in [142]. The most similar of these works to the present study was presented by Lee *et al.* [43]. The authors used a deep recurrent attentive writer (DRAW) generative network similar to [99] to predict unsteady flow fields over cylinders at different Reynolds numbers. The generative network used four sequential flow field predictions to predict a fifth flow field to good effect. However, the predictions are based on CFD calculations at a similar resolution to the network predictions. Researchers have not applied generative modeling techniques to model reacting flow

fields or predict high resolution thermal flow fields from coarse CFD predictions from zone models.

There are two learning tasks associated with generative modeling: the data model (optimal vector to describe the data), and the generator (architecture to produce a detailed description from the input vector) [61]. Much of the work in the literature focuses on unsupervised learning of the data model [143, 144], or coupling these two tasks [87, 145, 146]. These type of approaches excel at developing classification schemes through unsupervised learning; however, in physics modeling regression problems are typically of interest which are not trivial to approach in unsupervised learning. This work was inspired by the approach presented by Dosovitskiy *et al.* [61, 62] who focused on developing an up-convolutional neural network approach to the generator task to create two-dimensional rendered images of three-dimensional objects using an assumed data model and supervised learning. The data model consisted of a vector describing the class of the object, perspective information, and transformation information.

The objective of this study was to develop a data-driven generative model to provide spatially resolved predictions of temperature and velocity. The vector of inputs to the generative model included geometric parameters and predictions of physical parameters from the zone fire model (such as mass flow rates and gas temperatures at doors). A total of 12 output channels were predicted which consisted of two-dimensional slices of temperature and three-component velocity through two vertical center-lines of the compartment, and one horizontal plane 0.1m below the ceiling. The network architecture used in this work was a transpose convolutional neural network (TCNN) similar to that of Dosovitskiy [61]. Data for use in training and testing the model was generated using Fire Dynamics Simulator (FDS) version 6.2 developed by NIST [40]. The following sections describe the network architecture, data generation, and results. This work represents a first step in creating a framework to predict spatially and temporally resolved flow fields in compartment fires at the same speed as coarse zone fire models.

4.3 Methods

A schematic showing a high level view of the solution algorithm is shown in Fig. 4.1. Initial predictions of a fire scenario are made using a zone fire model. These predictions and a description of the geometry are input to the TCNN. The TCNN then uses its prior training to predict the thermal flow-field within each compartment individually. These predictions are then post-processed to output final predictions. The following subsections describe the network architecture, data generation, and post-processing used in this work.

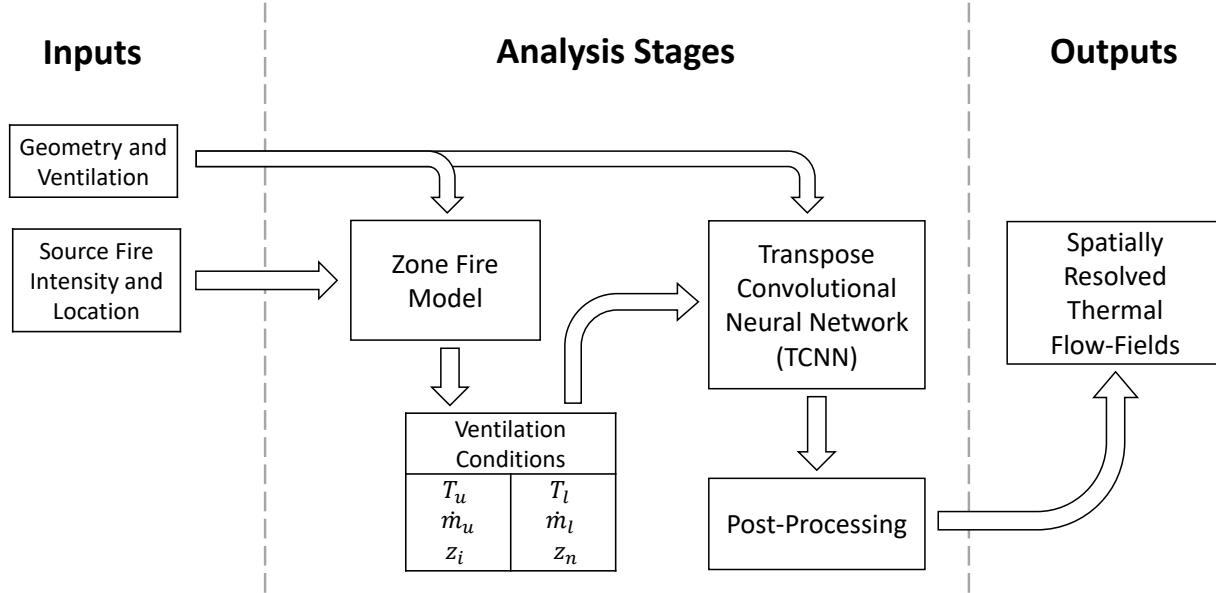


Figure 4.1: Schematic of solution algorithm.

4.3.1 Network Architecture

The artificial neural network approach presented in this work falls in the class of generative modeling in the machine learning community. The focus of generative modeling is to use a small number of input parameters to predict a significantly more detailed output. The specific inspiration for this work comes from the work of Dosovitskiy *et al.* [61]. The authors created a

transpose convolutional neural network which would render a two-dimensional image of different three-dimensional objects viewed from different angles and lighting from high level descriptors of the object. The authors observed after training that the TCNN was able to accurately interpolate new views of the objects as well as extrapolate beyond the views in the training set. For the TCNN architecture presented herein, the high level descriptors are replaced with coarse zone fire model predictions of compartment conditions, and the rendered images are replaced with spatially resolved slices of gas temperature and velocity.

An example CFD simulation of a compartment fire is shown Fig. 4.2a. Enclosures in a zone fire model are typically split into two zones, the hot gas layer and the lower layer. Each layer is assumed to be well mixed with a single mean value for each intensive property (e.g. temperature, concentration, pressure) representative of the space. Conservation laws are solved for each zone to predict the change in mean value over time. Typical predictions from a zone fire model include predictions of temperature in the upper and lower layer and the smoke layer height/interface height, in addition to mass flow rates and neutral plane predictions for each connection to other rooms. An example showing CFD predictions of temperature and interface height for a room are shown in Fig. 4.2b, and predictions of mass flow rates and neutral plane height through a door are shown in Fig. 4.2c.

The TCNN architecture used in this work is shown in Fig. 4.3. The architecture consists of a total of eleven hidden layers, of which nine are trained processing layers. The input layer consists of a vector of 35 points. Three numbers correspond to the basic compartment geometry (L , W , H). The remaining 32 points are split evenly between each wall surface (North, South, East, and West) for a total of eight numbers describing the junction to the neighboring space for each wall. These eight points describe the geometry and flow conditions of the ventilation. The points contained in the input vector are summarized in Table 4.1. The three dense (fully-connected) layers are used to develop a high dimensional representation of the input parameters prior to processing through

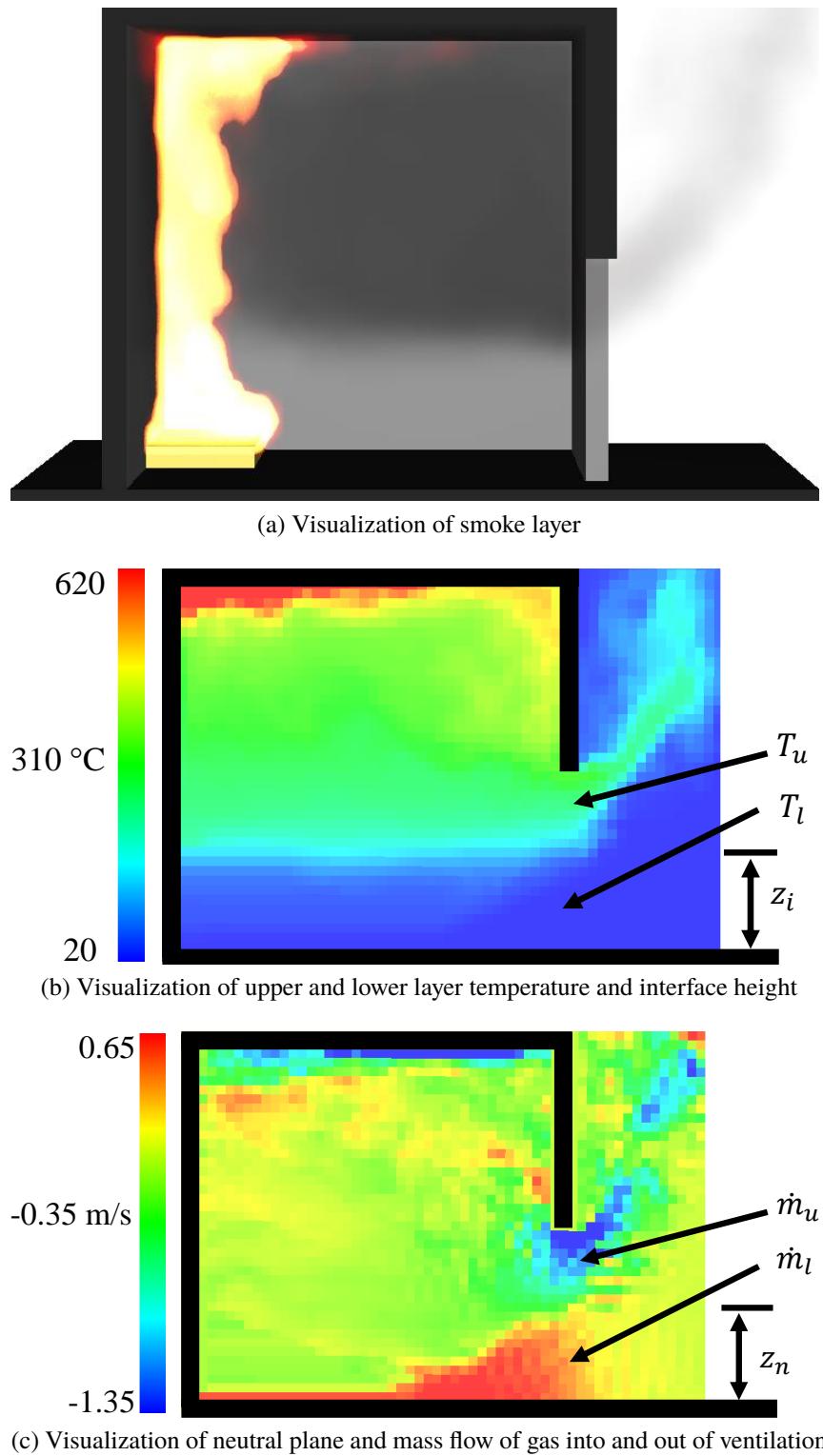


Figure 4.2: Example compartment fire simulation

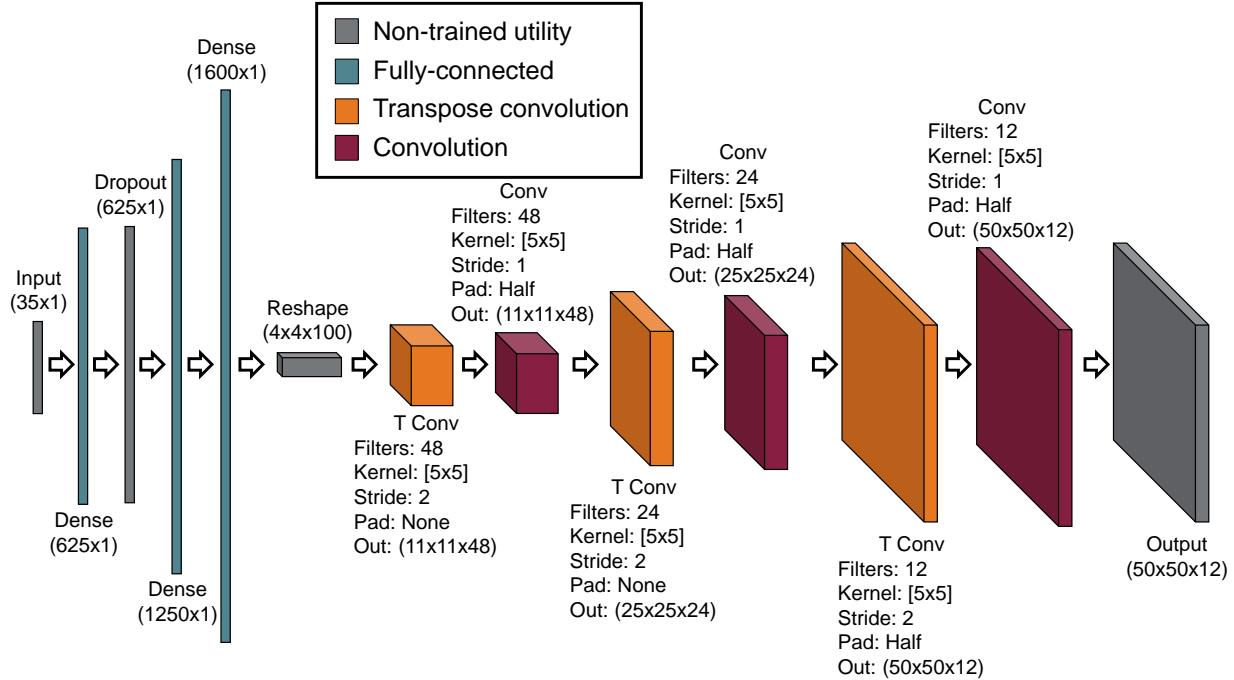


Figure 4.3: Schematic of the TCNN architecture used in this work.

the TCNN and CNN layers. A single dropout layer after this first fully-connected layer was used to prevent over-fitting during training [78]. Similar to Dosovitskiy *et al.*, the x-y resolution of the data was increased through 3 cycles of back to back TCNN and CNN layers. The output images were 50x50 pixels with twelve image channels corresponding to spatially resolved temperature, x-axis velocity (U-velocity), y-axis velocity (V-velocity), and z-axis velocity (W-velocity) through the room. To demonstrate the approach, two vertical slices were located along the center-line of the x-axis and y-axis as well as one horizontal slice was located 0.1m below the ceiling. The number of filters and step size in each CNN and TCNN layer and the number of neurons in the fully connected layers were specified to steadily increase the degrees of freedom of the system from 35 in the input layer to 30,000 in the output layer. All layers in the TCNN network used leaky rectified linear unit (ReLU) activation functions, as recommended by other researchers for regression problems [52, 53].

The network architecture was built using the Python 3 bindings for TensorFlow [106]. The model

Table 4.1: Description of points in input vector.

Variable	Name	Description
L	Length	Total extent of room along the x-axis
W	Width	Total extent of room along the y-axis
H	Height	Total extent of room along the z-axis
S	State	Whether ventilation is open or closed
D_S	Size	Class of ventilation size. 0: 0.625x1.9m 1: 1.25x1.9m 2: Full wall
T_u	Upper layer temperature	Temperature above the interface height
T_l	Lower layer temperature	Temperature below the interface height
z_i	Interface height	Temperature above the interface height
\dot{m}_u	Upper layer mass flow rate	Flow rate above the neutral plane
\dot{m}_l	Lower layer mass flow rate	Flow rate below the neutral plane
z_n	Neutral plane height	Height of inflection point in gas velocity

was trained using stochastic gradient descent with a batch size of 100 samples. All weights and biases were initialized from a uniform distribution between -1 and 1. The learning rate was fixed for all layers throughout training at 10^{-4} . The cost function used in training was based on square error, as recommended by Dosovitskiy. Over-fitting was reduced by using 75% dropout on the dropout layer and shuffling the order of the samples during training. The network was trained using 1,000 simulations (4,990 unique scenarios) for 20,000 cycles using a single NVIDIA Quadro K620. The total time to train the network was 48 hours. The total time to for the TCNN to predict the spatially resolved temperature and velocities for 5,000 scenarios was 8 seconds.

4.3.2 CFD Data Generation

It would be ideal to use high resolution fire simulations of entire structures when training the neural network. However, since numerous simulations are required (on the order of 1,000-10,000) to train the model, the computational cost of generating a data set using full structures was prohibitive.

Instead a two-room configuration was used to generate data for use in training the model. An example schematic of a two-room configuration is shown in Fig. 4.4. This simplification is reasonable as long as the range of initial conditions, opening flow conditions, and boundary conditions of the examined room in the two-room configuration capture the range of realistic conditions for the rooms in a full structure. This variation was modeled by varying the geometry of Room 1, the position of the source fire within the two-room configuration, and the intensity of the source fire. For this preliminary study the impact of thermal losses to the boundary were removed by considering all boundary surfaces as adiabatic.

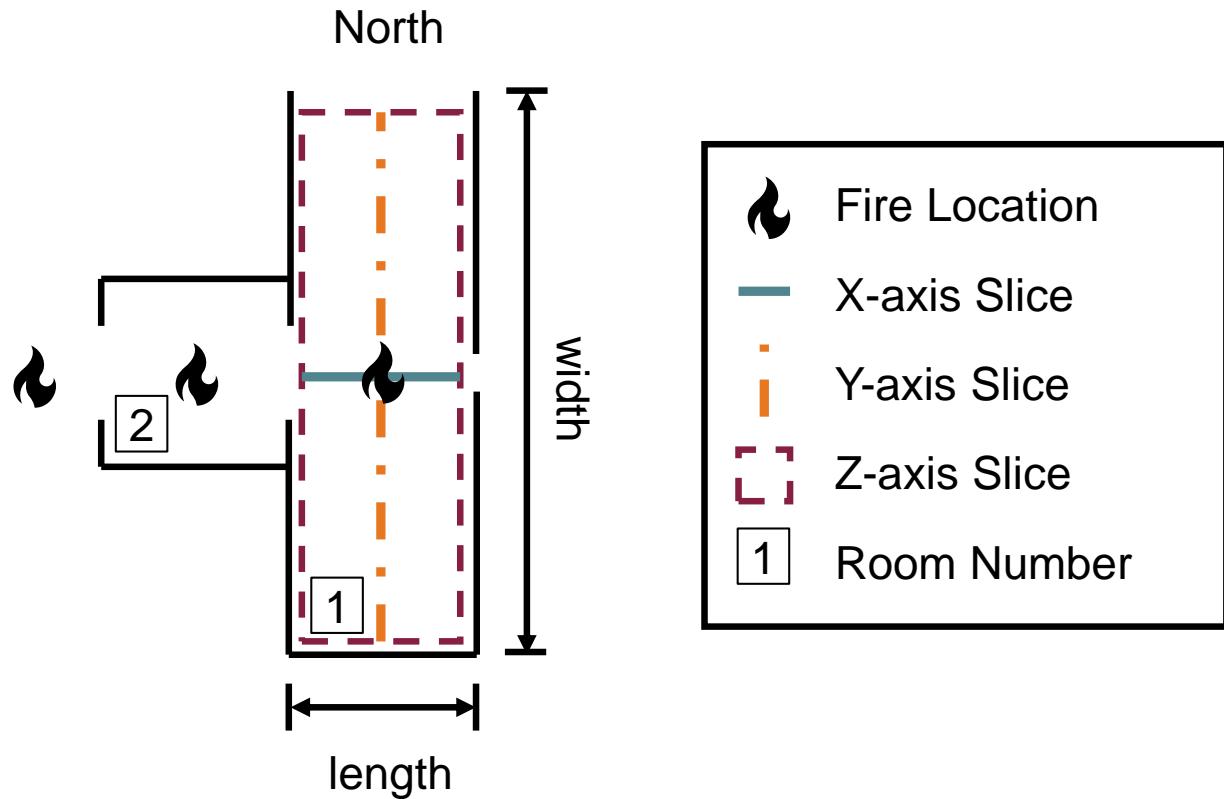


Figure 4.4: Schematic of two-room configuration. Predictions of temperature and velocity in Room 1 were used in training the model.

The parameters varied in this study are shown in Table 4.2. Either the length or width of Room 1 was fixed at 2.5 m and the other selected from the range of 2.5-10.0 m. The existence of a door on each wall was selected independently; however, at least one door was required to allow air flow. If a

door was present, it was selected to be one of three possible sizes: $0.625 \times 1.9 \text{ m}$, $1.250 \times 1.9 \text{ m}$, or the full extent of the wall (representing a hallway connection). The heat source was randomly selected to be either a fire in the center of Room 1, a fire in the center of Room 2, or a two-layer forced ventilation outside Room 2 (representing a distant fire). If the heat source was not in Room 1, a door was always open between Room 1 and Room 2. If the source was a fire, the heat release rate was selected from 50-1,000 kW . If the source was forced ventilation, the upper layer temperature was selected from 500-1,000 $^{\circ}\text{C}$, the velocity from 1-10 m/s , the interface height from 0-1.9 m , and the neutral plane from 0-1.9 m .

Table 4.2: Limits of each parameter in study.

Room Geometry	
Width	2.5-10.0 m
Length	2.5-10.0 m
Height	2.5 m
Wall Ventilation	North, East, South, West, West-West
Existence	True/False
Ventilation Size	0. $0.625 \times 1.9 \text{m}$ door 1. $1.250 \times 1.9 \text{m}$ door 2. Full wall
Source Configuration	
Location	0. Fire in Room 1, 1. Fire in Room 2, 2. Forced ventilation west of Room 2
Intensity	
If Fire	
Heat Release Rate	50-1000 kW
If Two-Layer Forced Ventilation	
Temperature	500-1000 $^{\circ}\text{C}$
Velocity	1-10 m/s
Interface Height	0-1.9 m
Neutral Plane	0-1.9 m

A total of 30s of exposure was simulated for 1,000 unique configurations using FDS. The grid

resolution was fixed at $0.1m$ along each axis for each simulation to limit the computational cost per simulation. Although this grid resolution does not fully resolve the source fire, the predictions were adequate for this preliminary analysis, where the focus was on reproducing CFD simulations from zone simulations rather than comparing to experimental results. For each simulation, the temperature, U-velocity, V-velocity, and W-velocity along the x-axis center-line, y-axis center-line, and z-axis just under the ceiling were averaged from 10-30s of exposure and output as two-dimensional data files. The size of the data-set was artificially increased through rotation up-sampling and ambient re-sampling. Rotation up-sampling was used to mitigate the impact of Room 2 being in a fixed position relative to Room 1 on the TCNN predictions. This process consisted of rotating the coordinate axes of the simulation without adjusting the numeric values. Since all FDS simulations included in the data-set originally included a heat source near the room of interest, preliminary predictions from the TCNN approach failed for rooms far from the initiating fire. The impact of this bias was mitigated through ambient re-sampling which physically corresponds to adding a non-flaming case for each geometric configuration. This process consisted of fixing all temperatures in input and output data to an ambient temperature of 20°C and fixing the mass flow rates in the upper and lower layer to zero. A total of 5,000 unique scenarios were in the database after rotation up-sampling and ambient re-sampling.

4.3.3 Zone Fire Model Data Generation

Estimates of zone model predictions were developed from the CFD predictions rather than using a zone model to predict the scenarios. This decision was made to ensure the errors in predictions were from the TCNN approach and not from differences in the CFD and zone modeling schemes.

The reduction to a two layer estimate was calculated using the method presented by Janssens as recommended in the FDS validation guide [40, 147]. The interface height was calculated using the

equations

$$I_1 = (H - z_{int}) T_u + z_{int} T_l = \int_0^H T(z) dz \quad (4.1)$$

$$I_2 = (H - z_{int}) \frac{1}{T_u} + z_{int} \frac{1}{T_l} = \int_0^H \frac{1}{T(z)} dz \quad (4.2)$$

$$z_{int} = \frac{T_l (I_1 I_2 - H^2)}{I_1 + I_2 T_l^2 - 2 T_l H} \quad (4.3)$$

where H was the total height in m , z_{int} was the interface height in m , T_u was the upper layer gas temperature, T_l was the temperature of the lowest grid cell in K , and $T(z)$ was the center-line temperature through the ventilation. The upper layer temperature was calculated using the equation

$$T_u = \frac{1}{(H - z_{int})} \int_{z_{int}}^H T(z) dz. \quad (4.4)$$

The neutral plane, z_n is defined as the height at which the direction of the air flow switches from an inflow to an outflow and was determined directly from the CFD predicted velocity profile. The mass flow rate of the upper and lower layer, \dot{m}_u and \dot{m}_l respectively, were calculated using the equations

$$\dot{m}_u = \int_{z_n}^H \int_0^x v(x', z) \rho(x', y) dx' dz \quad (4.5)$$

$$\dot{m}_l = \int_0^{z_n} \int_0^x v(x', z) \rho(x', y) dx' dz \quad (4.6)$$

where x is the width of the ventilation, and ρ is the density of the air. The density of the air was calculated using the equation

$$\rho = \frac{353.4}{T}. \quad (4.7)$$

4.3.4 Post Processing

There were times when the temperature predicted by the TCNN was less than ambient. Since this is non-physical for a fire scenario, any temperatures less than ambient were fixed to ambient in post-processing. Given enough data and training cycles, the TCNN would learn this relationship, but it was decided it was not necessary for this preliminary study. The raw temperature and velocity predictions from the TCNN were found to have some noise which was not present in the CFD predictions. This noise was minimized by using a Gaussian filter to smooth the predictions. The standard deviation in the temperature and velocity error from the training data was used to determine the optimal standard deviation for the filter, as shown in Fig. 4.5. The temperature and velocity errors were found to be a minimum with a kernel size of 0.50 and 1.00 for the Gaussian kernel, respectively. Since this post-processing operation is similar to a convolutional layer in the TCNN, the TCNN would learn this relationship given enough data and training cycles as well.

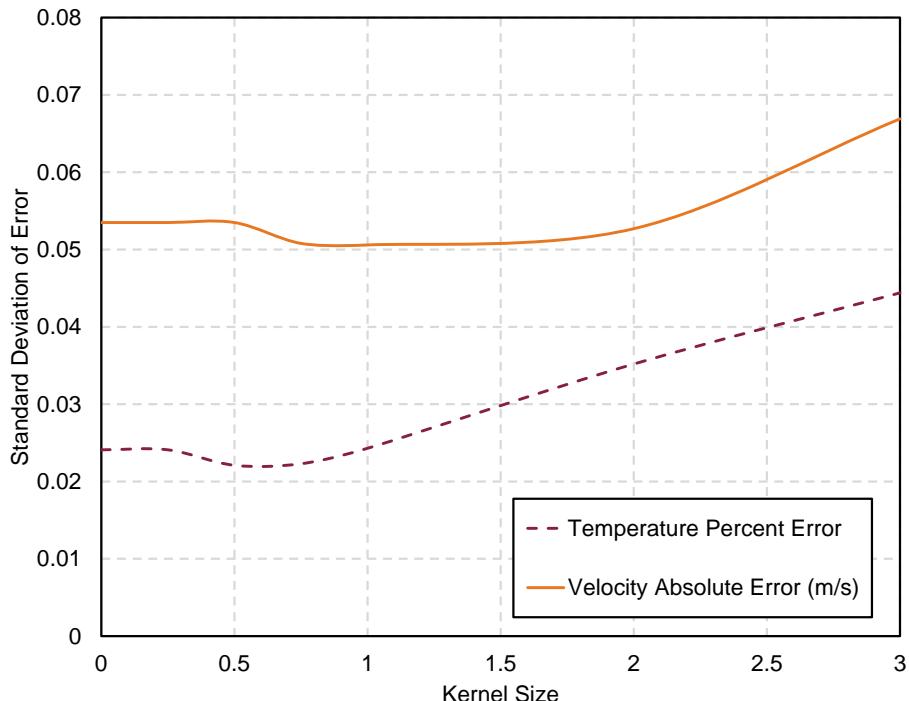


Figure 4.5: Standard deviation of error from training data with different standard deviation kernels for Gaussian filter.

4.4 Results

Neural network predictions of temperature and velocity are compared with CFD simulations in Fig. 4.6- 4.11 for two scenarios not included when training the network. In Figs. 4.6- 4.11 the rows correspond to the different two-dimensional slices predicted by the network. The left column of each image pair is the neural network prediction. The middle column of each image pair is the CFD prediction. The contour axis is scaled the same for each image pair. The histogram on the right shows the error in the neural network prediction compared to the CFD prediction. The titles correspond to the slice location and intensive property shown. Figure 4.6 shows the x-axis center-line temperature and three-component velocity comparison for Testing Case 1. Figure 4.7 shows the y-axis center-line temperature and three-component velocity comparison for Testing Case 1. Figure 4.8 shows the z-axis temperature and three-component velocity 0.1m below the ceiling comparison for Testing Case 1. Figure 4.9 shows the x-axis center-line temperature and three-component velocity comparison for Testing Case 2. Figure 4.10 shows the y-axis center-line temperature and three-component velocity comparison for Testing Case 2. Figure 4.11 shows the z-axis temperature and three-component velocity 0.1m below the ceiling comparison for Testing Case 2.

The percent error is shown for temperature predictions, calculated as

$$E_{T,PD} = \frac{T_{cf} - T_{TCNN}}{T_{cf}} \times 100 \quad (4.8)$$

where $E_{T,PD}$ is the percent error, T_{TCNN} is the temperature predicted by the neural network, and T_{cf} is the temperature predicted by CFD. Since many of the velocities are close to zero, the percent error was not a good metric to examine the robustness of the predictions. Instead the absolute error

is shown for the velocity predictions, calculated as

$$E_{V,ABS} = V_{cf} - V_{TCNN} \quad (4.9)$$

where $E_{V,ABS}$ is the absolute error, V_{TCNN} is the velocity predicted by the neural network, and V_{cf} is the velocity predicted by CFD. For the examples shown in Fig. 4.6- 4.11 temperatures are typically within $\pm 8.6\%$ and velocities within ± 0.18 m/s.

The discrete probability density function of temperature percent error and velocity absolute error for all training and test data is shown Fig. 4.12 and Fig. 4.13, respectively. The mean and standard deviation of error from the training and test data sets is summarized in Table 4.3. An error of zero corresponds to perfect agreement between the ANN and simulation predictions. Overall the error was low for both temperature percent error and velocity absolute error.

Table 4.3: Summary of performance of TCNN predictions on training and test data sets, values correspond to $\mu \pm \sigma$.

Intensive Property	Location	Training Set Error	Testing Set Error	Units
Temperature	x-center	-0.8 ± 2.5	0.0 ± 4.6	%
Temperature	y-center	0.3 ± 2.2	0.9 ± 4.8	%
Temperature	z-top	0.3 ± 1.7	1.3 ± 3.3	%
U-velocity	x-center	0.01 ± 0.05	0.01 ± 0.07	m/s
U-velocity	y-center	0.01 ± 0.06	0.01 ± 0.08	m/s
U-velocity	z-top	0.02 ± 0.05	0.02 ± 0.10	m/s
V-velocity	x-center	0.02 ± 0.06	0.03 ± 0.09	m/s
V-velocity	y-center	0.02 ± 0.05	0.01 ± 0.09	m/s
V-velocity	z-top	0.02 ± 0.05	0.00 ± 0.10	m/s
W-velocity	x-center	0.02 ± 0.05	0.03 ± 0.10	m/s
W-velocity	y-center	0.02 ± 0.05	0.01 ± 0.09	m/s
W-velocity	z-top	0.02 ± 0.05	0.02 ± 0.06	m/s
Temperature	all	0.00 ± 2.2	0.8 ± 4.3	%
Velocity	all	0.02 ± 0.05	0.02 ± 0.09	m/s

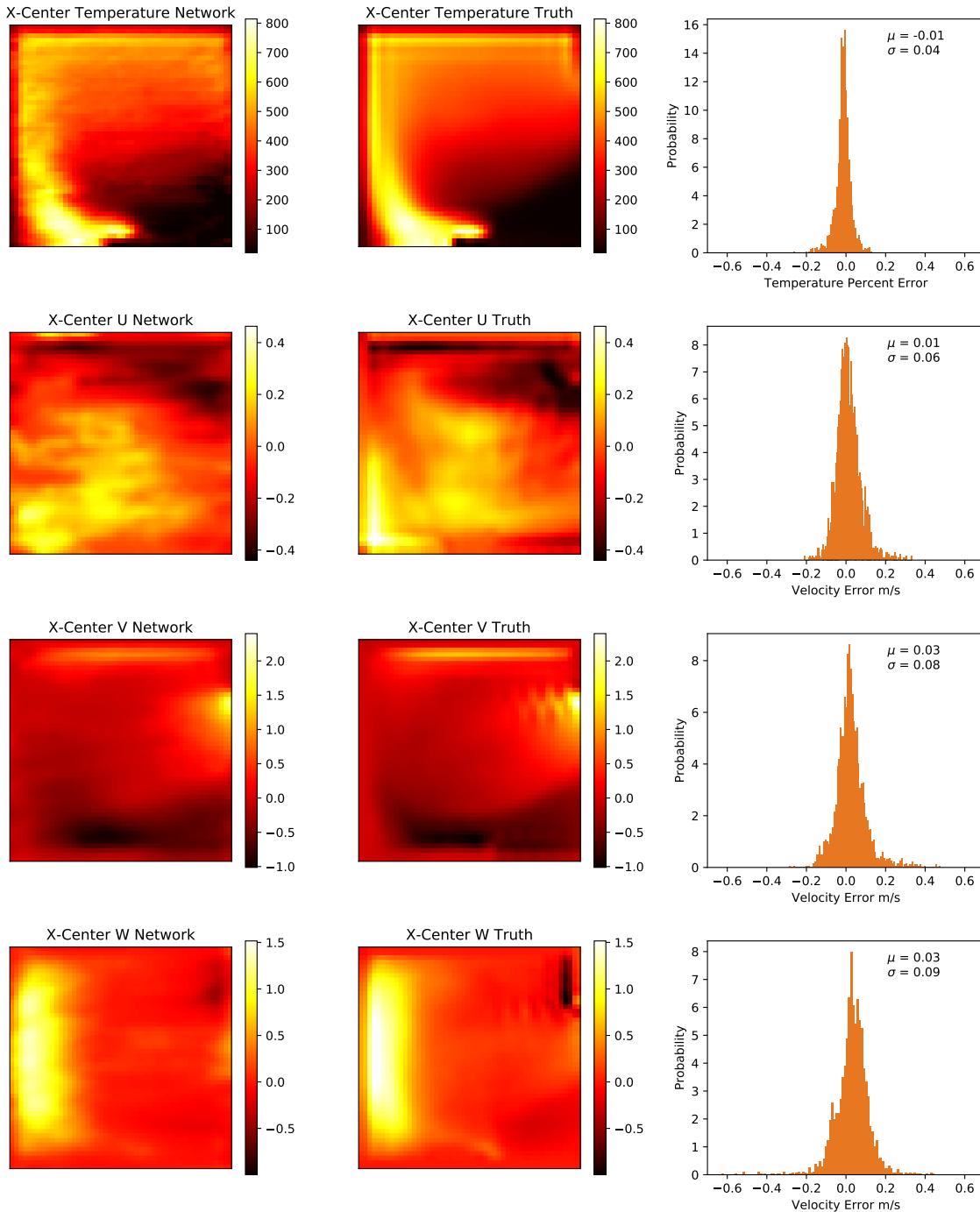


Figure 4.6: TCNN predictions of x-axis center-line temperature and velocity compared to CFD simulations for Testing Case 1.

