



FEDERAL UNIVERSITY OF SANTA CATARINA
TECHNOLOGY CENTER
AUTOMATION AND SYSTEMS ENGINEERING DEPARTMENT
UNDERGRADUATE COURSE IN CONTROL AND AUTOMATION ENGINEERING

André Padilha Alves

Accelerating Finite Element Simulations with Graph Neural Networks

Aachen
2025

André Padilha Alves

Accelerating Finite Element Simulations with Graph Neural Networks

Final report of the subject DAS5511 (Course Final Project) as a Concluding Dissertation of the Undergraduate Course in Control and Automation Engineering of the Federal University of Santa Catarina. Supervisor: Prof. Eric Aislan Antonelo, Dr.

Aachen
2025

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Alves, André Padilha
Accelerating Finite Element Simulations with Graph
Neural Networks / André Padilha Alves ; orientador, Eric
Aislan Antonelo, coorientador, Aakash Singh, 2026.
95 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia de Controle e Automação,
Florianópolis, 2026.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Engenharia de
Controle e Automação. 3. Graph Neural Network. 4. Graph
Convolutional Network. 5. Finite Elements Method. I.
Antonelo, Eric Aislan. II. Singh, Aakash. III.
Universidade Federal de Santa Catarina. Graduação em
Engenharia de Controle e Automação. IV. Título.

André Padilha Alves

Accelerating Finite Element Simulations with Graph Neural Networks

This dissertation was evaluated in the context of the subject DAS5511 (Course Final Project) and approved in its final form by the Undergraduate Course in Control and Automation Engineering

Florianópolis, March 4th, 2026.

Prof. Marcelo de Lellis Costa de Oliveira, Dr.
Course Coordinator

Examining Board:

Prof. Eric Aislan Antonelo, Dr.
Advisor
UFSC/CTC/EAS

Aakash Singh, M.Sc.
Supervisor
Company/University Fraunhofer IPT

Prof. Jean Panaioti Jordanou, Dr.
Evaluator
Institution UFSC/CTC/EAS

Prof. Edson Roberto De Pieri, Dr.
Board President
UFSC/CTC/EAS

I dedicate this work to my family, whose unwavering support, encouragement, and belief in my potential have sustained me throughout every stage of this journey. Above all, I dedicate it to my brother, who has been my greatest mentor and inspiration in choosing the path of engineering.

ACKNOWLEDGEMENTS

The completion of this thesis was made possible through the support and collaboration of many individuals and institutions. I am very grateful to Fraunhofer IPT for providing the infrastructure, resources, and technical environment necessary for the development of this work, and to the Digitalization and Process planning department for offering the tools and conditions that enabled its execution. I also want to thank my academic advisor, Eric Aislan Antonelo, and to my supervisor, Aakash Singh, whose guidance, expertise, and constructive feedback were essential throughout the research. Acknowledgment is also due to my second supervisor, Gustavo Laydner de Melo Rosa, as well as to my friends Bruno Mingoti, Carlos Eduardo Xavier Correia, Gustavo Portela Alves and Rodrigo Jost for their continuous encouragement, and motivation. Above all, heartfelt thanks go to my family for their constant support and confidence throughout this entire journey.

DISCLAIMER

Aachen, February 23th, 2026.

As representative of the Fraunhofer IPT in which the present work was carried out, I declare this document to be exempt from any confidential or sensitive content regarding intellectual property, that may keep it from being published by the Federal University of Santa Catarina (UFSC) to the general public, including its online availability in the Institutional Repository of the University Library (BU). Furthermore, I attest knowledge of the obligation by the author, as a student of UFSC, to deposit this document in the said Institutional Repository, for being it a Final Program Dissertation (*“Trabalho de Conclusão de Curso”*), in accordance with the *Resolução Normativa n° 126/2019/CUn*.

Aakash Singh

Digital unterschrieben von
Aakash Singh
Datum: 2026.03.18 10:00:48
+01'00'

Aakash Singh
Fraunhofer IPT

ABSTRACT

Milling of thin-walled components is highly susceptible to vibrations during the manufacturing process, where structural flexibility leads to pronounced vibration and instability, often leading to excessive rework and productivity losses. When it comes to process planning for such parts, accurate estimation of milling forces and structural dynamics is fundamental, as these variables influence the behaviour of technological models used for process optimisation. With the increasing digitalization of manufacturing, these models are progressively transitioning from academic prototypes to industrial CAM environments, enabling predictive and adaptive machining strategies. In this context, the Fraunhofer IPT has developed the dPart® digital twin framework, which integrates process simulations, machine data, and computational models to support real-time decision making. In industrial practice, accurate prediction of the dynamic behaviour of workpieces is essential for process planning, monitoring, and digital twin integration. Existing approaches rely heavily on repeated Finite Element Method (FEM) simulations, which are accurate but computationally expensive, time-intensive, and scale poorly with the large number of cutter locations required for full-operation evaluation. As the dPart® digital twin infrastructure evolves, these limitations can represent a bottleneck in the future. To overcome these limitations, this work investigates the use of machine learning techniques, specifically graph neural networks (GNNs) as surrogate models for FEM-based deformation prediction. A dataset of modal simulations was generated in ANSYS, from which nodal coordinates, boundary conditions, and deformation fields were extracted and converted into graph representations. Two architectures were tested, them being a graph convolutional network and a MeshGraphNet inspired network, both designed to operate directly on mesh connectivity and to learn spatial deformation patterns from modal data. The surrogate models were trained to predict node-level displacement fields for the first and second vibration modes of the workpiece. Results demonstrate that the proposed models are capable of reproducing the global deformation behaviour with less than 1% mean relative percentage error for the first deformation mode. These findings indicate that GNN-based surrogates hold promise for accelerating vibration analysis in milling applications and could support more efficient process planning and monitoring workflows.

Keywords: Finite Elements Method. Graph Neural Network. Graph Convolutional Network.

RESUMO

A usinagem de componentes de paredes finas é altamente suscetível a vibrações durante o processo de manufatura, uma vez que a flexibilidade estrutural resulta em vibrações e instabilidade significativas, frequentemente resultando em retrabalho excessivo e perdas de produtividade. Quando se trata de planejamento de processos para essas peças, a estimativa precisa das forças de fresagem e da dinâmica estrutural é fundamental, pois essas variáveis influenciam o comportamento dos modelos tecnológicos usados para a otimização do processo. Com a crescente digitalização da manufatura, esses modelos estão passando progressivamente de protótipos acadêmicos para ambientes CAM industriais, permitindo estratégias de usinagem preditivas e adaptativas. Nesse contexto, o Fraunhofer IPT desenvolveu a estrutura dPart® digital twin, que integra simulações de processos, dados de máquinas e modelos computacionais para dar suporte à tomada de decisões em tempo real. Na prática industrial, a previsão precisa do comportamento dinâmico das peças de trabalho é essencial para o planejamento do processo, o monitoramento e a integração do gêmeo digital. As abordagens existentes dependem muito de simulações repetidas do Método dos Elementos Finitos (FEM), que são precisas, mas computacionalmente caras, demoradas e pouco escalonáveis com o grande número de locais de corte necessários para a avaliação de toda a operação. À medida que a infraestrutura de gêmeos digitais dPart® evolui, essas limitações podem representar um gargalo no futuro. Para superar essas limitações, este trabalho investiga o uso de técnicas de aprendizado de máquina, especificamente redes neurais de grafos (GNNs) como modelos substitutos para a previsão de deformação baseada em FEM. Um conjunto de dados de simulações modais foi gerado no ANSYS, a partir do qual as coordenadas nodais, as condições de contorno e os campos de deformação foram extraídos e convertidos em representações gráficas. Duas arquiteturas foram testadas, sendo elas uma rede convolucional gráfica e uma rede inspirada na MeshGraphNet, ambas projetadas para operar diretamente na conectividade da malha e para aprender padrões de deformação espacial a partir de dados modais. Os modelos substitutos foram treinados para prever campos de deslocamento em nível de nó para o primeiro e o segundo modos de vibração da peça de trabalho. Os resultados demonstram que os modelos propostos são capazes de reproduzir o comportamento de deformação global com um erro percentual relativo médio inferior a 1% para o primeiro modo de deformação. Essas descobertas indicam que os substitutos baseados em GNN são promissores para acelerar a análise de vibração em aplicações de fresamento e podem dar suporte a fluxos de trabalho de monitoramento e planejamento de processos mais eficientes.

Palavras-chave: Método dos elementos finitos. Redes neurais de grafos. redes convolucionais de grafos.

LIST OF FIGURES

Figure 1 – Document Structure.	19
Figure 2 – Uncut chip thickness and cycloidal trajectory in 3-axis milling.	22
Figure 3 – Classical Oblique Cutting Geometry.	23
Figure 4 – Overview of natural frequencies in a milling machine.	26
Figure 5 – Classification of vibrations based on their temporal behavior.	27
Figure 6 – Causes of externally excited oscillations in milling.	28
Figure 7 – Oscillation amplitude as a function of depth of cut.	28
Figure 8 – Regenerative feedback mechanism in milling.	29
Figure 9 – Stability lobe diagram.	31
Figure 10 – Example of data to graph conversion.	36
Figure 11 – Examples of common GNNs tasks.	37
Figure 12 – Message passing with aggregation example.	38
Figure 13 – Dpart pipeline highlighting where the GNN surrogate will be used.	43
Figure 14 – Overview of the proposed GNN-based surrogate modeling pipeline.	47
Figure 15 – Overview of the proposed project steps.	49
Figure 16 – Example toolpath used for data generation.	51
Figure 17 – Example of modal analysis results.	52
Figure 18 – Example of data obtained from ANSYS.	53
Figure 19 – Example of data extraction from a mesh.	58
Figure 20 – Example of a point cloud display for a mesh.	59
Figure 21 – Resulting STL file without node matching.	60
Figure 22 – Corrected resulting STL file.	61
Figure 23 – GCN architecture.	64
Figure 24 – MAMGN architecture.	65
Figure 25 – Loss evolution over training epochs for GCN Mode 1.	70
Figure 26 – Loss evolution over training epochs for MAMGN Mode 1.	71
Figure 27 – Loss evolution over training epochs for GCN Mode 2.	71
Figure 28 – Loss evolution over training epochs for MAMGN Mode 2.	72
Figure 29 – MAPE evolution over training epochs for GCN Mode 1.	73
Figure 30 – MAPE evolution over training epochs for MAMGN Mode 1.	73
Figure 31 – MAPE evolution over training epochs for GCN Mode 2.	74
Figure 32 – MAPE evolution over training epochs for MAMGN Mode 2.	74
Figure 33 – MAE evolution over training epochs for GCN Mode 1.	75
Figure 34 – MAE evolution over training epochs for MAMGN Mode 1.	76
Figure 35 – MAE evolution over training epochs for GCN Mode 2.	76
Figure 36 – MAE evolution over training epochs for MAMGN Mode 2.	77
Figure 37 – MAPE over test set for GCN Mode 1.	79

Figure 38 – MAPE over test set for MAMGN Mode 1.	79
Figure 39 – MAPE over test set for GCN Mode 2.	80
Figure 40 – MAPE over test set for MAMGN Mode 2.	80
Figure 41 – Deformations ground truth for Mode 1.	82
Figure 42 – Deformations inferred by GCN for Mode 1.	82
Figure 43 – Deformations inferred by MAMGN for Mode 1.	83
Figure 44 – Deformations ground truth for Mode 2.	83
Figure 45 – Deformations inferred by GCN Mode 2.	84
Figure 46 – Deformations inferred by MAMGN Mode 2.	84

LIST OF TABLES

Table 3 – Comparison of machine learning paradigms and representative use cases.	32
Table 4 – Common benchmark graph datasets available in PyTorch Geometric.	35
Table 5 – System specifications used for all experiments.	66
Table 6 – Final prediction error metrics for both models and vibration modes. .	78

LIST OF ABBREVIATIONS AND ACRONYMS

CNNs	Convolutional Neural Networks
FEM	Finite Element Method
FRF	Frequency Response Function
GCNs	Graph Convolutional Networks
GNNs	Graph Neural Networks
MAE	Mean Absolute Error
MAMGN	Modal Analysis Mesh Graph Net
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLP	Multilayer Perceptron

LIST OF SYMBOLS

v_c	Cutting speed
v_f	Feed speed
n	Spindle speed
N_t	Number of teeth
f_z	Feed per tooth
d_{tool}	Tool diameter
φ_c	Contact angle
φ_s	Entry angle
φ_e	Exit angle
a_e	Radial depth of cut
λ	Inclination angle
κ	Cutting edge angle
F_a	Axial force
h_{sp}	Instantaneous uncut chip thickness
db	Differential cutting edge length perpendicular to cutting velocity
dz	Differential length in axial direction
m	Mass
k	Stiffness
D	Lehr's damping ratio
a_p	Axial depth of cut
$a_{p,cr}$	Critical depth of cut
$h(t)$	Dynamic chip thickness
τ	Tooth passing period
G_p	Regenerative feedback term (frequency domain)
ω	Excitation frequency
g	Static compliance
ω_n	Natural frequency (rad/s)
f_n	Natural frequency (Hz)
$m_{uv}^{(k)}$	Message from node u to node v at layer k
ϕ_m	Message function (MLP)
$h_v^{(k)}$	Feature vector of node v at layer k
e_{uv}	Edge feature between nodes u and v
$m_v^{(k)}$	Aggregated message at node v at layer k
$\mathcal{N}(v)$	Neighborhood of node v
ϕ_u	Node update function (MLP)
$\alpha_{uv}^{(k)}$	Attention weight from node u to v at layer k
W	Learnable weight matrix (attention)

$\mathbf{a}$	Learnable attention vector
d_i	Euclidean distance from node i to cutter
$\mathbf{x}_i$	Position vector of node i
$\mathbf{x}_{\text{cutter}}$	Cutter position vector
$\text{deg}_{\text{norm}}(i)$	Normalized node degree
$\text{deg}(i)$	Degree of node i (number of connected edges)
$L_{\text{mean}}(i)$	Mean edge length at node i
$\cos(\theta_i)$	Cutter direction cosine at node i
$\mathbf{c}$	Mesh centroid
d_{max}	Maximum radial distance from mesh centroid
N	Total number of mesh nodes
$\hat{u}_i$	Predicted nodal displacement at node i
u_i	Ground-truth nodal displacement at node i
β	Smooth L1 loss transition threshold

CONTENTS

1	INTRODUCTION	17
2	THEORETICAL BASIS	20
2.1	MILLING	20
2.2	TECHNOLOGY OF MILLING	20
2.2.1	Fundamentals of Milling	21
2.2.2	Cutter-Workpiece Engagement	23
2.2.3	Engagement Simulation	24
2.2.4	Process Simulation and Digital Twinning	24
2.3	VIBRATIONS IN MILLING	25
2.3.1	Occurrence of Vibrations in Machine Tools	25
2.3.2	Types of Vibrations	26
2.3.3	Regenerative Effect	28
2.3.4	Possibilities for Vibration Reduction	29
2.3.5	Mathematical Description of Vibration Characteristics	30
2.4	FUNDAMENTALS OF MACHINE LEARNING AND DEEP LEARNING	32
2.4.1	Supervised, Unsupervised, and Reinforcement Learning	32
2.5	GRAPH NEURAL NETWORKS	33
2.5.1	Graphs	34
2.5.2	Applications of Graph Neural Networks	35
2.5.3	Message Passing Mechanism	37
2.5.4	Similarities between message passing and FEM	39
3	PROBLEM DESCRIPTION AND TECHNICAL REQUIREMENTS . .	41
3.1	FRAUNHOFER-GESELLSCHAFT	41
3.1.1	High Performance Cutting Department	41
3.2	PROBLEM DESCRIPTION	42
3.2.1	Technical Requirements	43
4	PROPOSED SOLUTION AND METHODOLOGY	45
4.1	SCOPE, ASSUMPTIONS, AND CONSTRAINTS	46
4.2	HOW THE SOLUTION MEETS THE REQUIREMENTS	46
5	DEVELOPMENT AND RESULT ANALYSIS	49
5.1	SOLUTION DEVELOPMENT	49
5.1.1	Data Generation	50
5.1.2	Data Preprocessing	55
5.1.3	Model Definition and Training Strategy	62
5.1.4	Loss Function and Evaluation Metrics	66
5.2	RESULT ANALYSIS	69
5.2.1	Experimental Results and Model Comparison	70

5.2.2	Discussion and Result Synthesis	85
6	CONCLUSION	86
6.1	SUMMARY OF THE WORK AND MAIN RESULTS	86
6.1.1	Expectations, Objectives, and Limitations	86
6.2	EVALUATION OF FUNCTIONAL REQUIREMENTS	87
6.2.1	Data Processing and Graph Construction	87
6.2.2	Graph-Based Machine Learning Implementation	87
6.2.3	Model Evaluation Against FEM Results	87
6.2.4	Scalability Testing	88
6.2.5	Result Visualization	88
6.3	EVALUATION OF NON-FUNCTIONAL REQUIREMENTS	88
6.3.1	Computational Efficiency	88
6.3.2	Accuracy	88
6.3.3	Scalability	88
6.3.4	Impact of the Work	88
6.3.5	Future Works	89
	References	90

1 INTRODUCTION

Milling is a subtractive process where material is continuously removed through the relative motion between the tool and workpiece. During milling complex physical phenomena with thermal and vibrational aspects occur which influences process stability and productivity. Although every year machining technologies are more performant where accuracy, flexibility and productivity are concerned, vibrations are the cause of numerous problems. It includes tool breakage, poor surface quality, increased wear and this, when seen from the manufacturing industries point of view, leads to product rejections and rework. In context to rework, in Turbomachinery, components tend to have higher amplitude of vibrations be cause of their high aspect ratio and difficult to machine materials. For example, in the manufacturing of Blade Integrated Disks (BLISKs), which is a structurally complex workpiece, the major challenge faced by the industries is the amount of rework. The main challenge faced by manufacturing industries is the monitoring of the milling process, which when done effectively, could save reworks, which is estimated to be around 25%.

Through effective monitoring, vibrations and chatter, one of the major contributors to BLISK rework, could be effectively controlled and stabilized. Although cutting stability has been studied in detail over the past half-century, vibration predictions are difficult to make due to the continuous changes in stiffness of the thin-walled workpiece. On one hand, there is variation of the force application point due to the different tool positions and on the other hand, a change in stiffness and mass of the workpiece due to material removal. Currently this question is approached with finite element method Finite Element Method (FEM) simulations, but they are time-intensive, resource-heavy, and scale poorly with mesh size, and the need to do it for multiple cutter locations increases the time needed to create a good estimation of the workpiece dynamics with reasonable precision at all cutter locations. The FEM process has been optimized to a degree where further improvements are challenging to achieve. Additionally, interpolation fails to capture the complexity of the non-linear nature of partial differential equations whose solutions are approximated, which is a problem that machine learning can excel at.

In recent years, data-driven approaches have offered an alternative path for addressing these limitations. Machine learning techniques have demonstrated a strong capacity to approximate complex, nonlinear physical behaviours directly from data, enabling the construction of surrogate models that reproduce simulation results at a fraction of the computational cost. Among these approaches, Graph Neural Networks Graph Neural Networks (GNNs) have emerged as a particularly suitable class of models for problems defined on meshes or discretized physical domains.

A triangular mesh can be interpreted as a graph: nodes correspond to spatial

points of the discretized geometry, and edges represent the structural connectivity imposed by the mesh topology. GNNs operate directly on this structure through message-passing operations, propagating information across the mesh in a manner analogous to the propagation of physical quantities in numerical methods for partial differential equations. These similarities between GNNs and mesh-based numerical models makes it particularly well suited to the task of learning from FEM data.

Furthermore, recent studies have shown that GNN-based surrogate models can provide significant efficiency gains when approximating the behaviour of large-scale physical systems, particularly when the underlying domain is represented as a mesh or irregular grid. It is inferred from such works that replacing repeated FEM simulations with a trained GNN surrogate could reduce computational requirements while retaining sufficient accuracy for tasks such as process monitoring, toolpath evaluation, and vibration prediction.

In this context, the present work explores the development of a GNN-based surrogate model capable of predicting the deformation response of a thin-walled work-piece during milling, using FEM-generated modal data as ground truth. By leveraging the relational structure of the mesh and learning the vibrational characteristics of the component directly from simulation data, the proposed approach aims to enable faster, scalable, and more efficient prediction of dynamic behaviour across different tool positions.