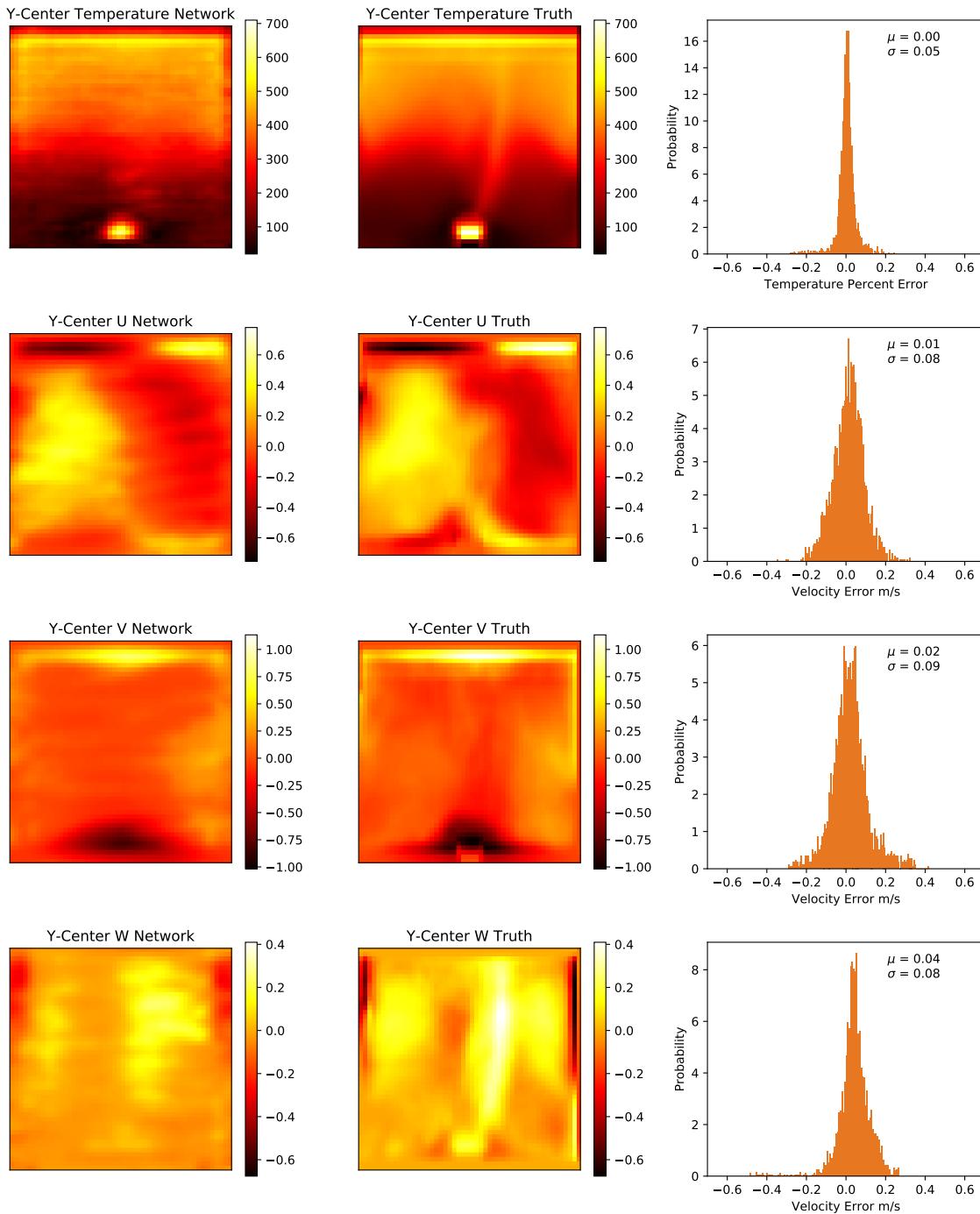


Figure 4.7: TCNN predictions of y-axis center-line temperature and velocity compared to CFD simulations for Testing Case 1.

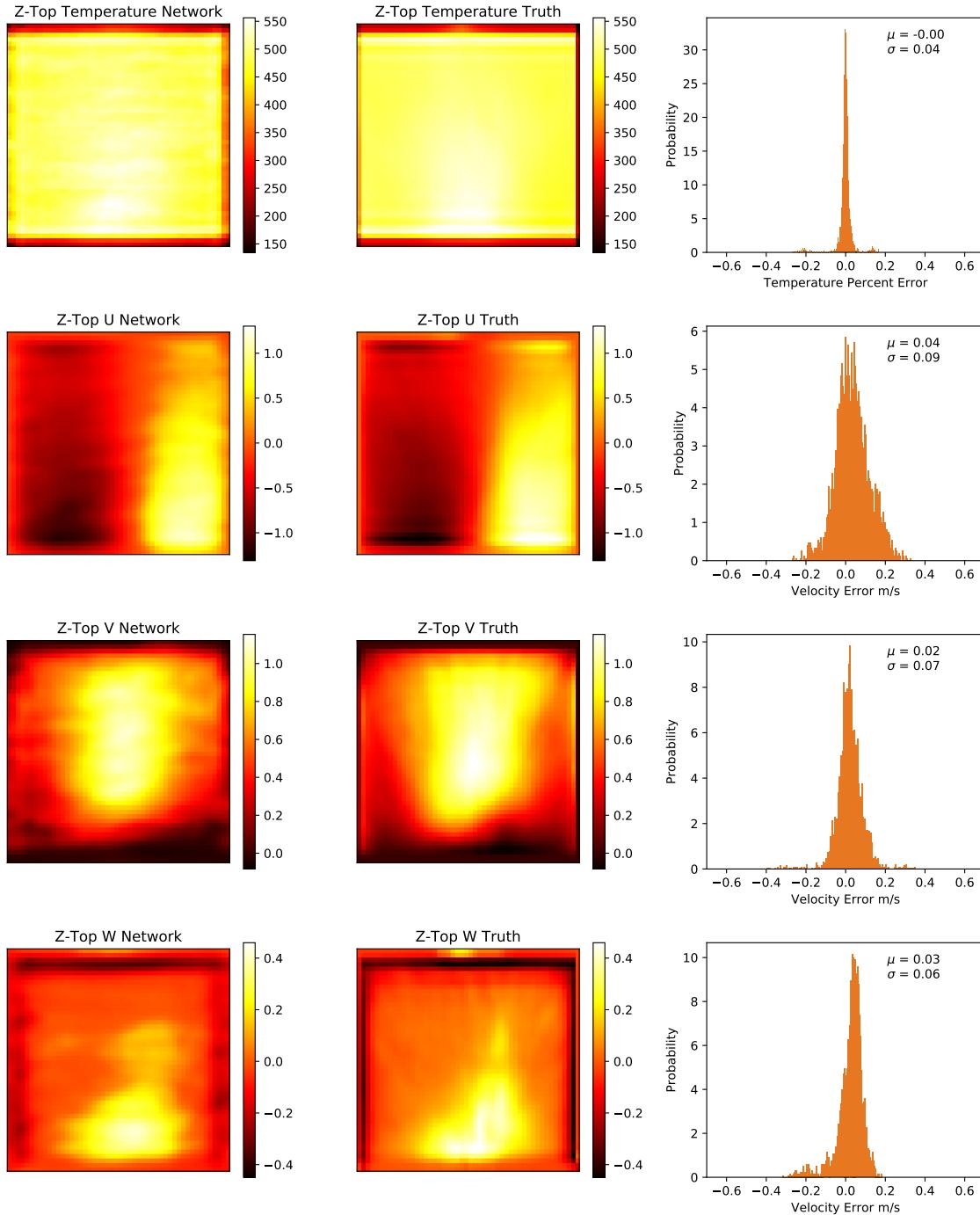


Figure 4.8: TCNN predictions of z-axis ceiling temperature and velocity 0.1m below the ceiling compared to CFD simulations for Testing Case 1.

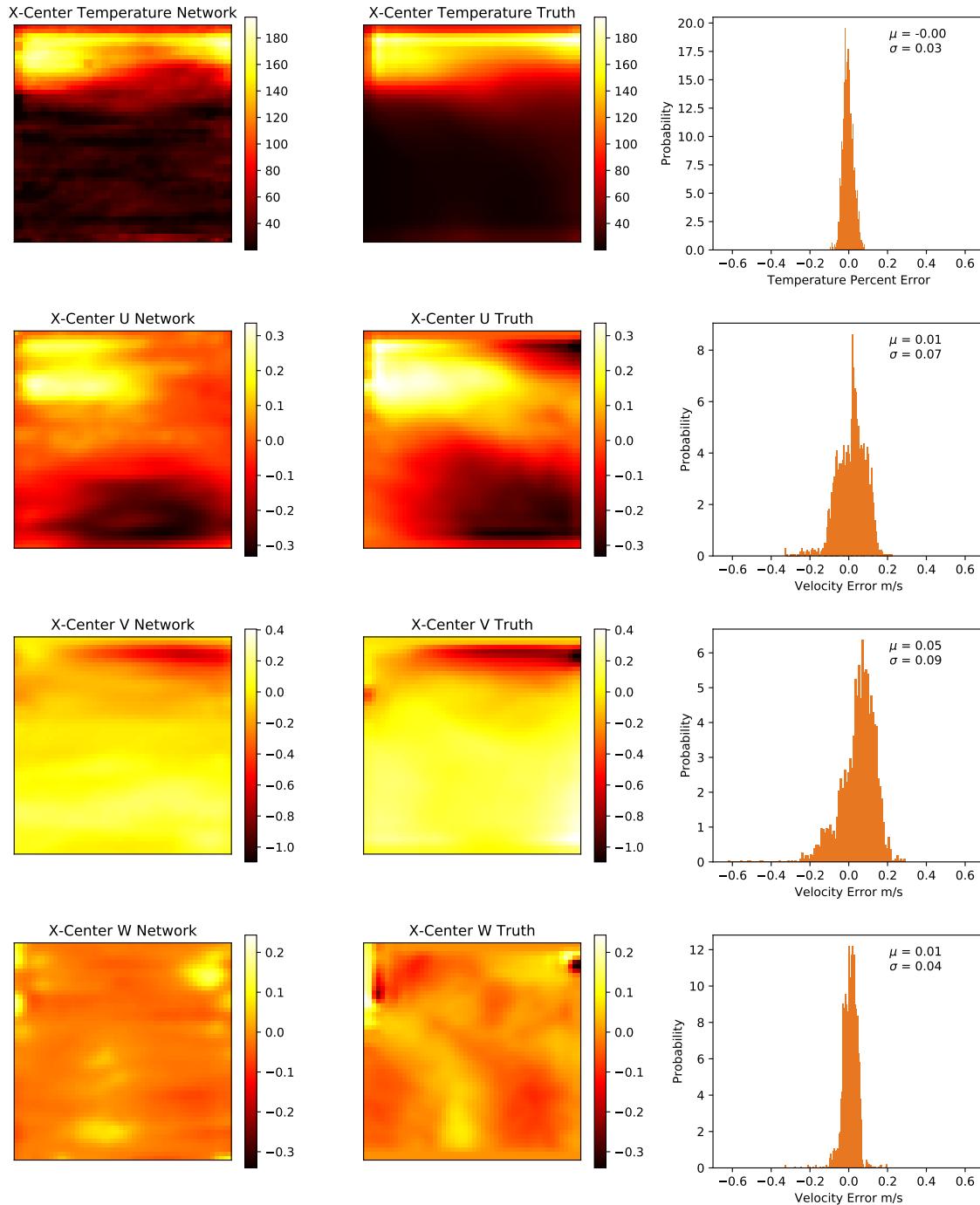


Figure 4.9: TCNN predictions of x-axis center-line temperature and velocity compared to CFD simulations for Testing Case 2.

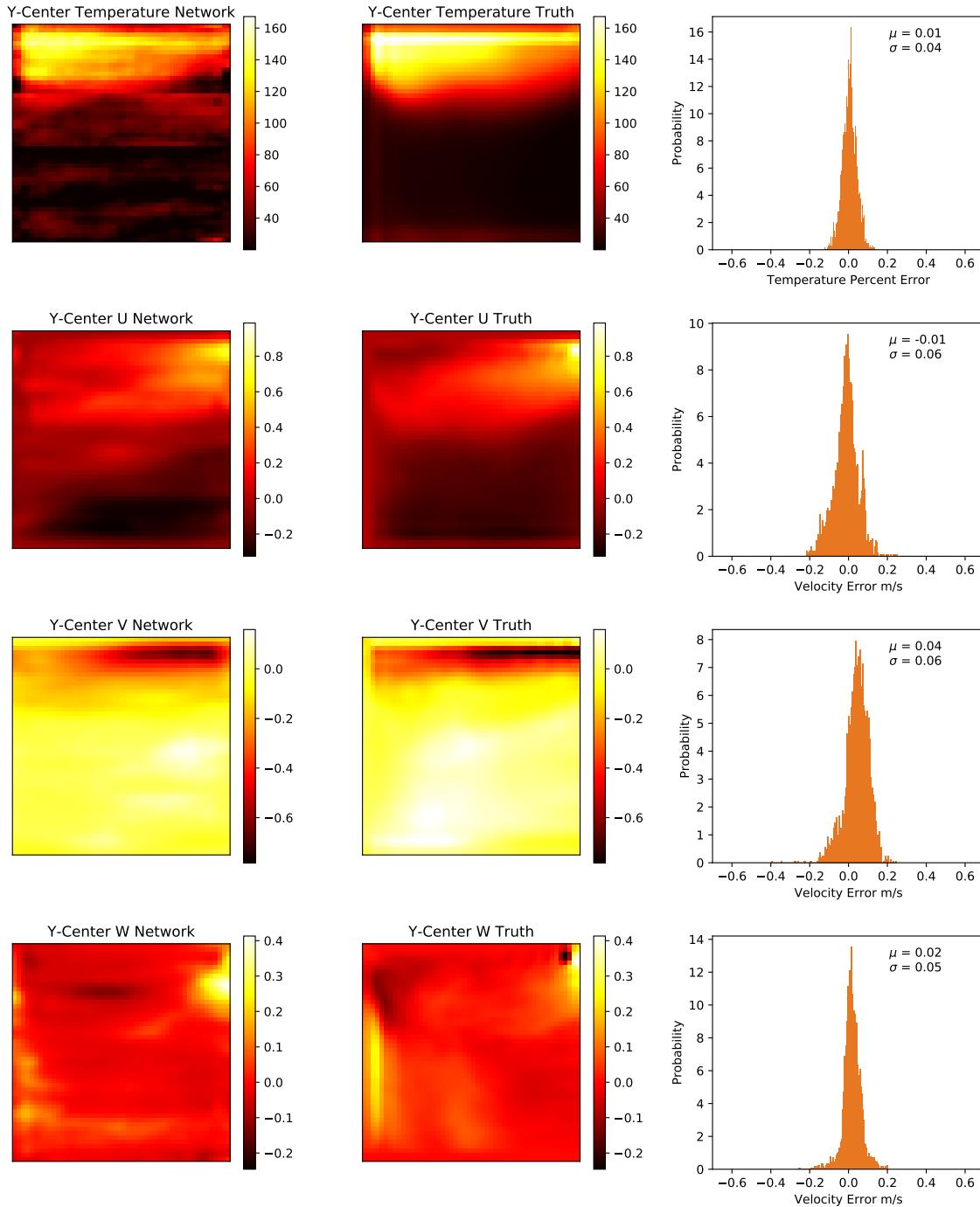


Figure 4.10: TCNN predictions of y-axis center-line temperature and velocity compared to CFD simulations for Testing Case 2.

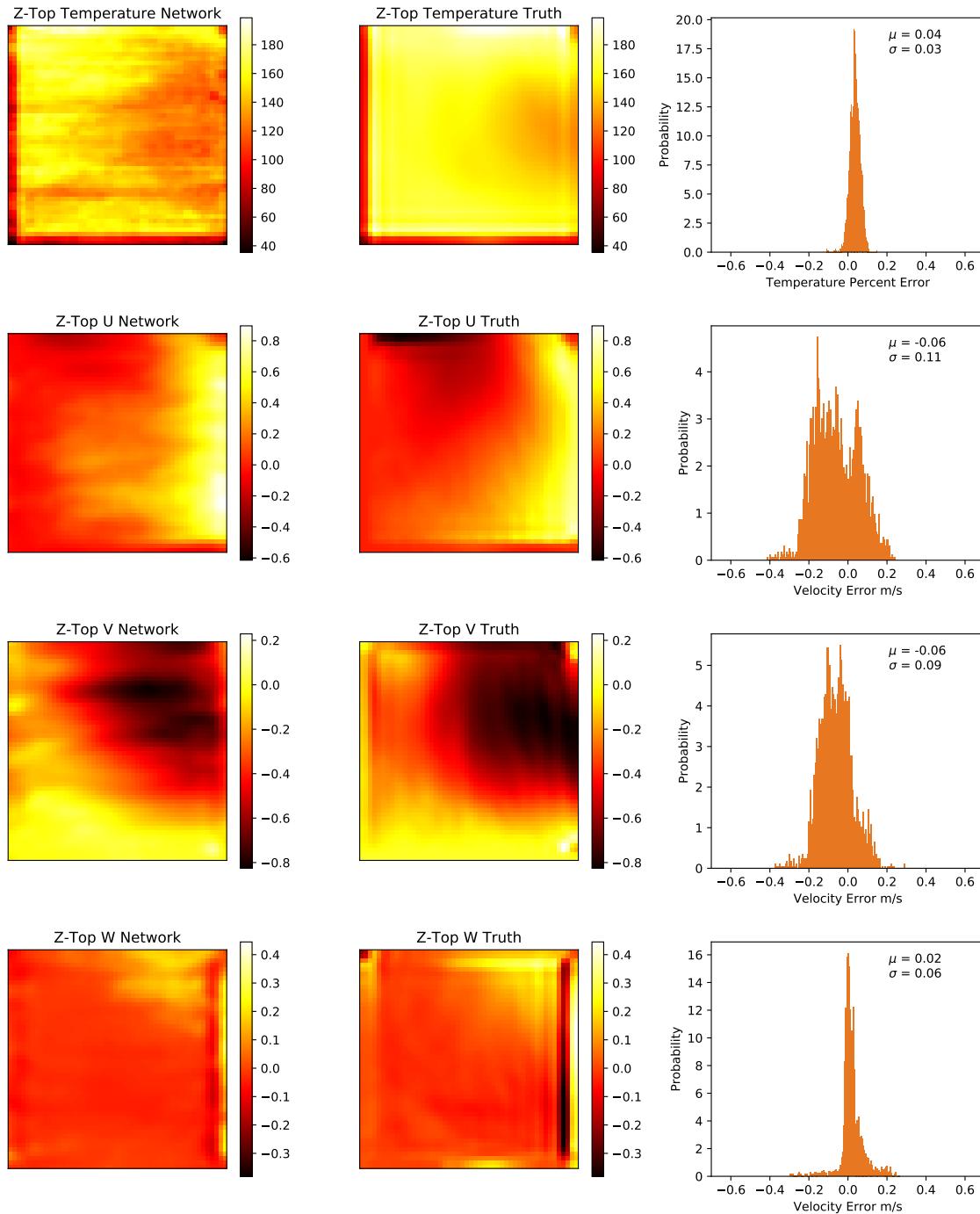


Figure 4.11: TCNN predictions of z-axis ceiling temperature and velocity 0.1m below the ceiling compared to CFD simulations for Testing Case 2.

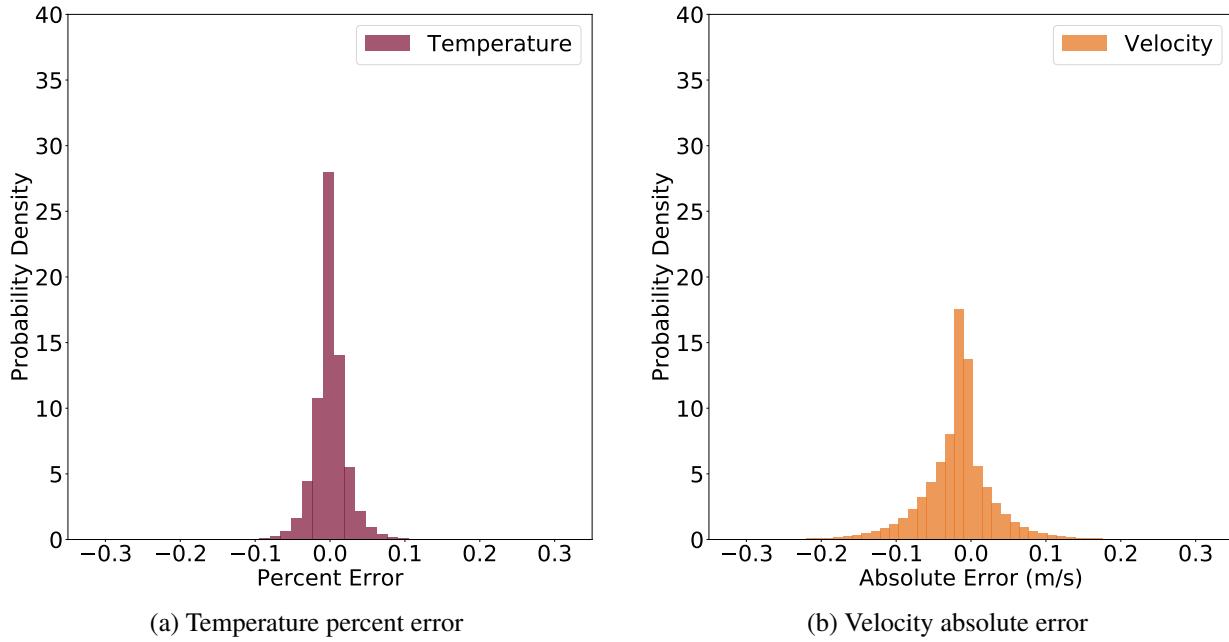


Figure 4.12: Discrete probability density functions of TCNN error from training data set

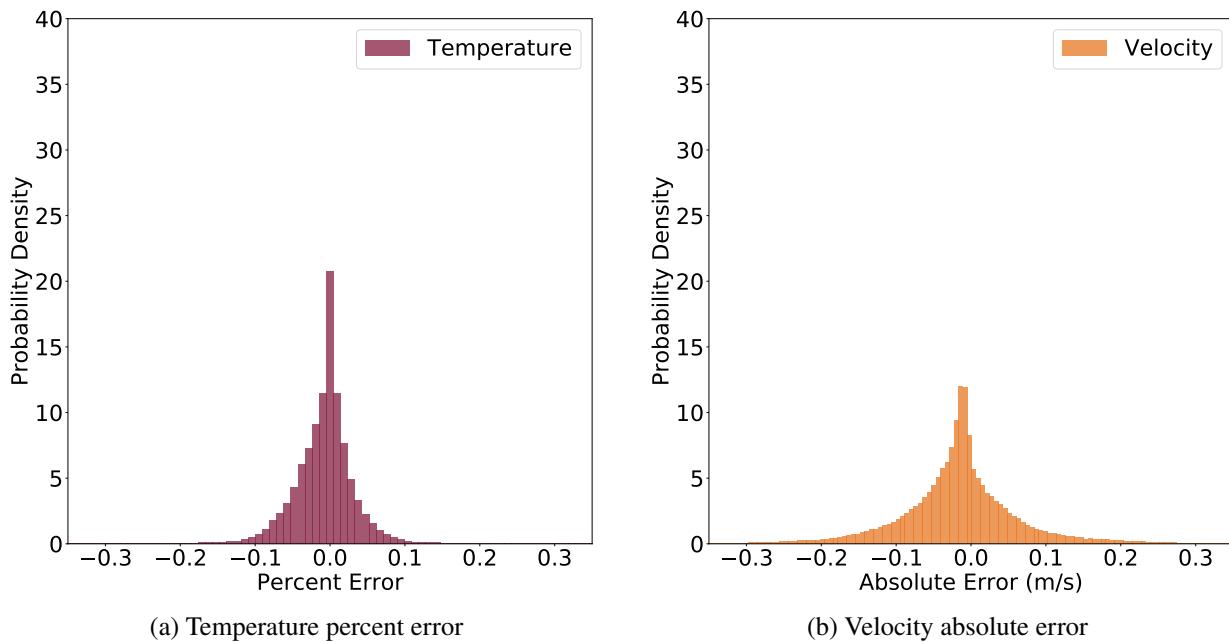


Figure 4.13: Discrete probability density functions of TCNN error from testing data set

4.5 Discussion

The example scenarios shown in Fig. 4.6- 4.11 show the overall temperature profiles predicted using the TCNN agree with the CFD predictions. The x-center temperature shown in Fig. 4.6 shows the TCNN was able to correctly identify the fire plume was in the room and that the plume was being tilted towards the left-hand side even though this information was not input to the network. The neural network was able to predict the spatially resolved change in interface height shown in the x-center and y-center temperature profiles in Fig. 4.9 and Fig. 4.10. Additionally the TCNN was able to predict the presence of a cooler region in front of the door on the right hand side of the z-top temperature in Fig. 4.11. Table 4.3 shows the mean temperature percent error is less than 1.5% for each spatial slice, and the standard deviation is 2.2% for the training set and 4.3% for the test set. Based on these results, 95% of TCNN temperature predictions should be within $\pm 8.6\%$ of the CFD predictions. Much of the temperature error was identified to come from two primary modes. It was observed that the peak temperature when the flame directly intersects the temperature field is generally under-predicted. Since the TCNN has no a priori knowledge of the presence of the flame, the peak temperature must be inferred directly from ventilation flows. When the interface height is high as in Testing Case 2 the temperature throughout the lower layer exhibits some non-physical noise, shown in Fig. 4.9 and Fig. 4.10. No significant difference in error between the x-center, y-center, and z-top temperatures was observed.

The example scenarios shown in Fig. 4.6- 4.11 highlight the capability of the method to predict complex flow fields even though the inputs to the TCNN are coarse. For example, the y-center U-velocity in Fig. 4.7 shows the TCNN was able to predict the multiple points of inflection in the velocity. Table 4.3 shows the mean velocity error is 0.02 m/s and a standard deviation of 0.05 m/s for the training set and 0.09 m/s for the test set. Based on these results, 95% of TCNN velocity predictions should be within ± 0.18 m/s of the CFD predictions. Much of the velocity error was

identified to come from two primary modes. Although most of the flow fields are predicted well, there are some regions where the network has difficulty, such as the recirculation region near the ceiling on the right hand side of the x-center U-velocity in Fig. 4.9. Additionally, the TCNN tends to under-predict the peak velocity near boundaries as can be seen in the x-center W-velocity contour in Fig. 4.6. No significant difference in error between the x-center, y-center, and z-top velocities was observed.

The computational time required for each FDS simulation ranged from 1,200-3,600 seconds dependent on the fire size and size of the computational domain. The computational time required for a similar geometry in a zone fire model such as CFAST is [148] is on the order of 1-30 seconds. The total time to predict spatially resolved temperatures and velocities with the TCNN with all 5,000 scenarios was 8 seconds, or approximately 0.0016 seconds per evaluation. For a multi-room configuration, the TCNN will evaluate each room individually which will linearly increase the computational cost. These results show the computational cost of applying the TCNN framework to zone fire model predictions is negligible compared to the base computational time.

The capability of the TCNN to predict fire scenarios beyond the two-room configuration was examined through the two multi-room validation case studies shown in Fig. 4.14. Since the TCNN presented in this work was not designed to capture transience, a longer simulation time of 120s was used to allow the smoke to fill the space for both validation cases. The temperature and velocity profiles were taken as the average from 10-120s of fire exposure.

Validation Case 1 shown in Fig. 4.14a consists of four rooms. The two-left rooms are 2.5x2.5x2.5m and the two-right rooms are 5.0x2.5x2.5m. All doors shown in Fig. 4.14a are 0.625x1.9m in size. A 0.25x0.25m burner with a heat release rate of 250 kW was located in the center of the south-west room. The temperature and velocity predictions in Validation Case 1 from the TCNN are compared with CFD predictions in Fig. 4.15. Overall temperature and flow profiles agree well with the CFD predictions. Adjacent predictions from neighboring rooms are fairly continuous, although a small

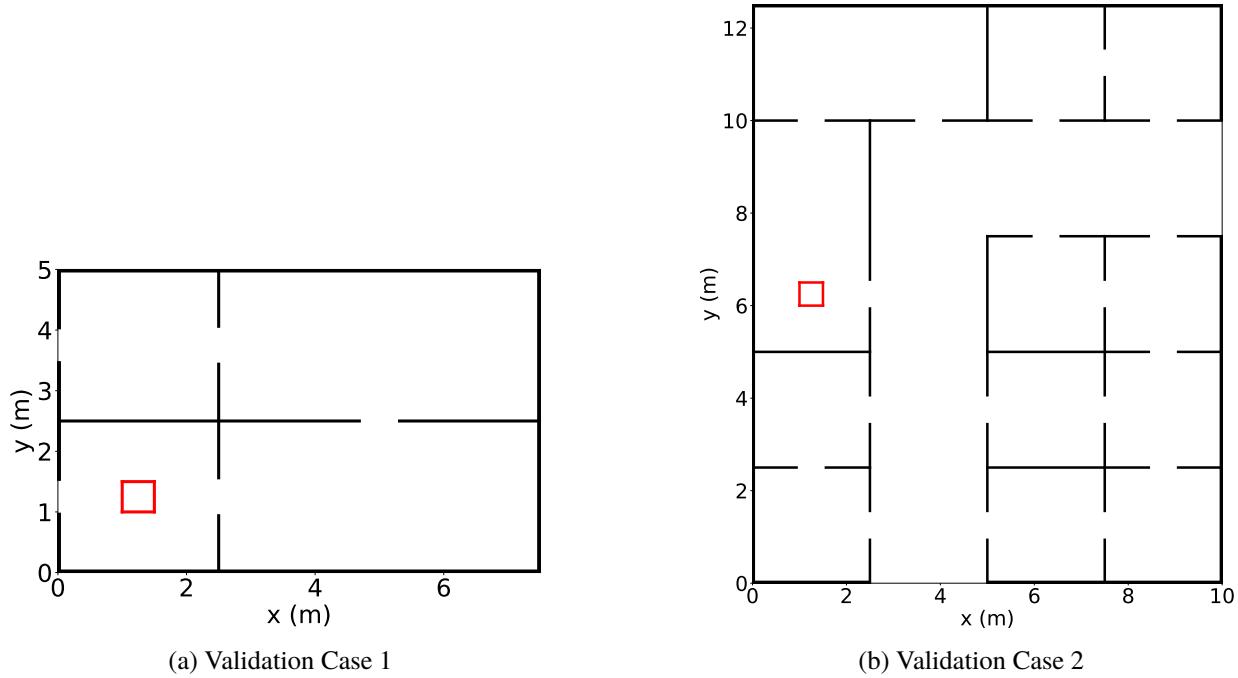


Figure 4.14: Geometry from multi-room configuration validation studies. The red square is the source fire position.

discontinuity can be seen in the region around $x = 2.5$ in Fig. 4.15a-Fig. 4.15d and $y = 2.5$ in Fig. 4.15e-Fig. 4.15f. The overall error in temperature and velocity is summarized for each room in Validation Case 1 in Fig. 4.16, where the rooms are sorted by distance to the initiating fire (0 is the south west room, 1 is the south east room, etc). The error bars in Fig. 4.16 correspond to σ . Figure 4.16 shows the error in temperature is greater in the rooms without the initiating fire, but is not increasing with distance. Additionally the error in velocity is not impacted by distance from the initiating fire.

Validation Case 2 shown in Fig. 4.14b represents a more complex geometric configuration than the others presented in this work. There are a total of ten $2.5 \times 2.5 \times 2.5$ m rooms, two $5.0 \times 2.5 \times 2.5$ m rooms, and a hallway which wraps around a corner. All doors shown in Fig. 4.14b are 0.625×1.9 m in size. A 0.25×0.25 m burner with a heat release rate of 250 kW was located in the position shown in Fig. 4.14b. Since the doors for the larger rooms and hallways are not always in the center of

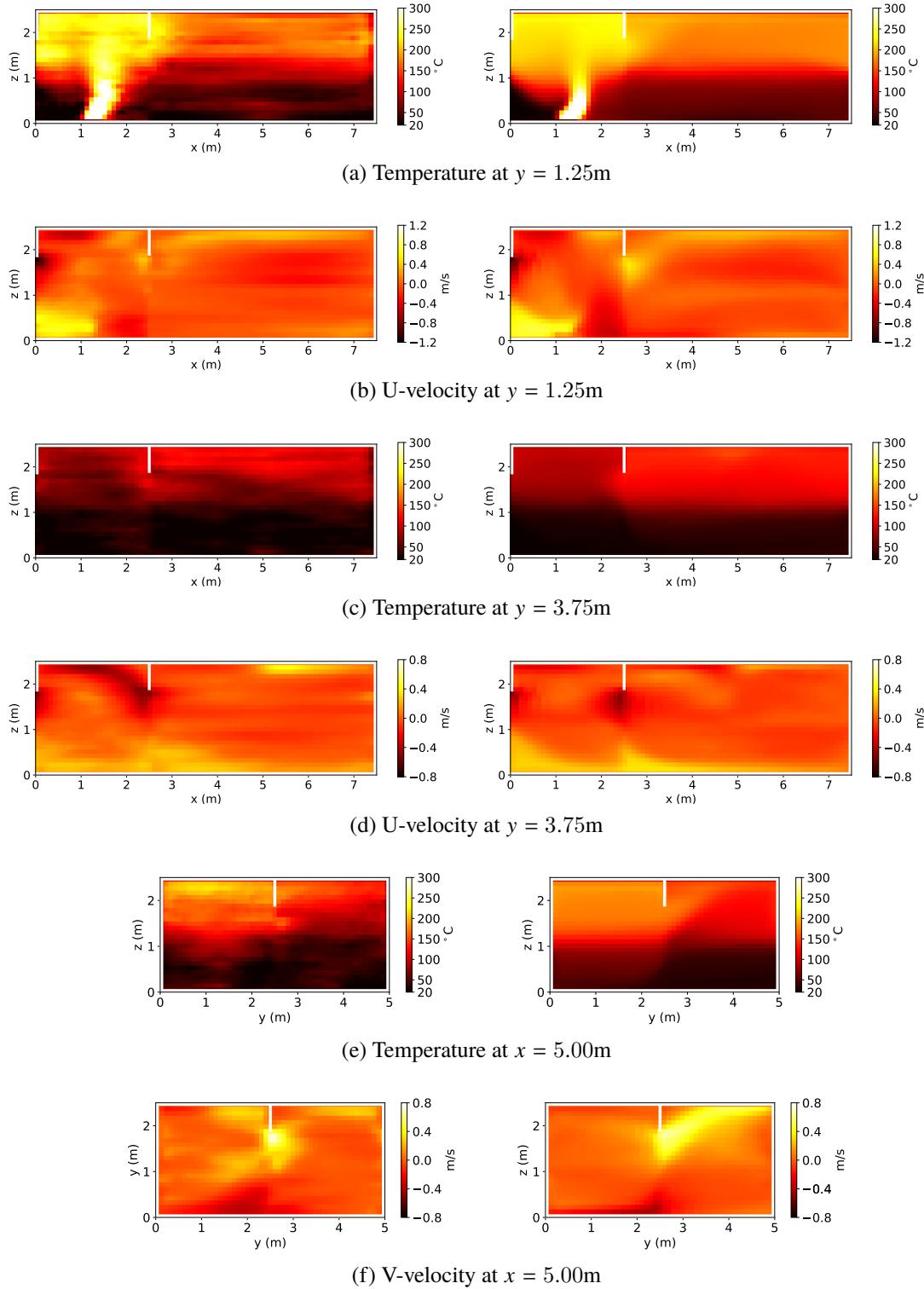


Figure 4.15: Comparison of TCNN (left) and CFD (right) predictions in Validation Case 1. The white borders correspond to obstructions to the flow.