DOCUMENT STRUCTURE

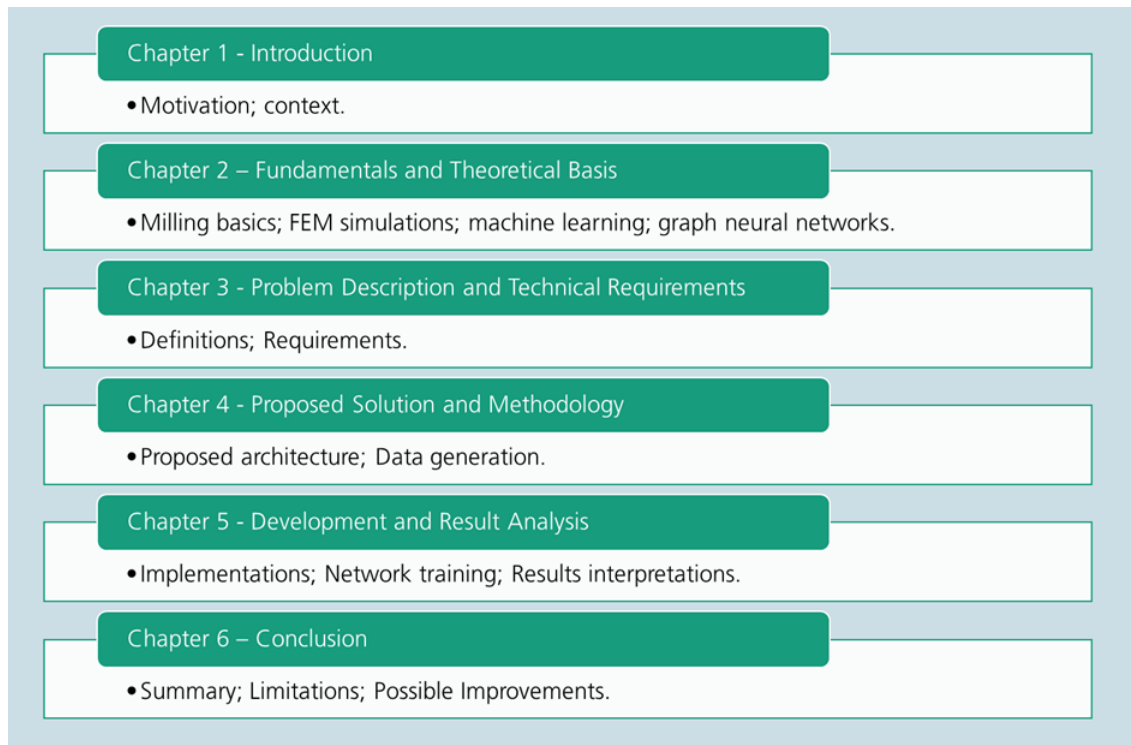
The structure of this document follows the workflow presented in 1. The document begins by introducing the concepts and motivation related to the project, establishing the industrial context associated with vibration prediction in milling and the limitations of current FEM-based approaches. Building upon this, the theoretical foundations and state of the art are presented, covering the fundamentals of machining dynamics, finite element modelling, and recent developments in machine learning and graph neural networks. This background provides the necessary basis to formalize the problem, define the technical requirements and constraints of the proposed system.

Following the adopted methodology is detailed, including the data generation workflow, the preprocessing steps used to transform FEM results into graph-based representations, and the architecture of the surrogate models explored in this project. Subsequently, the development process is described in detail, presenting the data processing steps, implementation of the surrogate models, their training procedures, and a comprehensive evaluation of their performance through quantitative metrics and mesh-based visual analyses.

Finally, the last chapter concludes the document by summarizing the contributions achieved, discussing the limitations encountered, and outlining future research

directions to further advance the integration of GNN-based surrogates into manufacturing process monitoring workflows.

Figure 1 – Document Structure.



2 THEORETICAL BASIS

This chapter presents the theoretical basis necessary for understanding the problem addressed, introducing the main concepts related to the milling process and the methods and definitions used in the simulations performed. Finally, the fundamentals of machine learning, which serve as theoretical support for the development of the project, are discussed.

2.1 MILLING

Mechanical components with tight geometric tolerances and high surface integrity are predominantly produced by machining operations. These machining processes have the material be selectively removed or plastically transformed in a controlled manner to obtain the desired part geometry. These processes serve multiple objectives, ranging from high material removal rates during rough shaping to the achievement of micron-level dimensional accuracy and superior surface quality in finishing operations.

From a manufacturing perspective, production technologies can be classified into subtractive, additive, and forming processes. Subtractive processes remove material from a workpiece (e.g., turning, drilling, milling, grinding), additive processes build components layer by layer, and forming processes reshape material without material removal. Among subtractive techniques, milling occupies a central role due to its flexibility in generating complex geometries and its capability to combine productivity with precision.

Despite its broad industrial relevance, milling is intrinsically associated with dynamic challenges. Process-induced vibrations arise from the interaction between the rotating cutter and the compliant machine-tool-workpiece system. Such dynamic effects compromise surface integrity, dimensional accuracy, and tool life. A comprehensive understanding of milling kinematics and process dynamics is therefore essential for predicting and mitigating adverse effects such as chatter and forced vibrations. This section introduces the technological and mechanical foundations of milling, emphasizing the mechanisms governing cutter-workpiece interaction.

2.2 TECHNOLOGY OF MILLING

The majority of engineered components are produced through machine-based operations that convert raw stock into functional parts. Although bulk forming processes such as forging, rolling, and casting are widely applied, final dimensional accuracy and surface finish are typically achieved through material removal operations (Altintas, 2012).

Material removal processes can be divided into cutting and abrasive operations. Primary cutting operations include milling, turning, and drilling, whereas grinding belongs to abrasive machining. Grinding is generally employed when superior surface quality and tight tolerances are required, while cutting operations allow higher material removal rates and greater flexibility in shaping.

Cutting operations are commonly structured into three sequential stages:

- **Roughing:** removal of large material volumes to approximate the final geometry;
- **Semi-finishing:** intermediate step to reduce residual material and prepare for final tolerances;
- **Finishing:** precision operation to meet dimensional and surface specifications (Klocke, 2018).

Depending on tolerance requirements, semi-finishing may already fulfill local specifications, making additional finishing unnecessary.

The theoretical analysis of machining mechanics traditionally begins with orthogonal cutting. Due to its simplified two-dimensional configuration and reduced parameter set, orthogonal cutting provides a tractable framework for understanding chip formation and force generation (Clausen, 2005). This conceptual foundation supports the modelling of more complex three-dimensional milling operations.

2.2.1 Fundamentals of Milling

According to standardized terminology, milling is defined as a machining process characterized by a rotating multi-edge tool performing a circular primary motion, combined with a feed motion that is perpendicular or inclined relative to the tool axis (ISO . . . , 1984). This superposition of rotational and translational motion produces complex tool trajectories.

For constant feed motion perpendicular to the spindle axis, the path of an individual cutting edge follows a cycloidal trajectory, as originally described by Reuleaux (1900) and shown in Figure 2.

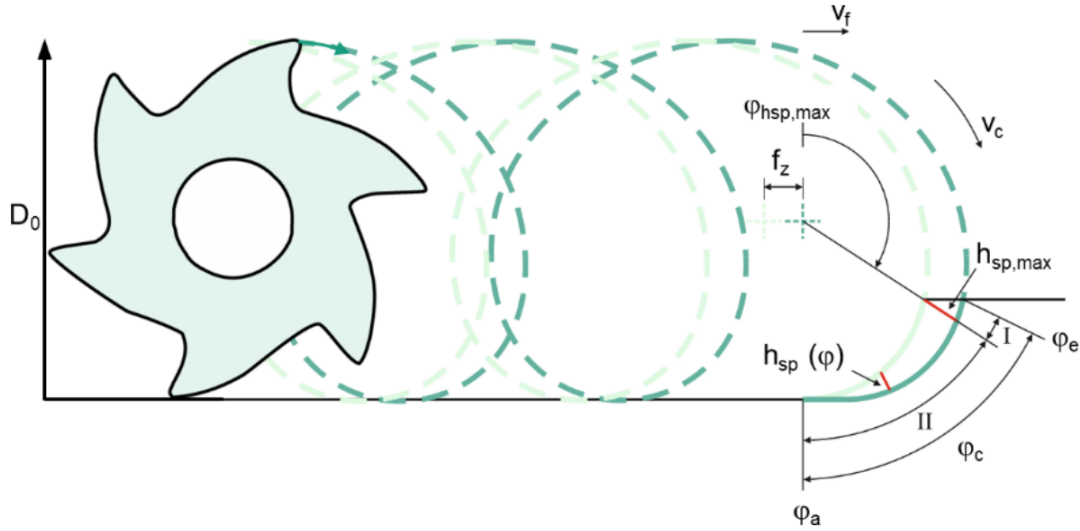
The rotational motion is described by the cutting speed v_c , whereas the translational motion is defined by the feed speed v_f . The feed speed can be expressed as:

$$v_f = nN_t f_z \quad (1)$$

where n denotes spindle speed, N_t the number of teeth, and f_z the feed per tooth.

The cutting speed depends on the tool diameter d_{tool} and spindle speed n :

Figure 2 – Uncut chip thickness and cycloidal trajectory in 3-axis milling.



Source: Adapted from (Reuleaux, 1900; Cabral, 2015).

$$v_c = \pi d_{\text{tool}} n \quad (2)$$

The angular immersion of a cutting edge is described by the contact angle φ_c , defined as

$$\varphi_c = \varphi_e - \varphi_s \quad (3)$$

with φ_s and φ_e representing entry and exit angles, respectively (Meinecke, 2009).

Assuming the tool radius r is significantly larger than the radial depth of cut ($r \gg a_e$), analytical expressions for entry and exit angles can be derived (Markworth, 2005):

$$\varphi_s = \frac{\pi}{2} - \arcsin \left(1 - \frac{a_e}{r} \right) \quad (4)$$

$$\varphi_e = \frac{\pi}{2} + \arcsin \left(1 - \frac{a_e}{r} \right) \quad (5)$$

For high cutting speeds relative to feed motion ($v_c \gg v_f$), the edge trajectory can be approximated as circular (Martellotti, 1945). Under this condition, the instantaneous uncut chip thickness varies sinusoidally with the angular position.

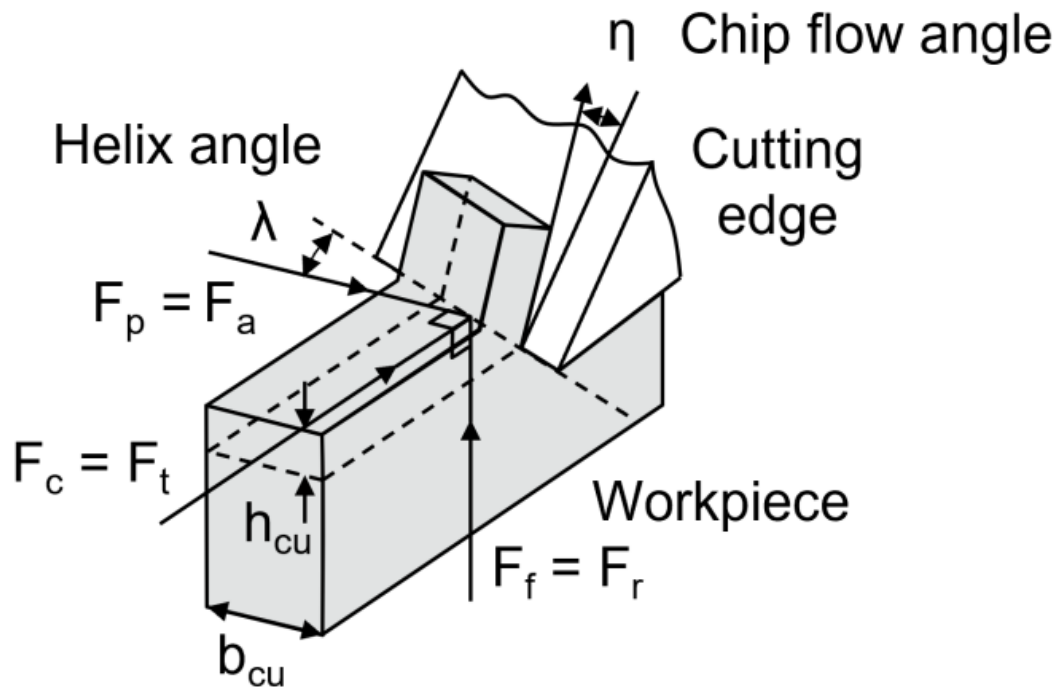
Two principal configurations are distinguished in milling: up-milling and down-milling. These differ in chip formation sequence, direction of the resultant force, and surface generation timing (Klocke, 2018). Down-milling is often preferred in aerospace

applications due to improved dimensional stability and reduced risk of undercutting (Altintas, 2012; Gey, 2003).

From Orthogonal to Oblique Cutting

Although orthogonal cutting simplifies analysis, practical milling processes are inherently oblique due to helix angles and edge inclination. In orthogonal cutting, the inclination angle $\lambda = 0$ and the cutting edge angle $\kappa = 90^\circ$ (Klocke, 2018). Oblique cutting introduces a non-zero inclination angle, resulting in a three-dimensional force system (Altintas, 2012).

Figure 3 – Classical Oblique Cutting Geometry.



Source: Adapted from Phillip Ganser (2025).

In oblique cutting, the passive force component becomes significant and corresponds to the axial force F_a , while the tangential and radial forces relate to cutting and normal force components. According to Stabler's chip flow rule (Stabler, 1964), the chip flow angle approximates the inclination angle.

2.2.2 Cutter-Workpiece Engagement

Milling is characterized by time-varying engagement conditions due to the rotating helical cutting edges. The geometric contact between tool and workpiece evolves continuously during cutter rotation. Understanding these engagement conditions is essential for force modelling and process optimization (Kienast, 2021).

Macroscopic engagement considers the overall geometric penetration of the tool envelope into the workpiece, whereas microscopic engagement describes local chip thickness variation along the cutting edge (Meinecke, 2009).

For small feed per tooth ($f_z \ll r$), the instantaneous uncut chip thickness can be expressed as (Altintas, 2012):

$$h_{sp}(\varphi_j, \kappa) = f_z \sin(\varphi_j) \sin(\kappa) \quad (6)$$

The differential chip width is defined as

$$db = \frac{dz}{\sin \kappa} \quad (7)$$

The product of h_{sp} and db yields the differential chip cross-sectional area, which forms the basis of mechanistic force models (Altintas; Lee, 1996).

2.2.3 Engagement Simulation

Virtual machining requires accurate determination of cutter-workpiece engagement along the tool path (Altintas *et al.*, 2014). Most industrial approaches rely on macroscopic engagement metrics such as chip volume and immersion angle. However, high-fidelity prediction of undeformed chip geometry in multi-axis machining remains challenging.

Hybrid modelling approaches combine dixel-based macroscopic engagement with analytical microscopic chip reconstruction and semi-empirical force models (Cabral, 2015; Ganser, Phillip, 2025). Such integrated methods enable simulation of five-axis milling of free-form surfaces and support digital twin frameworks.

2.2.4 Process Simulation and Digital Twinning

Digitalization has become a driving force in advanced manufacturing sectors such as aerospace, automotive, medical engineering, and mold & die production. A digital twin represents a virtual counterpart of a physical manufacturing system and allows monitoring, analysis, and prediction of process behavior (Ganser, Philipp *et al.*, 2021; Ladj *et al.*, 2020).

An example is the dPart® framework developed at the Fraunhofer IPT and RWTH Aachen University, integrating CAD/CAM/CAE environments with physics-based technology models (Ganser, Philipp *et al.*, 2021). Within such environments, simulations of engagement, cutting forces (Kienzle/Altintas), structural deflection, and dynamic stability are coupled iteratively.

Thin-walled components are particularly sensitive to static deflection and regenerative chatter (Rudel *et al.*, 2022; Liu, D. *et al.*, 2021). Finite element-based modal

updating methods have therefore been integrated into digital twins to account for changing structural dynamics during material removal (Maslo *et al.*, 2020; Maslo, 2022).

Accurate force modelling remains central to predicting tool deflection, workpiece deformation, chatter stability, and tool wear (Budak; Altintas, 1993; Ratchev; Liu, S.; Becker, 2005; Koike; Matsubara; Yamaji, 2013; Si; Wang, L., 2019). A precise estimation of cutter–workpiece interaction is thus indispensable for knowledge-based process planning and optimization (Ganser, Phillip, 2025).

2.3 VIBRATIONS IN MILLING

According to DIN 1311-1, vibrations are defined as temporal variations of a system state variable that alternate between increasing and decreasing values. The term may also encompass transient phenomena such as impacts and creep processes in a broader interpretation (DIN. . . , 2000). Within milling operations, the relevant phenomena are mechanical vibrations, which can be described through state variables such as displacement, velocity, or angular motion of structural components (Kuttner, 2015).

During machining, vibrations originate from the dynamic interaction between the cutting process and the structural compliance of the machine-tool-workpiece system. Their presence directly affects dimensional accuracy, surface integrity, and tool life. A systematic understanding of vibration mechanisms is therefore fundamental for stable and productive milling processes.

2.3.1 Occurrence of Vibrations in Machine Tools

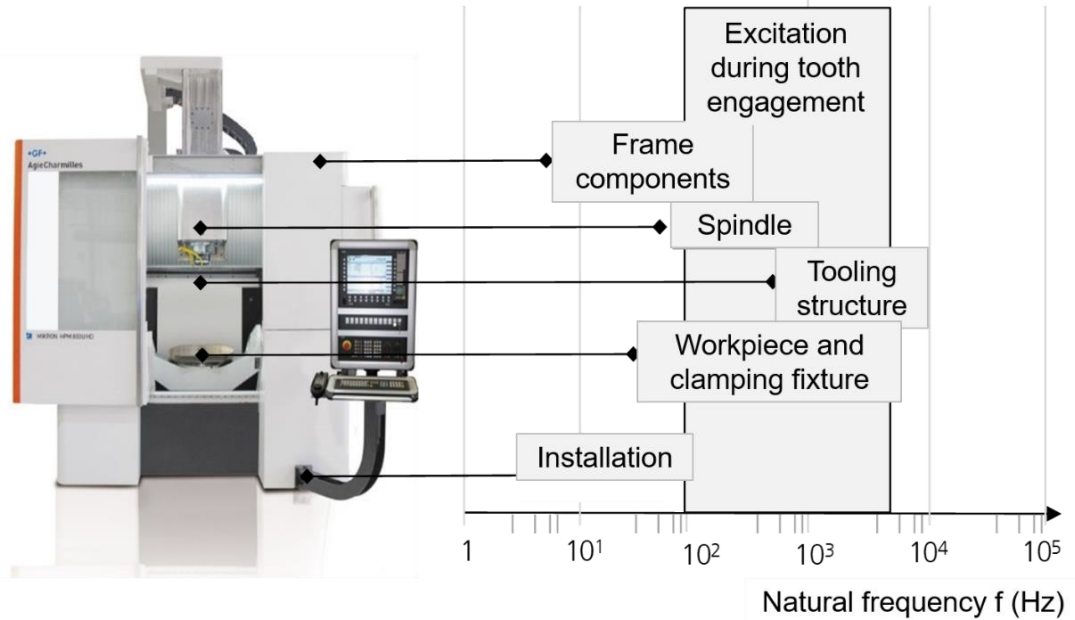
Every structural element of a milling machine exhibits inherent natural frequencies determined by its mass and stiffness distribution. During milling, periodic excitation forces are introduced by the rotating cutter teeth at the spindle speed and its harmonics. If these excitation frequencies approach the natural frequencies of structural components, resonance amplification may occur.

The machine tool forms a closed structural loop consisting of the workpiece, tool, tool holder, spindle, bearings, and machine frame. The excitation forces propagate through this loop, affecting each component to varying degrees. Elements located near the cutting zone - particularly the workpiece, cutting tool, and clamping system - experience the highest dynamic loads. Components further removed from the cutting interface are subjected to attenuated vibrations due to structural damping and compliance.

Figure 4 provides an overview of typical frequency ranges associated with the main structural constituents of a milling system. The tooth passing frequency during milling typically lies between 100 and 2500 Hz. Spindle natural frequencies generally occur in the range of 100-1000 Hz, while machine frames exhibit lower natural frequencies. Consequently, structural design must ensure that operational excitation frequencies do

not coincide with dominant eigenfrequencies.

Figure 4 – Overview of natural frequencies in a milling machine.



Source: Adapted from (Maslo, 2022).

From a mechanical standpoint, system behavior can be separated into static and dynamic responses. Static behavior describes elastic deformation under constant loads, such as gravitational forces. Dynamic behavior arises from time-varying loads, including periodically varying cutting forces during machining (Weck, 2006a).

A key static parameter is structural stiffness, which depends on material properties (Young's modulus), geometry (moment of inertia, overhang length), and boundary conditions. Under constant loading, this characteristic is referred to as static stiffness.

2.3.2 Types of Vibrations

In vibrating tool-workpiece systems, stiffness is frequency-dependent. This frequency-dependent characteristic is commonly termed dynamic stiffness. The dynamic response of a mechanical system is governed by three primary parameters:

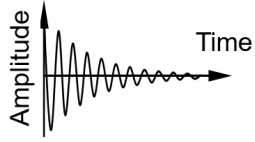
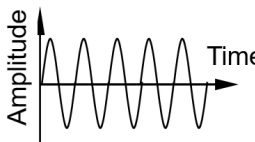
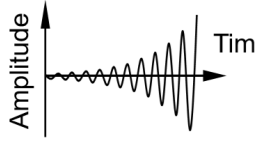
- Mass m and its spatial distribution,
- Stiffness k ,
- Damping D (Lehr's damping) (Weck, 2006b).

Based on their origin, mechanical vibrations in milling are categorized into:

1. Externally excited vibrations,

2. Self-excited vibrations.

Figure 5 – Classification of vibrations based on their temporal behavior.

	Externally excited vibrations		Self excited vibrations “Chatter”
	Free vibrations	Forced vibrations	
Excitation	Impulse	Periodic	Periodic
Deflection	Decays with natural frequency 	Dynamically stable and constant with excitation frequency 	Dynamically unstable and vibrations occur with natural frequency 

Source: Adapted from (Weck, 2006b, 2006a).