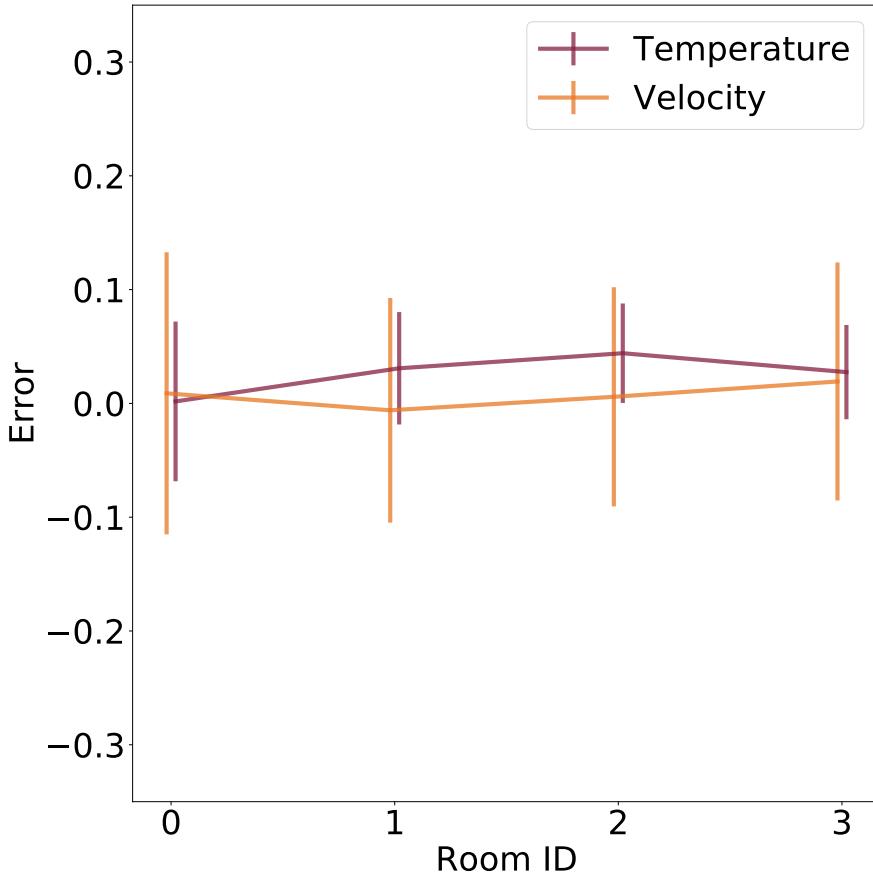


Figure 4.16: Validation Case 1 comparison of change in error with distance from the fire.

the room, these rooms were split into smaller zones before processing with the TCNN. Effectively, the two larger rooms were split into two adjacent smaller rooms with a full-wall ventilation path connecting them. Similarly the hallway was split into six segments.

The temperature and velocity predictions in Validation Case 2 from the TCNN are compared with CFD predictions in Fig. 4.17- 4.20. Overall the TCNN and CFD predictions of temperature are comparable to those in development and in Validation Case 1. Figure 4.17a- 4.17b show the TCNN predicted a hot spot near the floor in both zones which made up the source fire room. Otherwise, the temperature and velocity fields were well described. This shows the TCNN was able to make good predictions of the flow field even though the TCNN had not been trained with a fire source in the

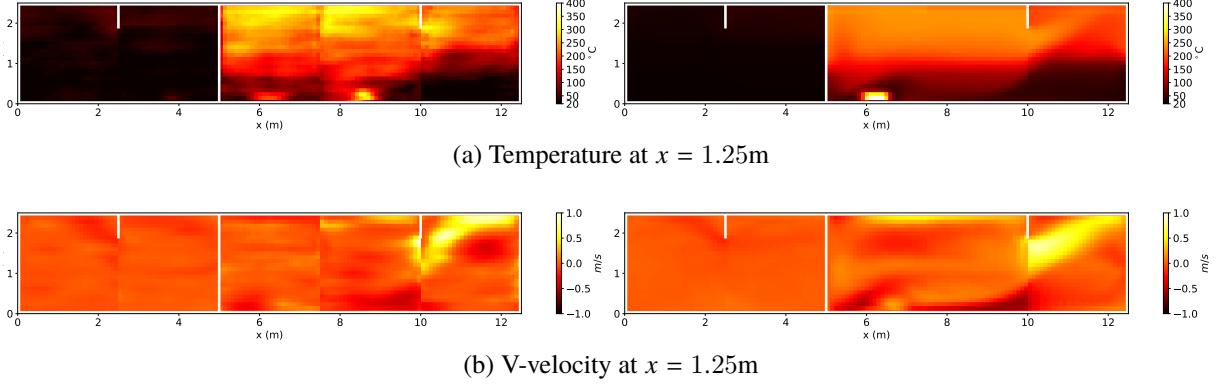


Figure 4.17: Comparison of vertical slices TCNN (left) and CFD (right) predictions in Validation Case 2. The white borders correspond to obstructions to the flow

large split room configuration. The TCNN predicted upper layer temperatures in Fig. 4.18 were able to capture the higher temperature regions where smoke was entering the hallway. Figure 4.19- 4.20 show the TCNN had difficulty predicting the velocity field below the ceiling in the source fire room and large room north of the fire source as well as in the corner zone of the hallway. These errors are likely due to significant vorticity in the flow which was not captured in the database from the two-room configuration. To minimize these effects, the data-set could be augmented with scenarios with hot gases entering the room from multiple sources such as occurs in the hallway corner zone. The discontinuities in the velocity predictions between neighboring rooms are more significant in Validation Case 2 than in Validation Case 1, as shown in Fig. 4.19- 4.20. These discontinuities are likely an artifact of the sampling used in generating the two-room configurations (there were few simulations used in training where the forced ventilation configuration was selected and both the west and east wall of Room 2 was a hallway-type connection).

The error for each temperature and velocity field for Validation Case 1 and 2 is shown in Table 4.4. The mean temperature percent error was higher in Validation Case 1 at 2.6% than in Validation Case 2 at -0.5%. The standard deviation of the temperature error was the same in both Validation Case 1 and Validation Case 2 at 5.5%. The velocity error was not significantly different between the two cases. The mean absolute velocity error was the same between the two cases at 0.01 m/s .

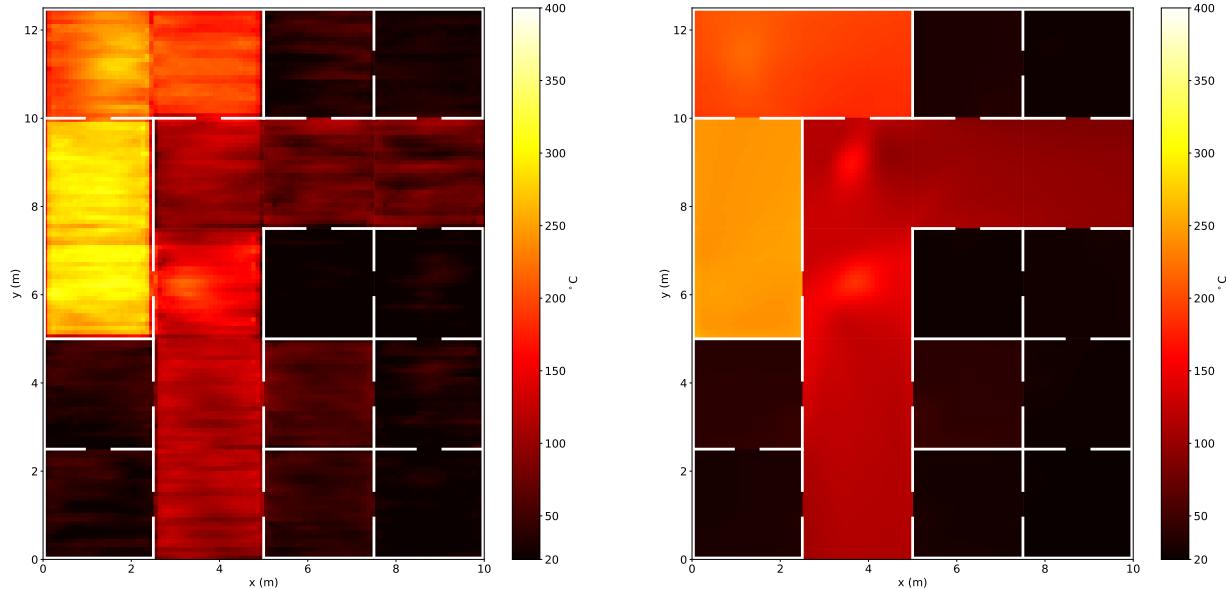


Figure 4.18: Comparison TCNN (left) and CFD (right) predictions of temperature 0.1m below the ceiling in Validation Case 2. The white borders correspond to obstructions to the flow

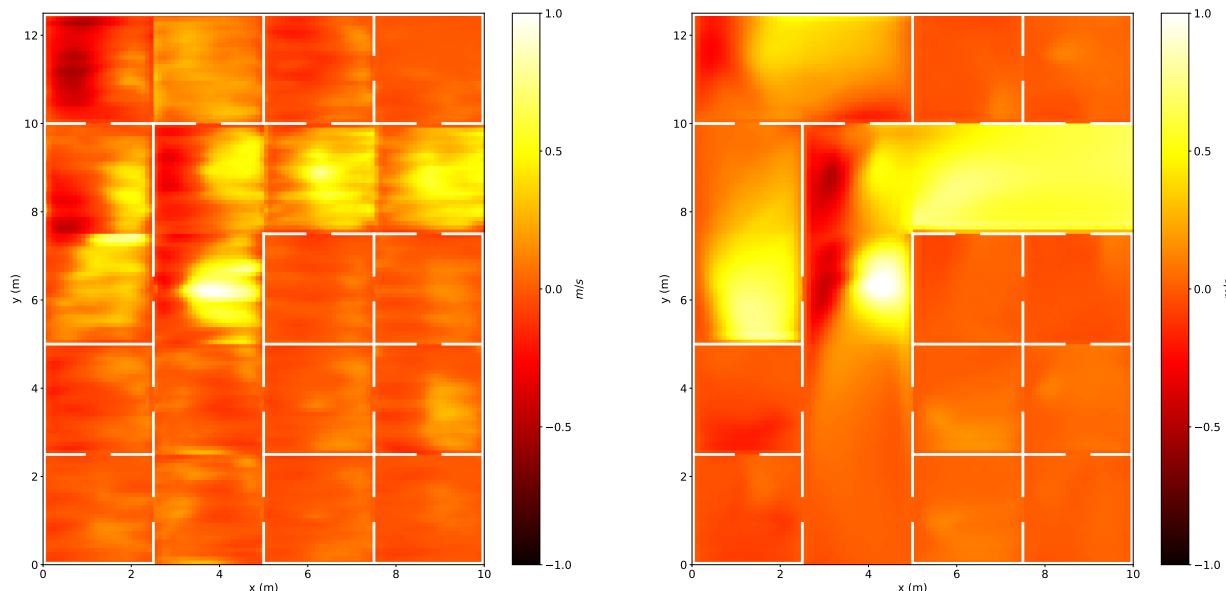


Figure 4.19: Comparison TCNN (left) and CFD (right) predictions of U-velocity 0.1m below the ceiling in Validation Case 2. The white borders correspond to obstructions to the flow

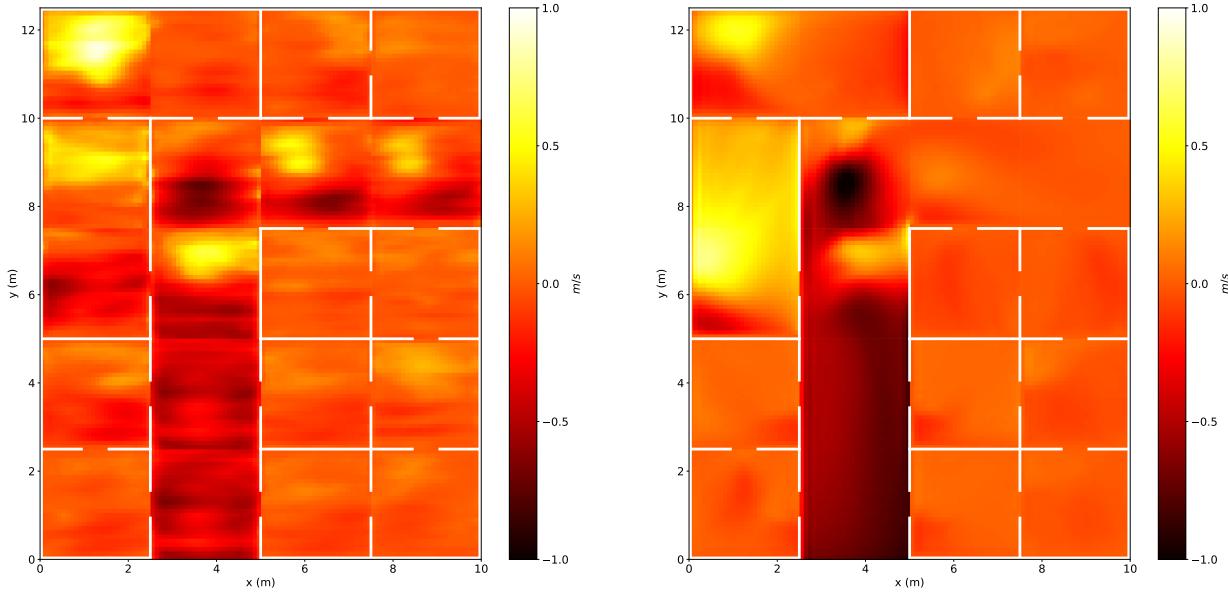


Figure 4.20: Comparison TCNN (left) and CFD (right) predictions of V-velocity 0.1m below the ceiling in Validation Case 2. The white borders correspond to obstructions to the flow

The standard deviation of the absolute velocity error was 0.11 m/s in Validation Case 1 and 0.12 m/s in Validation Case 2. The standard deviation of U-velocity and V-velocity error in the z-top slice was higher than the two vertical slices. The total computational time required to evaluate the validation cases using the CFD model and the TCNN model are shown in Table 4.5. Overall the TCNN is significantly faster, by a factor of 10^4 .

4.6 Conclusion

A novel data-driven approach was presented to predict full-field temperature and velocity within a compartment based on coarse zone fire modeling using a transpose convolutional neural network (TCNN). The network was trained using 1000 CFD simulations of a simple two-compartment configuration with different fire locations, fire sizes, ventilation configurations, and compartment geometries. The robustness of the approach was tested using ten two-compartment CFD simulations

Table 4.4: Summary of performance of TCNN predictions on validation cases, values correspond to $\mu \pm \sigma$.

Intensive Property	Location	Validation Case 1	Validation Case 2	Units
Temperature	x-center	3.0 ± 5.6	-1.2 ± 4.9	%
Temperature	y-center	2.9 ± 5.3	0.0 ± 5.4	%
Temperature	z-top	1.9 ± 5.4	-0.3 ± 6.1	%
U-velocity	x-center	0.01 ± 0.11	0.01 ± 0.10	m/s
U-velocity	y-center	0.00 ± 0.10	0.00 ± 0.05	m/s
U-velocity	z-top	0.04 ± 0.15	0.07 ± 0.17	m/s
V-velocity	x-center	0.01 ± 0.10	0.00 ± 0.11	m/s
V-velocity	y-center	0.00 ± 0.10	0.00 ± 0.10	m/s
V-velocity	z-top	0.00 ± 0.15	-0.01 ± 0.22	m/s
W-velocity	x-center	0.01 ± 0.07	0.00 ± 0.08	m/s
W-velocity	y-center	-0.02 ± 0.09	0.00 ± 0.11	m/s
W-velocity	z-top	0.02 ± 0.06	0.02 ± 0.06	m/s
Temperature	all	2.6 ± 5.5	-0.5 ± 5.5	%
Velocity	all	0.01 ± 0.11	0.01 ± 0.12	m/s

Table 4.5: Comparison of computational time for CFD and TCNN predictions of validation cases.

Case	Computational time (seconds)	
	CFD Model	TCNN Model
1	9310	0.76
2	22841	0.90

not included when training the network. In the two-compartment test cases 95% of TCNN predicted temperatures were within $\pm 8.6\%$ of CFD predictions, and 95% of TCNN predicted velocities were within ± 0.18 m/s of CFD predictions. The largest temperature error was due to under-prediction in peak temperatures and noise in the prediction of the lower layer temperatures. The largest velocity error was due to under-predicted peak velocities near boundaries and difficulty to predict some recirculation regions near ventilation. The computational time required for each CFD simulation in the database ranged from 1,200-3,600 seconds dependent on the size of the compartment. The computational time to evaluate 5,000 scenarios with the TCNN framework was 8 seconds.

Although the model was trained and tested using a simple two-compartment configuration, the TCNN approach was validated with two more complex multi-compartment CFD simulations by processing each room individually. Overall the flow fields in the multi-compartment tests agreed well with CFD predictions with 95% of TCNN predicted temperatures within $\pm 11\%$ of CFD predictions, and 95% of TCNN predicted velocities within ± 0.24 m/s of CFD predictions. The increase in error observed in the multi-compartment tests is attributed to the simple two-room configuration not capturing the impact of high vorticity due to the mixing of multiple hot gas flows, and a small number of hallway-style scenarios in the database. The computational time to evaluate the multi-compartment test cases using CFD were on the order of 10,000 seconds, while the computational time using the TCNN approach was on the order of 1 second, a factor of 10^4 faster.

This work represents a first step in creating a framework to predict spatially and temporally resolved flow fields in compartment fires at the same speed as coarse zone fire models. The next steps in this process are to expand the training data-set to remove many of the artificial limitations placed on the network (such as adiabatic walls, and limited ventilation sizes and positions), and redesign aspects of the network to handle time varying predictions and solid obstructions within a compartment.

4.7 Acknowledgements

This research was supported by Contract No. 200-2014-59669, awarded by the National Institute for Occupational Safety and Health (NIOSH). The findings and conclusions in this report are those of the authors and do not reflect the official policies of the Department of Health and Human Services; nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

Chapter 5

Conclusions and Future Work

5.1 Summary

This dissertation focused on advancing the state-of-the art in data-driven fire modeling using machine learning. A design framework was developed to create artificial neural networks (ANNs) to predict full-field conditions in new fire scenarios. This framework was applied to create a new ANN model predict the two-dimensional spatially resolved spread of a wildland fire over a large domain. The first research objective was to develop a design procedure to create artificial neural networks (ANNs) for new fire scenarios. The second research objective was to apply this framework to create a new ANN model to predict the spread of a wildland fire over a large domain. The third research objective was to apply this framework to create a new ANN model to predict the thermal flow fields in a compartment fire from coarse zone fire model predictions.

A general methodology to design an ANN to predict new fire scenarios was developed and used to develop an ANN to predict the peak standard heat flux on a wall from a fire. The fundamentals of the data model and predictive model were discussed in detail and applied to develop an ANN to make regression predictions in a supervised learning context. The type of data, format, and encoding of data were addressed and the implications on the required network capacity discussed. Recommendations of baseline network architectures to use for different input and output types were presented. A recommended process for manual tuning of the network hyper-parameters was presented in the context of overall network performance and generalization error. The predictions

of standard heat flux from the ANN were within 10% error of the simulated values for 95% of scenarios.

The developed framework was used to create a novel data-driven approach to predict the spread of a wildland fire. Since the inputs and outputs to the neural network consisted of spatially resolved data with a grid like topology, a deep convolutional inverse graphics network (DCIGN) architecture was used. Data for use in training, and testing the DCIGN approach was generated using 10,000 wildland fire spread simulations using the phenomenological model of Rothermel for homogeneous spatial conditions. Noise in the input parameters was found to have a minimal affect on the DCIGN predictions. The computational cost of the method was found to be comparable to simulating a phenomenological model for homogeneous spatial conditions, and a factor of $10^2 - 10^5$ faster for heterogeneous spatial conditions. Although trained on predictions six hours apart, the DCIGN-based approach is capable of predicting burn maps further in the future by recursively using previous predictions as inputs to the model. The cases where F-measure was observed to be less than 0.80 were found to have input burn maps of less than nine pixels. This indicates as the fire continues to grow, the predictions will continue to improve. A preliminary study predicting spread over heterogeneous landscapes showed the method has the potential to provide rapid estimates of fire spread; however, fire spread over heterogeneous landscapes should be made at a higher resolution to better capture the spatial variation. Although the data used to train the CNN in this work was generated using a phenomenological model, the model does not have any information about whether the data is from a computational fluid dynamics model, phenomenological model, or even experimental measurements of burn maps.

The developed framework was used to create a novel data-driven approach to predict spatially resolved thermal flow fields within a compartment fire based on coarse zone fire modeling. Since the inputs to the network had no topological structure and the outputs to the network did have a grid like topology, a transpose convolutional neural network architecture (TCNN) was used. The

network was trained using 1,000 CFD simulations of a simple two-compartment configuration with different fire locations, fire sizes, ventilation configurations, and compartment geometries. The robustness of the approach was tested using ten two-compartment CFD simulations not included when training the network. Overall temperature errors were within $\pm 8.6\%$ of CFD predictions velocities were within ± 0.18 m/s of CFD predictions. The largest temperature error was due to under-prediction in peak temperatures and noise in the prediction of the lower layer temperatures. The largest velocity error was due to under-predicted peak velocities near boundaries and difficulty to predict some recirculation regions near ventilation. The computational time required for each CFD simulation in the database ranged from 1,200-3,600 seconds dependent on the size of the compartment. The computational time to evaluate 5,000 scenarios with the TCNN framework was 8 seconds. Although the model was trained and tested using a simple two-compartment configuration, the TCNN approach was validated with two more complex multi-compartment CFD simulations by processing each room individually. Overall the flow fields in the multi-compartment tests agreed well with CFD predictions, although slightly more error was observed. The increase in error observed in the multi-compartment tests is attributed to the simple two-room configuration not capturing the impact of high vorticity due to the mixing of multiple hot gas flows, and a small number of hallway-style scenarios in the database. The computational time to evaluate the multi-compartment test cases using CFD were on the order of 10,000 seconds, while the computational time using the TCNN approach was on the order of 1 second, a factor of 10^4 faster.

This work represents a significant advancement in the field of data-driven fire modeling. Prior to the work presented herein data-driven approaches in fire modeling were limited to predictions of individual points of data. Using the techniques presented in this work spatially resolved neural network predictions of important fire parameters is possible. Together this represents a first step in creating a framework in developing super real-time fire predictions across multiple applications.

5.2 Conclusions

This dissertation focused on advancing the state-of-the art in data-driven fire modeling using machine learning. During the development, training, and testing process of the ANNs to predict fire behavior in wildland fires and structure fires with multiple compartments, the following conclusions were identified:

- Spatially resolved feature based neural networks (such as networks containing convolutional and transpose convolutional layers) are well posed to model multi-physics problems including fire dynamics. This was evident in the overall low error in the validation cases in wildland and structure fire modeling work.
- Simplified computational models can be used to train and test ANNs for more complex scenarios. This was the fundamental hypothesis of the two-room simulation framework used in the structure fire modeling work. The overall low error in the multi-room structure fire validation cases show the potential for this approach.
- Simplified computational models which are used to train and test ANNs must ensure the necessary physics are well represented. The two-room simulations used in the structure fire modeling were able to represent a wide range of scenarios observed in the validation simulations including complex ventilation configurations and split-room configurations. However, the network had difficulty predicting the flow field when there was significant vorticity at the ventilation due to not including cases with these physics in the training data.
- Sufficient spatial resolution is needed in the input data to capture the important physics of the system. This was seen in the wildland fire modeling work where down-sampling the input data sometimes led to non-burnable surfaces not being represented in the input data.
- The reduction in computational time for an ANN based approach to model fire behavior is

significant, on the order of $10^2 - 10^5$ for the applications investigated in this work.

Overall, the ANN methodology was determined to be successful at predicting the multi-physics problems. With additional training data and generalization of the geometry, ANNs are a viable method to provide high speed predictions.

5.3 Future Work

This work represents a first step in creating a framework in developing high fidelity super real-time fire predictions. Continual work in this area has the potential to revolutionize fire protection engineering and emergency response to fires. There are many areas this work can be extended in the near future.

The wildland fire spread work showed the feature learning aspect of the network based method is well posed to learn how to predict fire spread over heterogeneous spatial conditions. Improving the spatial and temporal resolution of the inputs and outputs to the network will improve the predictions of the network. Incorporating experimental measurements (such as satellite imagery, forestry personnel walking the fire perimeter, or drone imagery) into the training process could allow the network based approach to generate better predictions than the phenomenological models. In addition, incorporating a recurrent layer to the network architecture similar to the deep recurrent attentive writer (DRAW) framework could lead to overall better predictions.

The compartment fire thermal flow field work showed the neural network based method is capable of learning to predict the spatially resolved thermal flow field from coarse zone fire modeling predictions. One aspect of the network approach which should be explored in the near future is incorporating time varying predictions to the model. This could be accomplished through a recursive method similar to the wildland fire model or through the development of a recurrent

neural network architecture. Once the temporal aspect is added it will be important to add material properties to the network as well as the time evolution of the flow field will depend on the energy absorbed by the boundaries. In addition, augmenting the database with additional CFD simulations would lead to overall better predictions.

Another aspect which has not been explored in the thermal flow field predictions or the standard heat flux predictions is incorporating experimental measurements into the training and testing of the network. Using experimental measurements in the training and testing of the network could potentially improve the data-driven predictions beyond the capability of current modeling approaches as the experimental measurements contain all the physics of the problem. Spatially resolved velocity fields can be measured through particle imaging velocimetry. Spatially resolved gas temperatures can be measured through an array of thermocouples. Spatially resolved wall temperatures and standard heat flux can be measured using IR thermography and inverse heat transfer, using the method presented in Appendix D. Although this process is not trivial due to the number of experiments needed, it could add value to the data-driven approach in the future.

Often in structural fire research the coupled thermo-mechanical response of a sample is useful to determine when failure has occurred. Although the approaches presented in this dissertation focused on the thermal flow field and did not investigate displacement, the general approach could be used to predict full-field displacement as well. Researchers and fire safety engineers have developed the capability to couple a finite element structural solver with the computational fluid dynamics predictions of fire exposure to predict the coupled thermo-mechanical response of a structure to a fire [149, 150, 151, 152, 153]. In addition, recent advancements in experimental fire testing have developed techniques to measure the coupled displacement and temperature fields using Thermography Digital Image Correlation (TDIC) [153, 154, 155, 156]. A series of simulations or experiments similar to those shown in Appendix B and Appendix C could be used to develop a neural network approach to predict structural behavior based on sample geometry, fire size, and

distance to fire.

Bibliography

- [1] Jun Zhuang, Vineet M Payyappalli, Adam Behrendt, and Kathryn Lukasiewicz. Total cost of fire in the united states. *Fire Protection Research Foundation*, 2017.
- [2] Ben Evarts. Fire loss in the united states during 2017. *National Fire Protection Association Quincy, MA*, 2018.
- [3] Roger B Hammer, Volker C Radeloff, Jeremy S Fried, and Susan I Stewart. Wildland–urban interface housing growth during the 1990s in California, Oregon, and Washington. *International Journal of Wildland Fire*, 16(3):255–265, 2007.
- [4] Philip E Dennison, Simon C Brewer, James D Arnold, and Max A Moritz. Large wildfire trends in the western United States, 1984–2011. *Geophysical Research Letters*, 41(8):2928–2933, 2014.
- [5] William Matt Jolly, Mark A Cochrane, Patrick H Freeborn, Zachary A Holden, Timothy J Brown, Grant J Williamson, and David MJS Bowman. Climate-induced variations in global wildfire danger from 1979 to 2013. *Nature Communications*, 6:7537, 2015.
- [6] Albert Simeoni. Wildland fires. *SFPE Handbook of Fire Protection Engineering*, pages 3283–3302, 2015.
- [7] Youssef Safi and Abdelaziz Bouroumi. Prediction of forest fires using artificial neural networks. *Applied Mathematical Sciences*, 7(6):271–286, 2013.
- [8] Ales Popovic, Mauro Castelli, and Leonardo Vanneschi. Predicting burned areas of forest fires: an artificial intelligence approach. *Fire Ecology*, 11(1):106–118, 2015.

- [9] Jeremy Storer and Robert Green. PSO trained neural networks for predicting forest fire size: A comparison of implementation and performance. *International Joint Conference on Neural Networks (IJCNN)*, pages 676–683, 2016.
- [10] Hariharan Naganathan, Sudarshan P Seshasayee, Jonghoon Kim, Wai K Chong, and Ju-Sheng Chou. Wildfire predictions: Determining reliable models using fused dataset. *Global Journal of Computer Science and Technology*, 2016.
- [11] Yinxue Cao, Ming Wang, and Kai Liu. Wildfire susceptibility assessment in southern China: A comparison of multiple methods. *International Journal of Disaster Risk Science*, 8(2):164–181, 2017.
- [12] Ronald J McCormick, Thomas A Brandner, and Timothy FH Allen. *Toward a Theory of Meso-scale Wildfire Modeling: A Complex Systems Approach Using Artificial Neural Networks*. PhD thesis, University of Wisconsin–Madison, 2001.
- [13] Ronald J McCormick. On developing a meso-theoretical viewpoint of complex systems by exploring the use of artificial neural networks in modeling wildfires. *ForestSAT Symposium, Edinburgh*, 2002.
- [14] Gregory P Johnson and Frank C Lin. Hurricane tracking via backpropagation neural network. In *IEEE International Conference on Neural Networks*, volume 2, pages 1103–1106. IEEE, 1995.
- [15] T-L Lee. Neural network prediction of a storm surge. *Ocean Engineering*, 33(3-4):483–494, 2006.
- [16] Chih-Ming Tseng, Chyan-Deng Jan, Ji-Shang Wang, and CM Wang. Application of artificial neural networks in typhoon surge forecasting. *Ocean Engineering*, 34(11-12):1757–1768, 2007.

- [17] MC Deo. Artificial neural networks in coastal and ocean engineering. *Indian Journal of Geo-Marine Science*, pages 589–596, 2010.
- [18] Scott C James, Yushan Zhang, and Fearghal O’Donncha. A machine learning framework to forecast wave conditions. *Coastal Engineering*, 137:1–10, 2018.
- [19] Marina Campolo, Paolo Andreussi, and Alfredo Soldati. River flood forecasting with a neural network model. *Water Resources Research*, 35(4):1191–1197, 1999.
- [20] P Coulibaly, F Anctil, and B Bobee. Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. *Journal of Hydrology*, 230(3-4):244–257, 2000.
- [21] CW Dawson and RL Wilby. Hydrological modelling using artificial neural networks. *Progress in Physical Geography*, 25(1):80–108, 2001.
- [22] Puma C Nayak, KP Sudheer, DM Rangan, and KS Ramasastri. A neuro-fuzzy computing technique for modeling hydrological time series. *Journal of Hydrology*, 291(1-2):52–66, 2004.
- [23] Holger R Maier, Ashu Jain, Graeme C Dandy, and K P Sudheer. Methods used for the development of neural networks for the prediction of water resource variables in river systems: current status and future directions. *Environmental Modelling & Software*, 25(8):891–909, 2010.
- [24] Masoud Bakhtyari Kia, Saied Pirasteh, Biswajeet Pradhan, Ahmad Rodzi Mahmud, Wan Nor Azmin Sulaiman, and Abbas Moradi. An artificial neural network model for flood simulation using GIS: Johor River Basin, Malaysia. *Environmental Earth Sciences*, 67(1):251–264, 2012.
- [25] Shahab Araghinejad, Nima Fayaz, and Seyed-Mohammad Hosseini-Moghari. Development

- of a hybrid data driven model for hydrological estimation. *Water Resources Management*, pages 1–14, 2018.
- [26] I Seginer, T Boulard, and B J Bailey. Neural network models of the greenhouse climate. *Journal of Agricultural Engineering Research*, 59(3):203–216, 1994.
- [27] Sovan Lek and Jean-François Guégan. Artificial neural networks as a tool in ecological modelling, an introduction. *Ecological Modelling*, 120(2-3):65–73, 1999.
- [28] Pedro M Ferreira, EA Faria, and AE Ruano. Neural network models in greenhouse air temperature prediction. *Neurocomputing*, 43(1-4):51–75, 2002.
- [29] Hamed Ashouri, Kuo-Lin Hsu, Soroosh Sorooshian, Dan K Braithwaite, Kenneth R Knapp, L Dewayne Cecil, Brian R Nelson, and Olivier P Prat. PERSIANN-CDR: Daily precipitation climate data record from multisatellite observations for hydrological and climate studies. *Bulletin of the American Meteorological Society*, 96(1):69–83, 2015.
- [30] Jean F Mas and Juan J Flores. The application of artificial neural networks to the analysis of remotely sensed data. *International Journal of Remote Sensing*, 29(3):617–663, 2008.
- [31] Mariana Belgiu and Lucian Drăguț. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:24–31, 2016.
- [32] Zachary L Langford, Jitendra Kumar, and Forrest M Hoffman. Convolutional neural network approach for mapping arctic vegetation using multi-sensor remote sensing fusion. *IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 322–331, 2017.
- [33] Shuhui Li, Donald C Wunsch, Edgar A O’Hair, and Michael G Giesselmann. Using neural networks to estimate wind turbine power generation. *IEEE Transactions on Energy Conversion*, 16(3):276–282, 2001.