Externally Excited Vibrations

Externally excited vibrations are generated by forces imposed on the system from either the environment or the machining process itself (Litwinski, 2010). Depending on excitation characteristics, they can be classified as free or forced vibrations.

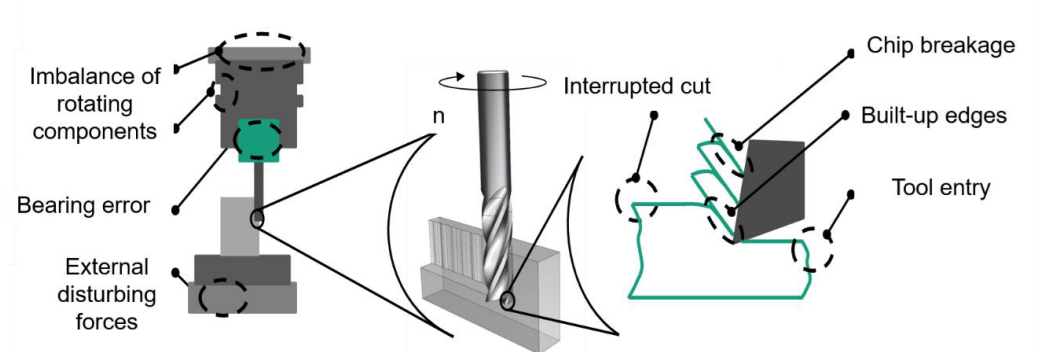
Free vibrations occur after short impulse-like disturbances. Following excitation, the system oscillates at its natural frequency while the amplitude decays exponentially due to damping. Examples include sudden axis accelerations, foundation disturbances, or interrupted cuts. Such vibrations can be mitigated through damping measures in machine foundations (Baur, 2013).

Forced vibrations arise from periodic excitation, such as tooth entry impacts, imbalance of rotating components, or bearing imperfections (Weck, 2006a). The system responds at the excitation frequency rather than its natural frequency. However, when excitation approaches resonance conditions, significant amplitude amplification occurs. Parameter adjustments (spindle speed, feed rate, depth of cut) can prevent transition toward instability.

Self-excited Vibrations

Self-excited vibrations develop when energy input from the cutting process exceeds structural damping capacity (Waibel, 2013). Unlike forced vibrations, no external

Figure 6 – Causes of externally excited oscillations in milling.



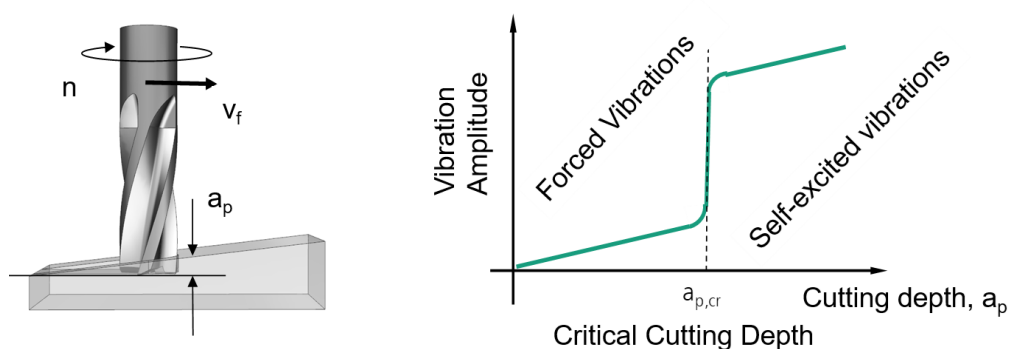
Source: Adapted from (Weck, 2006b, 2006a).

periodic forcing is required. Instead, instability originates from internal feedback between tool motion and chip formation.

The most prevalent form is regenerative chatter. It arises from the modulation of chip thickness caused by surface waviness generated in previous tooth passes. If process damping is insufficient, oscillation amplitudes grow exponentially until reaching a limit cycle.

The depth of cut a_p plays a decisive role. For constant spindle speed, increasing a_p leads to proportional growth in vibration amplitude. At the critical depth of cut $a_{p,cr}$, the system transitions from stable to unstable behavior (Figure Figure 7).

Figure 7 – Oscillation amplitude as a function of depth of cut.



Source: Adapted from (Maslo, 2022).

2.3.3 Regenerative Effect

The regenerative mechanism can be described as a feedback loop between structural deflection and chip formation. Cutting forces deform the tool-workpiece sys-

tem, producing surface undulations. During the subsequent tooth engagement, the instantaneous chip thickness depends on both current and delayed displacements (Akesson, 2009).

The dynamic chip thickness is expressed as:

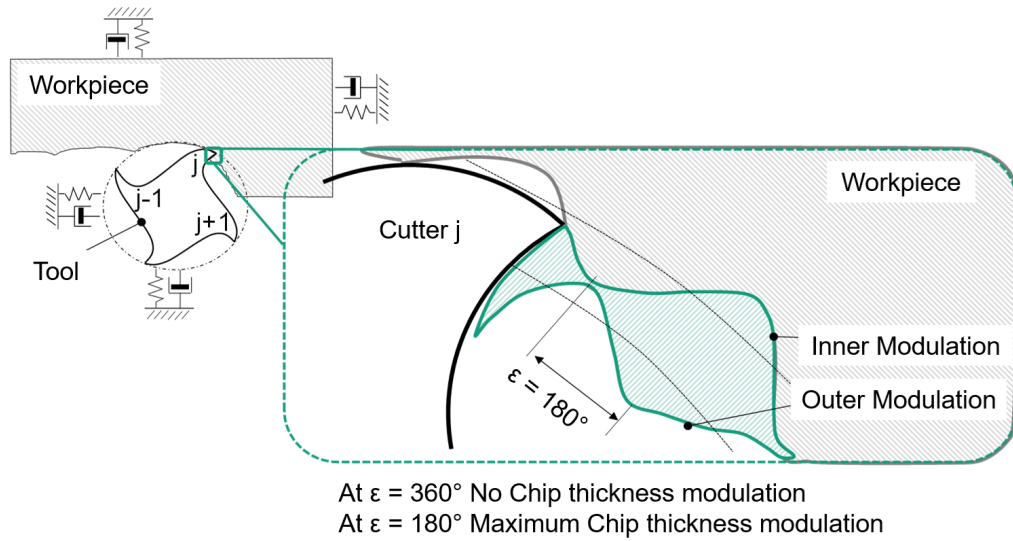
$$h(t) = h_0(t) + x(t - \tau) - x(t) \quad (8)$$

where τ denotes the tooth passing period. The phase difference between $x(t)$ and $x(t - \tau)$ determines stability. At a phase shift of 360° , chip thickness remains constant and the process is stable. At 180° , modulation is maximal and instability may occur (Weck, 2006b).

The regenerative feedback term can be expressed in frequency domain form as (Weck, 2006b):

$$G_p = (e^{-i\omega T_t} - 1)K_c a_p \quad (9)$$

Figure 8 – Regenerative feedback mechanism in milling.



Source: Adapted from (Altintas *et al.*, 2014).

2.3.4 Possibilities for Vibration Reduction

The dynamic behavior of the milling system can be represented by a delay differential equation:

$$m\ddot{x} + d\dot{x} + kx(t) = F_s + K_t a_p A(t)(x(t) - x(t - \tau)) + F_{pd}(t, \dot{x}(t)) \quad (10)$$

The right-hand side comprises the stationary periodic force F_s , the regenerative component, and the process damping force. Based on this formulation, several mitigation strategies can be derived:

Enhancement of Process Damping

Process damping arises from ploughing effects and velocity-dependent mechanisms at the tool flank (Feng *et al.*, 2018). Increasing flank contact or using serrated or variable-pitch tools can delay chatter onset. However, process damping is primarily effective at low frequencies.

Increase of Structural Stiffness

Improving system stiffness shifts natural frequencies and reduces compliance. This can be achieved by optimizing geometry, minimizing overhang, increasing cross-sectional inertia, or adding support elements. Unlike process damping, stiffness improvements influence the entire speed range.

Disruption of Regenerative Feedback

Modulation of spindle speed disturbs phase synchronization between successive cuts. Variable pitch cutters and active control strategies further suppress instability. Sensors such as accelerometers and dynamometers enable in-process monitoring (Zhang; Sims, 2005). Nevertheless, practical limitations exist due to spindle acceleration constraints and transient responses (Schmitz; Smith, 2009).

2.3.5 Mathematical Description of Vibration Characteristics

Machine tools are modeled as multi-degree-of-freedom systems, often approximated by decoupled single-mass oscillators.

$$m\ddot{x} + d\dot{x} + k(x_{stat} + x_{dyn}) = F_{stat} + F_{dyn} \quad (11)$$

Static compliance is defined as:

$$g = \frac{1}{k} \quad (12)$$

The natural frequency is:

$$\omega_n = \sqrt{\frac{k}{m}} \quad (13)$$

and

$$f_n = \frac{\omega_n}{2\pi} \quad (14)$$

For underdamped systems ($D < 1$):

$$x(t) = Ae^{-D\omega_n t} \sin(\omega_d t + \phi) \quad (15)$$

The Frequency Response Function (FRF) is given by:

$$G(j\omega) = \frac{1/k}{1 - \frac{\omega^2}{\omega_n^2} + 2jD\frac{\omega}{\omega_n}} \quad (16)$$

The FRF provides dynamic compliance, resonance frequencies, and damping characteristics (Kuttner, 2015). Near resonance, damping dominates; at low frequencies stiffness governs; at high frequencies mass effects prevail (Braña *et al.*, 2013).

Stability boundaries can be derived using the real part of the FRF:

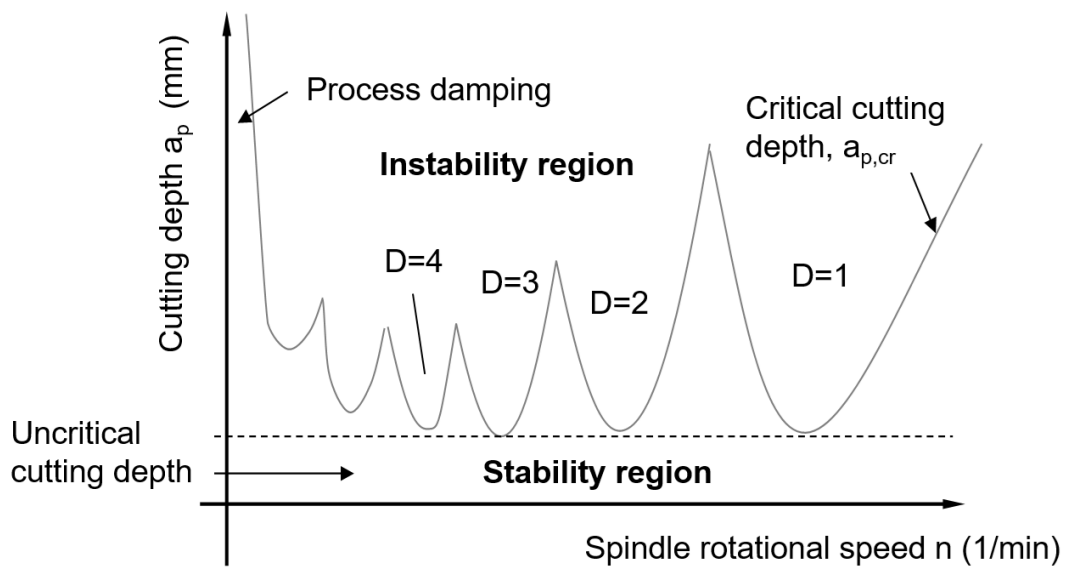
$$a_{p,cr} = -\frac{1}{2K_c \text{Re}(G_m)} \quad (17)$$

The corresponding spindle speed is:

$$n = \frac{60f}{z \left(\frac{1}{\pi} \tan^{-1} \left(-\frac{\text{Im}(G_m)}{\text{Re}(G_m)} \right) + D \right)} \quad (18)$$

Plotting these limits yields the stability lobe diagram, which separates stable and unstable cutting parameter combinations (Figure 9) (Brecher; Weck, 2019).

Figure 9 – Stability lobe diagram.



Source: Adapted from (Maslo, 2022).

2.4 FUNDAMENTALS OF MACHINE LEARNING AND DEEP LEARNING

Machine learning Machine Learning (ML) is a subfield of artificial intelligence focused on developing algorithms capable of learning patterns from data and making predictions without being explicitly programmed for specific tasks. This technique has been successfully applied across a wide range of domains, from computer vision and natural language processing to healthcare and scientific simulations. The choice of learning approach or model, depends on the nature of the problem and the structure of the available data.

2.4.1 Supervised, Unsupervised, and Reinforcement Learning

Machine learning tasks are typically categorized based on the type of supervision available during training, this is usually decided based on the available training data. In supervised learning, the model is trained on labeled data, where each input is paired with the correct output. This is the most common setting and includes tasks such as classification and regression. In contrast, unsupervised learning deals with unlabeled data, aiming to uncover hidden patterns or structures, such as in clustering or dimensionality reduction. Reinforcement learning represents a different paradigm where an agent learns to make decisions through interactions with an environment, receiving feedback in the form of rewards or penalties. Understanding these learning types is essential to situate more advanced models, such as GNNs, in the broader machine learning landscape.

Table 3 summarizes the main learning paradigms, their supervision mechanisms, and representative application domains, providing context for the supervised graph-based regression approach adopted in this work.

Table 3 – Comparison of machine learning paradigms and representative use cases.

Learning Paradigm	Training Signal	Representative Use Cases
Supervised Learning	Labeled input–output pairs	Image classification and regression; structural response prediction from FEM simulations; node-level regression in graph neural networks
Unsupervised Learning	Unlabeled data	Clustering and dimensionality reduction; graph representation learning; anomaly detection in sensor and manufacturing data
Reinforcement Learning	Reward signal from environment interaction	Robotics and control; game playing; adaptive process optimization and decision-making problems

Source: Adapted from Bishop (2006), LeCun, Bengio, and Hinton (2015) and Sutton and Barto (2018).

Training a neural network involves finding the optimal set of parameters (weights and biases) that allow the model to perform well on a given task. This is achieved by

minimizing a loss function, which computes the error between the model's predictions and the desired output. For example, in regression tasks, the mean squared error loss function (MSE) is commonly chosen, while in classification tasks, the cross-entropy loss is preferred. The loss function serves as a guide for updating the model's parameters through an optimization algorithm.

The optimization process relies on backpropagation, a method for computing the gradient of the loss function with respect to each parameter in the network. These gradients indicate the direction in which each parameter should be adjusted to reduce the loss. The core computational unit of most neural networks is the Multilayer Perceptron (MLP), a stack of fully connected layers interleaved with nonlinear activation functions such as ReLU or sigmoid. During training, each layer in the MLP contributes to the overall computation of the loss, and each parameter is updated iteratively based on its gradient contribution. This training framework is central to all neural network architectures, including Graph Neural Networks, which use similar principles but apply them to a graph-based dataset.

2.5 GRAPH NEURAL NETWORKS

Traditional machine learning models, like neural networks, often rely on fixed-size input representations, such as vectors or grids, which work well for structured data like images or time series. However, many real-world systems involve complex, relational, and irregular structures that cannot be naturally captured in these formats. As a result, these models often struggle to generalize when tasked with reasoning about structured and relational data.

Convolutional Neural Networks Convolutional Neural Networks (CNNs), which have achieved remarkable success across many scientific and engineering applications, face similar challenges. Despite their power, CNNs are fundamentally limited by their reliance on structured, uniform grids. Traditional CNN architectures operate on domains where there is an underlying euclidian structure (Bronstein *et al.*, 2017), making them not well suited for problems involving complex or irregular geometries—such as those encountered in mesh-based simulations of physical systems. In such cases, applying CNNs requires embedding the geometry into a dense uniform grid or interpolating the data to fit the grid structure. Both approaches introduce significant trade-offs, like increase in computational costs or sacrifices resolution and accuracy in regions of interest. To overcome these fundamental limitations, graph-based methods such as Graph Convolutional Networks (GCNs) have been proposed (Kipf; Welling, 2017; Defferrard; Bresson; Vandergheynst, 2017). Unlike CNNs, GCNs operate directly on irregular graph structures and naturally adapt to the topology of the data.

To better handle these data structures, GNNs have emerged as a powerful class

of models by incorporating relational inductive biases (Battaglia *et al.*, 2018), which is the architectural assumptions that guide learning about entities and their relationships. GNNs operate directly on graph-structured data, enabling computations over nodes (entities), edges (relations) and global attributes. This relational reasoning framework allows GNNs to model complex systems such as molecules, physical simulations, and social networks, where the interactions between components are as critical as the components themselves.

2.5.1 Graphs

Graphs are a flexible and powerful data structure used to model relationships between entities. Formally, a graph consists of a set of vertices representing entities and a set of edges representing the connections or relationships between them, commonly denoted as $G = [V, E]$ (Diestel, 2017). Unlike regular Euclidean data such as images or time series, graph data is inherently irregular and unordered, making it ideal for modeling complex systems where interactions are important, non-uniform and topologically rich. This structure appears naturally in many real-world applications—from social networks and molecules to physical simulations and transportation systems—where entities are best understood through their interconnections rather than their positions in a grid. Graph data also enables the encoding of both local and global relationships, offering a flexible framework for a wide array of tasks.

This diversity of applications also implies different ways of expressing and characterizing graphs. It can be used to represent static or dynamic systems, directed or undirected interactions, and homogeneous or heterogeneous node and edge attributes. Because the connectivity between graph elements is relevant to the problem environment, the need for a machine learning model that is able to capture it grew, motivating the development of Graph Neural Networks, a family of deep learning models designed specifically to operate on graph structures through message passing mechanisms. GNNs have shown good performance in areas as varied as chemistry, weather prediction, particle physics, recommendation systems, and simulation of physical processes (Sanchez-Gonzalez *et al.*, 2020). Their ability to aggregate information from a node’s neighborhood and generalize across different graph topologies makes them particularly suited for tasks involving relational reasoning and structured data.

Table 4 presents representative benchmark datasets commonly used in graph learning research, illustrating the diversity of graph structures, scales, and learning tasks supported by modern GNN frameworks.

Among this diverse datatype, an increasingly important area of research involves their use with mesh-based datasets, particularly in scientific computing, physical simulation, and engineering design, where the domain is often discretized using a mesh. Mesh-based graphs have its nodes correspond to mesh elements and the edges repre-

Table 4 – Common benchmark graph datasets available in PyTorch Geometric.

Dataset	Graph Type	Approx. Size		Typical Task / Domain
Cora	Directed, homogeneous	2.7k nodes,	5.4k edges	Node classification (citation network)
CiteSeer	Directed, homogeneous	3.3k nodes,	4.7k edges	Node classification (citation network)
PubMed	Directed, homogeneous	19.7k nodes,	44k edges	Node classification (biomedical citations)
MUTAG	Undirected, homogeneous	188 graphs	(~18 nodes/graph)	Graph classification (molecular structures)
ZINC	Undirected, homogeneous	~250k graphs		Graph regression (molecular properties)
QM9	Undirected, homogeneous	~134k graphs		Graph regression (quantum chemistry)
OGBN-Arxiv	Directed, homogeneous	169k nodes,	1.1M edges	Large-scale node classification

Source: Adapted from the PyTorch Geometric documentation (Fey; Lenssen, 2023).

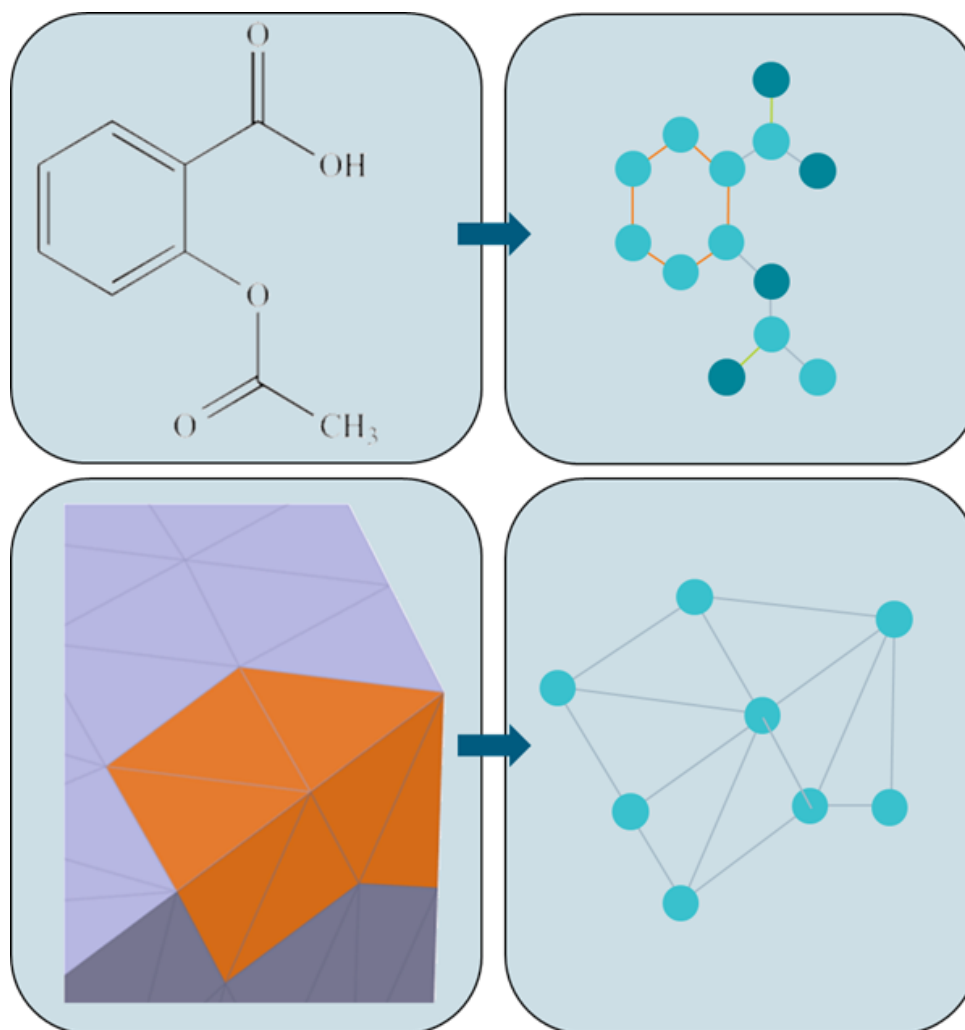
sent topological connections between them, thus defining the graph’s connectivity. This transformation enables the application of GNNs directly to the underlying geometric and physical structure of the domain. As demonstrated in recent works by Brandstetter, Worrall, and Welling (2023) and Gao, Zahr, and Jian-Xun Wang (2022), mesh-based GNNs can learn to approximate the solution of time-independent PDEs with high fidelity by encoding both geometric information and physical parameters into the graph structure. Approaches such as MeshGraphNets (Pfaff *et al.*, 2021) and its extensions build on this concept, allowing for efficient and generalizable learning across varying geometries, resolutions, and boundary conditions. These mesh-to-graph transformations have enabled GNNs to become powerful surrogates for traditional numerical solvers, offering advantages in speed, scalability, and adaptability.

In practice, graph-structured data often originates from the transformation of other data representations into nodes and edges. For example, molecular structures can be mapped to graphs by representing atoms as nodes and chemical bonds as edges, while mesh-based domains can be converted into graphs by associating nodes with mesh vertices or elements and edges with their topological connectivity. Figure 10 illustrates examples of data-to-graph conversions.

2.5.2 Applications of Graph Neural Networks

In the wide context of machine learning, the structure of the model, its objective and the dataset it is using all influence the design and application of the developed model. When the data is best represented as a graph, where relationships and structure play a central role, traditional models fail to achieve a suitable result, motivating the development of specialized approaches. Graph Neural Networks have rapidly evolved into a key framework for learning on graph-structured data, addressing the limitations of

Figure 10 – Example of data to graph conversion.



Source: Author.

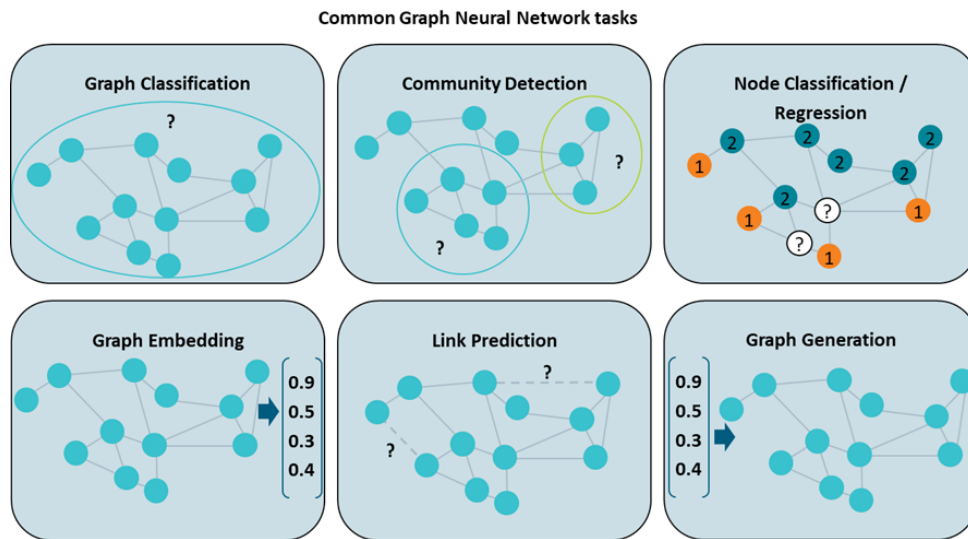
traditional machine learning models when applied to non-Euclidean domains. Inspired by classical neural network architectures such as CNNs and RNNs, GNNs generalize these paradigms by incorporating relational inductive biases to model interactions between entities and their relationships. The field now encompasses a variety of specialized architectures, each developed to address specific tasks, typically at the node level, edge level, or graph level. Beyond these, more advanced approaches support subgraph-level reasoning (Ying *et al.*, 2019), dynamic or spatio-temporal graph analysis, and hierarchical GNNs that capture multi-scale structure.

Node-level tasks such as classification (Velickovic *et al.*, 2018; Li; Han; Wu, 2018; Kipf; Welling, 2017) and regression tasks (Bayram, 2021) aim to predict properties of individual nodes within a graph, often using partial label information. Edge-level tasks include link prediction and edge classification (Lo *et al.*, 2022), which are used to infer missing connections or label interactions between entities. At a higher level, graph-level

tasks such as graph classification or regression (Yang *et al.*, 2023; Ling *et al.*, 2023) involve learning a holistic representation of an entire graph to predict outcomes such as molecular activity, structural properties, or behavior of dynamic systems. These tasks are often supported by architectural components like pooling and readout layers that aggregate node-level features into a graph-level representation. The versatility of GNNs in handling these tasks has contributed significantly to their widespread adoption across both academic research and industrial applications.

Figure 11 illustrates some of the most common learning tasks addressed by graph neural networks.

Figure 11 – Examples of common GNNs tasks.



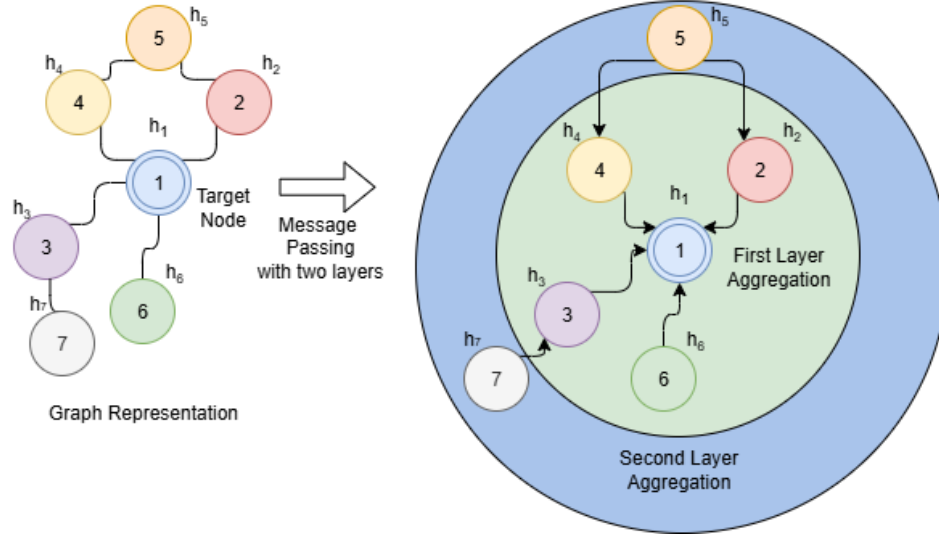
Source: Adapted from (DataCamp, 2021).

2.5.3 Message Passing Mechanism

At the core of most modern Graph Neural Networks lies the message passing method, a general framework for learning representations of graph-structured data. The central idea is that the feature vector of a node is iteratively updated by aggregating information from its neighbors, thereby capturing local topology and feature-level interactions. This process typically unfolds over multiple layers (or time steps), enabling information to propagate across the graph.

This mechanism is the main motivator for this project's case study, its success in problems involving physical applications indicates that its application could be a good solution for the problem proposed in chapter 4.

Figure 12 – Message passing with aggregation example.



Source: Author.

General Formulation

Formally, consider a graph $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. Let $h_v^{(k)} \in \mathbb{R}^d$ denote the feature vector of node v at layer k , and let $\mathcal{N}(v)$ denote the set of neighbors of node v . The general message passing framework at layer k can be expressed in three steps:

1. Message computation:

$$m_{uv}^{(k)} = \phi_m \left(h_v^{(k)} u[k-1], h_v^{(k)} v[k-1], e_{uv} \right) \quad (19)$$

2. Aggregation:

$$m_v^{(k)} = \text{AGGREGATE}^{(k)} \left(\left\{ m_{uv}^{(k)} : u \in \mathcal{N}(v) \right\} \right) \quad (20)$$

3. Update:

$$h_v^{(k)} = \phi_u \left(h_v^{(k)} v[k-1], m_v^{(k)} \right) \quad (21)$$

This general formulation follows the Message Passing Neural Network framework proposed by Gilmer *et al.* (2017). Here, ϕ_m is a message function that may optionally use edge features e_{uv} , and ϕ_u is the update function, often implemented as a MLP.

Aggregation Functions

The aggregation step must be permutation invariant (Zaheer *et al.*, 2018), meaning the result should not depend on the order in which neighbors are processed. Common aggregation functions include:

- **Mean Aggregation:**

$$m_v^{(k)} = \frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} h_v^{(k)} u[k-1] \quad (22)$$

- **Sum Aggregation:**

$$m_v^{(k)} = \sum_{u \in \mathcal{N}(v)} h_v^{(k)} u[k-1] \quad (23)$$

- **Max Aggregation:**

$$m_v^{(k)} = \max_{u \in \mathcal{N}(v)} h_v^{(k)} u[k-1] \quad (24)$$

- **Attention-based Aggregation:** (Velickovic *et al.*, 2018)

$$m_v^{(k)} = \sum_{u \in \mathcal{N}(v)} \alpha_{uv}^{(k)} h_v^{(k)} u[k-1] \quad (25)$$

where the attention weights $\alpha_{uv}^{(k)}$ are computed as:

$$\alpha_{uv}^{(k)} = \frac{\exp \left(\text{LeakyReLU} \left(a^\top [Wh_u^{(k-1)} \parallel Wh_v^{(k-1)}] \right) \right)}{\sum_{w \in \mathcal{N}(v)} \exp \left(\text{LeakyReLU} \left(a^\top [Wh_w^{(k-1)} \parallel Wh_v^{(k-1)}] \right) \right)} \quad (26)$$

where W and a are learnable parameters, and $\parallel$ denotes vector concatenation.

Layer Stacking

By stacking multiple message passing layers, a GNN can capture information from increasingly distant nodes, effectively increasing the receptive field. However, deep GNNs can suffer from the oversmoothing problem, where node embeddings become indistinguishable across the graph (Li; Han; Wu, 2018; Oono; Suzuki, 2021). Techniques such as residual connections, normalization, and skip connections are commonly used to mitigate this.

Moreover, the expressive power of message passing GNNs is closely tied to the Weisfeiler-Lehman (WL) test for graph isomorphism (Xu *et al.*, 2019). Standard GNNs are provably no more powerful than the 1-WL test in distinguishing certain non-isomorphic graphs. This has inspired the development of more expressive models, such as higher-order GNNs, subgraph-aware architectures, and GNNs with learnable structure.

2.5.4 Similarities between message passing and FEM

A fundamental justification for employing Graph Neural Networks as surrogate models for FEM simulations lies in the structural isomorphism between the finite element mesh and the graph topology. In FEM, continuous physical domains are discretized into a mesh consisting of nodes and elements, a structure that is mathematically equivalent to a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where vertices $\mathcal{V}$ represent mesh nodes and

edges $\mathcal{E}$ encode the connectivity derived from mesh elements. Unlike standard MLPs or CNNs, which often require mapping unstructured meshes onto regular grids, thereby losing geometric fidelity, GNNs operate directly on the unstructured data. This grants the surrogate model a strong geometric inductive bias, allowing it to naturally respect the underlying topology and boundary conditions of the physical system (Pfaff *et al.*, 2021; Gao; Zahr; Wang, J.-X., 2022). Beyond static representation, the iterative message passing mechanism in GNNs shares a striking computational parallel with the numerical solvers used in FEM, specifically the assembly of the global stiffness matrix from local elemental contributions. In classical FEM, the state of a node (e.g., displacement or temperature) is determined by solving equilibrium equations that sum contributions from adjacent nodes via shape functions. Analogously, a message passing neural network updates a node's hidden state by aggregating messages, learned feature vectors, from its immediate neighbors (Gilmer *et al.*, 2017; Pfaff *et al.*, 2021). Stacking multiple GNN layers allows this local information to propagate across the graph, effectively mimicking the iterative steps of a sparse linear solver or the time-stepping of an explicit dynamic analysis, where local forces propagate to affect the global deformation field.

3 PROBLEM DESCRIPTION AND TECHNICAL REQUIREMENTS

This chapter explains the motivation behind the problem of this thesis and how it is expected to integrate a bigger project under development at Fraunhofer IPT. It is composed by a description of the Fraunhofer IPT and the department responsible for the project in which the thesis is contextualized. Under this context, the problem and its requirements are presented, establishing the demands in terms of performance and implementation guidelines.

3.1 FRAUNHOFER-GESELLSCHAFT

The Fraunhofer-Gesellschaft, based in Germany, is one of the world's leading organizations for applied research with 75 institutes and independent research units. It plays a major role in innovation by prioritizing research on cutting-edge technologies and the transfer of results to industry (Gesellschaft, 2025).

The Fraunhofer Institute for Production Technology (IPT) develops system solutions for the networked, adaptive production of sustainable and resource-efficient products and the associated services. Our focus is on process technology, production machines, production quality and metrology, and technology management, and ranges from the fundamentals to the digital transformation of production. The institute develops and refines both new and optimizing methods, technologies, and processes aimed at shaping the future of production. Its work spans from highly specialized individual technologies to integrated system solutions, all with a focus on sustainable manufacturing (IPT, 2025c).

3.1.1 High Performance Cutting Department

The department is concerned with the development of technologies and solutions for machining operations throughout the entire process chain, from process design through real time data collection, to consulting and prototype manufacturing. The benefits of advanced technological expertise, integrated systems capabilities, and state-of-the-art machinery are used to conduct comprehensive, target-oriented research and development activities. Focus is placed on simultaneous multi-axis milling and turning of complex components manufactured from superalloys, ultra-hard steels, lightweight composites, and other specialized materials. These processes find typical application in fields such as turbomachinery, aerospace, toolmaking, and medical technology. Digitalization is considered a fundamental component in achieving a thorough understanding of manufacturing processes and in maintaining a sustainable competitive edge (IPT, 2025b).

DPart

High performance milling of thin-walled geometries has a very challenging nature due to its complexity, requiring highly advanced production environments. These environments cover the planning, manufacturing and quality assurance for the production of modern products, which are characterized by the interaction of multiple different software and hardware components. A digital twin framework has already been established within the company infrastructure, and this project is intended to be integrated into this existing system.

A digital twin is a virtual representation of a physical object, system, or process that mirrors its real-world counterpart in real time. Built using data collected from sensors, simulations, or historical records and is used to monitor, analyze, and optimize performance throughout the system. One key advantage of using a digital twin is the ability to simulate and predict behavior under different conditions without interfering with actual operations. This supports improved decision-making and reduces risks and reworking.

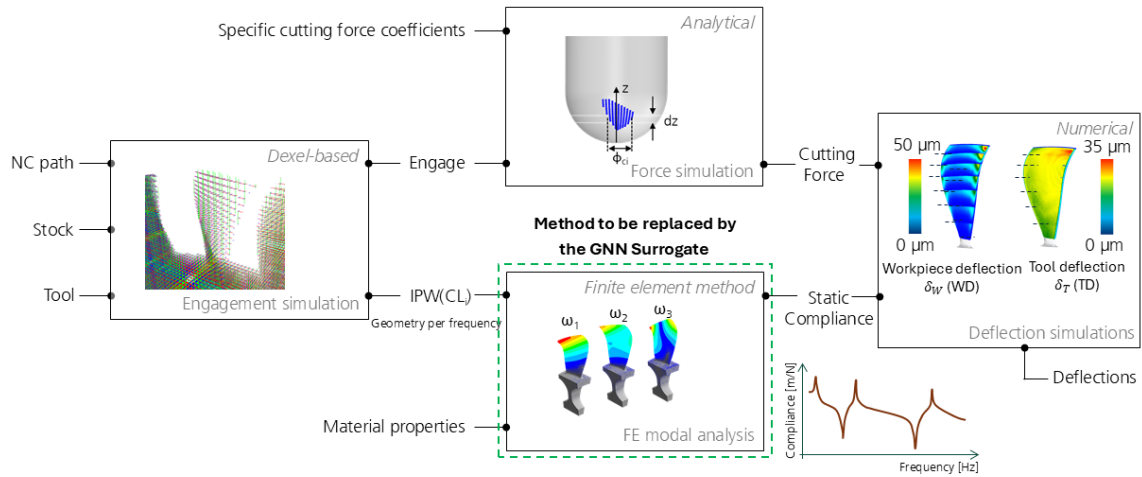
This digital twin environment was named dPart®, which represents a strategic framework as well as an implemented working appliance in the field of process technology at Fraunhofer IPT. The system is being continuously developed since 2018, and is already advancedly implemented regarding technology model microservices such as dexe-based engagement, modal analysis, milling force, tool static and workpiece deflections. The components also include computer-aided manufacturing (CAM) systems for computer-aided process planning, machine tools for carrying out the production task, and coordinate measuring machines for quality assurance (IPT, 2025a).

3.2 PROBLEM DESCRIPTION

In the context to rework, in Turbomachinery, components tend to have higher amplitude of vibrations because of their high aspect ratio and difficult to machine materials. For example, in the manufacturing of blade integrated disks, which is a structurally complex workpiece, the major challenge faced by the industries is the amount of rework. The main challenge faced by manufacturing industries is the monitoring of the milling process which when done effectively could save reworks, which is estimated to be around 25%. Through effective monitoring, vibrations and chatter, one of the major contributors to BLISK rework, could be effectively controlled and stabilised.

Although cutting stability has been studied in detail over the past half-century, vibration predictions are difficult to make due to the continuous changes in stiffness of the thin-walled workpiece. On one hand, there is variation of the force application point due to the different tool positions and on the other hand, a change in stiffness and mass of the workpiece due to material removal. Currently this question is approached

Figure 13 – Dpart pipeline highlighting where the GNN surrogate will be used.



Source: Adapted from (High Performance Cutting Department, 2025).

with finite element method simulations, but they are time-intensive, resource-heavy, and scale poorly with mesh size, and the need to do it for multiple cutter locations increases the time needed to create a good estimation of the workpiece dynamics with reasonable precision at all cutter locations. The FEM process has been optimized to a degree where further improvements are challenging to achieve. Additionally, interpolation fails to capture the complexity of the non-linear nature of partial differential equations whose solutions are approximated, which is a problem that machine learning can excel at.

The objective of this project is to develop a surrogate model for FEM simulations, supported by a pipeline that encompasses all key stages of the workflow. As part of this pipeline, the workflow includes data generation, preprocessing, graph construction and model training.

Figure 13 highlights the participation of this project inside the digital twin environment.

3.2.1 Technical Requirements

The following functional requirements were established for the project:

1. Data Processing and Graph Construction: Acquire and convert mesh and FEM data into graph-based representations. Develop scripts to extract node-level features, generate embeddings, and define graph structures.

2. Graph-Based Machine Learning: Implement various GNN models for surrogate modeling.
3. Model Evaluation: Compare predicted results from machine learning models against Ansys FEM simulation data.
4. Scalability Testing: Apply models to datasets of increasing size or complexity to assess scalability and generalization.
5. Result Visualization: Generate plots or visual outputs to illustrate model predictions.

In addition, there is an expectation that the project will meet the following non-functional requirements:

1. Computational Efficiency: The model must offer faster prediction times than traditional FEM simulations.
2. Accuracy: The model should reach a defined acceptable error margin compared to FEM results, ensuring that approximations are useful.
3. Scalability: The system should be capable of handling large mesh sizes and different cutter locations without increases in runtime or memory usage.

4 PROPOSED SOLUTION AND METHODOLOGY

The objective of this project is to develop a surrogate model for Finite Element Method (FEM) simulations, supported by a structured and modular pipeline encompassing all key stages of the workflow. This pipeline includes data generation, preprocessing, graph construction, model training, and evaluation.

The previous chapter presented the theoretical background and state of the art supporting this work, covering the fundamentals of finite element analysis, machining dynamics, and recent advances in machine learning for physics-based problems. Building upon this foundation, the present chapter details the proposed solution and methodology adopted to implement a Graph Neural Network based surrogate model, ensuring compliance with the technical and integration requirements established in Chapter 3.

The proposed solution aims to serve as an alternative to repeated FEM simulations by learning a data-driven approximation of the structural response of thin-walled workpieces under milling conditions. Rather than replacing FEM entirely, the surrogate model is designed to reproduce equivalent deformation responses with significantly reduced computational cost, enabling faster evaluation within digital twin environments.