- [34] Shu Fan, James R Liao, Ryuichi Yokoyama, Luonan Chen, and Wei-Jen Lee. Forecasting the wind generation using a two-stage network based on meteorological information. *IEEE Transactions on Energy Conversion*, 24(2):474–482, 2009.
- [35] Ercan Izgi, Ahmet Öztopal, Bihter Yerli, Mustafa Kemal Kaymak, and Ahmet Duran Şahin. Short–mid-term solar power prediction by using artificial neural networks. *Solar Energy*, 86(2):725–733, 2012.
- [36] Huan Long, Zijun Zhang, and Yan Su. Analysis of daily solar power prediction with data-driven approaches. *Applied Energy*, 126:29–37, 2014.
- [37] Chenhua Ni and Xiandong Ma. Prediction of wave power generation using a convolutional neural network with multiple inputs. *Energies*, 11(8):2097, 2018.
- [38] Raúl Rojas. *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.
- [39] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
<http://www.deeplearningbook.org>.
- [40] Kevin McGrattan, Simo Hostikka, Randall McDermott, Jason Floyd, and Marcos Vanella. Fire dynamics simulator (version 6), technical reference guide. *NIST Special Publication*, 1018(6), 2018.
- [41] Tharindu P Miyanawala and Rajeev K Jaiman. An efficient deep learning technique for the Navier-Stokes equations: Application to unsteady wake flow dynamics. *arXiv preprint arXiv:1710.09099*, 2017.
- [42] Sangseung Lee and Donghyun You. Prediction of laminar vortex shedding over a cylinder using deep learning. *arXiv preprint arXiv:1712.07854*, 2017.

- [43] Sangseung Lee and Donghyun You. Data-driven prediction of unsteady flow fields over a circular cylinder using deep learning. *arXiv preprint arXiv:1804.06076*, 2018.
- [44] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: A Navier-Stokes informed deep learning framework for assimilating flow visualization data. *arXiv preprint arXiv:1808.04327*, 2018.
- [45] Romit Maulik and Omer San. A neural network approach for the blind deconvolution of turbulent flows. *Journal of Fluid Mechanics*, 831:151–181, 2017.
- [46] Jinu Lee. Deep learning approach in multi-scale prediction of turbulent mixing-layer. *arXiv preprint arXiv:1809.07021*, 2018.
- [47] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.
- [48] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part ii): data-driven discovery of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10566*, 2017.
- [49] Siddhartha Mishra. A machine learning framework for data driven acceleration of computations of differential equations. *arXiv preprint arXiv:1807.09519*, 2018.
- [50] Joe H Scott and Robert E Burgan. Standard fire behavior fuel models: a comprehensive set for use with Rothermel’s surface fire spread model. Technical report, USDA Forest Service, 2005.
- [51] Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination Press USA, 2015.

- [52] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. *Proceedings of the 30th International Conference on Machine Learning*, 30:3, 2013.
- [53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.
- [54] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.
- [55] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.
- [56] Yi-Tong Zhou and Rama Chellappa. Computation of optical flow using a neural network. *IEEE International Conference on Neural Networks*, 1998:71–78, 1988.
- [57] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 111–118, 2010.
- [58] Y-Lan Boureau, Nicolas Le Roux, Francis Bach, Jean Ponce, and Yann LeCun. Ask the locals: multi-way local pooling for image recognition. *IEEE International Conference on Computer Vision (ICCV)*, pages 2651–2658, 2011.
- [59] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. *New York University*, 2010.

- [60] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.
- [61] Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, and Thomas Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):692–705, 2017.
- [62] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1538–1546, 2015.
- [63] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.
- [64] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.
- [65] Andrew R Barron. Complexity regularization with application to artificial neural networks. *Nonparametric Functional Estimation and Related Topics*, pages 561–576, 1991.
- [66] Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis. *Neural Computation*, 21(3):793–830, 2009.
- [67] Si Si, Dacheng Tao, and Bo Geng. Bregman divergence-based regularization for transfer subspace learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(7):929, 2010.
- [68] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide. KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. *IEEE*

- International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7893–7897, 2013.
- [69] Robert A Jacobs. Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1(4):295–307, 1988.
- [70] Matthew D Zeiler. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.
- [71] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [72] Léon Bottou. Large-scale machine learning with stochastic gradient descent. *Proceedings of COMPSTAT 2010*, pages 177–186, 2010.
- [73] Léon Bottou. Stochastic gradient descent tricks. *Neural Networks: Tricks of the Trade*, pages 421–436, 2012.
- [74] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.
- [75] Rudy Setiono. A penalty-function approach for pruning feedforward neural networks. *Neural Computation*, 9(1):185–204, 1997.
- [76] Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the 21st International Conference on Machine Learning*, page 78. ACM, 2004.
- [77] David I Warton. Penalized normal likelihood and ridge regularization of correlation and covariance matrices. *Journal of the American Statistical Association*, 103(481):340–349, 2008.

- [78] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [79] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using DropConnect. *International Conference on Machine Learning*, pages 1058–1066, 2013.
- [80] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.
- [81] Xavier Gastaldi. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017.
- [82] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- [83] Andrew Ng. Advice for applying machine learning. *Stanford University*, 2011. <http://cs229.stanford.edu/materials/ML-advice.pdf>.
- [84] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.
- [85] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, 39(1):43–62, 1997.
- [86] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

- [87] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616, 2009.
- [88] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.
- [89] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. *Advances in Neural Information Processing Systems*, pages 2539–2547, 2015.
- [90] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.
- [91] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. *Advances in Neural Information Processing Systems*, pages 469–477, 2016.
- [92] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.
- [93] Zili Yi, Hao (Richard) Zhang, Ping Tan, and Minglun Gong. DualGAN: Unsupervised dual learning for image-to-image translation. *ICCV*, pages 2868–2876, 2017.
- [94] Ken-ichi Funahashi and Yuichi Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6(6):801–806, 1993.
- [95] Ching-Hung Lee and Ching-Cheng Teng. Identification and control of dynamic systems using recurrent fuzzy neural networks. *IEEE Transactions on Fuzzy Systems*, 8(4):349–366, 2000.

- [96] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Recurrent neural network based language model. *11th Annual Conference of the International Speech Communication Association*, 2010.
- [97] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6645–6649, 2013.
- [98] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.
- [99] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.
- [100] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *International Conference on Machine Learning*, pages 2048–2057, 2015.
- [101] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.
- [102] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

- [103] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.
- [104] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.
- [105] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. DenseCap: Fully convolutional localization networks for dense captioning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4565–4574, 2016.
- [106] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [107] Kevin McGrattan, Simo Hostikka, Randall McDermott, Jason Floyd, and Marcos Vanella. Fire dynamics simulator (version 6), user’s guide. *NIST Special Publication*, 1019(6):1–367, 2018.
- [108] RO Weber. Modelling fire spread through fuel beds. *Progress in Energy and Combustion Science*, 17(1):67–82, 1991.
- [109] AL Sullivan. A review of wildland fire spread modelling, 1990-present 1: Physical and quasi-physical models. *arXiv:0706.3074v1[physics.geo-ph]*, 2008.

- [110] AL Sullivan. A review of wildland fire spread modelling, 1990-present 2: Empirical and quasi-empirical models. *arXiv:0706.4128[physics.geo-ph]*, 2013.
- [111] Richard C Rothermel et al. A mathematical model for predicting fire spread in wildland fuels. Technical report, USDA Forest Service, 1972.
- [112] Mark A Finney. Mechanistic modeling of landscape fire patterns. *Spatial Modeling of Forest Landscapes: Approaches and Applications*. Cambridge University Press, Cambridge, pages 186–209, 1999.
- [113] Mark A Finney et al. *FARSITE, Fire Area Simulator–model development and evaluation*, volume 3. US Department of Agriculture, Forest Service, Rocky Mountain Research Station Ogden, UT, 1998.
- [114] Ronald G Rehm and Randall J McDermott. *Fire-front propagation using the level set method*. US Department of Commerce, National Institute of Standards and Technology, 2009.
- [115] Chris Lautenberger. Wildland fire modeling with an Eulerian level set method and automated calibration. *Fire Safety Journal*, 62:289–298, 2013.
- [116] Alan Lattimer, Jeff Borggaard, Serkan Gugercin, Kray Luxbacher, and Brian Lattimer. Computationally efficient wildland fire spread models. *Proceedings of the 14th International Fire Science & Engineering Conference*, pages 305–315, 2016.
- [117] William Mell, Mary Ann Jenkins, Jim Gould, and Phil Cheney. A physics-based approach to modelling grassland fires. *International Journal of Wildland Fire*, 16(1):1–22, 2007.
- [118] Frank A Albini. Estimating wildfire behavior and effects. Technical report, USDA Forest Service, 1976.

- [119] Patricia L Andrews. Modeling wind adjustment factor and midflame wind speed for rothermel's surface fire spread model. *Gen. Tech. Rep. RMRS-GTR-266. Fort Collins, CO: Department of Agriculture, Forest Service, Rocky Mountain Research Station.* 39 p., 213, 2012.
- [120] CE Van Wagner. A simple fire-growth model. *The Forestry Chronicle*, 45(2):103–104, 1969.
- [121] DG Green, A Malcolm Gill, and IR Noble. Fire shapes and the adequacy of fire-spread models. *Ecological Modelling*, 20(1):33–45, 1983.
- [122] Patricia L Andrews. Behaveplus fire modeling system, version 5.0: Variables. *Gen. Tech. Rep. RMRS-GTR-213 Revised. Fort Collins, CO: Department of Agriculture, Forest Service, Rocky Mountain Research Station.* 111 p., 213, 2009.
- [123] Matthew G Rollins. LANDFIRE: a nationally consistent vegetation, wildland fire, and fuel assessment. *International Journal of Wildland Fire*, 18(3):235–249, 2009.
- [124] Kevin McGrattan and Stewart Miles. Modeling fires using computational fluid dynamics (CFD). *SFPE Handbook of Fire Protection Engineering*, pages 1034–1065, 2016.
- [125] Richard D Peacock, KB McGrattan, Glenn P Forney, and Paul A Reneke. CFAST-consolidated fire and smoke transport (version 7)-volume 3: Verification and validation guide. *NIST Technical Note 1889v3 (National Institute of Standards and Technology, Gaithersburg, MD, 2015).*, 2015.
- [126] L Wang and Q Chen. Theoretical and numerical studies of coupling multizone and CFD models for building air distribution simulations. *Indoor Air*, 17(5):348–361, 2007.
- [127] Yoshihiro Ishida and Shinsuke Kato. Method for coupling three-dimensional transient pollutant transport into one-dimensional transport simulation based on concentration response factor. *ASHRAE Transactions*, 114:259, 2008.

- [128] JA Clarke, JLM Hensen, and COR Negrao. Predicting indoor airflow by combining network, cfd, and thermal simulation. *16th AIVC Conf. "Implementing the Results of Ventilation Research"*, pages 145–154, 1995.
- [129] Cezar OR Negrão. Integration of computational fluid dynamics with building thermal and mass flow simulation. *Energy and Buildings*, 27(2):155–165, 1998.
- [130] M Bartak, I Beausoleil-Morrison, JA Clarke, J Denev, F Drkal, M Lain, IA Macdonald, A Melikov, Zbigniew Popielek, and P Stankov. Integrating CFD and building simulation. *Building and Environment*, 37(8-9):865–871, 2002.
- [131] E Djunaedy, JLM Hensen, and MGLC Loomans. External coupling between CFD and energy simulation: implementation and validation. *ASHRAE Transactions*, 111(1):612–624, 2005.
- [132] Kyosuke Hiyama and Shinsuke Kato. Integration of three-dimensional CFD results into energy simulations utilizing an advection–diffusion response factor. *Energy and Buildings*, 43(10):2752–2759, 2011.
- [133] Weirong Zhang, Kyosuke Hiyama, Shinsuke Kato, and Yoshihiro Ishida. Building energy simulation considering spatial temperature distribution for nonuniform indoor environment. *Building and Environment*, 63:89–96, 2013.
- [134] Liangzhu Wang and Qingyan Chen. Applications of a coupled multizone-CFD model to calculate airflow and contaminant dispersion in built environments for emergency management. *HVAC&R Research*, 14(6):925–939, 2008.
- [135] Francesco Colella, Guillermo Rein, Romano Borchellini, R Carvel, Jose L Torero, and Vittorio Verda. Calculation and design of tunnel ventilation systems using a two-scale modelling approach. *Building and Environment*, 44(12):2357–2367, 2009.

- [136] Jason Floyd. Coupling a network hvac model to a computational fluid dynamics model using large eddy simulation. *Fire Safety Science*, 10:459–470, 2011.
- [137] Francesco Colella, Guillermo Rein, Vittorio Verda, and Romano Borchiellini. Multiscale modeling of transient flows from fire and ventilation in long tunnels. *Computers & Fluids*, 51(1):16–29, 2011.
- [138] Francesco Colella, Guillermo Rein, Romano Borchiellini, and Jose L Torero. A novel multiscale methodology for simulating tunnel ventilation flows during fires. *Fire Technology*, 47(1):221–253, 2011.
- [139] Moohyun Cha, Soonhung Han, Jaikyung Lee, and Byungil Choi. A virtual reality based fire training simulator integrated with fire dynamics data. *Fire Safety Journal*, 50:12–24, 2012.
- [140] Izabella Vermesi, Guillermo Rein, Francesco Colella, Morten Valkvist, and Grunde Jomaas. Reducing the computational requirements for simulating tunnel fires by combining multiscale modelling and multiple processor calculation. *Tunnelling and Underground Space Technology*, 64:146–153, 2017.
- [141] Ali Haghigat, Kray Luxbacher, and Brian Y Lattimer. Development of a methodology for interface boundary selection in the multiscale road tunnel fire simulations. *Fire Technology*, pages 1–38, 2018.
- [142] Chih-Wei Chang and Nam T Dinh. Classification of machine learning frameworks for data-driven thermal fluid models. *International Journal of Thermal Sciences*, 135:559–579, 2019.
- [143] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

- [144] Roland Memisevic and Geoffrey Hinton. Unsupervised learning of image transformations. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [145] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.
- [146] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.
- [147] Marc Janssens and Hao C Tran. Data reduction of room tests for zone model validation. *Journal of Fire Sciences*, 10(6):528–555, 1992.
- [148] Richard D Peacock, W Jones, P Reneke, and G Forney. CFAST-consolidated model of fire growth and smoke transport (version 6) user’s guide. *NIST Special Publication*, 1041, 2005.
- [149] Changsong Luo, Liguo Chen, Jim Lua, and Philip Liu. ABAQUS fire interface simulator toolkit (AFIST) for coupled response prediction. *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 18th AIAA/ASME/AHS Adaptive Structures Conference 12th*, page 2768, 2010.
- [150] LIGUO Chen, CHANGSONG Luo, and J Lua. FDS and Abaqus coupling toolkit for fire simulation and thermal and mass flow prediction. *Fire Safety Science*, 10:1465–1477, 2011.
- [151] Antti Paajanen, Timo Korhonen, Sippola Merja, Simo Hostikka, Michal Malendowski, and Renaud Gutkin. FDS2FEM—a tool for coupling fire and structural analyses. *IABSE Symposium Report*, 100(4):218–224, 2013.
- [152] JA Feenstra. *FDS-2-Abaqus*. PhD thesis, Eindhoven University of Technology, 2016.

- [153] Jonathan L. Hodges, Christian M. Rippe, Scott W. Case, and Brian Y. Lattimer. Predicting the structural response of a compartment fire using full-field heat transfer measurements. *Fire Safety Journal*, 2017.
- [154] N Cholewa, PT Summers, S Feih, AP Mouritz, BY Lattimer, and SW Case. A technique for coupled thermomechanical response measurement using infrared thermography and digital image correlation (TDIC). *Experimental Mechanics*, 56(2):145–164, 2016.
- [155] Jonathan L. Hodges, Christian M. Rippe, Scott W. Case, and Brian Y. Lattimer. Thermo-graphic digital image correlation (TDIC) measurements of mechanically-loaded structures with fire in the field of view. *Proceedings of the 9th International Conference on Structures in Fire*, pages 811–818, 2016.
- [156] Jonathan L. Hodges, Christian M. Rippe, Nathan Cholewa, Patrick T Summers, Zenka Mathys, Adrian P. Mouritz, Scott W. Case, and Brian Y. Lattimer. Resolving obstructed views using multiple thermography digital image correlation systems. *Proceedings of the 14th International Fire Science & Engineering Conference*, pages 47–57, 2016.
- [157] Randall J McDermott. A velocity divergence constraint for large-eddy simulation of low-mach flows. *Journal of Computational Physics*, 274:413–431, 2014.
- [158] James W Deardorff. Stratocumulus-capped mixed layers derived from a three-dimensional model. *Boundary-Layer Meteorology*, 18(4):495–527, 1980.
- [159] Stephen B Pope. *Turbulent flows*. Cambridge University Press, 2000.
- [160] Ralph M Nelson Jr. An effective wind speed for models of fire spread. *International Journal of Wildland Fire*, 11(2):153–161, 2002.
- [161] Hubert Schreier, Jean-José Orteu, and Michael A Sutton. *Image correlation for shape, motion and deformation measurements*. Springer US, 2009.

- [162] Christian M Rippe. *Burnthrough Modeling of Marine Grade Aluminum Alloy Structural Plates Exposed to Fire*. PhD thesis, Virginia Tech, 2015.
- [163] Satoru Yoneyama, A Kitagawa, S Iwata, K Tani, and H Kikuta. Bridge deflection measurement using digital image correlation. *Experimental Techniques*, 31(1):34–40, 2007.
- [164] Bing Pan, Dafang Wu, and Yong Xia. An active imaging digital image correlation method for deformation measurement insensitive to ambient light. *Optics & Laser Technology*, 44(1):204–209, 2012.
- [165] Bing Pan, Dafang Wu, and Liping Yu. Optimization of a three-dimensional digital image correlation system for deformation measurements in extreme environments. *Applied Optics*, 51(19):4409–4419, 2012.
- [166] Ryan B Berke and John Lambros. Ultraviolet digital image correlation (UV-DIC) for high temperature applications. *Review of Scientific Instruments*, 85(4):045121, 2014.
- [167] Correlated Solutions. Vic-3d user manual. *Columbia, SC: Correlated Solutions*, 2005.
- [168] Shengbo Shi, Linjie Li, Guodong Fang, Jun Liang, Fajun Yi, and Guochang Lin. Three-dimensional modeling and experimental validation of thermomechanical response of frp composites exposed to one-sided heat flux. *Materials & Design*, 99:565–573, 2016.
- [169] Stuart W Churchill and Humbert HS Chu. Correlating equations for laminar and turbulent free convection from a vertical plate. *International Journal of Heat and Mass Transfer*, 18(11):1323–1329, 1975.
- [170] Haukur Ingason and John de Ris. Flame heat transfer in storage geometries. *Fire Safety Journal*, 31(1):39–60, 1998.
- [171] S E Dillon. Analysis of the ISO 9705 room/corner test: Simulations, correlations and heat flux measurements. Technical report, National Institute of Standards and Technology, 1998.

- [172] NR Keltner. Directional flame thermometers-a tool for measuring thermal exposure in furnaces and improving control. *Proceedings of the 11th International Fire Science & Engineering Conference*, 2007.
- [173] PF Lennon and GWH Silcock. An investigation of the ability of a thin plate heat flux device to determine the incident heat fluxes during enclosure fires. *International Journal on Engineering Performance-Based Fire Codes*, 3(1):1–15, 2001.
- [174] Christian M Rippe and Brian Y Lattimer. Full-field surface heat flux measurement using non-intrusive infrared thermography. *Fire Safety Journal*, 78:238–250, 2015.
- [175] Dat Duthinh, Kevin McGrattan, and Abed Khaskia. Recent advances in fire–structure analysis. *Fire Safety Journal*, 43(2):161–167, 2008.
- [176] Andrew R Gifford, David O Hubble, Pullins A Clayton, Thomas E Diller, and Scott T Huxtable. Durable heat flux sensor for extreme temperature and heat flux environments. *Journal of Thermophysics and Heat Transfer*, 24(1):69–76, 2010.
- [177] B Lattimer and Thomas Diller TE. Quantifying thermal boundary condition details using a hybrid heat flux gage. *Fire Safety Science*, 10:987–1000, 2011.
- [178] NA Dembsey, Patrick J Pagni, and Robert Brady Williamson. Compartment fire experiments: comparison with models. *Fire Safety Journal*, 25(3):187–227, 1995.
- [179] Peter S Veloo and James G Quintiere. Convective heat transfer coefficient in compartment fires. *Journal of Fire Sciences*, 31(5):410–423, 2013.
- [180] Paul E Sincaglia and Jonathan R Barnett. Development of a glass window fracture model for zone-type computer fire codes. *Journal of Fire Protection Engineering*, 8(3):101–117, 1996.

- [181] Patrick T Summers, Yanyun Chen, Christian M Rippe, Ben Allen, Adrian P Mouritz, Scott W Case, and Brian Y Lattimer. Overview of aluminum alloy mechanical properties during and after fires. *Fire Science Reviews*, 4(1):3, 2015.
- [182] Brian Y Lattimer. Heat fluxes from fires to surfaces. *SFPE Handbook of Fire Protection Engineering*, pages 745–798, 2015.

Appendices

Appendix A

Fire Modeling Using Computational Fluid Dynamics

A.1 Foreword

The contents of this appendix are intended to provide additional detail regarding the high resolution CFD models used in the other chapters of this work. The discussion regarding Fire Dynamics Simulator (FDS) is based largely on McDermott's low-Mach flows derivation [157], the Society of Fire Protection Engineering (SFPE) handbook [124], and the FDS technical reference guide [40]. The discussion regarding two-dimensional fire spread modeling using the model of Rothermel is based largely on the FARSITE model development paper [113].

A.2 Introduction

Fire dynamics is a complex process involving multi-mode heat transfer, reacting fluid flow, and the reaction of combustible materials. A fire releases heat to the environment which heats the combustion products resulting in a buoyancy driven plume. The released energy is transported to the environment through convection and radiation. Energy which is transferred to fuel in the environment (solid or liquid) may result in the gasification of more fuel which may ignite.

Computational fluid dynamics (CFD) fire models are often used in fire protection engineering to predict complex flow fields in structure fire scenarios. Wildland fires are typically modeled using a phenomenological model due to the large domain of interest. The applicability of each model to new fire scenarios is directly related to how well the modeling assumptions are met.

The following sections provide more detail on the equations used in the CFD packages used in this work. Section A.3 discusses the low-Mach number flow equations used in FDS. Section A.4 discusses the two-dimensional implementation of the model of Rothermel in FARSITE.

A.3 Summary of Conservation Equations in Low-Mach Number Flows

The governing equations of fire modeling include conservation of mass, species mass concentration balance, conservation of momentum, conservation of energy, and ideal gas equation of state. A brief overview of these equations and their applicability to fire modeling is presented in this section. The majority of this section is drawn from McDermott's low-Mach flows derivation [157], the Society of Fire Protection Engineering (SFPE) handbook [124], and the FDS technical reference guide [40]. Additional details regarding the derivation and numerical schemes can be found therein.

Following the method of McDermott, the pressure is decomposed into two components in the low-Mach formulation such that

$$p(\mathbf{x}, t) = \bar{p}(z, t) + \tilde{p}(\mathbf{x}, t) \quad (\text{A.1})$$

where \mathbf{x} is the position within the domain, z is the height within the domain, t is time, p is the overall pressure field, $\bar{p}(z, t)$ is the background thermodynamic pressure used in the equation of

state and $\tilde{p}(\mathbf{x}, t)$ is the fluctuating hydrodynamic pressure which drives fluctuations in the flow field.

Conservation of mass can be expressed as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0 \quad (\text{A.2})$$

where ρ is the density of the gas, and \mathbf{u} is the velocity. This equation represents the fundamental principle that mass is neither created nor destroyed. In fire modeling it is often necessary to consider the transport of individual species such as fuel and oxygen. Conservation of mass for a mixture of gases can be written as the set of mass balance equations

$$\frac{\partial (\rho Y_\alpha)}{\partial t} + \nabla \cdot (\rho Y_\alpha \mathbf{u}) = \nabla \cdot (\rho D_\alpha \nabla Y_\alpha) + \dot{m}_\alpha''' \quad (\text{A.3})$$

where the left-hand side of the equation corresponds to the transport of the species, α , and the right-hand side corresponds to the diffusion and production of α . When the conservation of mass equations for all species are added together, the diffusion and production terms will sum to zero. The production term for each species is a result of on-going chemical reactions.

Conservation of momentum is expressed as

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = -\nabla \tilde{p} + \nabla \cdot \tau + (\rho - \rho_0) \mathbf{g} \quad (\text{A.4})$$

where τ is the deviatoric stress tensor, ρ_0 is the reference density at ambient conditions, and \mathbf{g} is the acceleration due to gravity. Since the impact of pressure on the flow field is based on pressure differences, the impact of the \bar{p} is neglected and only \tilde{p} is retained in the conservation of momentum equation.

Conservation of energy based on sensible enthalpy is expressed as

$$\frac{\partial(\rho h)}{\partial t} + \nabla \cdot (\rho h \mathbf{u}) = \frac{D\bar{p}}{Dt} + \dot{q}''' - \nabla \cdot \dot{\mathbf{q}}'' \quad (\text{A.5})$$

where h is the sensible enthalpy per unit mass, \dot{q}''' is the heat release per unit volume from reactions, and $\dot{\mathbf{q}}$ is the net energy flux into a point. Since the impact of pressure on the sensible enthalpy is based on the mean pressure, the impact of \tilde{p} is neglected and only \bar{p} is retained in the conservation of energy equation. The equation of state is used to close the system of equations. Typically, it is sufficient in fire applications to represent the gas as an ideal gas such that

$$\rho = \frac{\bar{p}\bar{W}}{RT} \quad (\text{A.6})$$

where \bar{W} is the mixture molecular weight, R is the universal gas constant, and T is the temperature of the gas.

FDS uses a large-eddy simulation (LES) approach to solving Eq. A.2 - Eq. A.6. The premise of LES is to apply a low pass filter to the transport equations of mass, momentum, and energy. The predicted flow fields capture the structure of large eddies in the flow; however, the smallest length scales are not resolved. Since smaller length scales have a significant impact on the dissipation of kinetic energy, the impact of these scales on the flow must be modeled. Several sub-grid turbulence models have been proposed in the literature. The turbulence model used in FDS is the eddy viscosity subgrid closure model of Deardorff [40, 158, 159]. FDS uses implicit filtering where the filter width, Δ , is equal to the grid size. The FDS simulations presented in Chapter 2 used a grid size based on the characteristic fire diameter

$$D^* = \left(\frac{\dot{Q}}{\rho_\infty c_p T_\infty \sqrt{g}} \right)^{\frac{2}{5}} \quad (\text{A.7})$$

where D^* is the characteristic fire diameter, \dot{Q} is the heat release rate of the fire, ρ_∞ is the air density at T_∞ , c_p is the specific heat capacity of the air at T_∞ , and T_∞ is the ambient air temperature. Grid spacing of $D^*/dx = 10$ was found to be sufficiently converged for the simulations in Chapter 2. The FDS simulations presented in Chapter 4 used a fixed grid spacing of 0.1m which is typically sufficient for evaluating the spread of smoke and heat through a building [107]. Additional details regarding the computational algorithm used in FDS can be found in the FDS technical reference guide [40].

The system of equations in Eq. A.2 - Eq. A.6 represent the underlying physics of low-speed buoyancy driven reacting flows. Solving these equations at a high spatial-temporal resolution comes at a computational cost, where simulation times are often measured in hours, days, or even weeks depending on the simulation. More rapid predictions over large domains are possible with zone based compartment fire models which apply the transport equations to a single grid point per room in a structure. The work presented in Chapter 4 demonstrates that a feature based neural network is capable of using coarse predictions from a zone based fire model to predict high resolution intensive properties from Eq. A.2 - Eq. A.6. By training the weights and biases embedded in the network based on high resolution predictions, the neural network becomes an optimal approximation function of Eq. A.2 - Eq. A.6 for compartment fires.

A.4 Summary of Two-Dimensional Rothermel Predictions

In the Rothermel/Albini phenomenological fire spread model, the peak surface fire spread rate, $V_{s,peak}$ is calculated using the equation

$$V_{s,peak} = \frac{\dot{Q}''\zeta}{\rho\epsilon Q_{ig}} (1 + \phi_s + \phi_w) \quad (\text{A.8})$$

where Q'' is the heat release rate per unit area, ρ is the fuel density, Q_{ig} is the heat of pre-ignition, ζ is the propagating flux ratio (percentage of heat released which pre-ignites fuel), ϵ is the effective heating number (percentage of fuel which is involved in ignition), ϕ_w is the wind coefficient, and ϕ_s is the slope coefficient.

Various researchers have developed empirical relationships for the different parameters in Eq. A.8. A commonly used approach in the literature is to specify Q'' , ρ , ζ , ϵ based on classifying the primary fuel in a region into a fuel model. A total of 53 fuel models were considered in this work including 13 developed by Rothermel/Albini [111, 118], and 40 developed by Scott [50]. Rothermel presented an empirical relationship for Q_{ig} based on the fuel model and moisture content, and Scott extended the relationship to handle dynamic fuel models. Rothermel presented empirical relationships for ϕ_w and ϕ_s based on fuel model, midflame wind speed, and slope. Andrews presented an algorithm to adjust typical atmospheric wind measurements (10m or 20ft) to midflame wind speed based on three additional parameters describing the upper story vegetation (canopy cover, canopy height, and crown ratio) [119].

The model of Rothermel is designed to predict the rate of spread in the peak direction of spread. Predictions in two-dimensions using this model generally rely the empirical observation that wildland fires generally grow in an ellipsoidal shape for homogeneous spatial conditions based on $V_{s,peak}$ and wind speed [112, 120, 121]. Using this type of approach, the two-dimensional propagation of the fire from a discrete point is predicted by calculating the heading fire spread velocity and estimating the eccentricity of the ellipse. This corresponds to predicting the two-dimensional spread of a wildland fire over homogeneous landscape and vegetation.

The heading spread rate, or rate of spread in the peak spread direction, is calculated using Eq. A.8. The heading spread direction is calculated using the equation

$$\theta = \text{atan} (V_y, V_x) \quad (\text{A.9})$$

where θ is the peak spread direction relative to the slope, V_y is the y component of the vector combination of wind spread rate, V_w , and slope spread rate, V_s , and V_x is the x component of the vector combination of V_w and V_s . The slope spread rate is calculated using Eq. A.8 with ϕ_w equal to zero. Similarly, the wind spread rate is calculated using Eq. A.8 with ϕ_s equal to zero [122].

The backing spread rate, or rate of spread in the opposite direction of the heading spread, is calculated from ϵ using the equation

$$V_{s,min} = V_{s,peak} \times \frac{(1.0 - \epsilon)}{(1.0 + \epsilon)} \quad (\text{A.10})$$

where $V_{s,min}$ is the backing spread rate, and ϵ is the eccentricity of the ellipse. The eccentricity of the ellipse is calculated using the equation [122]

$$\epsilon = \frac{\sqrt{r^2 - 1.0}}{r} \quad (\text{A.11})$$

where r is the length-to-width ratio. The length-to-width ratio is calculated using the equation

$$r = 1.0 + 0.25U_{eff} \quad (\text{A.12})$$

where U_{eff} is the effective wind speed in the heading direction. The effective wind speed is the magnitude of the vector combination of the ambient wind velocity and the buoyant velocity generated by the fire perimeter [160]. The buoyant velocity generated by the fire perimeter is calculated using the equation

$$U_b = \left(\frac{2gQ_c''}{\rho_a c_{p,a} T_a} \right)^{1/3} \quad (\text{A.13})$$

where U_b is the buoyant velocity, g is the acceleration due to gravity, Q_c'' is the convection component of the heat release rate per unit area, ρ_a is the air density, $c_{p,a}$ is the air specific heat capacity, and T_a is the air temperature.

The discussion up to this point describes the methodology to calculate the two-dimensional rate of spread using the model of Rothermel in homogeneous conditions. In heterogeneous conditions, the differences in fuel and elevation across the landscape will result in different points along the fire perimeter growing at different rates. One methodology to predict the changing shape of the fire perimeter in heterogeneous conditions is to consider the fire perimeter as a propagating wave, often characterized by Huygens' principle. The overall propagation of the wave can be characterized by the propagation of each point along the wavefront. This is the methodology adopted in FARSITE, which was used in this work [112].

The first stage in predicting the fire spread over a heterogeneous landscape is to discretize the current fire perimeter into a series of points. The spatial resolution of the discretization must be sufficient to capture the spatial variation in the landscape; however, the computational time increases significantly with increased spatial resolution. Typically, the spatial resolution of the fire perimeter in FARSITE is fixed at 30-60m. The two-dimensional rate of spread of each point in the fire perimeter is calculated using the homogeneous rate of spread methodology. The rate of spread for each point propagates the fire perimeter for a fixed duration. Typically, the duration of the propagation in FARSITE is fixed at 30-60 minutes. At the end of each time-step, the updated fire perimeter is re-discretized. Details regarding the FARSITE implementation and error handling are presented in [112].