To achieve this objective, the methodology reformulates FEM simulation outputs as a graph-based learning problem. Mesh connectivity, nodal geometry, and simulation-derived quantities are encoded as graph features, allowing GNNs to exploit both local neighborhood interactions and global structural topology. The methodology is structured into the following main stages:

1. **Data Generation and Acquisition:** FEM simulations are performed using ANSYS Mechanical to generate high-fidelity modal data. Simulations are executed at multiple toolpath positions corresponding to different material removal states, capturing the evolving dynamic behavior of thin-walled structures. The solver outputs natural frequencies and associated mode shapes, which serve as ground truth for supervised learning.
2. **Preprocessing and Graph Construction:** Simulation outputs are processed using custom scripts to extract mesh connectivity, nodal coordinates, boundary conditions, and deformation fields. These quantities are converted into graph representations, where nodes correspond to mesh vertices and edges encode topological adjacency. Node embeddings are constructed to combine geometric and physical information into a format suitable for GNN training.
3. **Model Design and Training:** Two GNN architectures are implemented and evaluated, a conventional GCN and a MeshGraphNet-inspired encoder-processor-decoder architecture. The training framework supports configurable architectures

and hyperparameters without direct code modification, enabling systematic experimentation. Models are trained to predict nodal displacement fields for selected vibration modes, and performance is assessed against FEM ground truth.

4. **Post-Processing, Validation, and Integration:** Predicted deformation fields are denormalized to physical units and exported in engineering-compatible formats. Quantitative error metrics and qualitative visualizations are used to evaluate prediction fidelity. Outputs are structured to ensure compatibility with the existing Datagen and ANSYS workflows, enabling seamless integration into the dPart® digital twin framework.

A schematic overview of the complete pipeline is shown in Figure 14, illustrating the flow of information from FEM simulations to the trained GNN surrogate and back to engineering formats.

4.1 SCOPE, ASSUMPTIONS, AND CONSTRAINTS

The methodology targets single-part, thin-walled workpieces represented by surface meshes with consistent meshing conventions. Boundary conditions and simulation metadata are assumed to be available at preprocessing time. To constrain problem complexity and ensure feasibility within the project timeline, the analysis is limited to the first two vibration modes, which capture the dominant structural deformation behavior during machining.

Higher-order modes are excluded due to their reduced influence on the overall deformation response. Nevertheless, the proposed pipeline is fully extensible, and the same preprocessing, training, and validation procedures could be applied to additional modes if required.

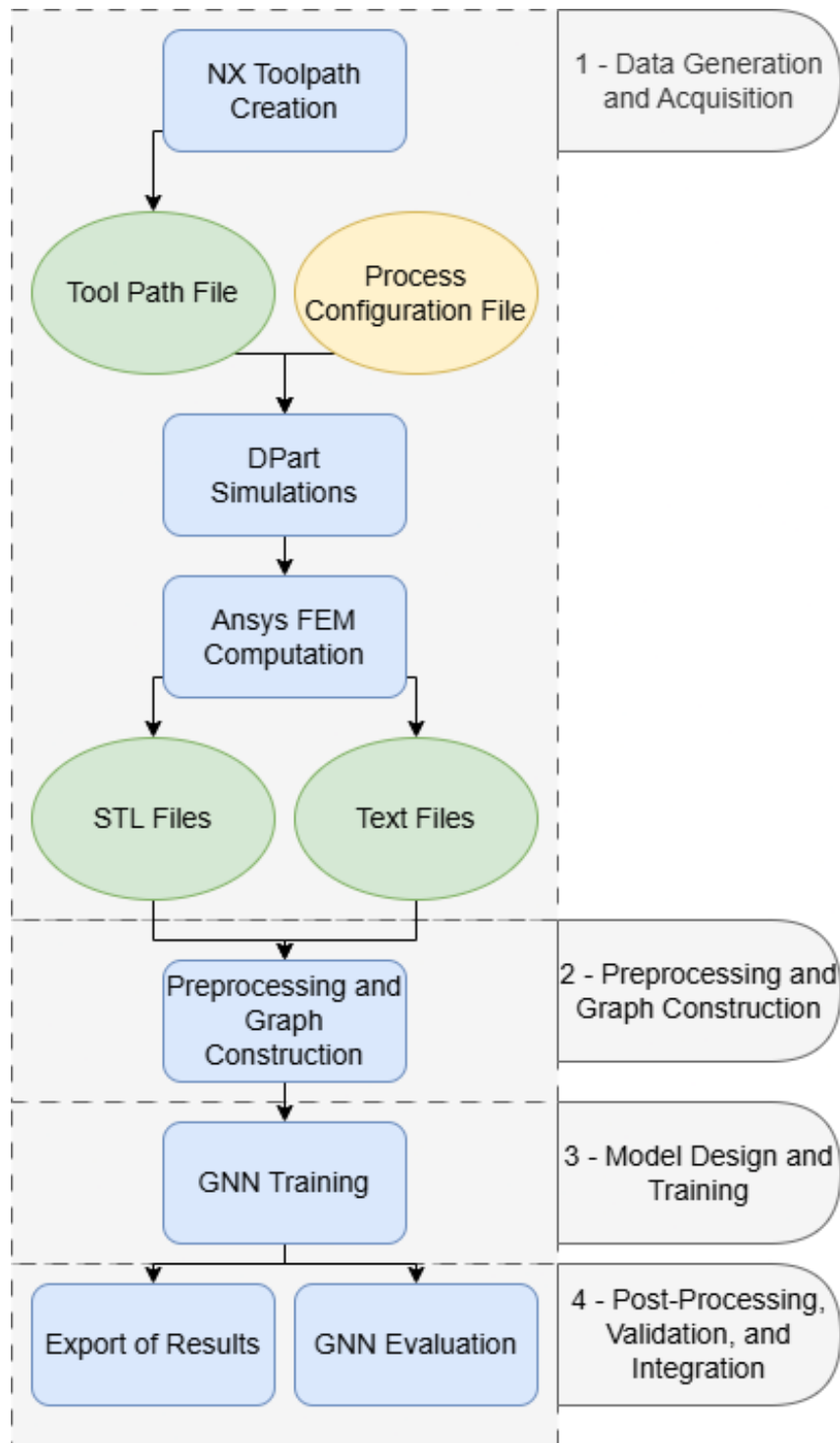
Architectures requiring materially different input representations or training paradigms were excluded to preserve comparability and integration with the existing Datagen framework.

4.2 HOW THE SOLUTION MEETS THE REQUIREMENTS

Technical adequacy:

- Graph representations preserve mesh topology and local neighborhoods critical for structural response prediction.
- Node embeddings encode geometric, physical, and boundary-related information relevant to deformation behavior.
- Normalization strategies reduce scale bias and improve numerical stability during training.

Figure 14 – Overview of the proposed GNN-based surrogate modeling pipeline.



Source: Author.

Computational efficiency: Once trained, the surrogate model produces deformation predictions orders of magnitude faster than full FEM simulations, supporting rapid evaluation across multiple toolpath locations.

Integrability: All intermediate and final data formats (graphs, STL meshes, text-

based nodal fields) are consistent with existing Datagen and ANSYS conventions, minimizing disruption to upstream and downstream processes.

With this presented methodology, the project should achieve a modular and scalable methodology for transforming ANSYS FEM simulation outputs into graph-based datasets and training mesh-aware GNN surrogate models. The design choices regarding data representation, normalization, model architecture, and output formats were guided by accuracy, stability, and integration into existing digital twin workflows.

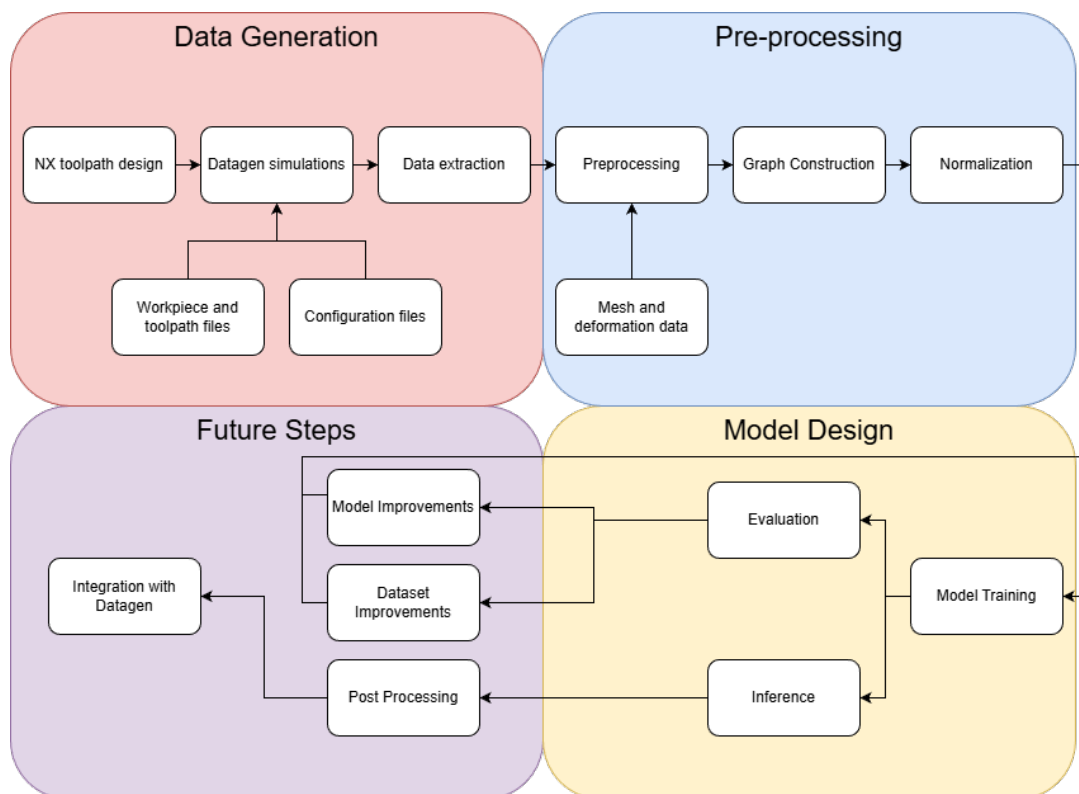
5 DEVELOPMENT AND RESULT ANALYSIS

5.1 SOLUTION DEVELOPMENT

This chapter presents the development and results obtained throughout the implementation of the proposed project. Building upon the concepts presented in Chapter 2 and the requirements introduced in Chapter 3, the following sections describe how the solution was constructed, how the data and models were integrated, and how the performance of the implemented architectures was evaluated.

The development is organized into four stages of the workflow, beginning with the generation of simulation data, followed by preprocessing and graph construction, model training and concluding with evaluation. Each stage plays a critical role in ensuring that the surrogate model can accurately reproduce the deformation behaviour predicted by the finite element method.

Figure 15 – Overview of the proposed project steps.



Source: Author.

To clarify the overall structure of the implementation, Figure 15 illustrates the complete pipeline adopted in this project, highlighting the data flow and the contribution of each component to the final solution and the future steps that are going to be describe in Chapter 6.

5.1.1 Data Generation

The initial stage of the development process consists of generating the geometric and process-related data required for the numerical simulations and learning tasks. The necessary inputs for this stage are the workpiece geometry, the fixed base defining the boundary conditions, and the machining toolpath. Together, these elements define both the physical domain under analysis and the mechanical loading conditions imposed during the milling process.

The core problem addressed in this work is the dynamic behaviour of very thin-walled workpieces during milling. Such components exhibit low structural stiffness, making them highly susceptible to vibrations and large deformation amplitudes. In order to increase these effects, the problem was approached by intentionally increasing the degree of material removal, reducing the structural stiffness of the component. This reduction in stiffness ensures that the deformation patterns are sufficiently pronounced to be captured by both FEM simulations and the surrogate models.

Geometric Modelling and Toolpath definition

The geometric model and toolpath were generated using Siemens NX, a computer-aided design and manufacturing (CAD/CAM) environment. Within NX, the workpiece geometry is defined by specifying the base region, which will be defined as the boundary condition in the simulations, and the machined surface. The surrounding block geometry is designed to replicate the industrial setup used in practice.

Subsequently, a milling toolpath is generated to reproduce the planned manufacturing process. Although the geometry and tool motion correspond to a realistic and simplified industrial scenario, the amount of material removed was increased when compared to the original finishing operation. This was done to highlight the material removal aspect of the operation, increasing the difference between different cutter locations.

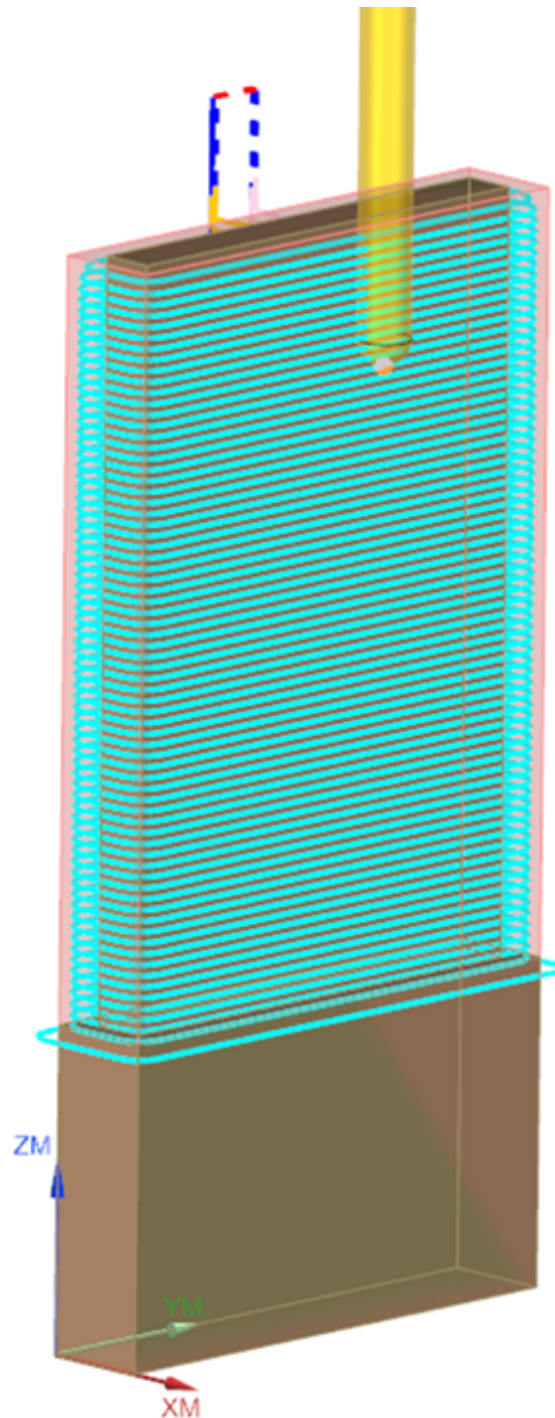
Figure 16 displays the sample toolpath generated for this work, later exported and used as an input for data generation.

Mesh and embeddings extraction

Following the geometric and process definition, the prepared workpiece and toolpath data are introduced into the internal Datagen pipeline, which is based on ANSYS Workbench. This pipeline integrates all simulation stages required to obtain structural dynamic data relevant to the milling process.

The focus of this work is on the ANSYS Mechanical component of the pipeline, where FEM simulations are performed to evaluate the structural response of the workpiece. ANSYS Mechanical is a finite element analysis (FEA) environment designed for

Figure 16 – Example toolpath used for data generation.



Source: Author.

structural mechanics simulations under prescribed boundary and loading conditions, making it well suited for data generation.

The simulations are executed at multiple tool positions along the milling path, capturing variations in dynamic response associated with progressive material removal.

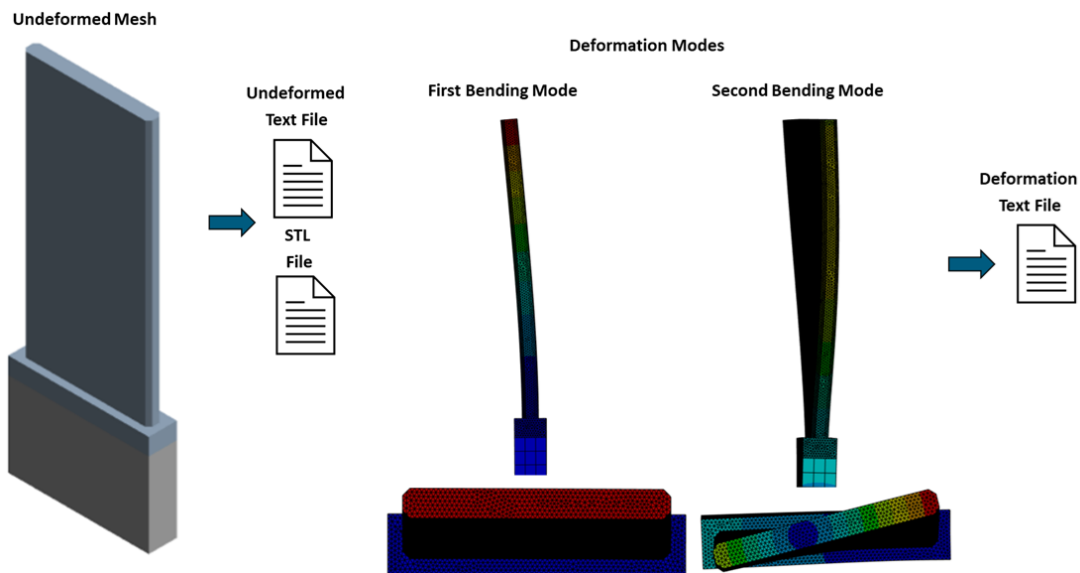
Modal Analysis

The modal analysis is conducted using ANSYS Harmonic, as the project's interest lies in the vibrational characteristics of the workpiece. The simulation outputs a set of eigenfrequencies and their associated mode shapes, which describe the natural vibration behaviour of the structure.

Each eigenfrequency corresponds to a characteristic oscillation frequency of the workpiece, while its associated mode shape represents the spatial deformation pattern across the mesh. These mode shapes are expressed as three-dimensional nodal displacement fields, containing both amplitude and direction of deformation at each mesh node.

In this study, only the first two eigenmodes are considered, as they capture the dominant structural behaviours of the workpiece and are expected to pose different levels of difficulty for the predictive models. Higher-order modes usually contribute less to the overall deformation response and are therefore excluded to restrict the scope of this project. It should be noted, however, that the same data processing, training, and evaluation pipeline could be directly applied to these modes if required.

Figure 17 – Example of modal analysis results.



Source: Author.

Exported Data and Graph Representation

To automate data extraction during the finite element simulations, a custom ANSYS scripting routine is executed as part of the Datagen simulation workflow. This

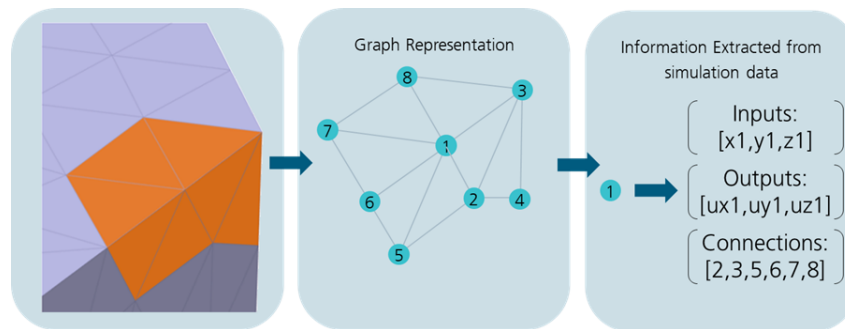
routine exports both the undeformed mesh geometry and the deformation data associated with each vibration mode.

The undeformed mesh is exported in Standard Tessellation Language (STL) format, while the deformation data are stored in text files. Each text file contains nodal information for a specific eigenmode, including the node identifier, spatial coordinates (x, y, z) , and the corresponding displacement vector components (u_x, u_y, u_z) . As a result, a single undeformed mesh is associated with multiple deformation fields, each corresponding to a different eigenfrequency obtained from the modal analysis.

The STL format was selected due to its standardized and lightweight representation of surface meshes as collections of triangular elements. This format enables straightforward extraction of mesh connectivity, which is later used to define graph edges. Additionally, it is widely supported by robust Python libraries, such as *Trimesh*, which facilitate efficient parsing, manipulation, and preprocessing of mesh data during the graph construction stage.

In the resulting graph representation, mesh vertices are mapped to graph nodes, while edges are defined based on mesh adjacency. At this stage, node-level features include geometric information, such as spatial coordinates. This formulation allows the graph neural network to operate directly on the underlying geometric and physical structure of the workpiece.

Figure 18 – Example of data obtained from ANSYS.



Source: Author.

Rationale for Data Format Selection

The combination of STL files for mesh geometry and text files for nodal deformation data was chosen to satisfy both practical and methodological requirements of the proposed workflow. Firstly, this data structure provides a clear separation between geometric connectivity and physical response quantities. The mesh topology, extracted from the STL file, defines the graph structure, while the deformation data stored in text files serve as node-level features and learning targets. This separation aligns naturally

with graph-based learning formulations and simplifies data preprocessing. Secondly, the selected formats are lightweight and widely supported, enabling efficient storage, transfer, and manipulation of large datasets. The use of standard file formats also ensures compatibility with existing simulation pipelines and post-processing tools. Finally, this representation supports scalability and extensibility, allowing additional physical quantities or simulation outputs to be incorporated without altering the underlying mesh representation.

Material and Process Description

The simulations performed in this study are based on a milling process designed to represent a thin-walled machining scenario with pronounced structural flexibility. The main characteristics of the simulated setup are summarized below.

Workpiece Dimensions

The workpiece geometry is defined by a rectangular prismatic domain with overall dimensions of 10 mm $\times$ 50 mm $\times$ 130 mm (thickness $\times$ width $\times$ height). The lower 30 mm of the total height corresponds to the base region, which is rigidly fixed and serves as the clamping interface and boundary condition. The remaining 100 mm constitutes the thin-walled, free-standing section subjected to material removal and dynamic excitation during the milling process. This geometric configuration was selected to accentuate structural flexibility and vibration sensitivity, providing a good representative case for the surrogate model training.

Material and Machining Strategy

The workpiece material is titanium alloy Ti-6Al-4V, which is widely used in aerospace and turbomachinery applications due to its high strength-to-weight ratio and resistance, but is also known for its difficult machinability and hardness, having a tendency to induce vibrations during the process. The machining strategy chosen for this workpiece corresponds to a top-down milling operation, in which material removal is applied from the free end of the workpiece toward the clamped base, this technique was employed due to the low complexity of the chosen workpiece. The operation consists of removing 2.5 mm uniformly along the lateral surfaces in the XY plane, extending over the full 100 mm height of the top section. This machining configuration results in a progressively thinning structure with reduced stiffness, amplifying dynamic effects and increasing the sensitivity of the workpiece to vibrational modes as material is removed.

The selected configuration approximates a milling process done in the department, while intentionally emphasizing thin-walled behaviour. This design choice ensures that the resulting dataset captures deformation patterns that are representative of the

machining conditions and suitable for training and evaluating graph-based surrogate models. The material properties, boundary conditions, and process parameters defined in this section form the basis for the finite element simulations and directly influence the modal characteristics and deformation fields analyzed in subsequent sections.

5.1.2 Data Preprocessing

The preprocessing stage is of great importance in the context of machine learning and is equally essential in this project. In this stage, the data collected previously from ANSYS is read and transformed into graph structures, which are subsequently embedded for use in training. The relevant information associated with each graph element is stored in a format compatible with PyTorch Geometric.

The initial procedure consists of separating and reading the STL files. These files are separated based on their identification numbers, and each undeformed part is paired with its corresponding deformation text files. Considering the use of Python for this stage, the Trimesh library was selected due to its efficiency and convenience in converting three-dimensional meshes into graph representations. With both geometries loaded, the nodal displacements are computed by matching the node ID from the undeformed mesh to its corresponding deformation found at the deformation text file.

The resulting dataset is then structured for training, where each mesh is represented as a graph, and each graph has local node features and edges.

Node Input Features

Each node in the mesh graph is represented by nine input features that combine geometric, structural, and process-related information. These features provide the graph neural network with both the physical and topological context necessary to learn deformation behavior.

1. Cartesian Coordinates

These three features define the spatial position of each node in the mesh. They provide the model with geometric awareness, allowing it to distinguish between regions of the component and establish a connection between node positions and their deformed result. Since the deformation field varies spatially, these coordinates are essential for establishing positional relationships within the network.

2. Boundary Condition Flag

This binary variable identifies whether a node is constrained by boundary conditions, meaning it is a fixed or clamped node. Nodes with a value of 1 are immobile,

while those with 0 are free to deform. This feature should allow the network to correctly infer areas where deformation is physically restricted.

3. Near Cutter Location Flag

This feature indicates whether a node lies within a predefined radius of the cutting tool. Nodes close to the machining zone experience higher deformation due to direct tool contact, receiving 1 as value, while distant nodes experience less vibrations, receiving 0. Including this feature can help the model distinguish stressed from unstressed regions.

4. Distance to Cutter Location

This continuous feature represents the Euclidean distance between each node and the cutting tool position:

$$d_i = \|x_i - x_{\text{cutter}}\|_2$$

It provides a quantitative measure of proximity to the tool, complementing the binary near-cutter flag. This could assist the network in learning that deformation magnitude typically decreases with distance from the machining area.

5. Degree Normalization

The degree normalization represents the local connectivity of each node within the mesh, scaled by the maximum degree of the graph:

$$\text{deg}_{\text{norm}}(i) = \frac{\text{deg}(i)}{\max_j \text{deg}(j)} \quad (27)$$

where $\text{deg}(i)$ is the number of edges connected to node i . This feature encodes the local geometric density or stiffness of the mesh, giving the model information about structural variations in regions with different connection densities.

6. Mean Edge Length

This feature corresponds to the average length of all edges incident to a node, given by:

$$L_{\text{mean}}(i) = \frac{1}{\text{deg}(i)} \sum_{j \in \mathcal{N}(v_i)} \|x_i - x_j\|_2 \quad (28)$$

This value reflects the local mesh resolution, where smaller values indicate finely meshed regions (often near complex and important parts) and larger values correspond to coarser regions. Providing this information could help the model understand variations in geometric scale and local stiffness.

7. Cutter Direction Cosine

This feature represents the cosine of the angle between the vector from each node to the cutter and the global machining axis (in the use case of this project, the Z -axis):

$$\cos(\theta_i) = \frac{(x_{\text{cutter}} - x_i) \cdot \hat{z}}{\|x_{\text{cutter}} - x_i\|_2} \quad (29)$$

It encodes the orientation of each node relative to the cutter's movement direction, distinguishing between nodes located above, below, or lateral to the machining axis. This is meant to give more information about machining forces, could provide more relevant information for data obtained from five-axis machining processes.

Normalization of Coordinates and Distances

All distance-related quantities (nodal coordinates, distance to cutter, and mean edge length) are normalized by the maximum radial distance from the mesh centroid:

$$x_i' = \frac{x_i - c}{d_{\max}}, \quad \text{where } c = \frac{1}{N} \sum_{i=1}^N x_i, \quad d_{\max} = \max_i \|x_i - c\|_2 \quad (30)$$

This ensures that all meshes are scale-invariant, preserving geometric proportions and stabilizing network training. The final input feature vector for each node thus has dimension $[N, 9]$, combining spatial, mechanical, and process-based information.

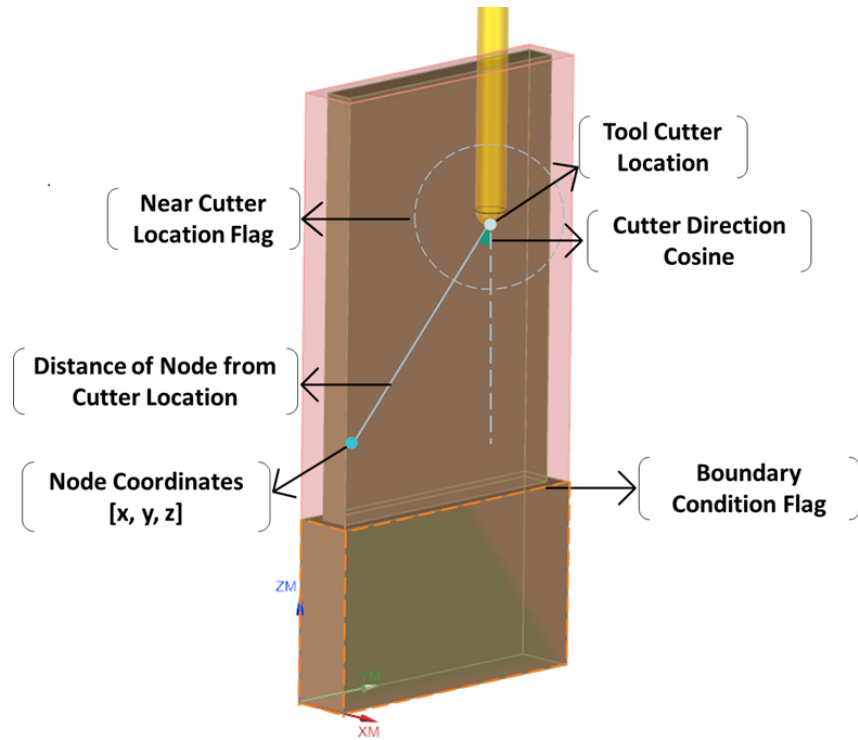
Preprocessing Code Description

This stage is meant to transform simulation data into data used by machine learning models by performing mesh parsing, feature extraction, graph construction, and dataset serialization. Two main components are involved in this process: a primary preprocessing script that performs all necessary operations and supports data validation and experimentation, and a dataset class designed for scalable training.

The main preprocessing file serves as an exploratory and validation-oriented implementation of the pipeline. Its primary purpose is to load the raw FEM outputs, verify data integrity, and test different strategies for mesh handling and feature construction before consolidating them into a reusable dataset format.

The undeformed workpiece geometry is imported in STL format and processed using the Trimesh library. The STL file represents the surface discretization of the workpiece as a set of triangular faces, from which vertex coordinates and face connectivity are directly obtained. Since the indexing of STL vertices does not correspond to the node numbering used in the finite element solver, a nearest-neighbor mapping is performed to associate each STL vertex with the closest undeformed FEM node based on Euclidean distance. This mapping enables the transfer of nodal deformation values

Figure 19 – Example of data extraction from a mesh.



Source: Author.

computed in ANSYS to the corresponding STL vertices, this data is saved in text files for easier handling.

The graph topology is then derived directly from the mesh connectivity encoded in the STL faces. Each mesh vertex is represented as a graph node, while edges are constructed from the triangular adjacency information provided by the STL mesh, such that all triangle edges contribute to the graph structure. This approach preserves the original surface discretization of the finite element model, ensuring that the resulting graph faithfully reflects the physical connectivity of the workpiece. The preprocessing script also includes intermediate visualization and checks, allowing rapid inspection of the data being transformed.

The dataset class file encapsulates the finalized preprocessing logic into a structured dataset class compatible with graph learning frameworks. It is designed to provide a clean interface for loading, storing, and accessing graph samples during model training and evaluation.

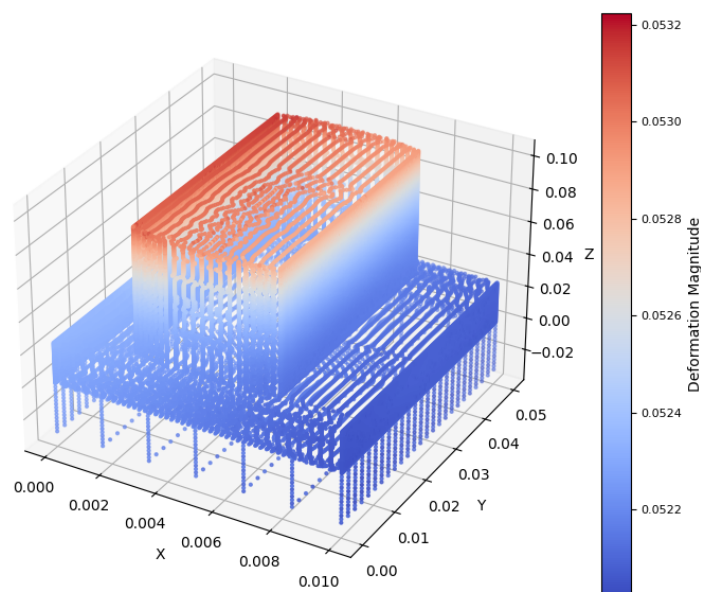
This class loads preprocessed graph objects, each representing a specific eigenmode. Each graph contains node features, edge indices defining mesh connectivity, and node-level target values corresponding to displacement vectors. The class also stores metadata such as the number of nodes, feature dimensionality, and output dimensions, enabling dynamic model configuration.

To ensure efficient training, the dataset implements standardized normalization and denormalization procedures based on dataset-level statistics. These statistics are computed once and reused across training and validation splits, ensuring numerical stability and consistent scaling across experiments. The dataset structure supports batching, which is essential when working with large meshes and multiple simulation instances. By separating exploratory preprocessing from the finalized dataset implementation, this two-stage design improves reproducibility, scalability, and maintainability.

Node Mismatch Issue

The initial preprocessing method presented concerning problems related to data consistency. However, a significant problem emerged during the process of subtracting the nodes from the STL files. This was identified due to a high degree of inconsistency, which was attributed to a small discrepancy in the number of nodes between the files. Upon examination of the point cloud, as illustrated in Figure 20, the component in question was observed to be unaltered and accurate. However, in subsequent developments, due to suboptimal training outcomes, it was decided to take a more comprehensive verification of the dataset, verifying the connections. The graph components were then reconstructed in an STL file from the training data, and the same process can be applied to reconstruct the component from the results of network inference. The reconstruction revealed that the connections were inconsistent, resulting in a poor dataset, as illustrated in Figure 21.

Figure 20 – Example of a point cloud display for a mesh.



Source: Author.

Figure 21 – Resulting STL file without node matching.

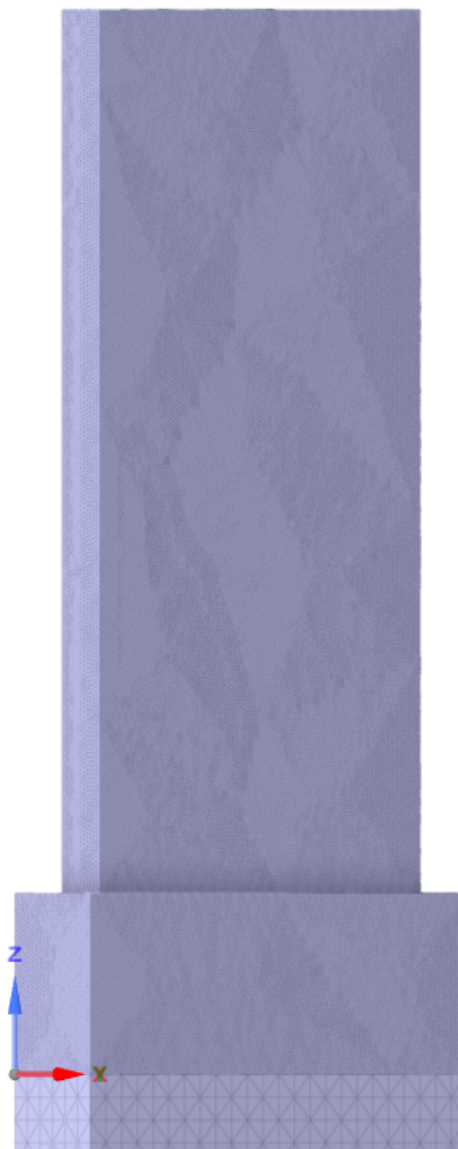


Source: Author.

To address this problem, it was necessary to explore alternative methods of exporting workpiece data through ANSYS that were also feasible within Datagen's existing pipeline. This was due to the fact that STL file export is managed internally by the software and is not easily customizable. To circumvent this, the node data utilized

internally by ANSYS was acquired. In contrast to an STL file, which represents only the surface of the part, the node data also contains the mesh volumetric information. This created the necessity of filtering the surface nodes. The utilization of the recently obtained data, stored in a text file, facilitated the preprocessing part. This process enabled the identification of the corresponding deformations of each node within its designated file. This procedure ensured that the STL file of the reconstructed deformed part was consistent with the expected outcome, as shown in Figure 22.

Figure 22 – Corrected resulting STL file.



Source: Author.

Additionally, due to how boundary conditions are established in Ansys, they could not be automatically introduced, which required additional code implementations to preprocessing.

5.1.3 Model Definition and Training Strategy

The objective of the learning stage is to construct a graph-based surrogate model capable of approximating the deformation fields obtained from finite element simulations. Given the mesh-structured nature of the input data and the node-level regression objective, graph neural networks are a natural choice, as they are explicitly designed to exploit relational structure and local neighborhood interactions. In this work, the focus is placed on models that have previously demonstrated effectiveness as learned simulators or surrogate models for physical systems discretized on meshes.

Model Selection Rationale

The original MeshGraphNets architecture employs GraphNet blocks, in which each block performs a complete message-passing cycle consisting of an edge update multilayer perceptron, an aggregation step, and a node update MLP. While this design enables explicit learning of edge-level interactions, it significantly increases computational cost and memory consumption, especially for large finite element meshes with dense connectivity. Moreover, it requires the definition and storage of edge features, which would require more preprocessing steps, and not fit the dataset considered in this work.

To address these limitations, the MeshGraphNet architecture was adapted into a simplified variant referred to as Modal Analysis Mesh Graph Net Modal Analysis Mesh Graph Net (MAMGN). In this formulation, the original GraphNet blocks are replaced by node-centric graph convolutional layers, eliminating the need to maintain and update explicit edge features. This modification substantially reduces memory usage and accelerates training. Given that the deformation response in this application is primarily governed by geometric configuration, boundary conditions, and machining-related node features, rather than heterogeneous or time-varying edge attributes, this simplification is well aligned with the underlying physics of the problem.

In addition to MAMGN, a conventional graph convolutional network was selected as a baseline model. The GCN provides a simpler architecture with fewer message-passing steps and serves as a reference point to evaluate the benefits of deeper, iterative message-passing schemes. Together, these two models enable a controlled comparison between a shallow feed-forward GNN and a mesh-aware surrogate model inspired by physical simulation principles.

Graph Neural Network Architectures

Although both models employed in this work are based on graph convolutional layers, their internal structure and learning behavior differs.

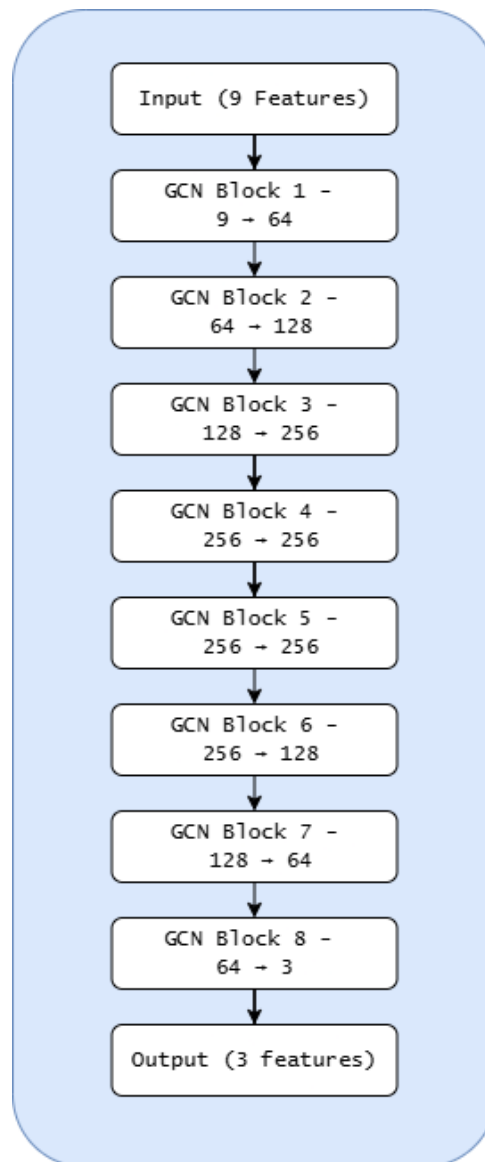
The GCN model follows a conventional feed-forward graph neural network design, in which successive graph convolutional layers transform node features through a predefined sequence of hidden dimensions. Each layer performs neighborhood aggregation based on the mesh connectivity, and the prediction is obtained in a single forward pass through the network. Due to the limited number of message-passing layers, information propagation is restricted to relatively local neighborhoods, which reduces the risk of oversmoothing and limits the influence of distant nodes. As a result, this GCN model acts primarily as a feature extractor and provides an efficient baseline for node-level deformation prediction.

The MAMGN architecture adopts an encoder-processor-decoder architecture inspired by the original MeshGraphNets framework. The encoder projects the nine-dimensional input node features into a latent space of dimension L , where the message-passing operations are performed. The processor stage consists of K repeated message-passing blocks, each implemented as a GCNConv layer followed by a residual connection. This structure enables iterative refinement of the latent node representations, allowing information to propagate across the entire mesh over multiple message-passing steps. Such iterative updates should resemble the relaxation behavior of finite element solvers, where local interactions progressively influence the global deformation state.

Finally, the decoder maps the refined latent representations back to a three-dimensional displacement vector at each node. Because the latent dimension remains constant throughout the processor stage, the model effectively performs K successive correction steps in the same feature space. This distinguishes MAMGN from standard GCNs and enables it to behave as a learned physical simulator rather than a purely feed-forward predictor.