The focus of this type of wildland fire modeling is to predict the changing shape of the fire perimeter as a function of time. Data-driven modeling using convolutional neural networks uses the idea of feature recognition to classify or generate images. The work presented in Chapter 3 demonstrates that a feature based neural network is capable of learning the impact of the complex interaction of the spatially varying landscape and vegetation on the growth of a wildland fire perimeter.

Appendix B

Thermographic Digital Image Correlation (TDIC) Measurements of Mechanically Loaded Structures

B.1 Foreword

The contents of this appendix are published in the proceedings of the 9th International Conference on Structures in Fire at Princeton University, reproduced by permission [155]. The published work has four author's including Jonathan L. Hodges, Christian M. Rippe, Scott W. Case, and Brian Y. Lattimer. Jonathan conducted the experiments and was the primary author of the paper. Christian assisted with the experiments. Scott and Brian advised on experimental design and provided feedback on the paper.

B.2 Abstract

This research examined the use of Thermographic Digital Image Correlation (TDIC) in fire testing with flames in the field of view. The error and uncertainty in the deformation measurements obtained via TDIC with fire in the field of view were reduced by ensuring the gray-scale histogram of the

CCD images was neither under-saturated nor over-saturated once fire was present. Quantitatively, this corresponded to the gray-scale histogram having a mean value between 13-91 out of 255 and a kurtosis between 2.7-10.7 without the fire present. Results obtained from testing using this setup were found to reduce data loss due to overexposure when compared to results obtained using arbitrary light conditions. Thus, future investigators may reduce the error in the system for live fire testing before exposing the sample to fire by setting up lighting and camera settings to obtain histograms in this range.

B.3 Introduction

In structural experiments involving fire it is often desirable to measure both deformation and temperature fields of load-carrying members being tested. Traditionally these measurements are obtained using physical methods such as string potentiometers and thermocouples. Obtaining full field measurements requires an array of these devices along the specimen. Unfortunately it can be difficult to attach these devices without affecting the physical system. In addition, it is often difficult to determine where failure may occur so that instrumentation can be appropriately positioned. Non-contact measurement methods are desirable to reduce setup issues and ensure the appropriate data is captured.

Numerous non-contact measurement methods have been proposed for obtaining either deformation or temperature field measurements; however, few have been proposed which simultaneously measure both quantities. Thermographic Digital Image Correlation (TDIC) is an optical method for measuring full field 3-D deformation and temperature on surfaces by mapping simultaneously captured CCD and IR images to a common coordinate system and tracking anisotropic speckle patterns [154]. The fusion of the DIC and thermography measurements through the calibration process allows temperature and deflections to be aligned and tracked, even in cases with large

deflections.

Traditionally DIC has been limited to laboratory conditions since the DIC algorithms depend on consistent lighting [161], which has limited the use of this method in fire testing to measurements without fire in the field of view (FOV) of the cameras [154, 162]. Recent studies have been performed testing this limitation. Yoneyama *et al.* [163] used DIC to measure the deformation of an in-service bridge where ambient conditions may change. Pan *et al.* [164] investigated the potential use of bandpass filters and monochromatic lighting to minimize the effects of ambient lighting on DIC, and this system has been tested in high temperature applications [164, 165, 166]. However, the effects of active flames in the FOV (which will affect the sample and sensor in different ways over time) have not been quantified. The purpose of this study was to determine ideal camera settings to minimize the error and uncertainty in TDIC measurements of displacement for structural elements exposed to diffusion flames that are in the FOV of the measurement cameras.

B.4 TDIC Setup

The TDIC system used in this testing consists of two CCD cameras and one IR camera. The CCD cameras used were Allied Vision Technologies Prosilica GE 4000 CCD cameras (12-bit, 4008 x 2672 pixels, up to 10 Hz). Nikon AF Nikkor lenses with a focal length of 28mm were used with various f-stop values. The IR camera used was a FLIR A655sc (16-bit, 640 x 480 pixels, up to 100 Hz). Simultaneous imaging was achieved by using a hardware trigger to synchronize capture. The calibration procedure presented by Cholewa *et al.* [154] was used. The TDIC setup is shown in Fig. B.1. Images were sampled at 2 Hz, and deformation measurements were obtained using Vic-3D [167].

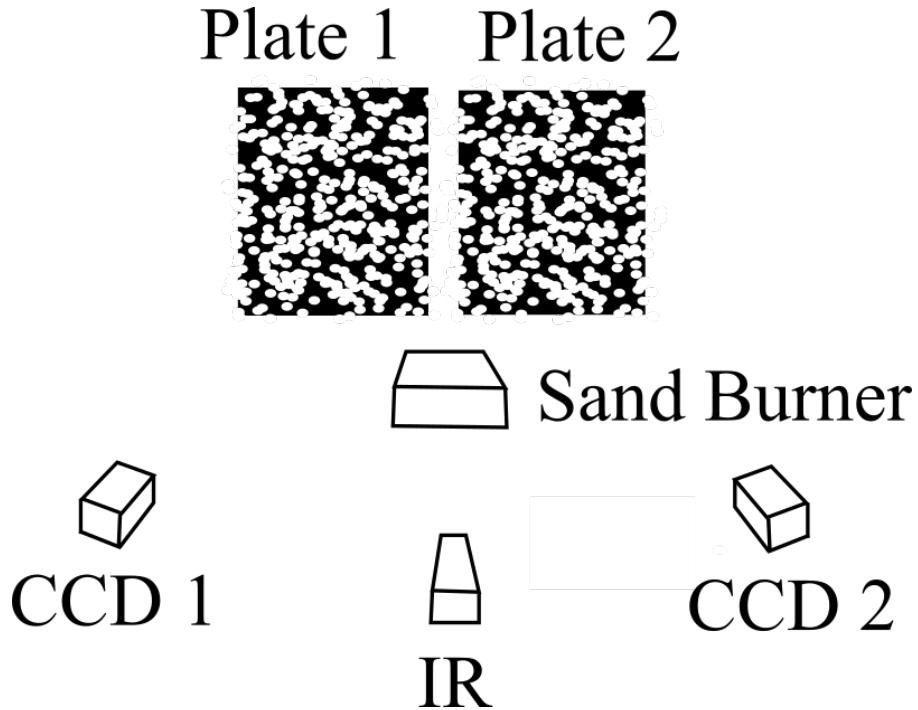


Figure B.1: Schematic of fire in FOV experiment

B.5 Fire in Field of View Testing

Aluminum 6061-T6 samples were exposed to 20 kW fires produced by a 0.3 m square sand burner. Two 0.60 m x 0.80 m plates were located 0.7 m behind the sand burner, and uniformly lit with 2 Hilio LED light panels, each providing a luminous output equivalent to a 1000W traditional light, set to maximum, as shown in Fig. B.2a. When a fire is in the FOV as in Fig. B.2b, the samples were occluded by overexposure from the flame, the overall intensity of the lighting on the plate increased, and the lighting was no longer consistent across the plate. An example of the change in contrast on the sample due to fire in the FOV is shown in Fig. B.2c which contains the gray-scale histogram of the sample images shown in Fig. B.2a-Fig. B.2b. Figure B.2c shows the percentage of over-saturated pixels (i.e., values at 255) increases drastically when the fire is in the FOV. Additionally, if all overexposed pixels are ignored in Fig. B.2c, the mean and standard deviation gray-scale intensity on the samples increases from 91 to 108 and 61 to 68, respectively.

Because a fire in the FOV will typically be transient, the effect on the intensity of reflected light at an individual location will change over time.

Because the contrast on the sample depends on aperture, exposure time, and fire size, a parametric study was performed to determine the optimum camera settings for TDIC testing with fire in the FOV. The metrics used to compare settings were projection error, uncertainty, and percentage of data loss. Projection error is defined as the root mean square euclidean distance in pixels of each point viewed from one CCD camera to the epipolar line of the point viewed from the other CCD camera. The uncertainty results from the DIC algorithm matching of speckle subsets. The percentage data loss is defined as the percentage of pixels which had data available at the start of the test from which measurements of deformation could not be obtained.

The results of testing with and without fire in the FOV for an f-stop (the ratio of focal length of the lens to the diameter of the entrance pupil) of 11 are shown in Fig. B.3. As expected, each metric is worse in fire testing than the corresponding baseline due to the transient pixel intensities caused by the fire exposure. Considering Fig. B.3 as a whole, exposure times in the range of 7.5-30 ms were found to correspond to the ideal contrast levels for testing with a 20 kW fire in the FOV with an f-stop of 11. Gray-scale histograms of images taken at the lower and upper contrast levels identified as ideal (7.5ms and 30ms exposure time for this test) are shown with and without fire in the FOV in Fig. B.4a and Fig. B.4b respectively. No significant variation was observed for various f-stop values with similar histograms.

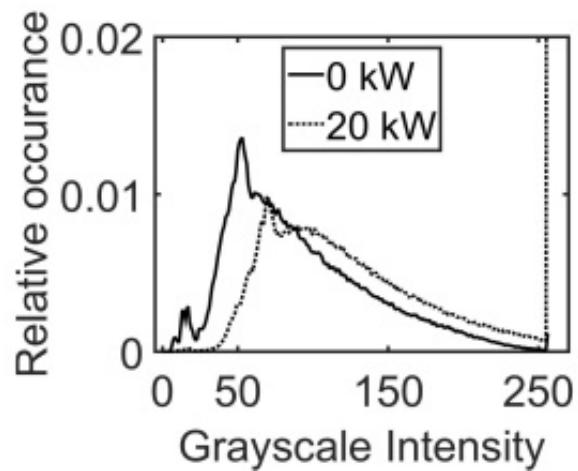
As the exposure time is increased the mean value of the histogram increases and the histogram becomes more flat. It was found that the ideal measurement regions had a mean gray-scale intensity between 13-91 out of 256 and a kurtosis (sharpness of peaks) between 2.7-10.7 without fire in the FOV. Additionally, less than 10% of the total pixels on the sample were overexposed once the fire was present. Thus, when initially calibrating the system, it is a best practice to sample an initial data set with varying contrast levels and comparing the gray-scale histograms of the samples



(a) Sample without fire



(b) Sample with fire



(c) Gray-scale histograms of a-b where the relative occurrence is the percentage each intensity value occurs on the sample

Figure B.2: Sample images and gray-scale histograms with and without fire exposure

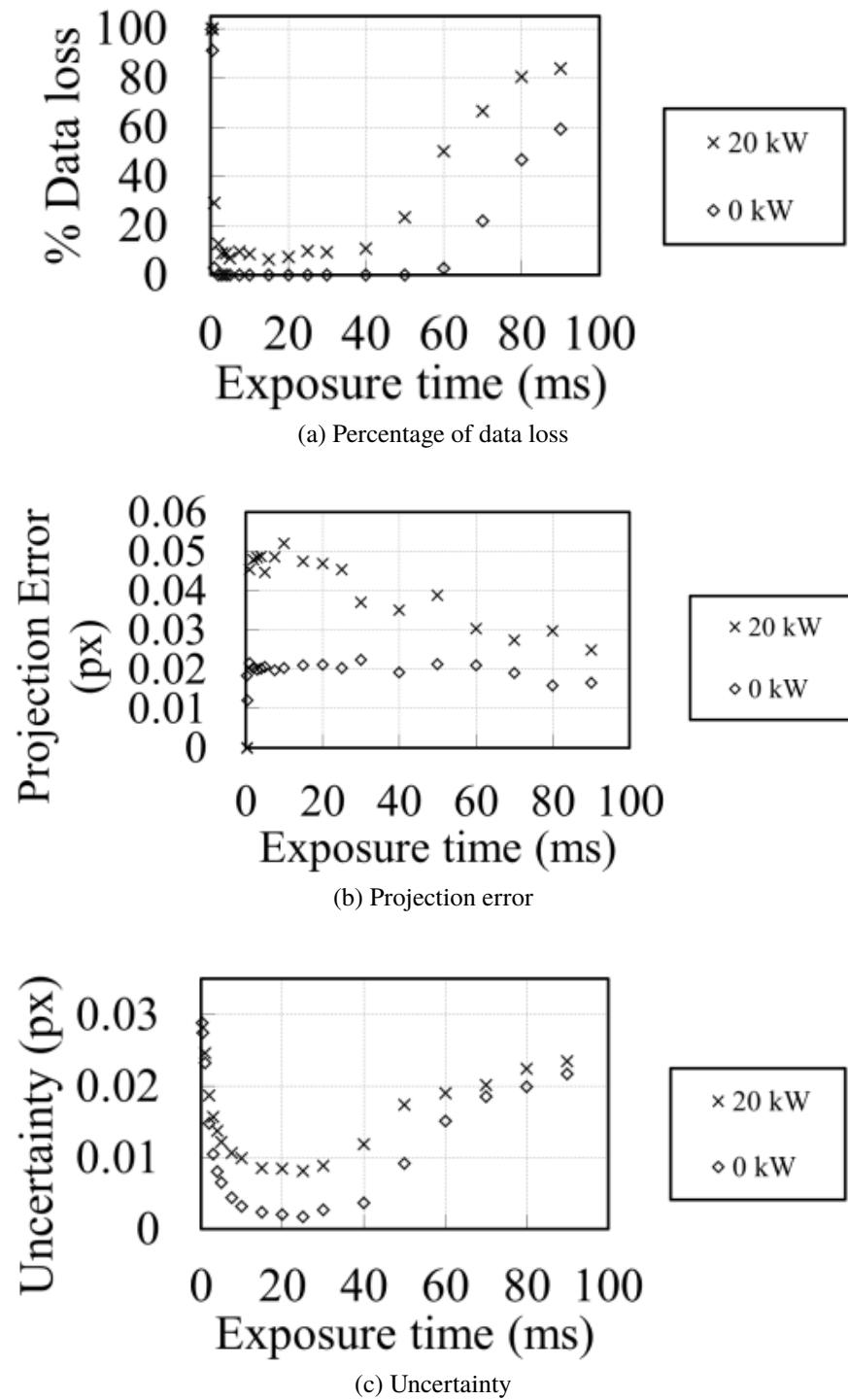
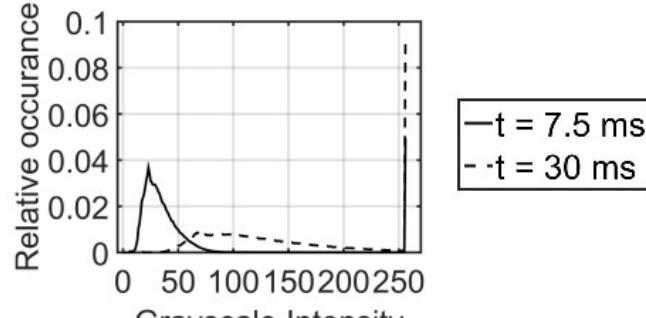


Figure B.3: Performance metrics of measurement from sample under various contrast levels with and without fire exposure

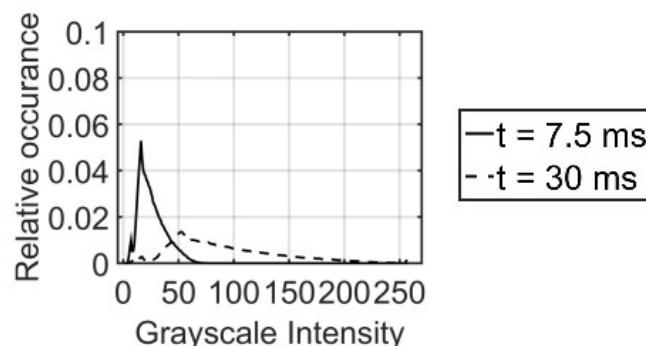
at each level. Choosing an exposure time such that the gray-scale histogram of the sample in the CCD images falls within the limits shown in Fig. B.4b before taking measurements with fire will result in lower error and less pixels overexposed by the fire. To examine the effects of using these camera settings on deformation measurements, TDIC measurements were taken as Plate 1 in Fig. B.1 was moved a known distance on a stage with 0.1 mm resolution. Between each movement, images were sampled with no fire, 20kW, 50kW, and 100kW fires, sampled at both 7.5ms and 30ms exposure times using the method presented herein. While better results could have been achieved by matching the histogram of the sample to that shown in Fig. B.3a with each fire size, the tests were conducted with the same settings across multiple fire sizes to test the method presented in this work. As shown for each camera setup in Fig. B.5 with different fire sizes, the most significant difference in performance is in the percentage of data loss. The average data loss increases as the fire size increases due to an increase in the percentage of over-saturated pixels on the sample. Additionally, better results for larger fire sizes are obtained when the gray-scale histograms of the sampled images are closer to the lower end of the range. This is due to the larger fire sizes shifting the higher end gray-scale histogram out of the ideal range. Using the camera setup presented herein improves performance by reducing the percentage of neighboring pixels over-saturated by the fire.

B.6 I-Beam Testing Using Ideal Gray-scale Histogram

To examine the utility of this method in structural testing, the camera settings identified above were used for testing of Aluminum 6061-T6 structural I-beams subjected to four point bending. The samples were supported 0.3 m from each end of the beam and loaded with 500 pounds split evenly between two points 0.25 m from the center of the beam. Baseline measurements were taken prior to mechanical loading; however, the system was allowed to rest with the static load for 15 minutes before fire testing began. Timestamps were taken to be zero at the start of fire testing.



(a) Gray-scale histograms with fire in the FOV



(b) Gray-scale histograms with the same camera setting as previous with no fire in the FOV

Figure B.4: Boundary sample gray-scale histograms showing viable contrast levels for TDIC measurements with fire in the FOV from exposure times identified in Fig. B.3

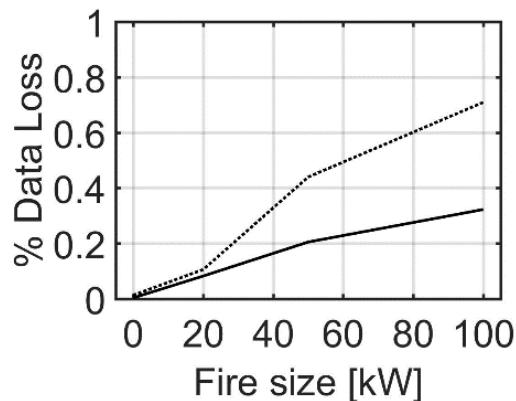
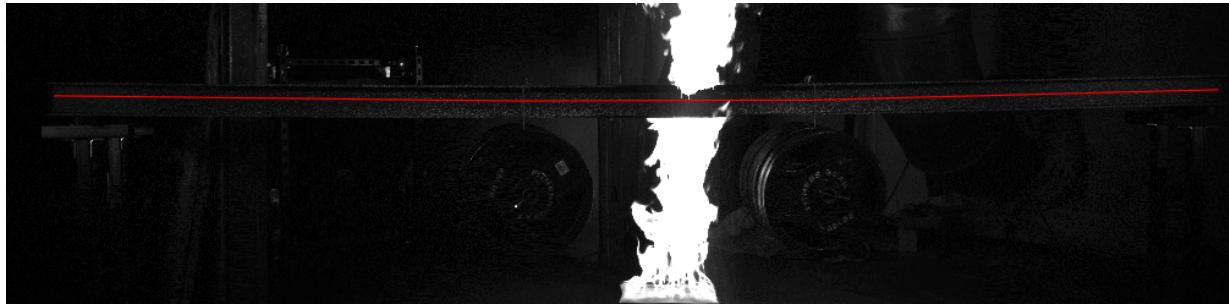
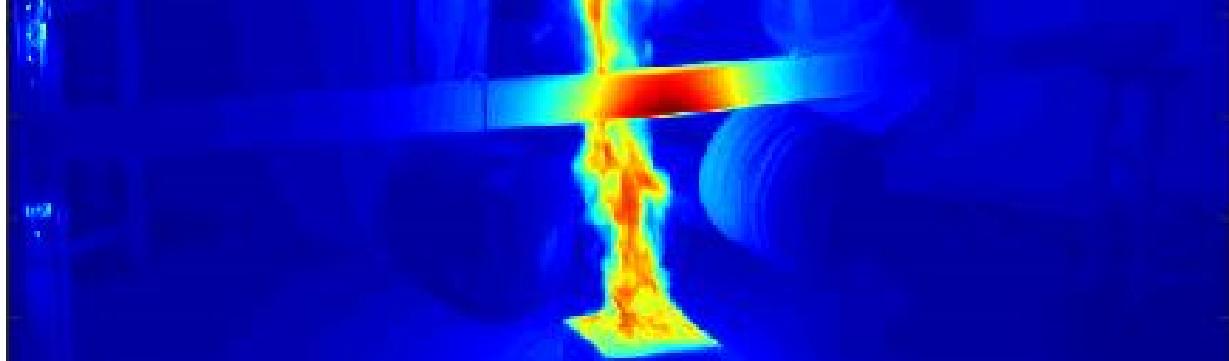


Figure B.5: Percentage of data loss with fire in the FOV using typical TDIC setup and the recommendations presented in this work for various fire sizes

An example image of an unloaded beam from a CCD camera is shown in Fig. B.6a, and an example of the beam exposed to fire viewed from the IR camera is shown in Fig. B.6b. The deformation and temperature of the beam along the line shown in Fig. B.6a throughout testing is shown in Fig. B.7. The empty points in Fig. B.7 correspond to times when the flame occluded the beam. The last point in Fig. B.7 shows a loss of correlation on the right side of the beam. This occurred due to the beam yielding into the fire and the CCD camera being unable to track the anisotropic speckle pattern.



(a) CCD camera



(b) IR camera

Figure B.6: Example images of a structural I-beam test from different cameras

A baseline I-beam test was performed with traditional TDIC setup parameters and the results compared to testing with adjusted contrast levels. An example image from each test is shown in Fig. B.8. The overlaid contour map corresponds to the deformation measurements from the TDIC system. Figure B.8a shows the pixels near the center of the beam lose correlation when the fire is present. This is due to the speckle pattern becoming over-saturated. In contrast, Fig. B.8b shows the pixels near the flame maintain correlation and provide deformation measurements in

the reduced contrast case. Thus, additional measurements of deformation are obtained using the adjusted contrast settings which otherwise would have been unavailable.

B.7 Conclusion

The application of TDIC measurement to open fire testing of structural members was examined. Initial results show that it is most important to ensure the gray-scale histogram of the speckling on the sample have as little as possible pixels over or under saturated. The mean and kurtosis of the gray-scale histogram were identified as two important parameters which can be used to quantify the contrast on a sample prior to fire testing. Calibrating the system without fire in the FOV such that the mean gray-scale intensity falls within 13-91 out of 255 and the kurtosis between 2.7-10.7 yielded improved results for a 20 kW fire in the field of view. Testing of additional fire sizes showed staying closer to the lower end of this range is beneficial as the presence of larger fires will result in the histogram on the sample shifting outside the identified ideal contrast range. Setting the cameras using this method, full field deformation measurements were obtained through a variety of intensity changes due to fire in the FOV.

B.8 Acknowledgements

The authors appreciate the sponsorship related to this work from N00014-13-1-0894 DURIP for the IR camera and DIC equipment. The work in this paper was supported by the Department of the Navy, contract N65540-10-C-0003.

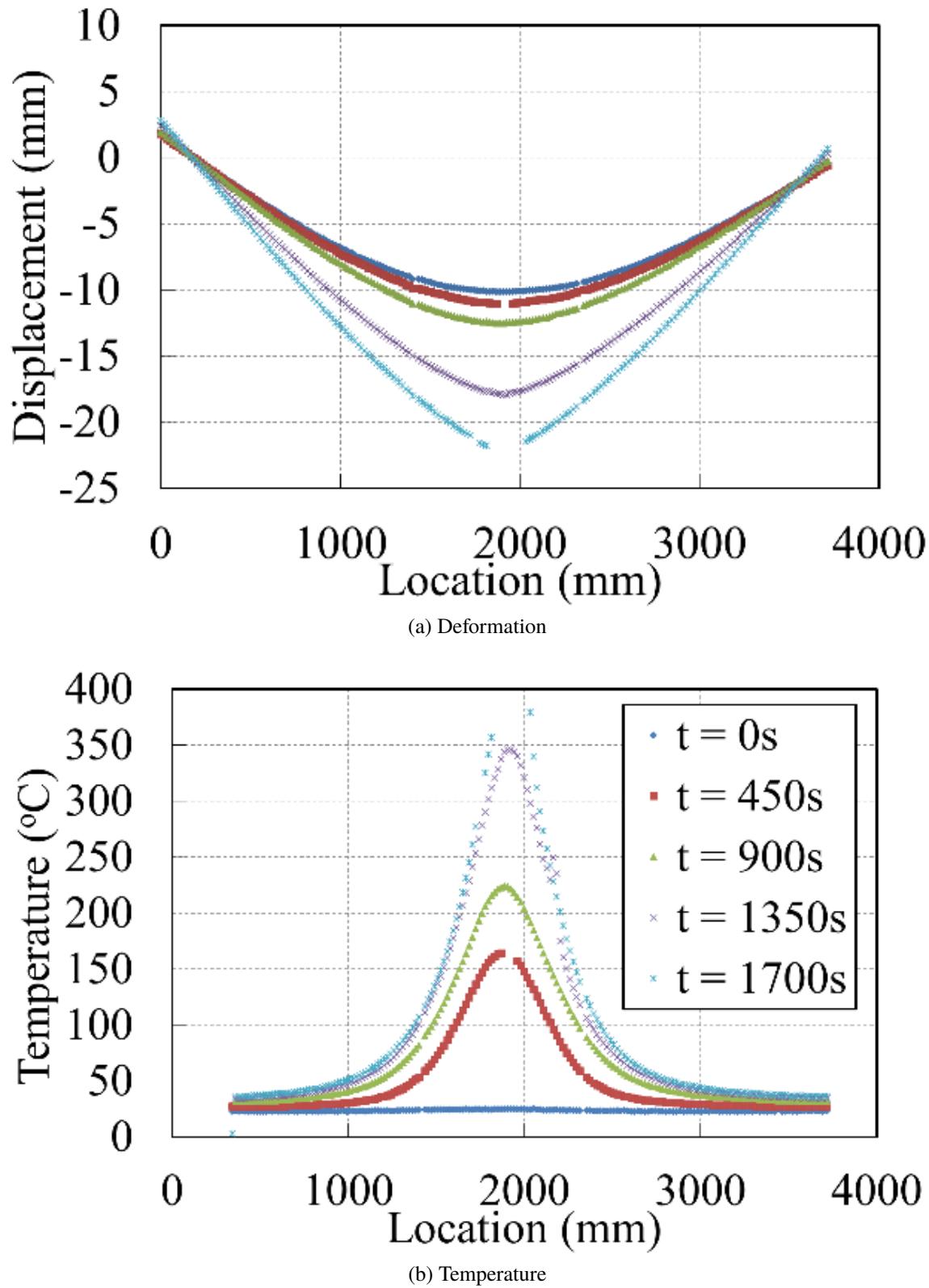
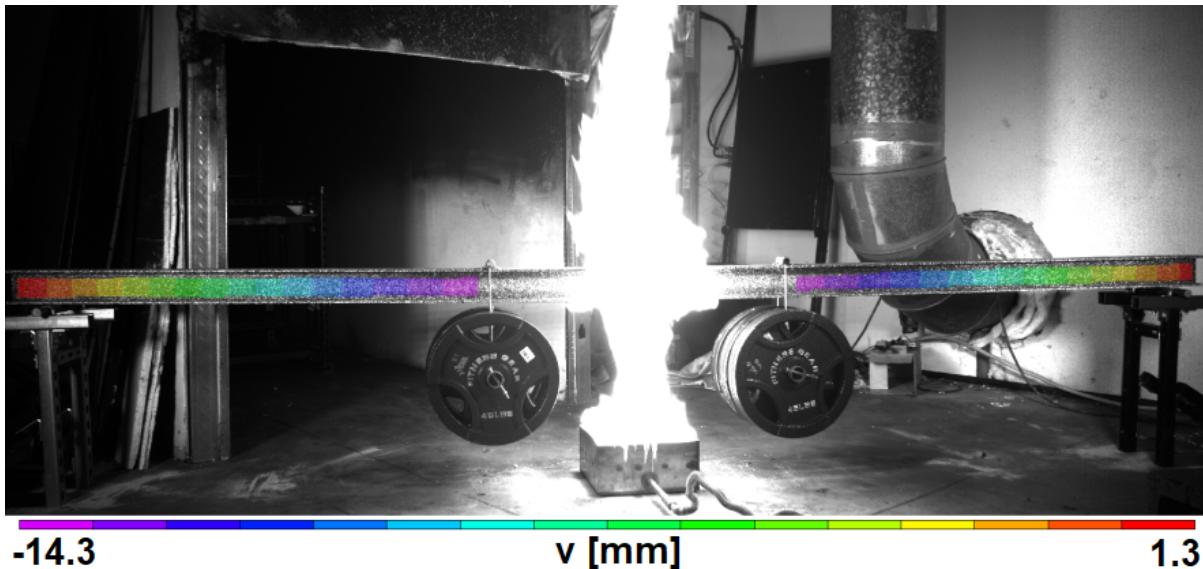
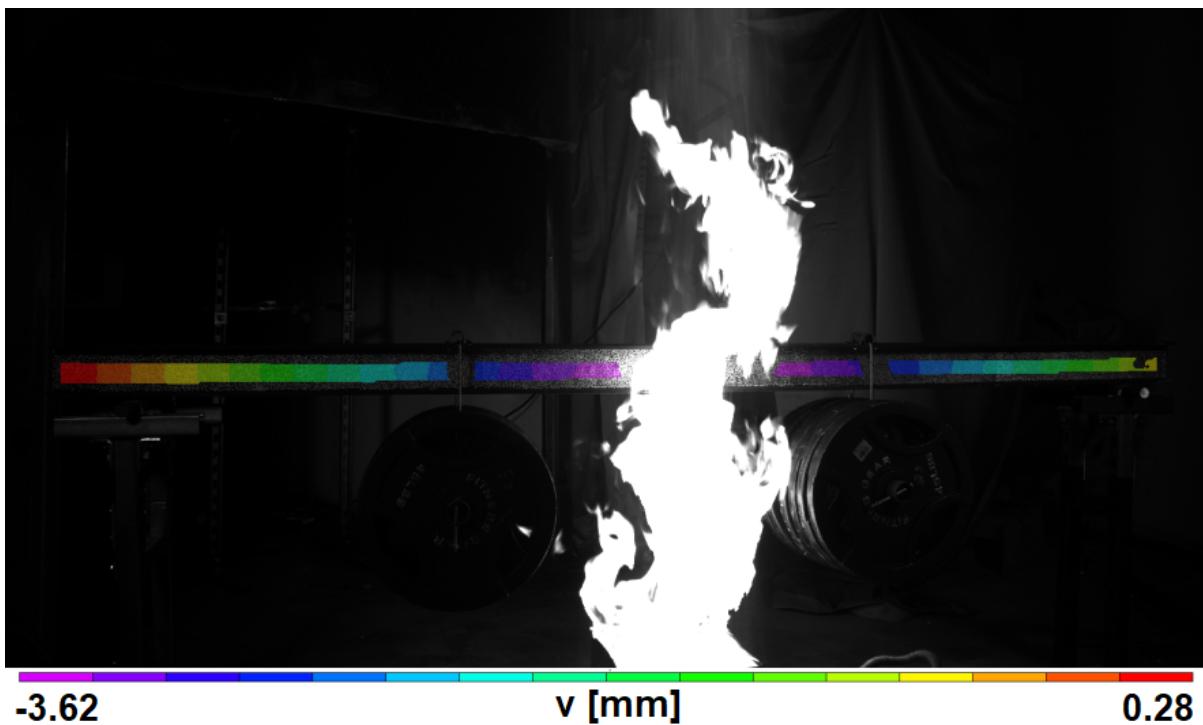


Figure B.7: Example of deformation and temperature data fusion from a structural I-beam testing for the red line slice shown in Fig. B.6a at various times throughout testing



(a) Traditional TDIC settings



(b) Modified settings based on gray-scale histogram

Figure B.8: Example of deformation measurements of structural I-beams exposed to 100 kW fire imaged with different exposure settings

Appendix C

Resolving Obstructed Views Using Multiple Thermography Digital Image Correlation Systems

C.1 Foreword

The contents of this appendix are published in the proceedings of the 14th International Conference and Exhibition on Fire Science and Engineering at the Royal Holloway College, Nr Windsor, UK, reproduced by permission [156]. The published work has eight author's including Jonathan L. Hodges, Christian M. Rippe, Nathan M. Cholewa, Patrick T. Summers, Zenka Mathys, Adrian P. Mouritz, Scott W. Case, and Brian Y. Lattimer. Jonathan post-processed the results of the experiments, conducted the analysis, and was the primary author of the paper. Christian assisted with the experiments and ran the Abaqus simulations. Nathan and Patrick ran the experiments. Zenka, Adrian, Scott and Brian advised on experimental design. Scott and Brian and provided feedback on the paper.

C.2 Abstract

Full-field measurements of the thermal and displacement response of a stiffened aluminum plate subjected to combined compression loading and one-sided heat flux were obtained using a fused thermography digital image correlation (TDIC). One limitation of this method with a single TDIC system is the inability to measure complex shapes where portions of the sample are obscured from the cameras' view. By stitching measurements from two TDIC systems together using a rigid body transform, measurements were obtained which reduced the effects of this limitation. The obtained temperature and displacement measurements were subsequently used as validation for finite element (FE) simulations.

C.3 Introduction

Full-field measurements of the thermal and displacement response are needed to quantify behavior of complex elements or structures during fire scenarios. Typically, these are obtained using an array of point measurements obtained from sensors such as thermocouples or string potentiometers. However, from a practical viewpoint, it can be difficult to attach these sensors without affecting the sample. Additionally, the selection of points for instrumentation on complex elements or structures where the failure location is uncertain *a priori* and the spatial distribution in the response varies widely can be difficult. The use of non-contact measurement methods reduces the effects of instrumentation on the sample and has the potential to allow full-field measurements on a surface.

One recently developed non-contact method for simultaneously obtaining full-field thermal and displacement responses is thermography digital image correlation (TDIC) [154]. The TDIC method utilizes a pair of stereoscopic charge-coupled device (CCD) digital cameras and one infrared (IR) camera normal to a surface to measure full-field three-dimensional deformation and temperature.

Through the coupled calibration procedure, the temperature and deflection measurements are mapped to the same coordinate system, which has the benefit of tracking temperatures with the deforming structure [154]. To date TDIC has been used for high temperature composite testing [154, 168], open fire testing of structural elements [155], and burn-through testing of marine grade aluminum structural plates [162].