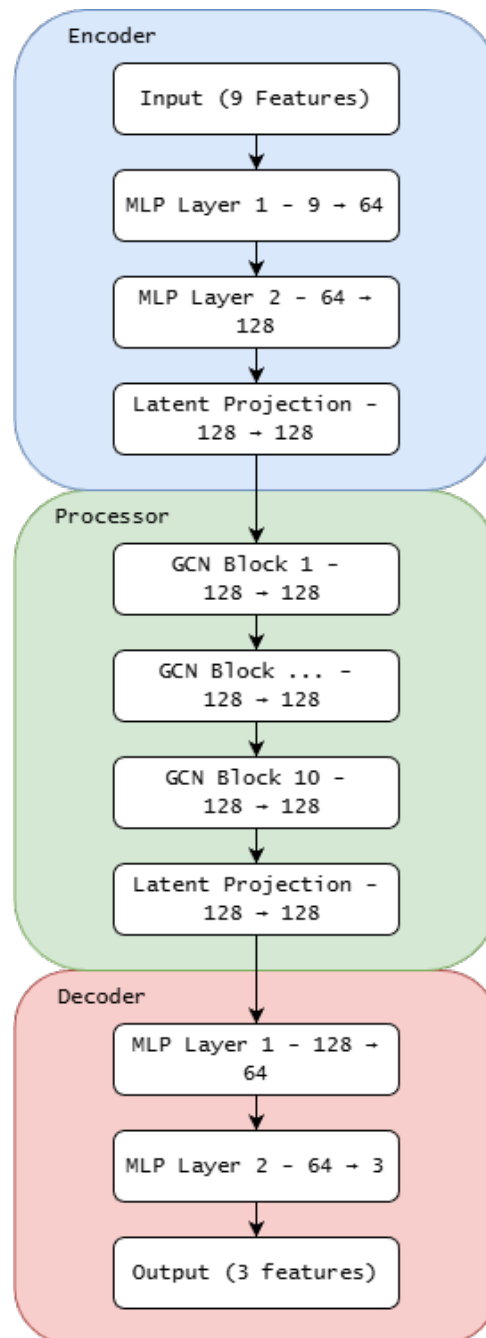
The idea behind having two models is to provide a comparison between two different message-passing strategies in the context of mesh-based surrogate modeling, with MeshGraphNets being chosen due to showing promising results as a surrogate model for fluid dynamics simulation.

Figure 23 – GCN architecture.



Source: Author.

Figure 24 – MAMGN architecture.



Source: Author.

Training Pipeline

Both models are trained in a supervised learning setting, where the input consists of graph-structured representations of the undeformed mesh and associated process-related node features, and the target output is the nodal displacement field obtained from FEM simulations. Training is performed independently for each vibration mode considered in this work, allowing the models to specialize in learning the spatial deformation patterns associated with individual eigenmodes.

The training pipeline includes dataset splitting into training and validation subsets, feature normalization, batch-wise graph loading, and periodic checkpointing of model parameters. Model selection is based on the validation loss, and the best-performing checkpoint is retained for subsequent evaluation. This pipeline enables reproducible experimentation and supports rapid comparison between architectures without modifying the underlying codebase.

All experiments were conducted on a system equipped with an Intel Xeon processor based on the Skylake microarchitecture, featuring 8 physical cores operating at 2.0 GHz. The system architecture is 64-bit and supports standard x86_64 instruction sets. Model training was performed using an NVIDIA A100 GPU with 80 GB of on-board memory. System memory consisted of 16 GB of ECC-enabled RAM.

Table 5 – System specifications used for all experiments.

Component	Specification
CPU	Intel Xeon (Skylake microarchitecture), 8 physical cores @ 2.0 GHz
Architecture	64-bit (x86_64 instruction set)
GPU	NVIDIA A100, 80 GB memory
System Memory	16 GB ECC-enabled RAM

The dataset used for training and evaluation was generated from the finite element simulations described previously, each graph contains information of a different cutter location, corresponding to a different material removal state. The complete dataset contains 100 graph samples, which were split into training, validation, and test sets using a 80/10/10 ratio. The average graph size is 88965 nodes and 266883 edges. Due to the size of individual graphs, training was performed using small batch sizes, to remain within GPU memory limits.

Training was carried out for 1000 epochs using the Adam optimizer, with an initial learning rate of 0.0005. In addition to loss monitoring, the previous mentioned evaluation metrics are obtained every 50 epochs.

5.1.4 Loss Function and Evaluation Metrics

The learning objective is defined as a node-level regression problem, where the model predicts three-dimensional displacement vectors for each node in the mesh. To ensure robustness to outliers while maintaining sensitivity to small prediction errors,

a Smooth L1 loss function is employed during training. This loss function combines mean squared error and mean absolute error losses and is particularly well-suited for regression tasks involving physical quantities with varying magnitudes.

Model performance is evaluated using multiple complementary metrics, including coordinate-wise errors, total deformation, and percentage-based error metrics. These metrics allow for a detailed view of directional accuracy, deformation magnitude, and relative error distribution across the workpiece. In addition to numerical metrics, visualization of predicted deformation fields on the mesh is used to verify spatial consistency of the predictions.

To assess the performance of the proposed surrogate model, it is essential to analyze its global error, how accurately it reproduces the spatially varying deformation field predicted by the FEM simulations and where these errors occur, as errors near cutter location are not a good result. Since the output consists of three-dimensional nodal displacements defined over a mesh, making the most relevant metrics be: coordinate-wise errors to quantify directional accuracy, total deformation (Δ) metrics evaluate the magnitude of total deformation, and spatial distributions reveal localized regions of underperformance. These metrics capture different aspects of result quality and help identify the strengths and weaknesses of the model. Because deformation amplitudes vary across the workpiece, percentage-based metrics are required to contextualize prediction error relative to the local ground-truth values. Finally, mesh visualizations are used as support to verify fidelity, spatial location of the errors and irregularities.

Loss

The surrogate model is trained using the Smooth L1 loss function, a variant of the Huber loss. This loss combines a quadratic penalty for small errors with a linear penalty for larger deviations, which makes it particularly suitable for deformation prediction where abrupt variations may occur near the cutter or at boundary regions. For a predicted displacement $\hat{u}_i$ and the corresponding ground truth value u_i , the Smooth L1 loss is defined as:

$$L_{\beta}(\hat{u}_i, u_i) = \begin{cases} \frac{1}{2}(\hat{u}_i - u_i)^2, & \text{if } |\hat{u}_i - u_i| < \beta \\ \beta|\hat{u}_i - u_i| - \frac{1}{2}\beta^2, & \text{otherwise} \end{cases} \quad (31)$$

The surrogate model is trained using the SmoothL1Loss function available at the PyTorch library. This criterion is closely related to the classical Huber loss, as both share the same structure where a quadratic penalty is applied for small errors and a linear penalty for larger ones. However, it is important to mention that the PyTorch implementation is not mathematically identical to the original Huber formulation. PyTorch applies a different scaling in the quadratic region and uses a threshold parameter β that

influences the transition between the two regimes. Therefore, the Smooth L1 function used in this work is a variant of the Huber loss, with slightly modified behaviour tailored for stability in deep learning applications.

The PyTorch Smooth L1 loss is defined as:

$$L_{\beta}^{\text{SmoothL1}}(\hat{u}_i, u_i) = \begin{cases} \frac{1}{2\beta}(\hat{u}_i - u_i)^2, & \text{if } |\hat{u}_i - u_i| < \beta \\ |\hat{u}_i - u_i| - \frac{\beta}{2}, & \text{otherwise} \end{cases} \quad (32)$$

where β is the transition threshold (set to $\beta = 0.01$ in this work). This differs from the classical Huber loss, given by:

$$L_{\delta}^{\text{Huber}}(\hat{u}_i, u_i) = \begin{cases} \frac{1}{2}(\hat{u}_i - u_i)^2, & \text{if } |\hat{u}_i - u_i| \leq \delta \\ \delta \left(|\hat{u}_i - u_i| - \frac{\delta}{2} \right), & \text{otherwise} \end{cases} \quad (33)$$

The main differences are the normalization factor $\frac{1}{2\beta}$ in the quadratic region and the different shifting term in the linear region. Despite these differences, both formulations share the same purpose: combining the smoothness of MSE near zero with the robustness of MAE for large deviations.

This loss function offers several practical advantages for the task, providing stable and well-behaved gradients and avoiding gradient explosion that will later be analyzed in the results presented at section 5.2.

Metrics

The evaluation of the proposed surrogate model requires more than a numerical comparison of predicted and simulated deformation values. Because the task involves learning complex, spatially varying structural behaviour on irregular meshes, evaluation must be done using metrics that take into account the fidelity of the final result. Considering this necessity, multiple quantitative and qualitative evaluation tools were adopted, which were explored during the model development and training refinements.

To assess prediction quality, the training and validation loops generate a series of diagnostic plots over the dataset.

Axis-wise MAE curves:

These plots track the Mean Absolute Error (MAE) in meters for the x , y , and z coordinates across training epochs. They indicate whether the model captures deformation consistently in each spatial direction and reveal imbalances (e.g., a tendency to learn the dominant deformation axis first). This is particularly relevant to analyze which components are exhibiting larger errors and identify biases in the model. The same equation can be used for coordinates Y and Z .

$$\text{MAE}_x = \frac{1}{N} \sum_{i=1}^N |\hat{u}_{i,x} - u_{i,x}| \quad (34)$$

MAPE and Standard Deviation curves:

The Mean Absolute Percentage Error Mean Absolute Percentage Error (MAPE) and the standard deviation of the error were computed for each axis and for the vector of total deformation. These metrics complement the MAE by expressing performance relative to the magnitude of deformation, which is essential in regions near the cutter where small displacements cannot tolerate large percentage errors.

Standard Deviation of the Error per Axis

$$\text{STD}_x = \sqrt{\frac{1}{N} \sum_{i=1}^N (|\hat{u}_{i,x} - u_{i,x}| - \text{MAE}_x)^2} \quad (35)$$

Mean Absolute Percentage Error per Axis

$$\text{MAPE}_x = \frac{1}{N} \sum_{i=1}^N \left(\frac{|\hat{u}_{i,x} - u_{i,x}|}{|u_{i,x}| + \varepsilon} \right) \times 100\% \quad (36)$$

Together, these visual and numerical tools should enable a comprehensive evaluation of the GNN surrogate model. They are meant to identify convergence issues, feature importance, model weaknesses, and the conditions under which the predictions aligned most closely with the FEM ground truth.

5.2 RESULT ANALYSIS

This section presents and discusses the results obtained for the two trained architectures across the first and second vibration modes. With the evaluation methodology and metrics previously defined, the focus here is on analyzing the behaviour of the models, comparing their predictive capabilities, and illustrating how well they reproduce the deformation fields generated by the FEM simulations.

For each experiment, the analysis will account for global model behaviour and learning dynamics, through the evolution of the loss function and errors along the training epochs. Accuracy, evaluated quantitatively using the coordinate-wise and total deformation errors over the test set. And finally, spatial and geometric fidelity, by analysing deformation fields, error heatmaps, and reconstructed STL predictions This

way it is possible to analyze and select the models based on their training behaviour, convergence properties, predictive performance and visually to validate the final results.

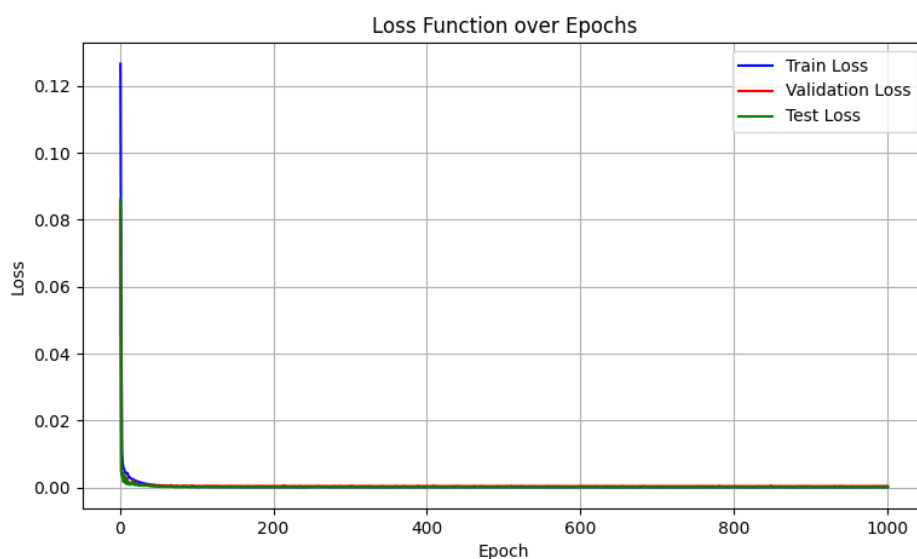
5.2.1 Experimental Results and Model Comparison

Training Behaviour and Convergence

This subsection evaluates the training dynamics of the proposed graph neural network architectures, with emphasis on convergence behaviour, optimization stability, and the relationship between numerical loss reduction and physically meaningful prediction accuracy. Both the GCN and the MAMGN models were trained under identical conditions for the first and second vibration modes, enabling a direct comparison of their learning behaviour.

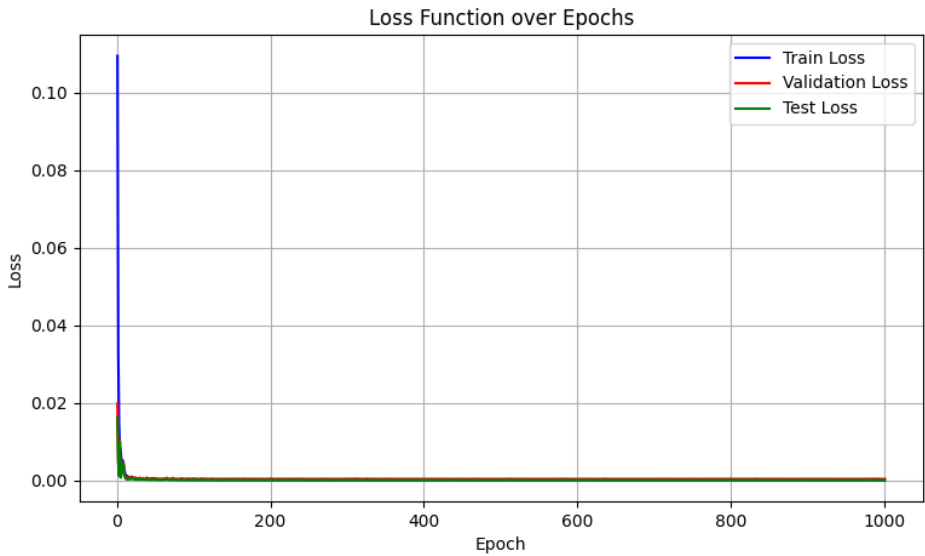
For vibration mode 1, both models converge relatively quickly, suggesting that the dominant deformation pattern is easier to learn and can be captured with a limited number of message-passing steps. In contrast, mode 2 exhibits slower convergence, higher residual loss and higher variation, which is consistent with the increased spatial complexity and higher-frequency deformation characteristics of this mode. The close overlap between training, validation, and test losses demonstrates strong generalization and the absence of overfitting. The final convergence toward a low plateau confirms that the surrogate models successfully approximates the FEM deformation field with consistent behaviour across the dataset.

Figure 25 – Loss evolution over training epochs for GCN Mode 1.



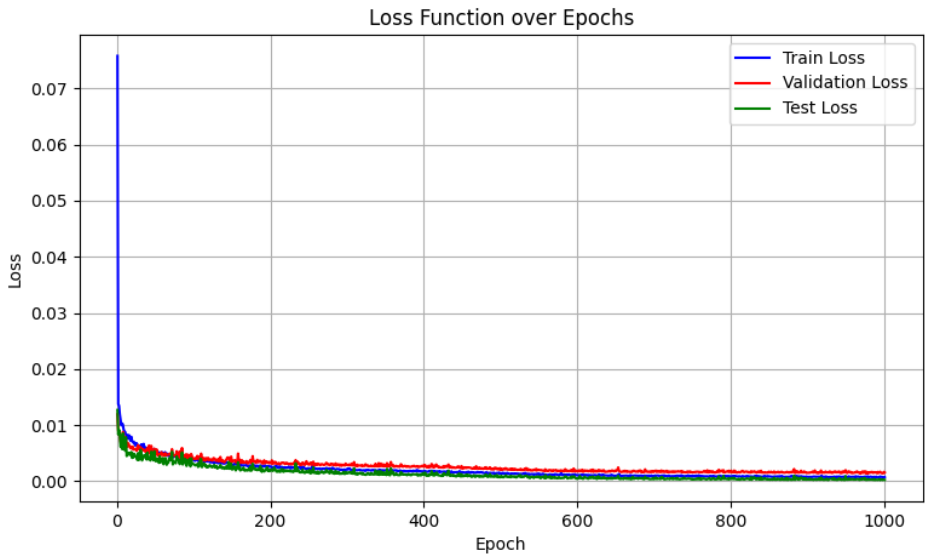
Source: Author.

Figure 26 – Loss evolution over training epochs for MAMGN Mode 1.



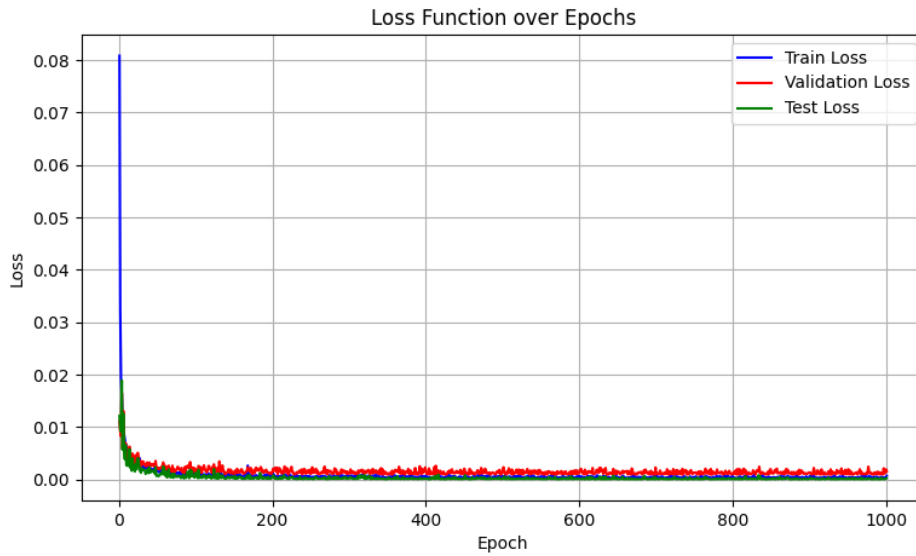
Source: Author.

Figure 27 – Loss evolution over training epochs for GCN Mode 2.



Source: Author.

Figure 28 – Loss evolution over training epochs for MAMGN Mode 2.



Source: Author.

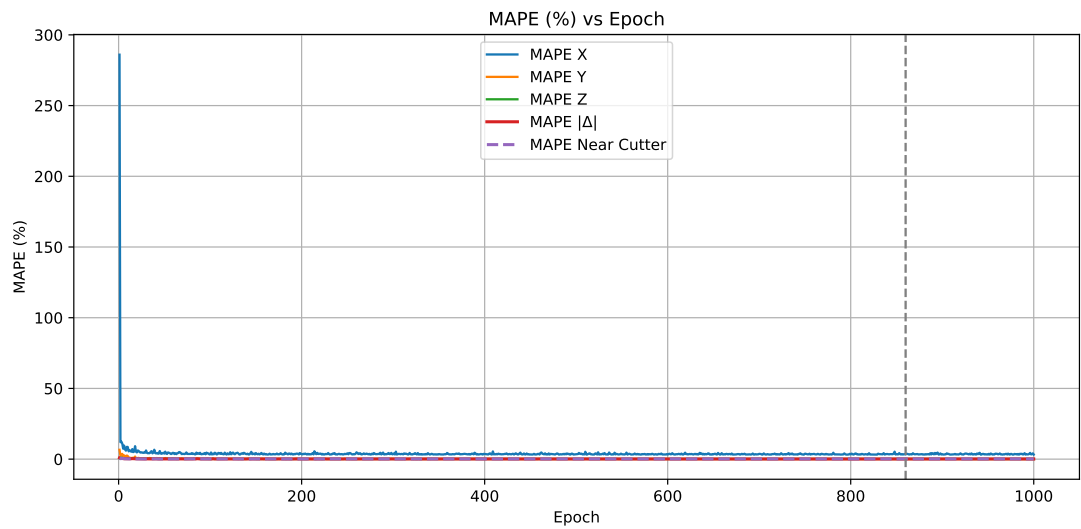
Figures 25, 26, 27 and 28 illustrate the evolution of the training loss across epochs for the two vibration modes and architectures. In both cases, the loss decreases steadily, indicating stable gradient propagation and the absence of numerical instabilities such as exploding or vanishing gradients. Differences in convergence speed and final loss levels can also be observed between the two architectures, reflecting their distinct message-passing mechanisms and internal representations, with GCN being equal in the first mode and slightly inferior in the second mode.

Axis-Wise Mean Absolute Error Percentage

In addition to loss evolution, Figures 29, 30, 31, and 32 present the mean absolute percentage error progression over training epochs. The reduction in MAPE follows a trend similar to that of the loss curves, confirming that numerical optimization correlates with improved predictive accuracy in physical space. However, minor discrepancies between loss reduction and error stabilization can be observed, indicating that further training beyond a certain point yields diminishing returns in terms of physically relevant accuracy.