TDIC measurements in previous studies have been limited to a single surface due to limitations on camera field of view and obstruction of the sample. However, mechanically tested elements and structures are three dimensional in nature, and often it is desirable to obtain measurements on multiple surfaces. One way to bypass this limitation is to obtain measurements from multiple TDIC systems and develop a coordinate transform between the systems to map the measurements to the same coordinate system. This paper presents results from using two separate TDIC imaging systems to make measurements on a plate with T-stiffeners exposed to a constant heat flux while being subjected to mechanical load. A calibration technique is presented that allows for the results from the two systems to be stitched together, forming a spatial and time synchronized result on multiple surfaces. The measurements are compared with computational results.

C.4 Experimental Methods

Two separate TDIC systems were used in this testing. Each system consisted of two CCD cameras and one IR camera. The IR camera in both systems was a FLIR A655sc (16-bit, 640 x 480 pixels, up to 100 Hz). The CCD cameras in the primary TDIC system were Allied Vision Technologies Prosilica GE 4000 CCD cameras (12-bit, 4008 x 2672 pixels, up to 10 Hz). Nikon AF lenses with a focal length of 56 mm were used with an f-stop of 11. The CCD cameras in the secondary TDIC system were Allied Vision Technologies Prosilica GX 1660 CCD cameras (14-bit, 1600 x 1200 pixels, up to 66 Hz). Tokina lenses with a focal length of 100 mm were used with an f-stop of 11.

C.4.1 TDIC Calibration

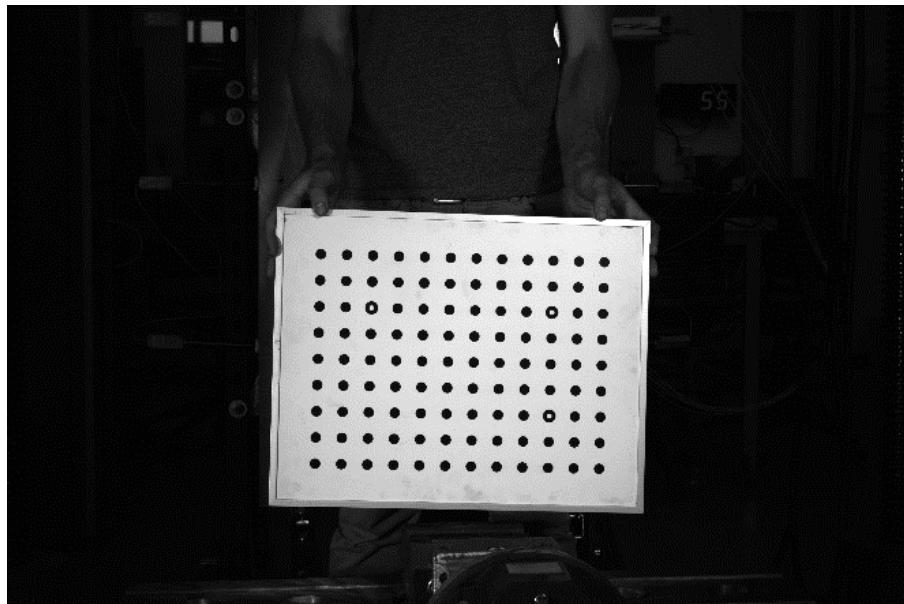
TDIC systems are calibrated using a two-dimensional array of dots with a known separation that are imaged in several orientations. In this study, the CCD cameras were calibrated using traditional dot grids while the IR camera was calibrated using anodized aluminum dot grids developed in this research that rely on emissivity differences to produce measured thermal contrast. All calibration images were processed in the commercial package Vic-3D [167].

The CCD cameras were calibrated by simultaneously capturing stereo pair images of the calibration grid at various locations and orientations within the field of view of the cameras. An example stereo pair of calibration images is shown in Fig. C.1. A set of 20-30 of these stereo pairs was used to generate the individual camera intrinsic and stereo calibration parameters. The intrinsic parameters of the IR camera were calibrated using an anodized aluminum grid with etched dots. A sample image of the anodized aluminum grid viewed from the IR camera is shown in Fig. C.2c.

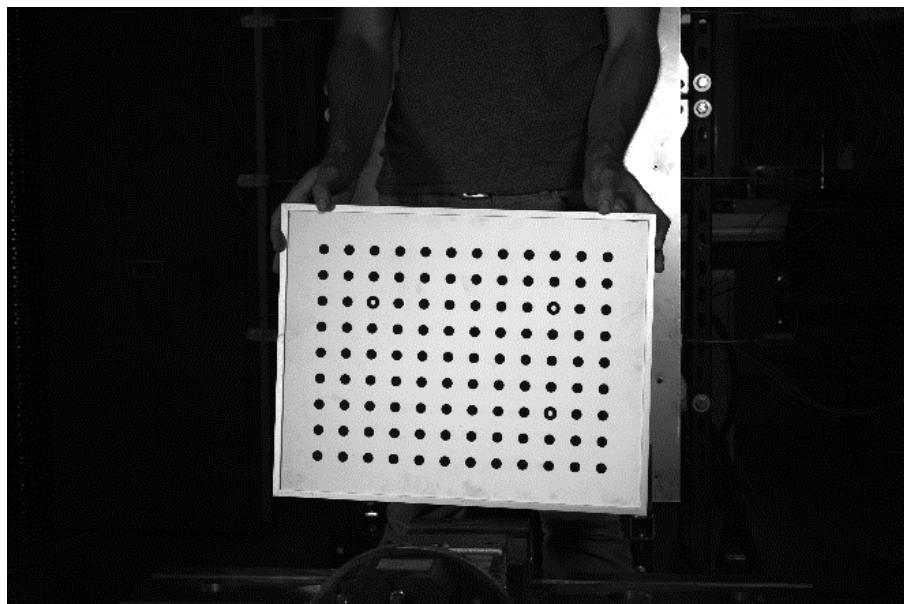
Mapping of the CCD and IR cameras to the same coordinate system was performed by simultaneously capturing the anodized aluminum grid with all cameras and calculating a rigid body transform between each. Simultaneous capture was enabled in the calibration and testing procedures through a hardware trigger. A sample set of simultaneous capture from each camera from the primary TDIC system of the anodized aluminum grid is shown in Fig. C.2. This same process was performed independently for each TDIC system.

C.4.2 Test Articles

Two stitched TDIC systems were implemented to measure the thermo-structural response of extruded T-stiffened AA6061 plates. The dimensions of the plate were 0.6 m x 0.75 m x 0.00635 m, with the two stiffeners mirrored about the center-line of the plate with a separation of 0.4 m. The

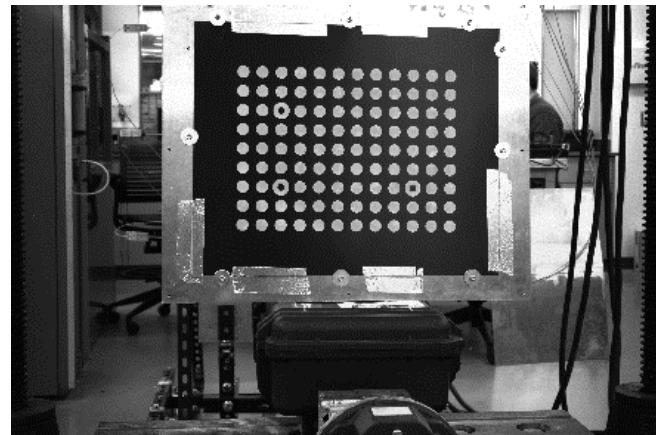


(a) Left camera

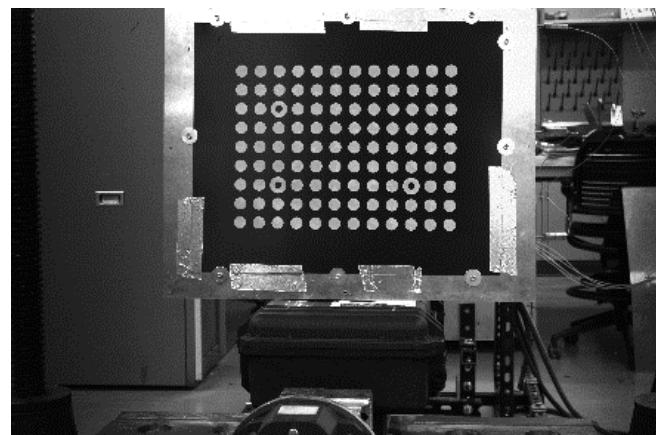


(b) Right camera

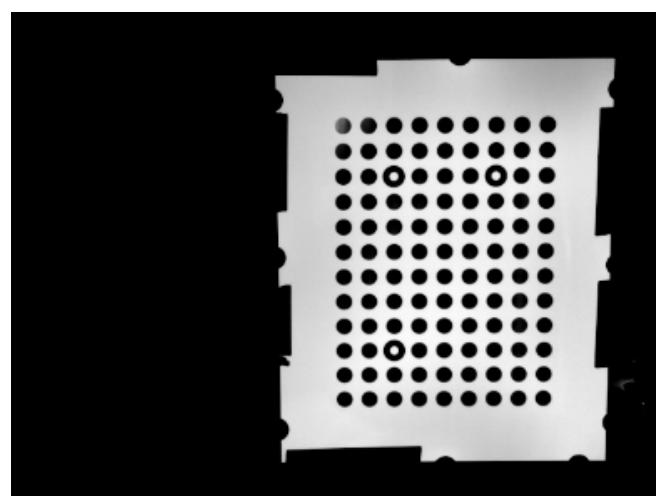
Figure C.1: Sample stereo pair of calibration images from the primary TDIC system



(a) Left CCD camera



(b) Right CCD camera



(c) IR camera

Figure C.2: Sample simultaneous captured image from each camera in a TDIC system used for rigid body transform

primary TDIC system was used to measure the thermo-structural response of the plate. Because the stiffener webs were obscured from the primary TDIC system by the stiffener flanges, the secondary TDIC system was used to measure the response of the left stiffener web and flange. Test article geometry and measurement surface locations are shown in Fig. C.3.

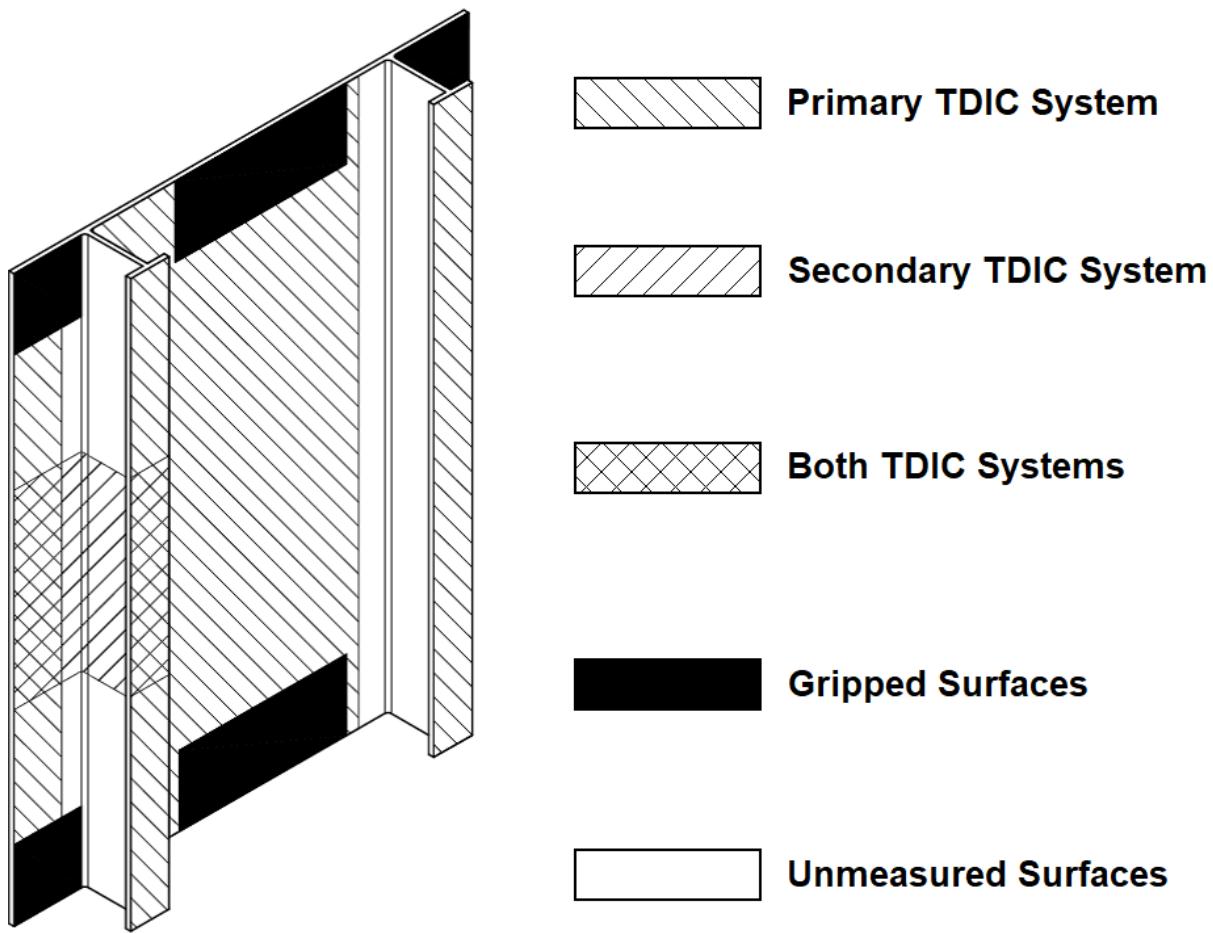


Figure C.3: Measurement surfaces on T-stiffened aluminum plate for two TDIC systems

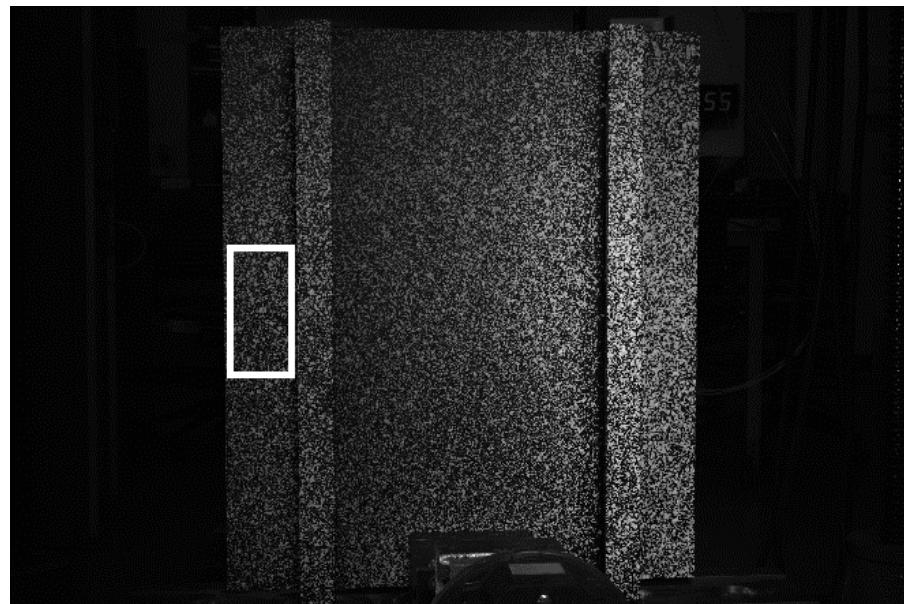
All surfaces of the sample were painted using Rust-Oleum® Specialty High Heat matte black and white paints for the background and speckles respectively. The background and speckles were applied using an industrial spray gun. The optical thickness for the IR camera, $11\mu m$, was achieved using the method presented in Cholewa et. al.[154]. An image from one of the CCD cameras showing the isotropic, non-repeating speckle pattern on the sample is shown in Fig. C.4.



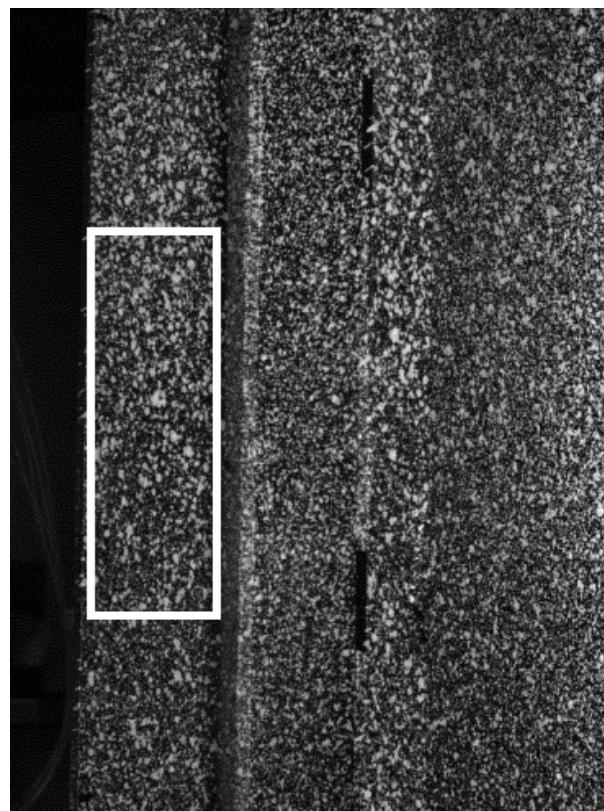
Figure C.4: Painted surface for TDIC measurements

C.4.3 Data Stitching Process

When visual access to all surfaces cannot be obtained by a single imaging system, full-field results can still be obtained by stitching data from multiple TDIC systems into a single coordinate system. Once each TDIC system has been independently calibrated, a rigid body transformation relationship between the two systems is generated by simultaneously capturing a single speckled measurement surface with both systems. This transformation relationship is subsequently used to spatially synchronize data from the secondary TDIC system to data from the primary TDIC system. Temporal synchronization is achieved by initiating TDIC measurements via a hardwired voltage trigger to both systems. (This trigger is the same used in the initial calibration to sync the FLIR and CCD images.) A sample pair of images showing the surface used for the coordinate transform is shown in Fig. C.5.



(a) Primary TDIC system



(b) Secondary TDIC system

Figure C.5: Same measurement surface viewed from one CCD camera in each TDIC system. The boxed region was used to create the coordinate system transform.

C.4.4 Testing Conditions

The test articles were mechanically loaded with 222kN of compressive force. Once the force was applied, articles were exposed to peak surface heat fluxes of 35 and 40 kW/m^2 using radiant heater panels to simulate fire exposure. Testing continued until failure.

C.5 Numerical Methods

Finite element (FE) simulations of the experiments were constructed using the commercial software package Abaqus 6.12-EF. The response of the test articles was simulated using sequentially coupled thermal and mechanical analysis. The stiffened plate, heater panel face, and representative grip geometries were modeled in the thermal analysis as seen in Fig.C.6a. The stiffened plate mesh was seeded at 25 mm in the plane of the plate and a single element was used through the thickness. The heater face meshed was seeded at 51 mm and the representative grip geometry mesh was seeded at 25 mm. All the geometries in the thermal model were meshed using quadratic hexahedral heat transfer elements (DC3D20). A cavity radiation model between the heater and plate was used to calculate the incident radiative heat flux onto the plate. Convection losses from both the exposed and unexposed surface of the plate were calculated using natural convection correlations for a vertical surface by Churchill and Chu [169]. A surface temperature of 300°C was used to calculate a convection heat transfer coefficient of $9 \text{ W/m}^2\text{K}$. The surface emissivity of the plate was taken as a constant 0.95 which is within 5% of values measured by Cholewa *et al.* [154] for temperatures observed in the model. Temperature dependent thermal conductivity and specific heat capacity were used in the thermal model.

Mechanical response was simulated using a nonlinear static analysis. Only the stiffened plate was modeled in the mechanical simulation. Figure C.6b contains the meshed geometry used in

the mechanical analysis. The plate mesh was seeded at 9.5 mm in the plane of the plate. Two elements were used through the thickness of the plate providing an average element aspect ratio of 3.2. Fully integrated quadratic hexahedral stress-displacement elements (C3D20) were used in the mechanical analysis. Elements in contact with the lower grip region were fixed from translation in all three directions. Elements in contact with the upper grip region were fixed from translation in the horizontal directions and constrained to move equally in the vertical direction to simulate the motion of the upper cross-head. Temperature-dependent mechanical properties including thermal softening, yield strength reductions, thermal expansion, and creep were used in the mechanical models.

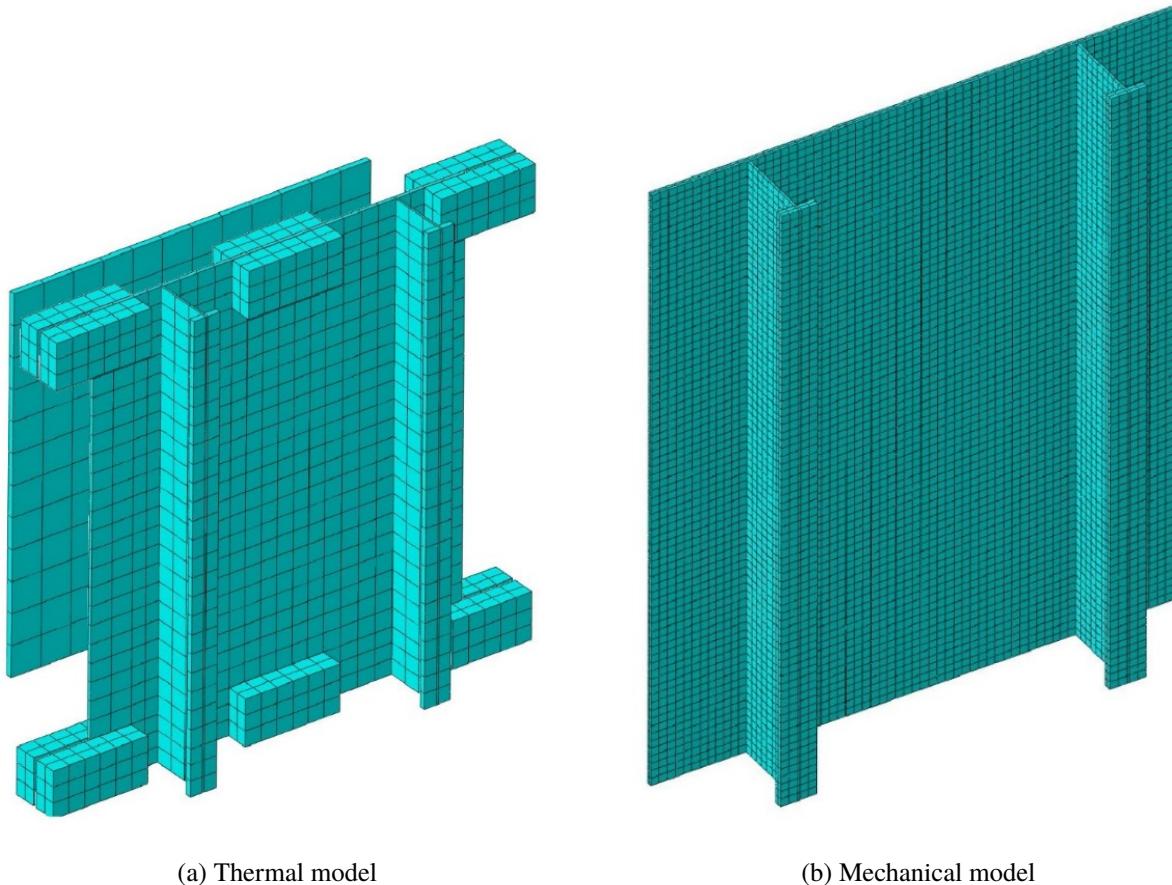


Figure C.6: Representative geometries and grid resolution in numerical modeling

C.6 Results

Two test articles were tested under the conditions described above. The sample tested with a peak surface heat flux of 35 kW/m^2 failed after 11.5 minutes, whereas the sample tested at 40 kW/m^2 failed after 8.3 minutes. Each sample collapsed after reaching a peak temperature of approximately 385°C . Full-field measurements of temperature, displacement, and strain on the base plate as well as the stiffener web and flange for two test articles were compared with numerical results. Sample contour plots of temperature near failure are shown in Fig. C.7. Figure C.7a and Figure C.7b show contour plots of temperature for the primary TDIC system, and the secondary TDIC system, respectively. This shows the limitations of measuring the sample with a single TDIC system, wherein either the plate or the web can be measured but not both simultaneously. Figure C.7c contains a contour plot after the stitching process, which shows it is possible to show experimental results in a unified image after stitching. From these unified results, a better understanding of the spatial distributions of measurement variables can be obtained. Additionally, quantitative single point data can be extracted from stitched spatial profiles of either TDIC system and compared with numerical results, shown for the same time step in Fig. C.7d.

A series of contour plots of temperature overlaid on a deformed test article exposed to a peak heat flux of 35 kW/m^2 are shown from various times during testing in Fig. C.8. Significant failure starts at $t = 600$ seconds, as shown in Fig. C.8f. Looking at the flange of the left stiffener, some observations can be made of the results after stitching. There is a difference in the temperature measurements between the two systems, as can be seen in Fig. C.8c and Fig. C.8d. At point P0 marked in Fig. C.9, the root mean square difference in the temperature between the systems is of $12\text{--}20^\circ\text{C}$ for the duration of testing. In thermography it is ideal to place the camera normal to the surface of interest. For the primary system, this surface was the plate and top of the stiffeners. For the secondary system, it was the web of the stiffener. Thus, the difference on the plate is due to the

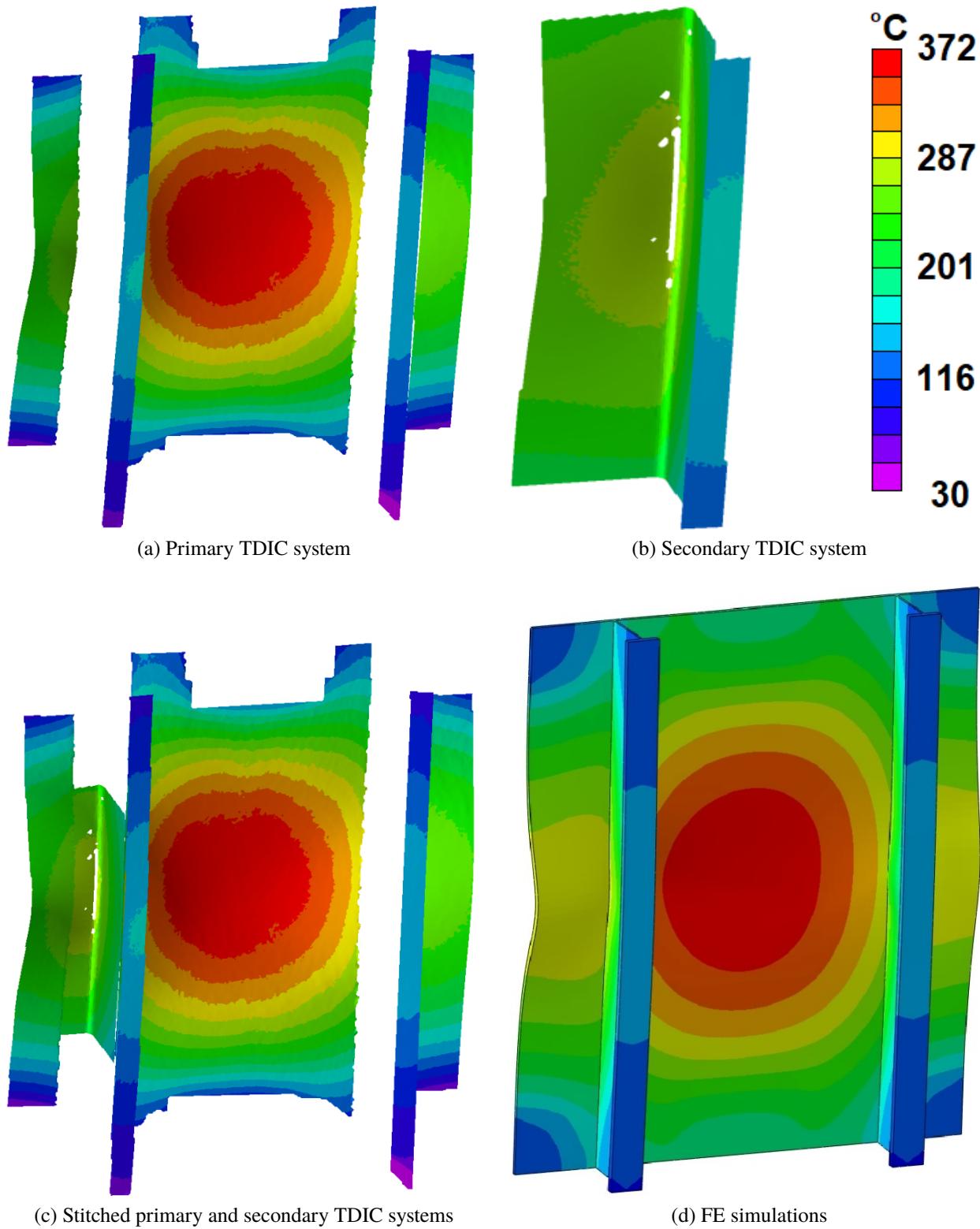


Figure C.7: Full-field temperature mapped to displaced geometry

2 separate IR cameras imaging the surface from different angles. The root mean square difference of deformation measured between the two systems at P0 is 0.24 mm.

To better compare the numerical and experimental results, measurements at the two line slices shown in Fig. C.9 were examined for a test article exposed to a peak heat flux of 35 kW/m^2 . Measurements of deflection and temperature from the experiment for the horizontal center-line, L0, are presented in Fig. C.10a and Fig. C.10b, and the measurements for the vertical line slice, L1, are shown in Fig. C.10c and Fig. C.10d. The gaps in deflection measurements in Fig. C.10a show sections of the article which were obscured from measurement by the primary TDIC system. By stitching the 2 systems together, the gap in between X=0-180 mm is filled by deflection measurements from the secondary TDIC system. Figure C.10b and Figure C.10d show the temperature across the sample reached steady state approximately 90 seconds before failure. The maximum temperature across the plate was the same in both center-lines at 385°C . The total range of observed temperatures between the stiffeners for the horizontal slice was $290^\circ\text{C}-385^\circ\text{C}$; whereas the range for the vertical slice was $160^\circ\text{C}-383^\circ\text{C}$. Figure C.10a and Figure C.10c show the sample began to deflect approximately 120 seconds after the steady state temperature was reached. The stiffeners buckled 660 seconds after the start of testing, as seen by the lines at X=100,500 mm in Fig. C.10a beginning to rotate and L1 starting to distort in the opposite direction at Y=100,450 mm in Fig. C.10c. The range of observed out of plane deflection in testing was $\pm 30\text{ mm}$.

Similarly, predictions of deflection and temperature from the numerical simulations for L0 and L1 are presented in Fig. C.11a- Fig. C.11b and Fig. C.11c- Fig. C.11d respectively. Comparing the results in Fig. C.10 and Fig. C.11, the observed trends of deflection and temperature are similar between the experimental and numerical results. Maximum predicted and observed temperatures at the center of the base plate were 380°C and 385°C , respectively. The temperature predictions were generally lower than observed values for the duration of testing due to idealizations of the heater panels and boundary conditions. As a result, the overall failure time predictions were generally

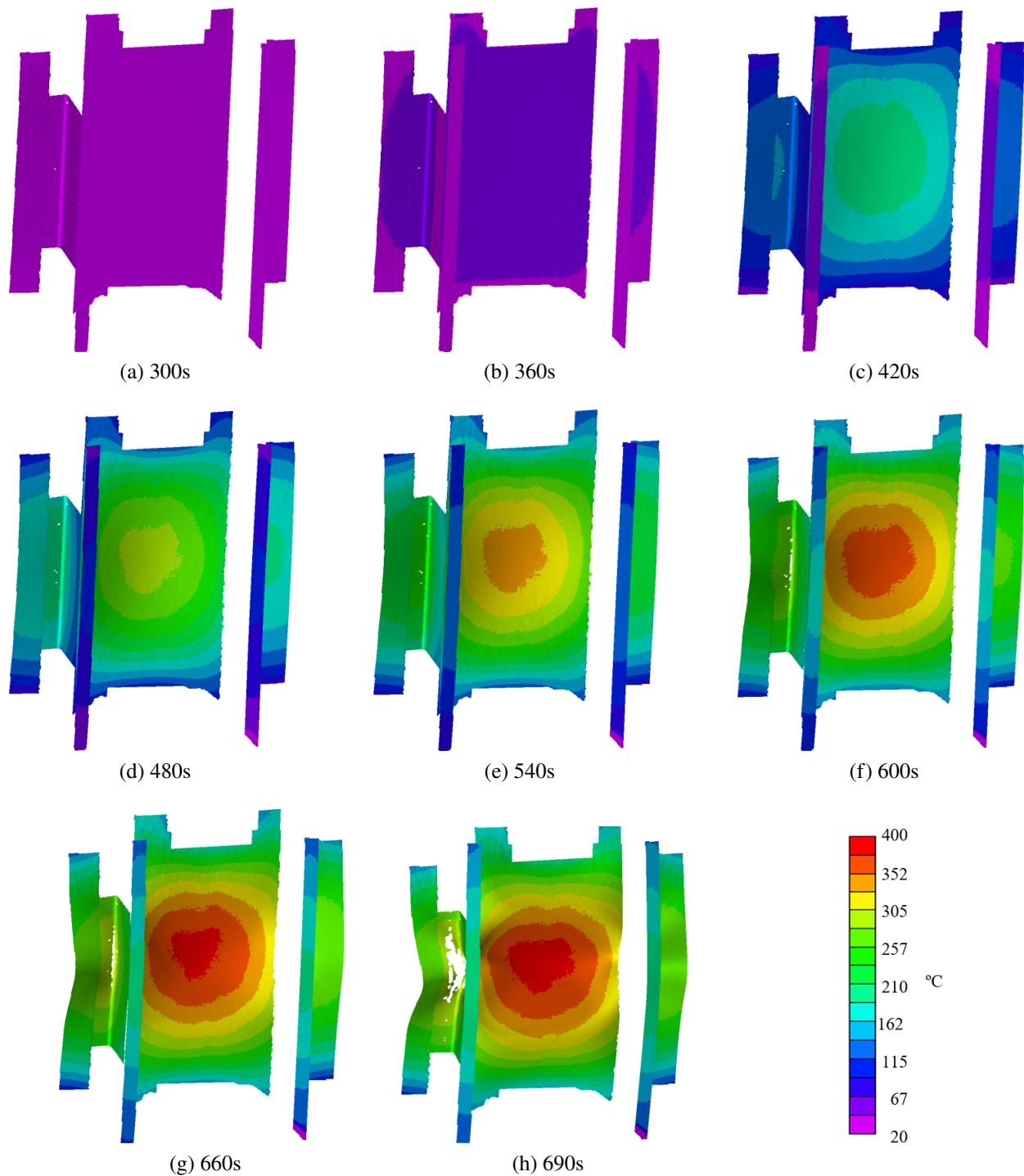


Figure C.8: Full-field temperature mapped to displaced geometry for a test article exposed to a peak heat flux of 35 kW/m^2 at various times throughout testing

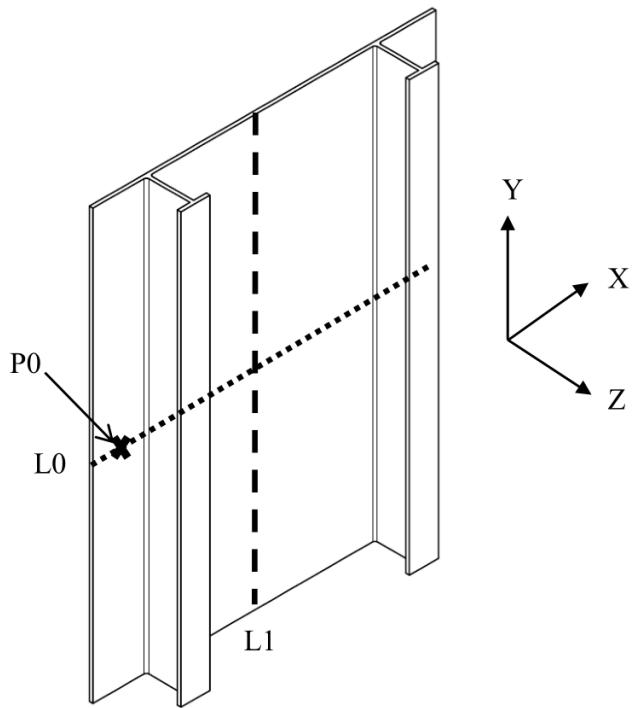


Figure C.9: Location of line slices for comparisons shown in Fig. C.10 and Fig. C.11

50% greater than experimental observation.

C.7 Conclusion

Full-field measurements of the thermal and displacement response of a stiffened aluminum plate were obtained using a fused thermography digital image correlation (TDIC). Using two systems it was possible to obtain measurements of the web of the stiffener which was obscured from the view of the primary TDIC system. Transforming the measurements of both systems to the same coordinate system, the measurements from both systems were combined into a single resulting data set, which allows for a straightforward extraction of individual points of interest anywhere on the sample. The utility of these full field measurements was shown by examining the temporal trends of deflection and temperature across the horizontal and vertical trend lines. Numerical results were

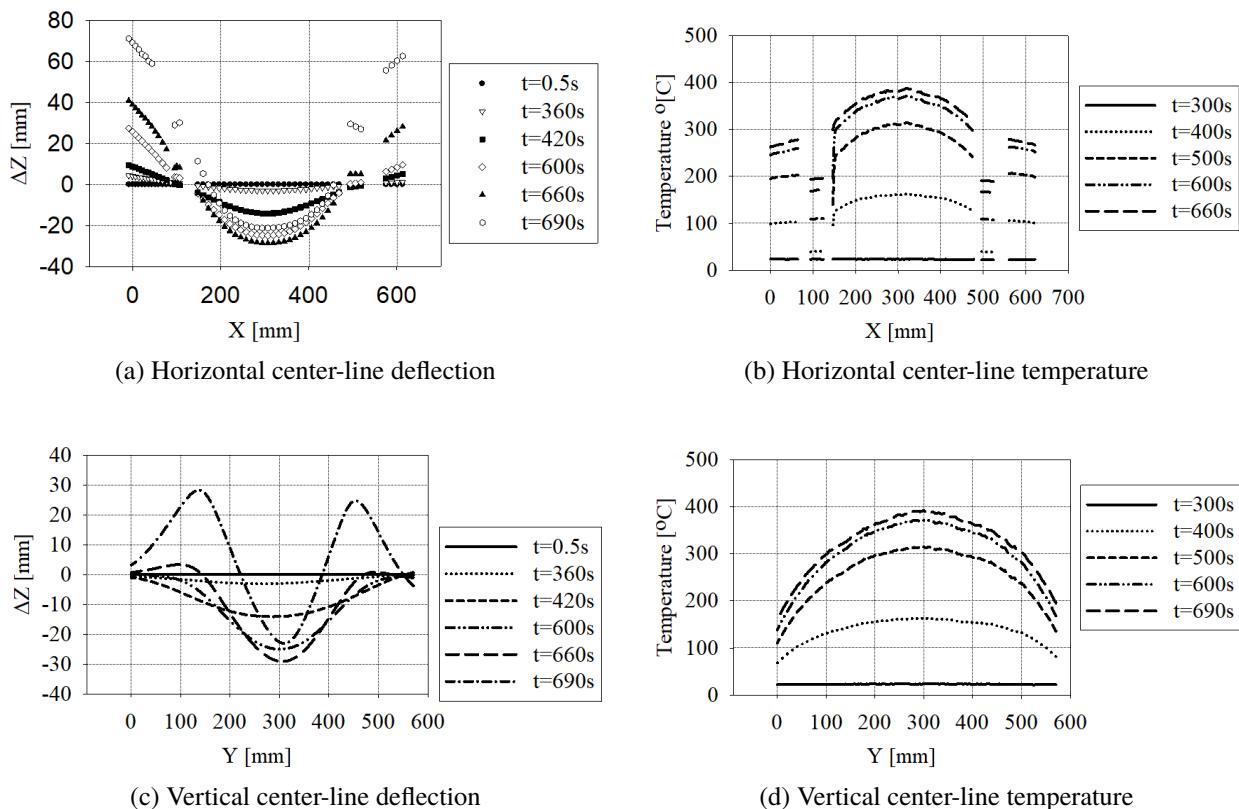


Figure C.10: Experimental measurements of deflection and temperature of a test article exposed to a peak heat flux of 35 kW/m^2 at various times throughout testing

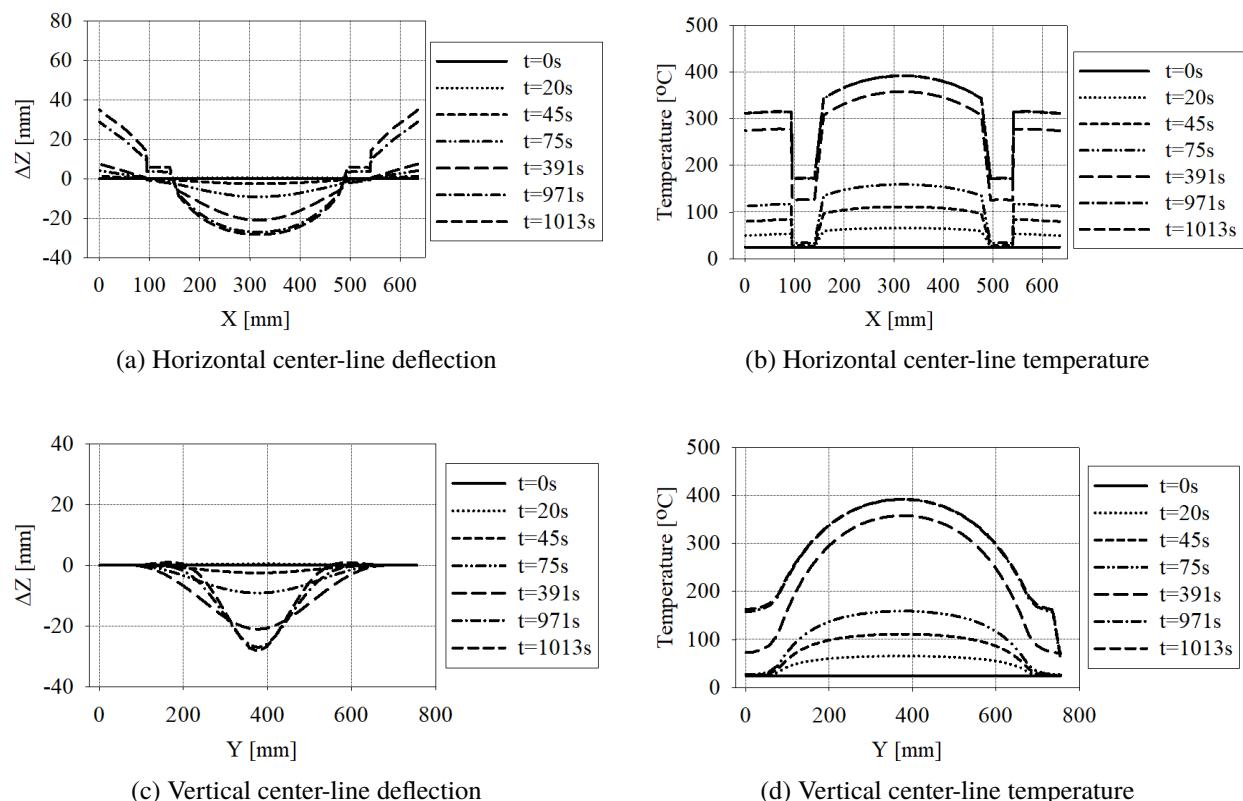


Figure C.11: Numerical measurements of deflection and temperature of a test article exposed to a peak heat flux of 35 kW/m^2 at various times throughout testing

verified by comparing the deflection and temperature at the maximum temperature region from the experimental results.

C.8 Acknowledgements

The authors appreciate the sponsorship related to this work from N00014-13-1-0590 scientific monitor Dr Tom Fu. The work in this paper was supported NEEC subcontract 2002542411-UT.

Appendix D

Predicting the Structural Response of a Compartment Fire using Full-field Heat Transfer Measurements

D.1 Foreword

The contents of this appendix are published in Fire Safety Journal, published by Elsevier, reproduced by permission [153]. The published work has four authors including Jonathan L. Hodges, Christian M. Rippe, Scott W. Case, and Brian Y. Lattimer. Jonathan ran the experiments, and performed the inverse heat transfer analysis. Jonathan and Brian contributed equally to the writing of the paper. Christian ran the Abaqus simulations. Scott and Brian advised on experimental design. Scott provided feedback on the paper.

D.2 Abstract

Inverse heat transfer analysis (IHT) was used to measure the full-field heat fluxes on a small scale (0.9m x 0.9m x 0.9m) stainless steel SS304 compartment exposed to a 100kW diffusion flame. The measured heat fluxes were then used in a thermo-mechanical finite element model in Abaqus to

predict the response of an aluminum 6061-T6 compartment to the same exposure. Coupled measurements of deflection and temperature using Thermographic Digital Image Correlation (TDIC) were obtained of an aluminum compartment tested until collapse. Two convective heat transfer coefficients, $h = 35 \text{ W/m}^2 - \text{K}$ and $h = \text{W/m}^2 - \text{K}$ were examined for the thermal model using the experimentally measured heat fluxes. Predictions of the thermal and structural response of the same compartment were generated by coupling Fire Dynamics Simulator (FDS) and Abaqus using the two values for h , $h = 35 \text{ W/m}^2 - \text{K}$ and h from convection correlations. Predictions of deflection and temperature using heat fluxes from IHT and FDS with $h = 35 \text{ W/m}^2 - \text{K}$ agreed with experimental measurements along the back wall. The temperature predictions from the IHT-Abaqus model were independent of h , whereas the temperature predictions from the FDS-Abaqus model were dependent on h .

D.3 Introduction

Fires produce temporal and spatial varying exposures onto structures, which affects how the structure deforms and fails during a fire. Computational models are needed to quantify the temporal and spatial fire exposures for input into finite element models to accurately predict the thermal and structural response of the enclosure. Previous studies that have measured the thermo-structural response have focused on measuring the heat transfer, temperature, and structural response at point locations.

Researchers have measured cold surface heat flux measurements using a one-dimensional inverse heat transfer on temperature measurements of steel plates [170, 171, 172, 173]. This method was used to measure the standard heat flux for fires in an ISO 9705 fire test room by extending the one-dimensional inverse heat transfer to two-dimensions [171]. More recently, a technique has been developed to predict the spatial and temporal variation to a two dimensional surface based on

IR thermography on a metal plate [174]. The method can be used to determine the net heat flux or standard heat flux (defined as the heat flux to a surface at the standard temperature of 298 K).

Some efforts have been performed to predict the thermal and structural response of elements using Fire Dynamics Simulator (FDS) to predict the thermal exposure and finite element models to predict the temperature and structural response [175, 176]. In these efforts, the adiabatic surface temperature was used to define the thermal boundary condition. The adiabatic surface temperature and the standard heat flux provide the same boundary condition information to the thermal model, just expressed differently [177]. In both cases, the heat transfer coefficient used in the FDS simulation must be used in the FE thermal model to ensure accurate results. Recently, thermography digital image correlation (TDIC) has been used to measure both the temperature and deflections in the same framework so that temperatures and deflections are tracked even in cases where excessive deflections occur [154].

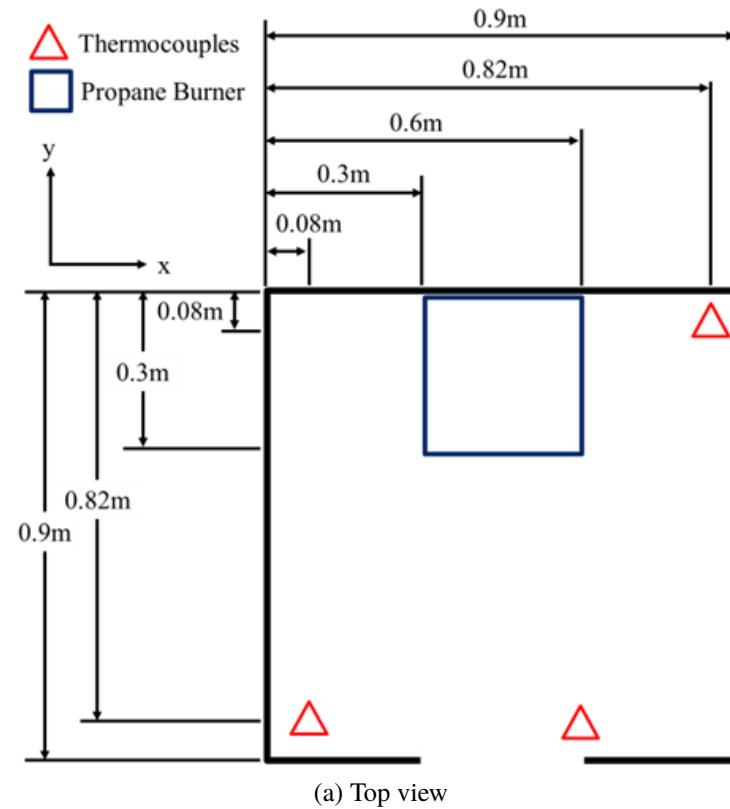
The focus of this paper is to conduct an experiment that accurately measures the response of three-dimensional collapse of a compartment containing a fire and predict the response of the compartment collapse. For this, methods were developed to spatially and temporally measure the heat transfer from the compartment fire to the boundaries. These results were input into a finite element model to predict the thermal and structural response. In addition, the fire dynamics were predicted using FDS and the predicted heat transfer were input into the finite element model to predict the thermal and structural response. The thermal and structural responses from the two approaches were compared and the sensitivity of the thermal boundary condition details (primarily the heat transfer coefficient) were evaluated.

D.4 Experimental Methods

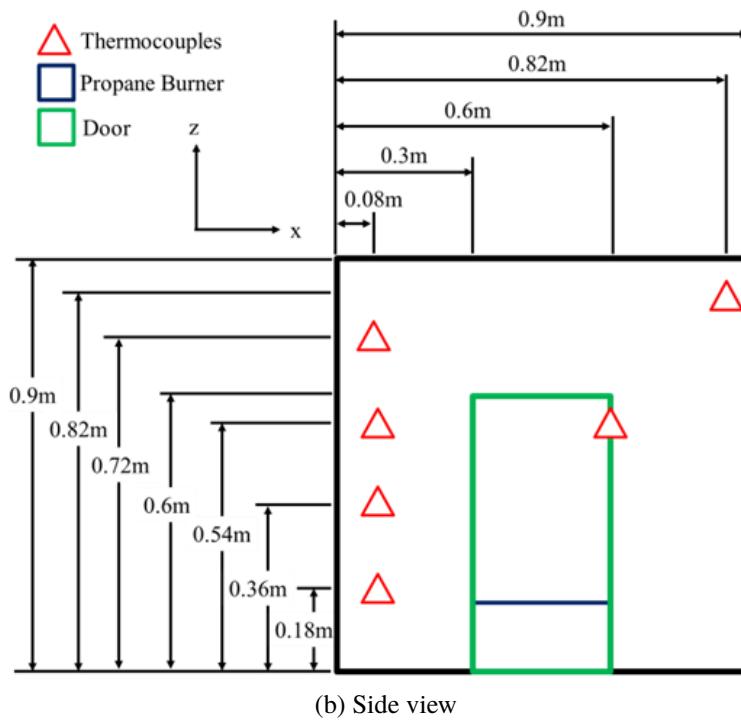
Compartment fires tests were performed using 0.9 m x 0.9 m x 0.9 m compartments with a 0.3 m x 0.6 m door on one side. A 0.3 m x 0.3 m propane sand burner was placed at the center of the wall opposite the door. All testing was performed with a 100 kW fire. Each compartment was made from 3 separate pieces of metal which were attached with screws. Silicone sealant was used to make the compartment air tight at the bolt holes and each air gap. The compartments were placed on a slab of drywall and sealed along the bottom to remove air leaks. The first compartment was built from SS304 stainless steel (0.8mm thickness) and was used to quantify the heat flux from the fire exposure. The second compartment was built from 6061-T6 aluminum (1.6mm thickness) and was used for structural collapse testing. The details of each experiment and data analysis are described in the following sections. A schematic of the experimental setup for the heat flux mapping experiments is shown in Fig. D.1.

D.4.1 Heat Flux Mapping Experiments

Full-field temperature measurements were obtained on each of the exterior surfaces of the stainless steel compartment through infrared (IR) thermography. All interior and exterior surfaces were painted with Rust-Oleum® Specialty High Heat matte black paint which was previously shown to have an emissivity of 0.95 by [154]. The IR cameras used in this work were FLIR SC655 (640x480 pixels, up to 50 Hz, 16-bit, 7.5 - 14.0 μ m). Two cameras were placed to provide an isometric view of the compartment, focusing on opposite corners of the top surface as shown in Fig. D.2a and Fig. D.2b. The last camera was placed orthogonal to the back wall where the sand burner was located, as shown in Fig. D.2c. Each image in Fig. D.2 has been scaled to the minimum and maximum temperatures measured by the camera within that frame. No mechanical load was applied



(a) Top view



(b) Side view

Figure D.1: Experimental setup for the heat flux mapping experiments

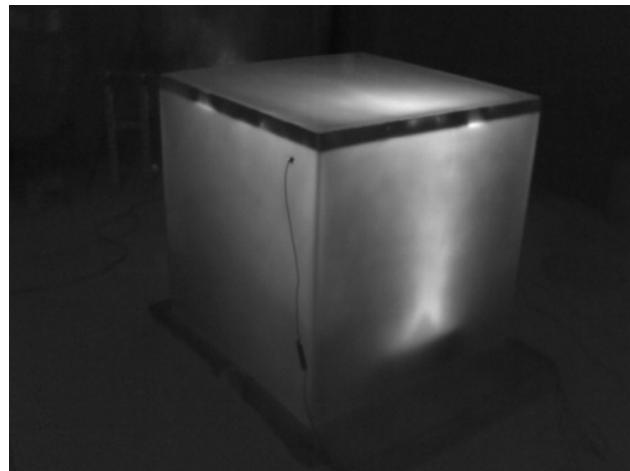
to the compartment during the heat flux mapping experiments. Gas temperature was measured at the left corner of the compartment by the door at elevations of 0.18 m, 0.36 m, 0.54 m and 0.72 m using 24 gauge ceramic braid, bare bead Type K thermocouples.

The heat release rate of the fire was set to 20kW for ten seconds and then increased to 100kW for the remaining duration of the experiment until the surface temperatures reached steady-state (approximately 50 seconds additional heating, for a total exposure of 60 seconds). The fire was then turned off and the compartment was allowed to cool.

Spatial and temporal variation in the heat transfer to the boundaries of the compartment were determined using inverse heat transfer (IHT) analysis on the thermography images using the method described by Rippe *et al.* [174]. Previously this was applied to a single two-dimensional surface. In this work, an isometric view of a three-dimensional object was imaged using the IR cameras (Fig. D.2). To determine the heat transfer to each surface, the three-dimensional images were rectified to create two-dimensional images of each surface with time. In the analysis, it was assumed that there was no conductive heat transfer along the edges between surfaces of the compartment; this assumption is reasonable due to the relatively small value of the plate thickness.

One unknown in the IHT method necessary to obtain the standard heat flux is the convective heat transfer coefficient, h . Prior published studies have presented temperature dependent relations for h for compartment fires. Correlations of h by Dembsky *et al.* [178], Veloo and Quintiere [179], and a modified version of the Emmons correlation presented in Sincaglia and Barnett [180] are provided in Fig. D.3. Since the gas temperatures in the compartment fire tests ranged from 300-800K, it was decided to complete two trials, one with the heat transfer coefficient on the interior and exterior was $h_{int} = h_{ext} = 10 \text{ W/m}^2 - K$ and one with $h_{ext} = 10 \text{ W/m}^2 - K$ and $h_{int} = 35 \text{ W/m}^2 - K$ where the subscripts correspond to the interior and exterior h , respectively.

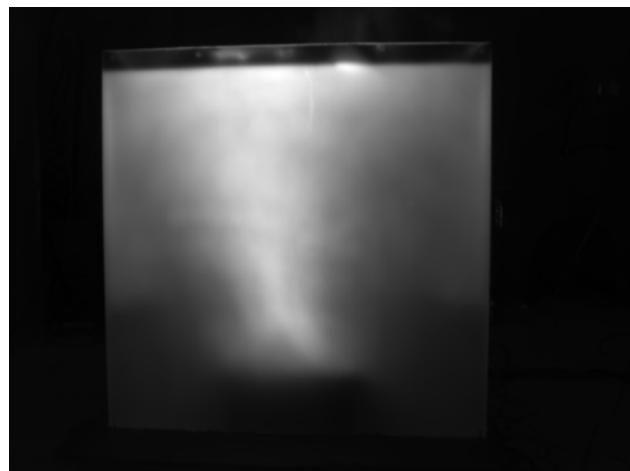
The heat fluxes obtained from the experimental measurements using the stainless steel compartment



(a) Isometric view of back wall



(b) Isometric view of front wall



(c) Isometric view of front wall

Figure D.2: Compartment heat flux testing IR camera perspectives

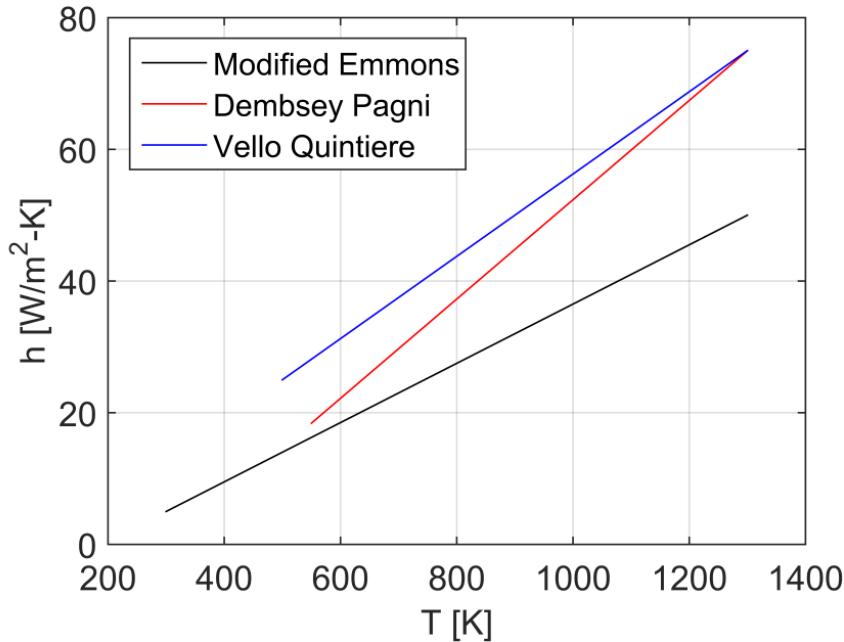


Figure D.3: Experimental setup for the heat flux mapping experiments

were imported into Abaqus to predict the thermal and structural response of the aluminum compartment. Since the surfaces in each compartment were painted with the same emissivity paint, the heat fluxes obtained from the stainless steel compartment can be directly applied to the aluminum compartment. As discussed below, the spatial and temporal measured standard heat fluxes, q_0 , (i.e., heat flux to a surface at the standard temperature of 298 K) were input into a thermal model to predict the temperature of the boundaries. Based on the formulation of the standard heat flux, it will be shown that the use of a different estimate of the heat transfer coefficient does not influence the predicted boundary temperature as long as the same heat transfer coefficient is used in the thermal modeling. Note, that the IHT method measures the net heat flux from the fire into the surface and is independent of the heat transfer coefficient. The relationship between the net heat flux and the standard heat flux is given by

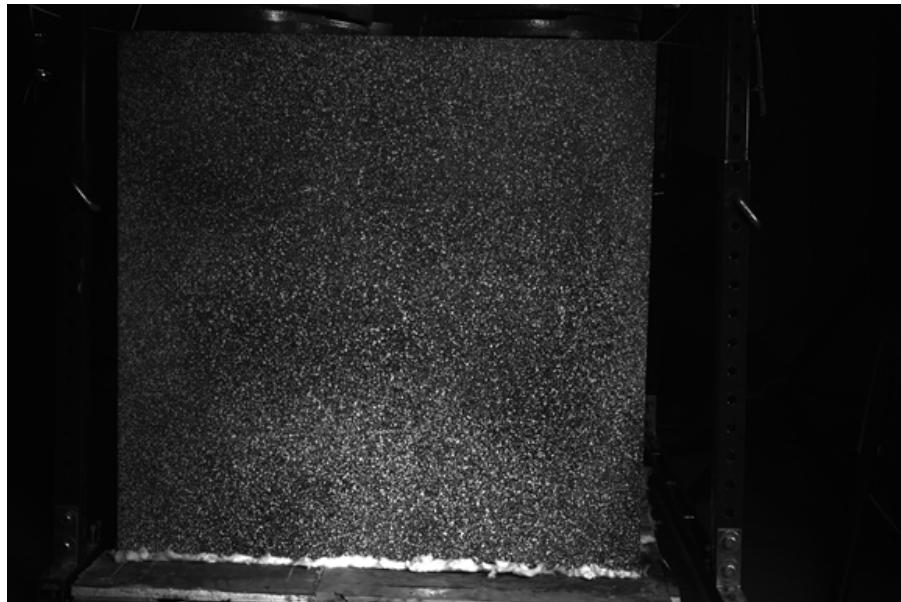
$$q''_{net} = q''_0 - h(T_s - T_0) - \epsilon\sigma(T_s^4 - T_0^4) \quad (\text{D.1})$$

where T_s is the surface temperature, T_0 is the reference temperature (298 K in this work), ϵ is the emissivity, and σ is the Stefan-Boltzmann constant.

D.4.2 Structural Collapse Experiments

Structural collapse tests were conducted using an aluminum sheet compartment. Detailed spatial and temporal temperature and deflection measurements were taken on the unexposed side of the back wall of the compartment where the fire was located. Coupled full-field measurements of deflection and temperature were obtained using Thermographic Digital Image Correlation (TDIC) [154]. The two CCD cameras used in this work were Allied Vision Technologies Prosilica GE 4000 CCD cameras (4008 x 2672 pixels, up to 10 Hz, 12-bit) with Nikon AF lenses with a focal length of 56 mm were used with an f-stop of 11. The IR camera was a FLIR SC655 (640x480 pixels, up to 50 Hz, 16-bit, 7.5 μm \times 14.0 μm). As described in Ref. [154], a special calibration grid was used to fuse the thermal measurements over the deflections measurements so that these measurements are aligned even when the wall experiences significant deflection. All interior and exterior surfaces were painted with a flat black background of Rust-Oleum® high heat paint. The exterior surfaces were then speckled with Rust-Oleum® high heat white paint which was also characterized by Cholewa *et al.* [154]. The surface preparation of the back wall of the compartment is shown in Fig. D.4a.

The aluminum compartment was mechanically loaded with 1.6 kN split evenly between the four corners on the top of the compartment as shown in Fig. D.4b. The compartment was exposed to a 20 kW fire for ten seconds followed by a 100 kW fire for the duration of the test until collapse was observed.



(a) Speckling for TDIC on back wall



(b) Mechanical loading on top of compartment

Figure D.4: Structural collapse experiment compartment setup prior to testing

D.5 Simulations

A series of simulations were conducted to predict the three-dimensional thermal and structural behavior of the aluminum compartment using the commercial finite element (FE) software Abaqus 6.12EF. The thermal and structural models were sequentially coupled, meaning that the thermal analysis was performed and then the predicted temperatures were passed to a separate structural analysis. The first series of simulations were performed using the IHT determined heat transfer as input into the thermal analysis. A second series of simulations were performed using the heat transfer determined using the computational fluid dynamics software Fire Dynamics Simulator (FDS). The details of the simulations for FDS and the Abaqus thermal and structural models are provided in the sections below.

D.5.1 FDS Model

FDS simulations were conducted to determine the heat transfer from the fire environment to the boundaries of the compartment. The fire source was generated using a plane with a constant heat release rate per unit area to represent the top of the propane sand burner. The mixing-controlled fast chemistry combustion model with default propane and air properties was used for the combustion reaction model. The computational domain consisted of the compartment interior as well as 0.72 m of space in front of the compartment door and 0.28 m around all other surfaces. The computational domain was seeded in every direction at 40 mm. Mesh convergence was checked using a model with a 20 mm mesh seed. Temperature dependent aluminum thermal properties were used to accurately capture the temperature response of the compartment surfaces. A surface emissivity of 0.95 was used on the aluminum because all of the aluminum surfaces were painted with high emissivity paint. The heat release rate (HRR) in the simulation included a 5 second ramp to 20kW and a 15 second ramp to 100kW.

In order to accurately predict the three-dimensional heat transfer in Abaqus, the standard heat flux (i.e., heat flux to a surface at the standard temperature of 298K) and the heat transfer coefficient used in FDS must be passed to Abaqus. Simulations were performed using two different estimates of the heat transfer coefficient for the interior and exterior of the compartment. One simulation was performed using the built-in FDS natural convection correlations (on the order of $h = 10 \text{ W/m}^2 - K$) on the interior and exterior surfaces of the compartment. The second simulation was conducted an exterior value of $h_{ext} = 10 \text{ W/m}^2 - K$ consistent with natural convection near room temperature and an interior value of $h_{int} = 35 \text{ W/m}^2 - K$, which as discussed above was based on the results found in the literature [178, 179, 180].

D.5.2 Abaqus Thermal Model

The spatially and temporally varying standard surface heat flux from the FDS simulation or from the IHT was used as a boundary condition for the Abaqus thermal model. The thermal model included a solid representation of the entire aluminum compartment geometry. For thermal models in which the built-in FDS correlations for h were used, the FDS calculated coefficient was also imported into the thermal model. For models in which constant values of h were used in FDS, these values were also directly specified within Abaqus. For all of the thermal models in which heat fluxes from an IHT were used, the same h as used in the IHT was used in the thermal model. Re-radiation losses from the aluminum surfaces were modeled in the thermal model with a surface emissivity of 0.95. All of the thermal models included temperature dependent material thermal properties.

The thermal model was meshed using fully integrated solid quadratic hexahedral heat transfer elements (DC3D20). Only a single element was used through the thickness of the aluminum sheets as no significant thermal gradient was expected to develop through the material due to aluminum's high thermal conductivity. This high thermal conductivity also limits in-plane thermal gradients.

Because of this, an in-plane mesh seed of 80 mm was used in every plane of the compartment. This provided 12 elements in each direction. Thermal model convergence was validated using a thermal model with in-plane mesh seed of 40 mm.

D.5.3 Abaqus Mechanical Model

Full-field, transient temperature results from the thermal models were input into the mechanical models to simulate the high-temperature mechanical response of the compartment. The mechanical boundary conditions at the base of the compartment were fixed at the compartment corners and pinned along the edges to best represent the mounting used in the testing. The mechanical load on the top of the compartment was represented as uniform pressure under the four circles shown in Fig. D.5 with 1.6kN evenly split. The self-weight of the compartment was also included in the mechanical analysis. For all of the mechanical models, temperature dependent mechanical properties including elasticity, plasticity, creep, and thermal expansion were used [181].

The compartment geometry was represented using two-dimensional elements as opposed to the three-dimensional elements used in the thermal models. This was done to eliminate simulation errors caused by element aspect ratios in mechanical analysis. The mechanical model was meshed using quadratic quadrilateral shell elements with reduced integration (S8R). The mesh seed was biased to the compartment corners and locations where failure events were observed during the experiment. At these locations a mesh seed of 8 mm was used. The mesh seed was reduced to 30 mm away from these locations.

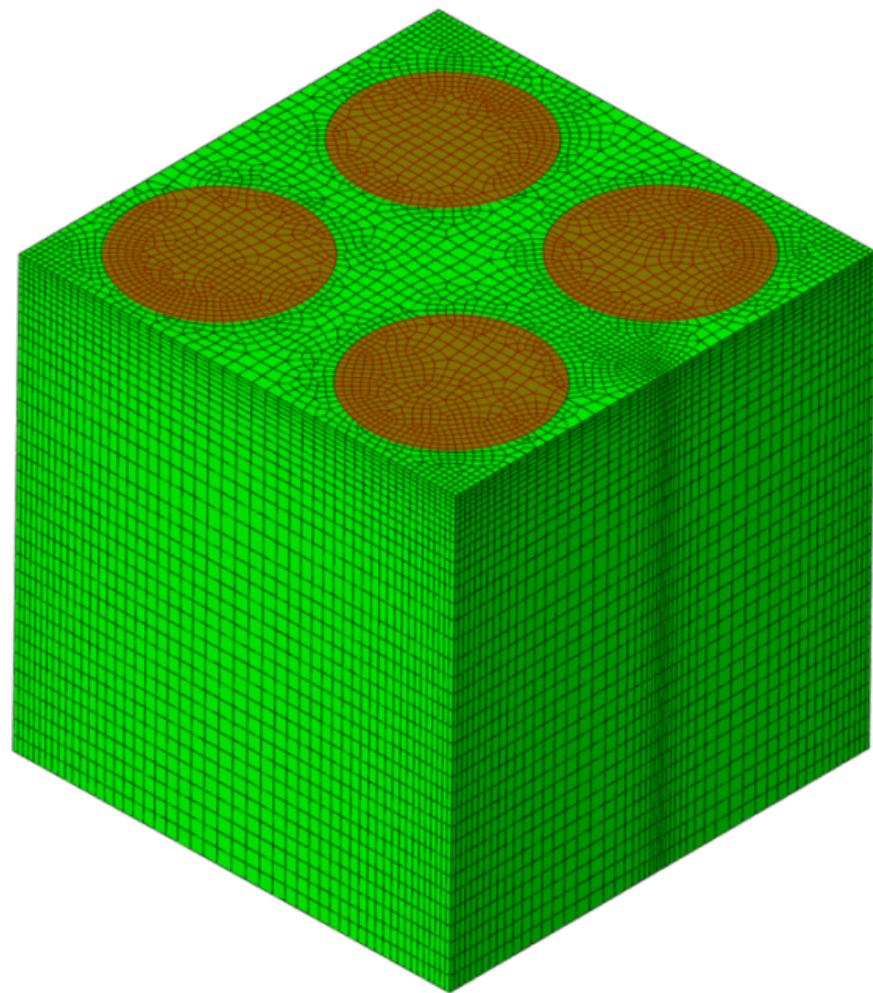


Figure D.5: Non-uniform mesh in Abaqus. The four red circles show the uniform pressure loading points

D.5.4 Simulation Matrix

A series of simulations were conducted to evaluate the ability of the FE model to predict the temperature and structural response of the aluminum compartment. In these simulations, the spatial and temporal heat transfer was input into the FE model based on the heat transfer measured in the experiments and the FDS predicted heat transfer. One aspect of uncertainty in both the measured and predicted heat transfer was the heat transfer coefficient. As a result, different combinations of heat transfer coefficients from Fig. D.3 were considered to evaluate the effects on the heat transfer and predicted temperatures. FE predictions were performed using the heat transfer from the following sources

1. Simulation using FDS with h_{int} and h_{ext} using built-in natural convection correlations,
2. Simulation using FDS $h_{int}=35 \text{ W/m}^2 - K$, $h_{ext}=10 \text{ W/m}^2 - K$,
3. Measured using $h_{int}=10 \text{ W/m}^2 - K$, $h_{ext}=10 \text{ W/m}^2 - K$, and
4. Measured using $h_{int}=35 \text{ W/m}^2 - K$, $h_{ext}=10 \text{ W/m}^2 - K$.

Predicted temperatures and heat transfer results from these different cases are compared to determine the sensitivity of the results on the heat transfer coefficient. In addition, structural models were conducted using heat transfer results from Cases 2 and 4 above.

D.6 Results

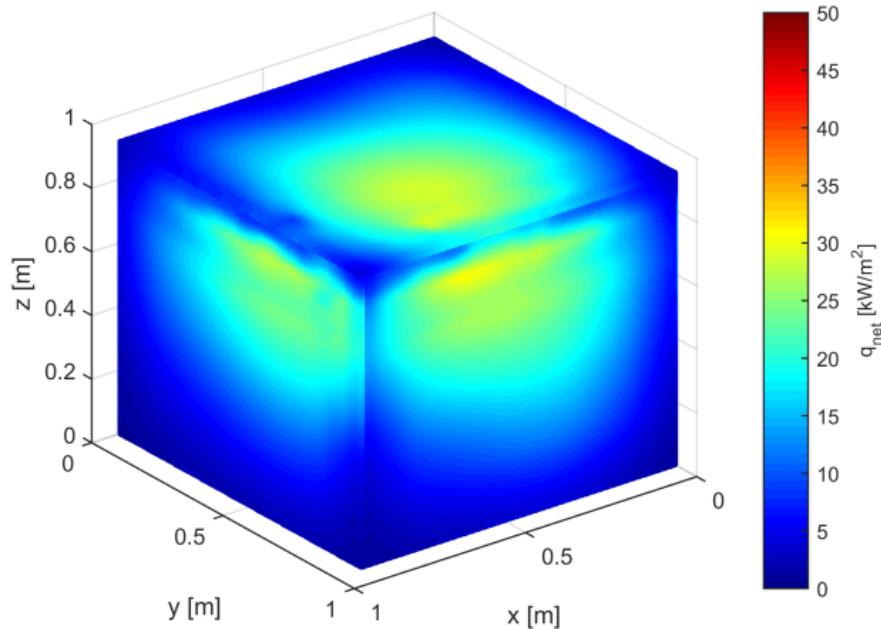
The net heat flux measured from IHT after 30 seconds of exposure to a 100 kW fire is shown in Fig. D.6. The maximum heat flux occurred at the top of the back wall and on the top near the back of the compartment. In the front view of the compartment with door, the higher heat flux toward

the right of the door is attributed to the hot gases exiting the door causing intermittent heating of the side of the door. The standard heat fluxes in the Fig. D.7 are in general agreement with those for a 100 kW fire against a flat wall [182].

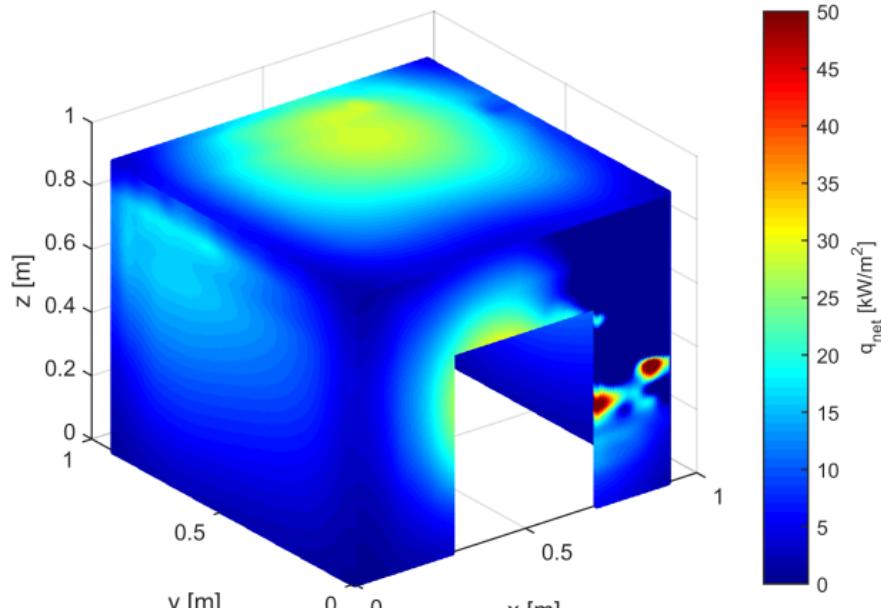
The net heat flux values measured using IHT were compared to those computed using FDS with $h_{int} = 35 \text{ W/m}^2 - K$ and $h_{ext} = 10 \text{ W/m}^2 - K$ and $h_{int} = h_{ext} = 10 \text{ W/m}^2 - K$. The net at the back wall top center [(x,y,z) = (0.5, 0.9, 0.8)] and the right wall top center [(x,y,z) = (0.9, 0.5, 0.8)] are shown in Fig. D.8a and Fig. D.8b, respectively. Full field measurements of net heat flux from IHT are compared with FDS predictions in Fig. D.8c, Fig. D.8d, Fig. D.8e, and Fig. D.8f at the time marked in Fig. D.8a and Fig. D.8b ($t = 30\text{s}$). The net heat transfer measured using the IHT are the same since the net heat transfer into the material responsible for the temperature rise is the same. However, changing the interior heat transfer coefficient in FDS has an effect on the net heat transfer.

The FDS predicted gas temperatures are compared with the measured gas temperatures in Fig. D.9. The measured and predicted gas temperatures in the upper layer and close to the floor are in good agreement. The largest difference was at $z = 0.36\text{m}$ which is close to the upper and lower layer interface height. This difference may be due to the fact that temperatures are sensitive to height changes in this region. As seen in this figure, reducing h_{int} causes the gas temperature to increase. The predicted temperatures from the simulation with $h_{int}=35 \text{ W/m}^2 - K$ agree more closely with the measured upper layer temperatures. This is due to the fact decreasing h_{int} decreases the heat losses to the boundaries causing the gas temperature to increase.

Measured surface temperature during the structural collapse experiment are compared with Abaqus predictions using IHT measured heat fluxes and FDS predicted heat fluxes in Fig. D.10 and Fig. D.11. The temperature from fire ignition to structural collapse at the back wall top center



(a) Back view



(b) Front view

Figure D.6: Inverse heat transfer measurements of net heat flux with $h_{int} = 35 \text{ W/m}^2 - K$, $h_{ext} = 10 \text{ W/m}^2 - K$

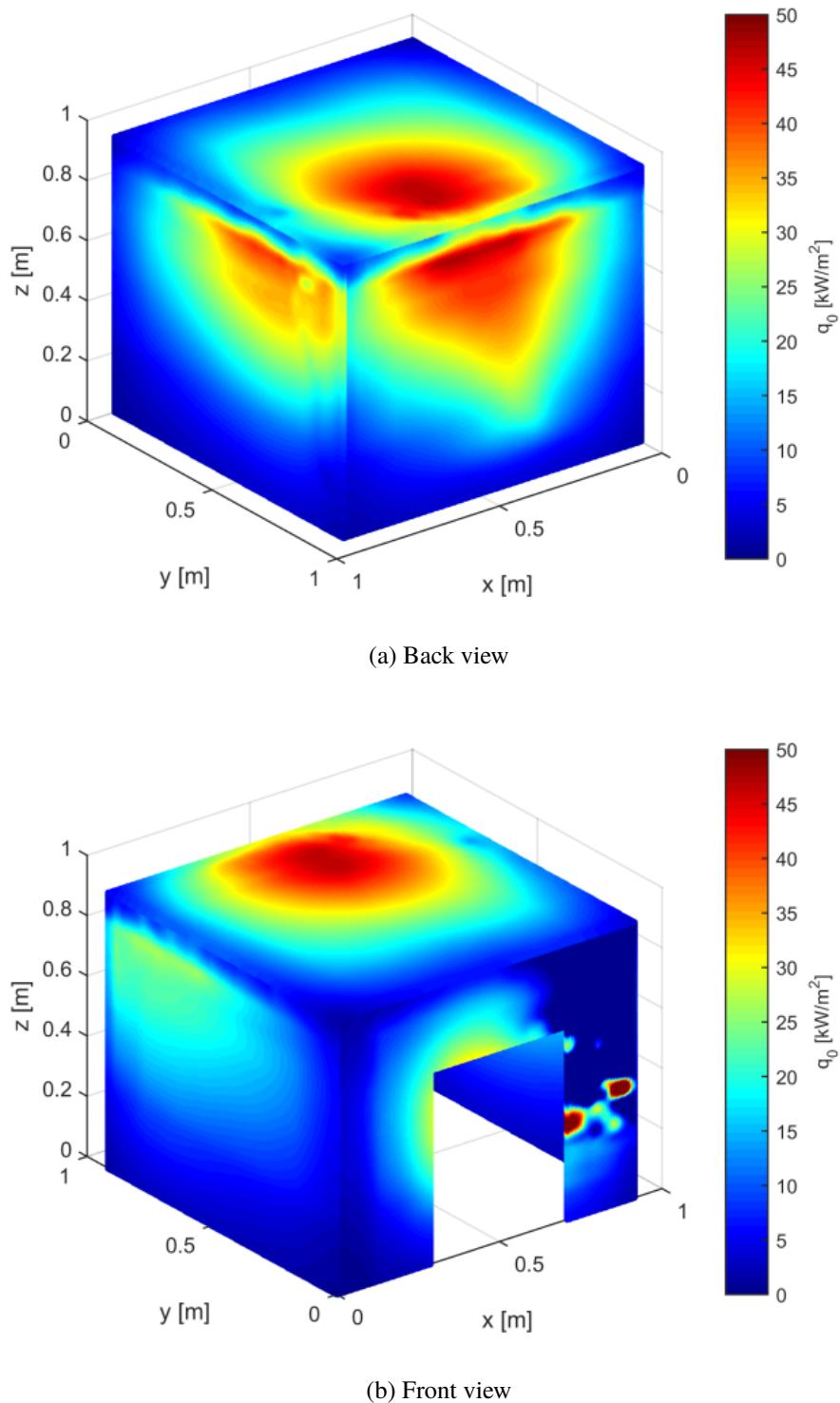


Figure D.7: Inverse heat transfer measurements of standard heat flux with $h_{int} = 35 \text{ W/m}^2 - K$, $h_{ext} = 10 \text{ W/m}^2 - K$

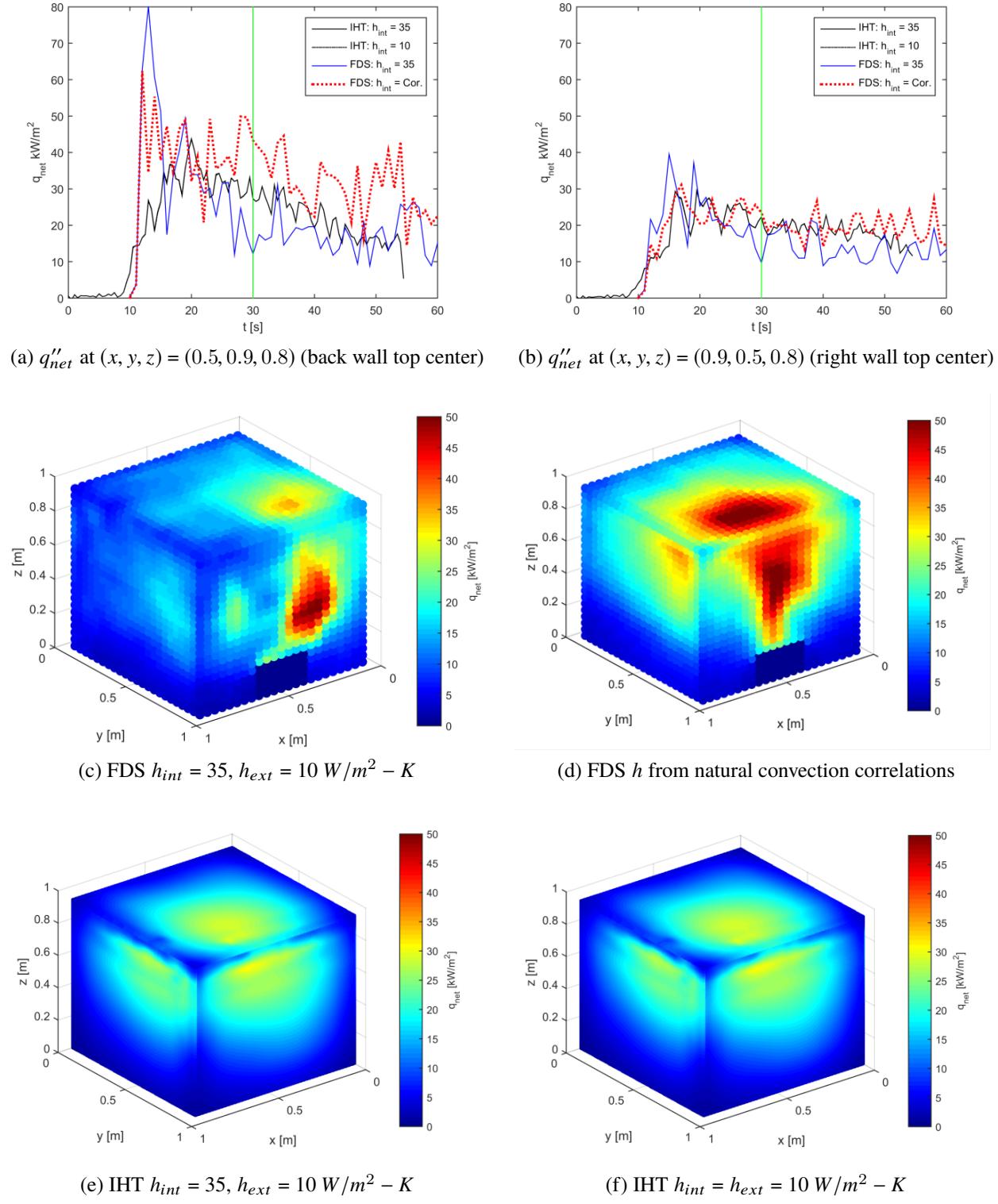


Figure D.8: Comparison of net heat fluxes from IHT method and FDS at various h

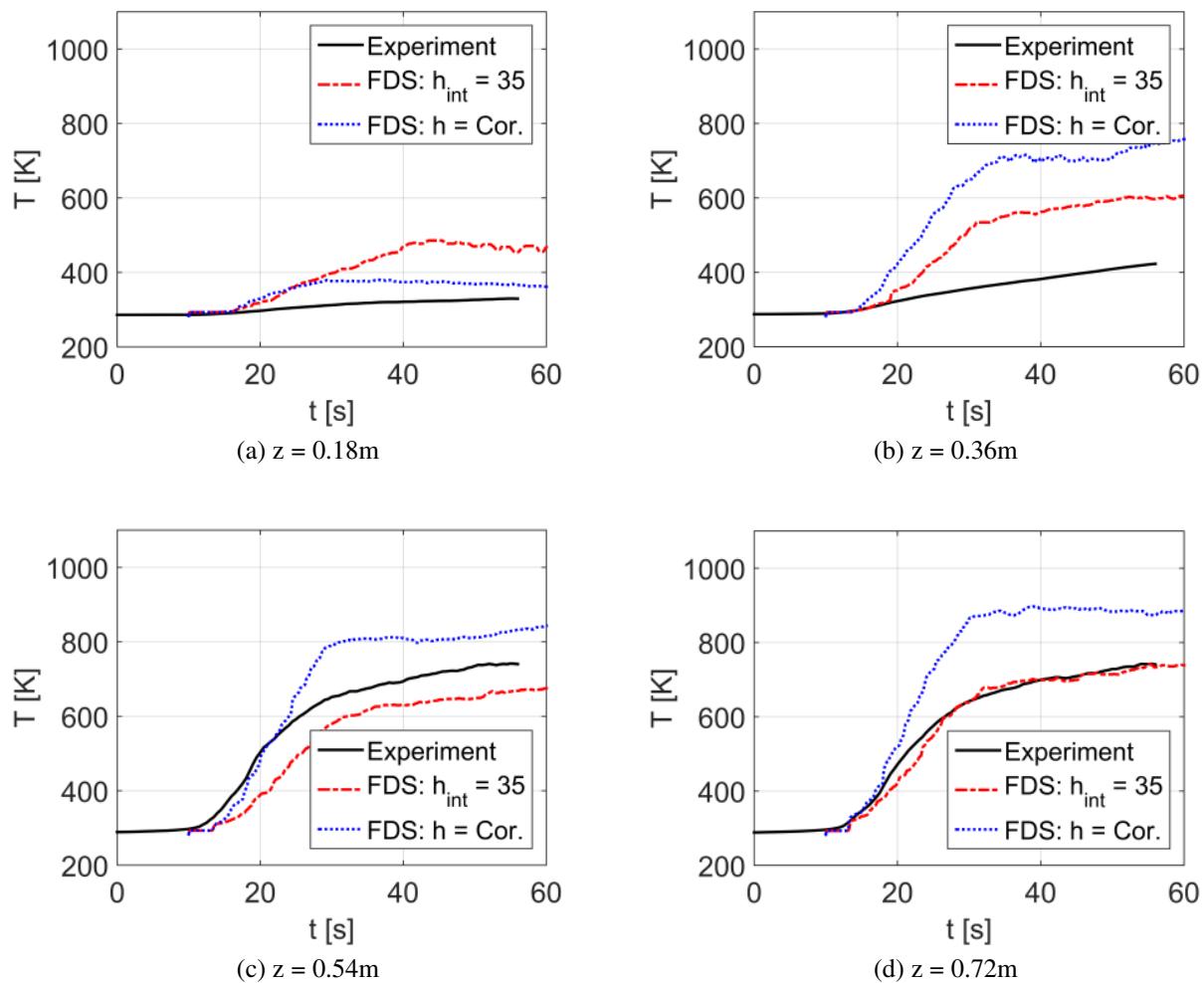


Figure D.9: Comparison of gas temperatures from heat flux testing with FDS predictions

$[(x,y,z) = (0.5,0.9,0.8)]$ and back wall top left $[(x,y,z) = (0.8,0.9,0.8)]$ are shown in Fig. D.10a and Fig. D.10b, respectively. Note the experimental and IHT predictions have been linearly offset to account for the initial 20kW fire which is not present in the FDS predictions. Fig. D.11 shows full field measurements and predictions of temperature from each method at the $t = 300$ s. The final collapse of the compartment began once the peak temperature of the back wall reached 800 K.

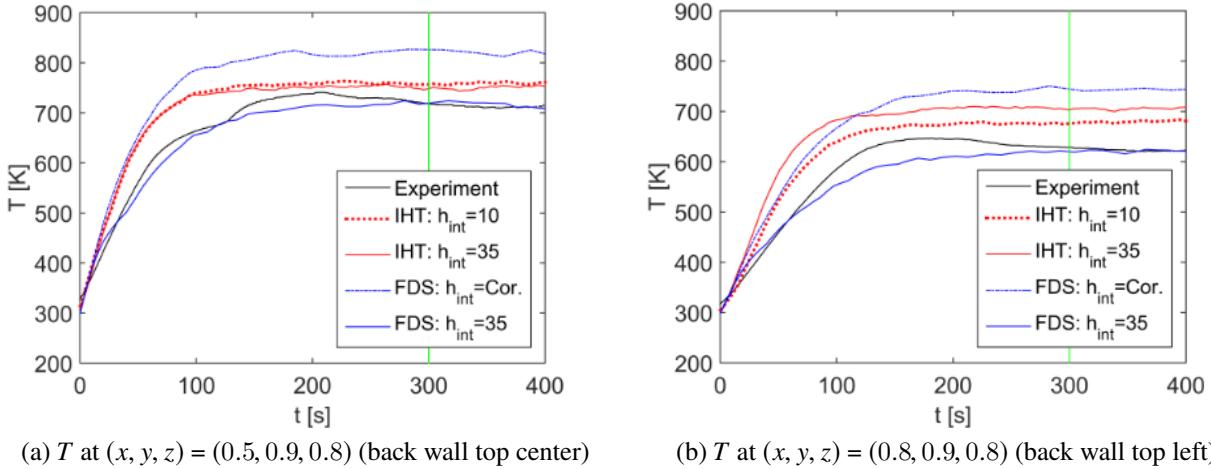
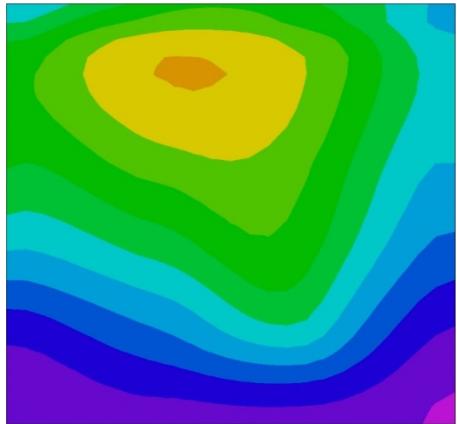
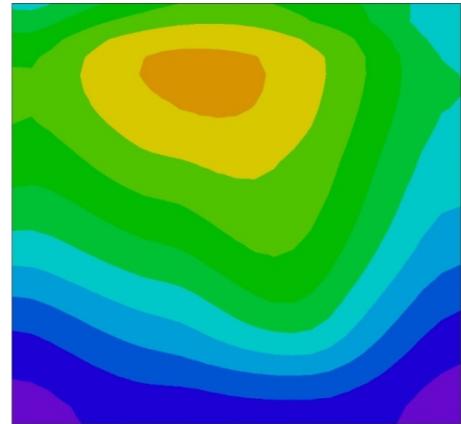


Figure D.10: Comparison of surface temperature predictions with experimental results at two points

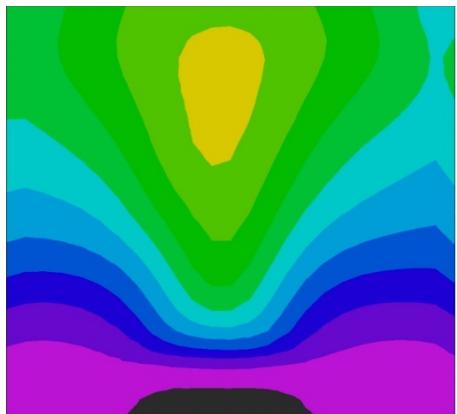
Full-field measurements of deflection along the back wall of the aluminum compartment from the structural collapse testing just before failure are shown in Fig. D.12. In all cases the center has roughly 30mm deflection towards the fire, while there is a 10-15mm deflection outward in the top corners. At the time of collapse (not shown), the center of the back wall had deflected out of plane 120mm towards the fire, and each of the top corners had deflected in plane along the x-axis 40mm.



(a) IHT $h_{ext} = 10 \text{ W/m}^2 - K$, $h_{int} = 35 \text{ W/m}^2 - K$



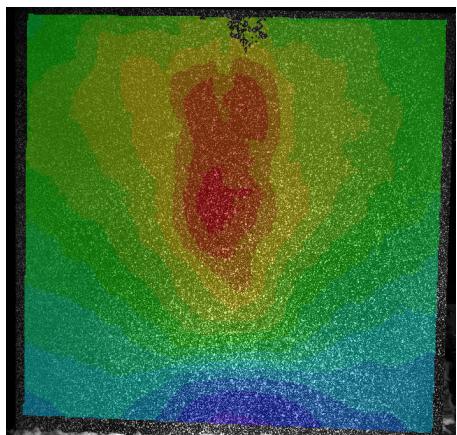
(b) IHT $h_{ext} = 10 \text{ W/m}^2 - K$, $h_{int} = 10 \text{ W/m}^2 - K$



(c) FDS $h_{ext} = 10 \text{ W/m}^2 - K$, $h_{int} = 35 \text{ W/m}^2 - K$



(d) FDS h from natural convection relations



(e) Experiment T (K)

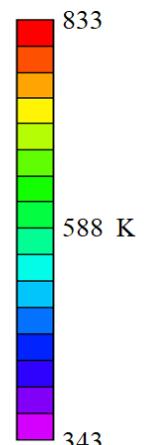


Figure D.11: Temperature in [K] of back wall after 300s of fire exposure

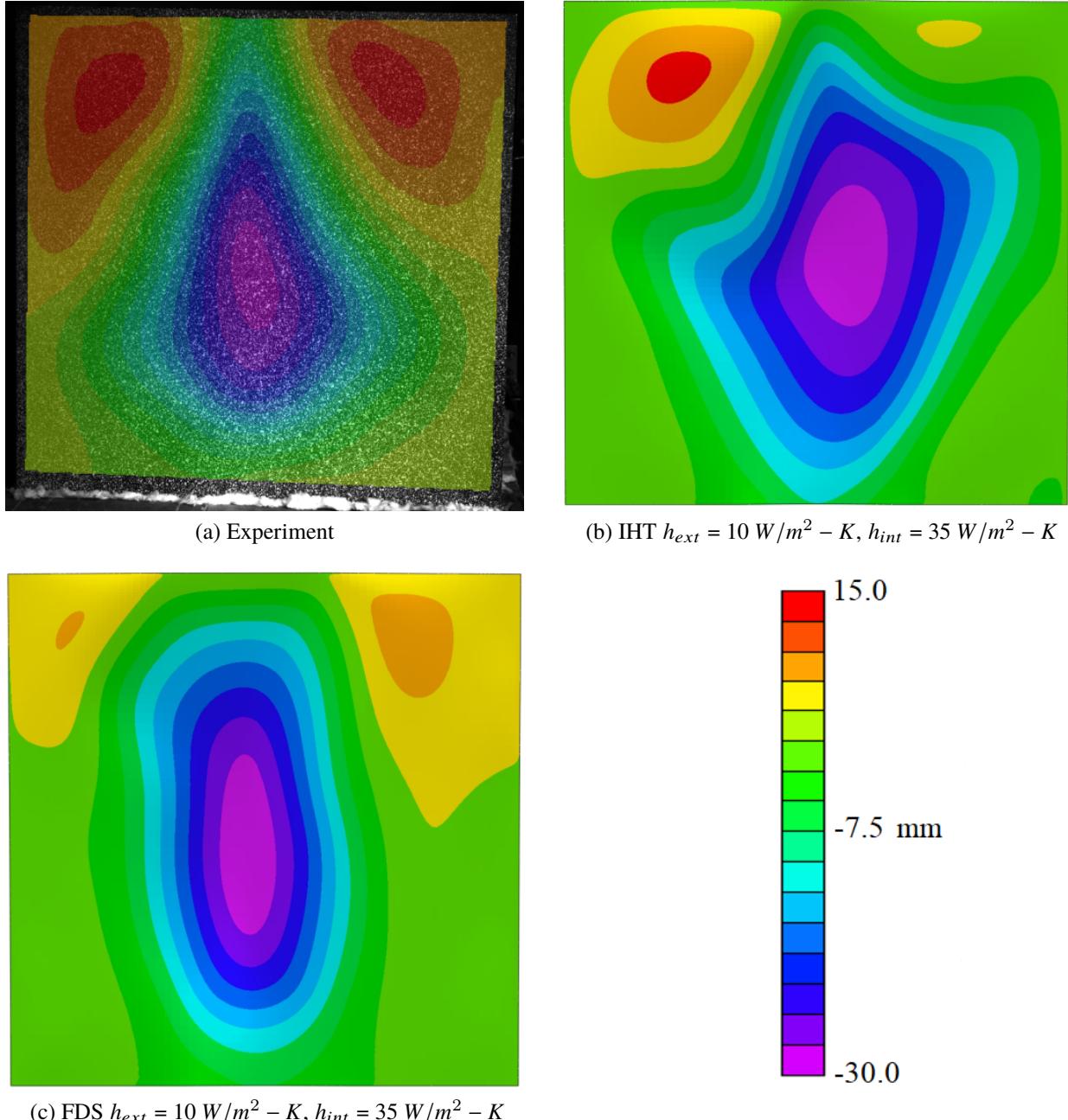


Figure D.12: Comparison of in-plane deflection predictions (y-direction in [mm]) with experimental measurements just prior to compartment collapse

D.7 Discussion

All of the thermal predictions of the compartment surface follow the same general trend as the measured surface temperatures, as is shown for two points in Fig. D.10. The surface temperatures predicted using the simulation results from FDS with the two different heat transfer coefficients resulted in significantly different predicted surface temperatures. This demonstrates the significance of h in obtaining accurate predictions of fire using FDS. The best prediction occurred using $h_{int} = 35 \text{ W/m}^2 - K$, $h_{ext} = 10 \text{ W/m}^2 - K$; whereas the worst prediction resulted from using the natural convection correlations for h (approximately $h = 10 \text{ W/m}^2 - K$). The difference in the predicted gas temperatures is a result of the difference in net heat flux into the surface of the boundaries between the two cases as seen in Fig. D.8.

The reason for the difference in the net heat transfer into the boundaries was determined to be due to FDS predicting different gas temperatures inside of the compartment in the two simulations, as seen in Fig. D.9c and Fig. D.9d. In FDS simulations with $h_{int} = 35 \text{ W/m}^2 - K$, the predicted gas temperatures are closer to measured gas temperatures. Decreasing h_{int} to approximately 10 $\text{W/m}^2 - K$ by using the natural convection correlation resulted in predictions with higher gas temperatures, which was attributed to decreasing the convection heat losses to the boundaries. Since the compartment fire was a radiation dominated phenomena, this increased the net heat transfer to the surface which resulted in a higher surface temperature.

The two predictions based on IHT heat fluxes for $h_{int} = 10 \text{ W/m}^2 - K$, $h_{ext} = 10 \text{ W/m}^2 - K$ and $h_{int} = 35 \text{ W/m}^2 - K$, $h_{ext} = 10 \text{ W/m}^2 - K$ produced temperature profiles in the middle of the two FDS predictions. As seen in Fig. D.10, the predicted temperatures are close but not exactly the same due to the difference in predicted temperatures and measured temperatures resulting in different losses. However, the thermal predictions using IHT were nearly independent of h . This is significant because h is often unknown in more complex fire scenarios.

The thermal modeling results demonstrate the spatially and temporally varying heat transfer pre-

dicted using FDS and measured using the IHT method are able to be used to generate accurate temperature predictions. However, a good approximation of the heat transfer coefficient is more important when predicting the heat transfer using FDS. The IHT method measures the net heat transfer to the surface; therefore, as long as a consistent heat transfer coefficient is used in the inverse analysis and the thermal analysis, the predicted temperature rise will be the same. The predicted deformed shape of the compartment just prior to collapse is seen in Fig. D.12 to match well with the experimental data. These predictions were performed using the FDS model results with $h_{int}=35 \text{ W/m}^2 - \text{K}$, which produces gas temperatures similar to that in the experiment. The similar predicted structural response is expected since the predicted temperatures using the heat transfer from FDS and IHT method are similar. In order to capture the non-uniform deflected shape, the spatial and temporal heat transfer from the fire must be accounted for in the modeling.

D.8 Conclusion

Numerical predictions of surface temperature and deflection of an aluminum 6061-T6 compartment using heat fluxes obtained from FDS and IHT were successfully validated against experimental results. Numerical simulations using FDS were found to be sensitive to changes in the h due to differences in the predicted temperature of the upper gas layer. However, predictions of temperature using IHT were not sensitive to changes in h . In situations where h is well understood either FDS or IHT may be used to obtain heat fluxes for structural modeling. However, application of IHT to experiments where h is not well understood may provide additional insight to the spatial and temporal variation in energy transfer through a sample without requiring extensive study to quantify the value of h .

D.9 Acknowledgements

The authors would like to thank Maggie Goetz and Joseph Starr for assisting in performing the experiments as well as Mohammad Nahid for assisting with some of the initial FDS simulations.