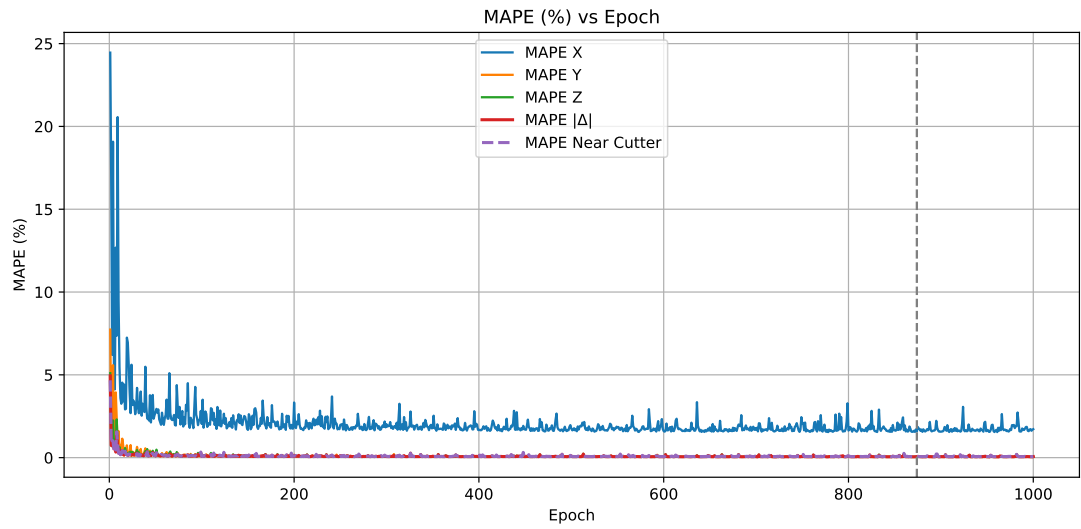
A more detailed inspection of axis-wise error evolution reveals uneven learning behaviour across spatial directions. Errors along the X axis remain consistently higher compared to Y and Z, which may be attributed to lower displacement magnitudes, increased sign changes in the deformation field, or reduced signal-to-noise ratio in this direction. These observations underline the importance of evaluating directional error

Figure 29 – MAPE evolution over training epochs for GCN Mode 1.



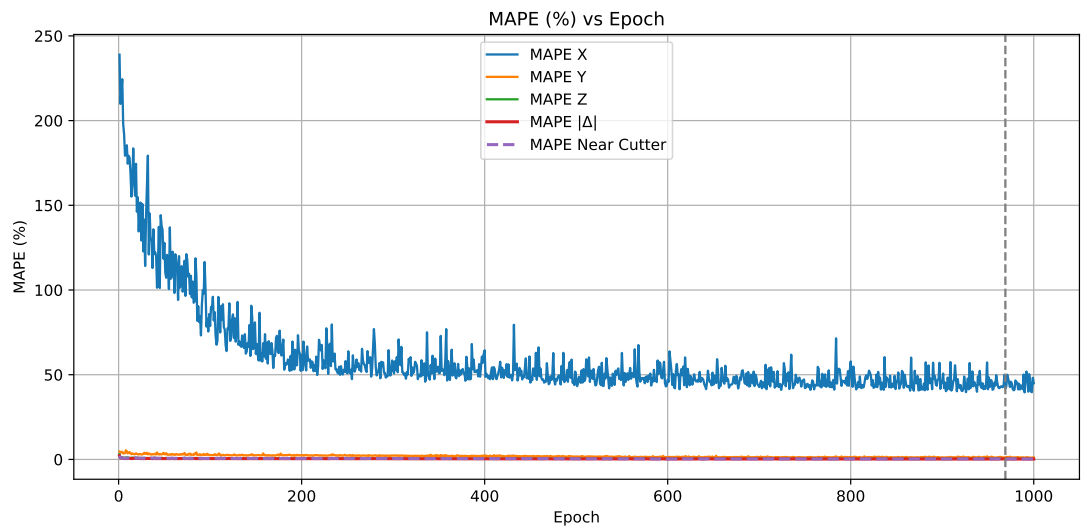
Source: Author.

Figure 30 – MAPE evolution over training epochs for MAMGN Mode 1.



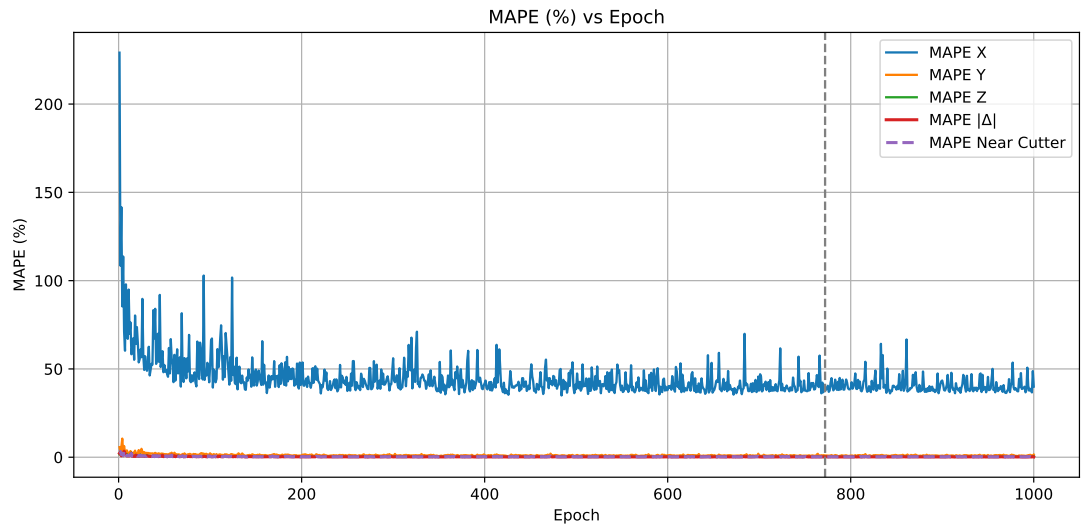
Source: Author.

Figure 31 – MAPE evolution over training epochs for GCN Mode 2.



Source: Author.

Figure 32 – MAPE evolution over training epochs for MAMGN Mode 2.



Source: Author.

components when assessing surrogate models for structural dynamics.

Overall, the training behaviour demonstrates that both models are capable of learning stable mappings between mesh geometry and deformation fields, while also revealing architectural differences in convergence efficiency and robustness across vibration modes.

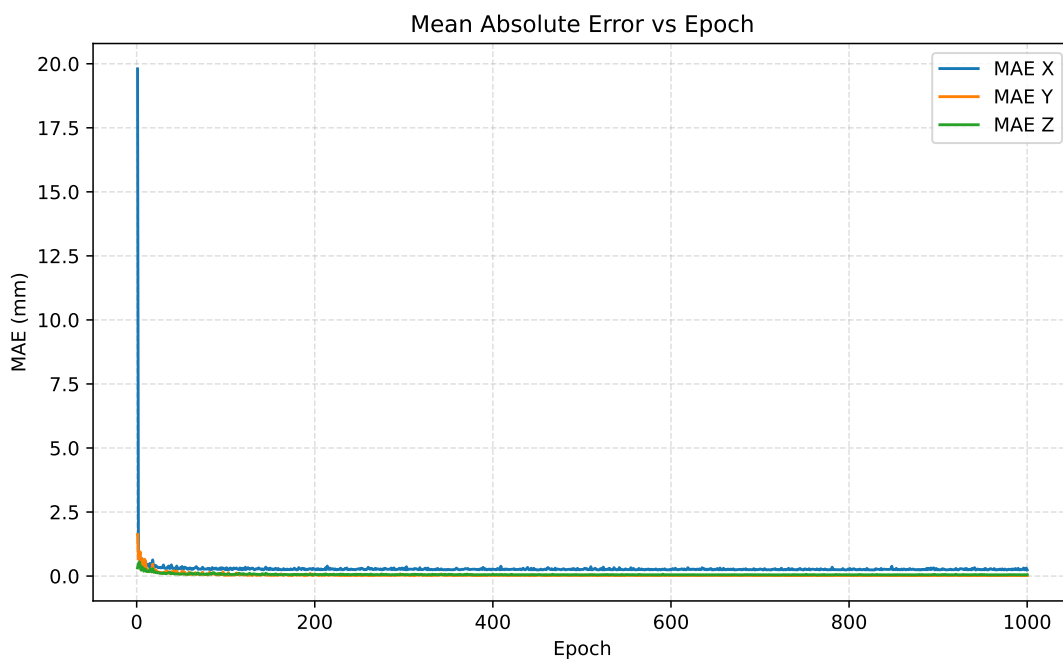
Axis-Wise Mean Absolute Error in Physical Units

In addition to the percentage-based metrics, the MAE was evaluated for each displacement component individually. All MAE values are expressed in millimeters, directly representing the physical deviation from the FEM reference solution.

Since the model predicts the displacement vector components independently, evaluating MAE per axis provides a detailed assessment of directional prediction behaviour. This decomposition enables identification of anisotropic learning effects and avoids masking axis-specific discrepancies through vector magnitude aggregation.

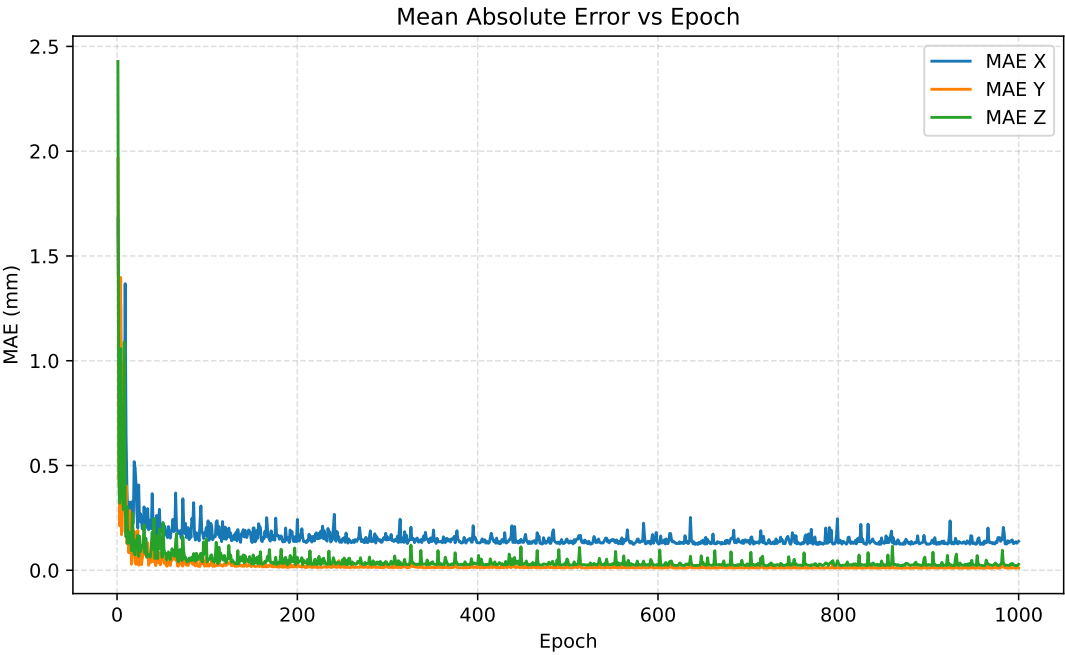
Although MAPE is evaluated both per axis and for the total deformation, MAE is computed only for the individual displacement components. The total deformation MAPE already reflects the relative accuracy of the total deformation magnitude. Computing an additional MAE on the total deformation would not provide useful information, as the directional MAE values already quantify the absolute physical deviation of the predicted displacement field.

Figure 33 – MAE evolution over training epochs for GCN Mode 1.



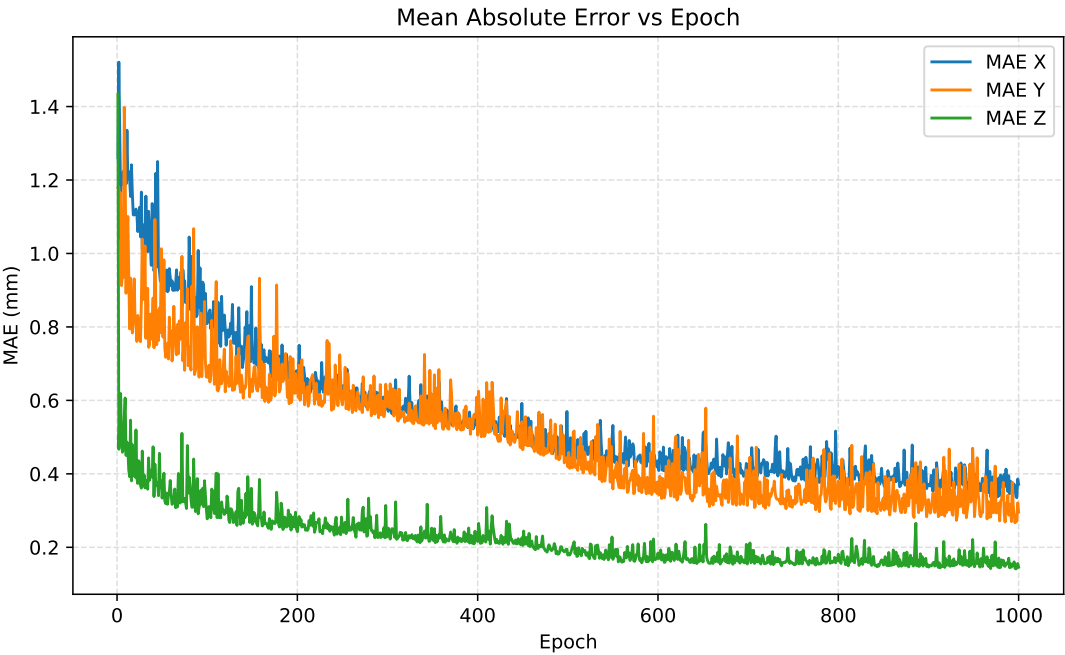
Source: Author.

Figure 34 – MAE evolution over training epochs for MAMGN Mode 1.



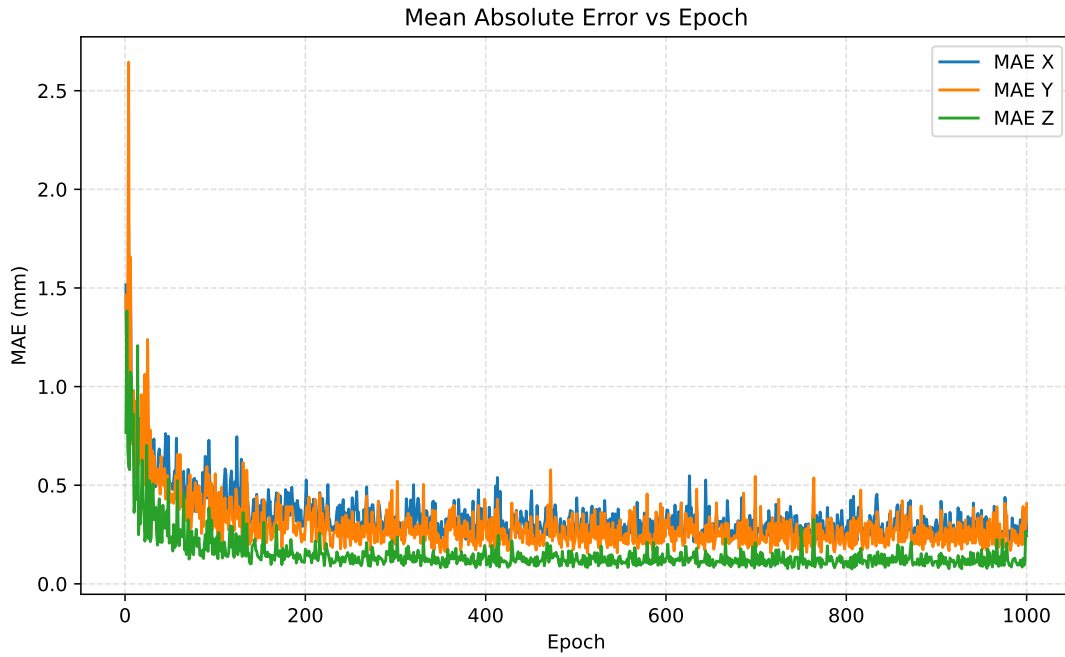
Source: Author.

Figure 35 – MAE evolution over training epochs for GCN Mode 2.



Source: Author.

Figure 36 – MAE evolution over training epochs for MAMGN Mode 2.



Source: Author.

Figures 33, 34, 35, and 36 illustrate the evolution of MAE over the training epochs for each spatial direction and architecture. For vibration mode 1, both models converge toward low MAE values across all axes, confirming that the dominant modal deformation pattern can be approximated with high physical precision. The final errors remain within a small fraction of a millimeter, indicating deviations in the micrometer range.

For vibration mode 2, higher MAE levels are observed, particularly along the X direction. This behaviour is consistent with the increased spatial complexity and higher-frequency characteristics of the second vibration mode. The elevated MAE values align with the previously observed trends in loss and MAPE evolution.

A consistent directional pattern can be identified across experiments, where the X-axis exhibits higher MAE values compared to Y and Z. This effect may be attributed to lower displacement magnitudes, more frequent sign changes in the modal field due to the torsion, or directional sensitivity in the graph aggregation mechanism.

When comparing architectures, MAMGN achieves systematically lower MAE values across both vibration modes. The improvement is higher for mode 2, suggesting that the extended message-passing structure enhances long-range spatial information propagation, which is essential for modeling higher-order deformation patterns.

Overall, the axis-wise MAE analysis confirms that both models achieve physically meaningful approximation quality while revealing architectural differences in directional robustness and stability.

Final Quantitative Performance

This subsection compares the final predictive performance of the trained models using quantitative error metrics evaluated on the test dataset. The objective is to assess how accurately each architecture reproduces the reference FEM displacement fields after training convergence, and to determine their suitability as surrogate models for modal deformation prediction.

Table 6 summarizes the final MAPE and additional relevant metrics for both models and both vibration modes. The results show that prediction accuracy varies significantly between modes, with consistently lower errors observed for the first vibration mode. This outcome aligns with expectations, as the first mode typically exhibits smoother and more globally coherent deformation patterns.

When comparing architectures, MAMGN achieves lower average errors across both modes, particularly for the second vibration mode. This suggests that the iterative message-passing structure of the processor stage enables more effective long-range information propagation across the mesh, which is critical for capturing higher-order deformation patterns. In contrast, the GCN model, which performs a limited number of message-passing steps, shows a tendency to under-represent global interactions.

Axis-wise error decomposition further highlights model-specific performance characteristics. While both architectures perform similarly along the Y and Z axes, discrepancies are more pronounced along the X axis, where errors remain comparatively higher. This behaviour may indicate that certain deformation components are more sensitive to graph connectivity and neighborhood aggregation depth.

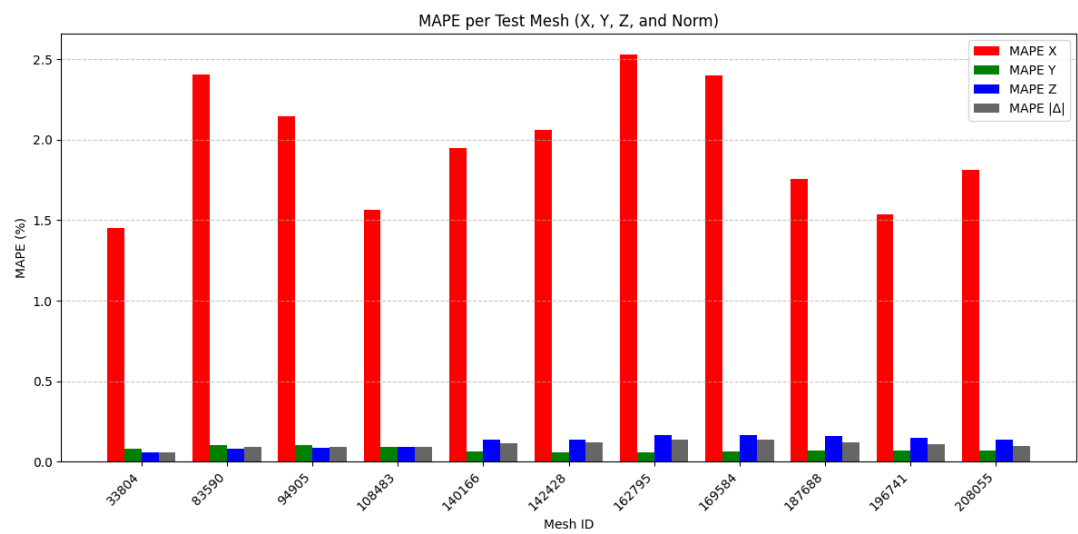
Despite these differences, both models achieve error levels that suggest meaningful approximation of the underlying FEM solutions. The quantitative results therefore support the feasibility of GNN-based surrogate models for modal deformation prediction, while also highlighting architectural trade-offs between simplicity and expressive power.

Table 6 – Final prediction error metrics for both models and vibration modes.

Model	Mode	MAPE	MAPE near Cutter Location	MAE X[mm]	MAE Y[mm]	MAE Z[mm]
GCN	Mode 1	0.1078%	0.1449%	0.2416	0.0200	0.0476
MAMGN	Mode 1	0.0533%	0.0648%	0.1248	0.0102	0.0235
GCN	Mode 2	0.2336%	0.2979%	0.3374	0.2823	0.1421
MAMGN	Mode 2	0.1340%	0.1952%	0.2128	0.1598	0.0829

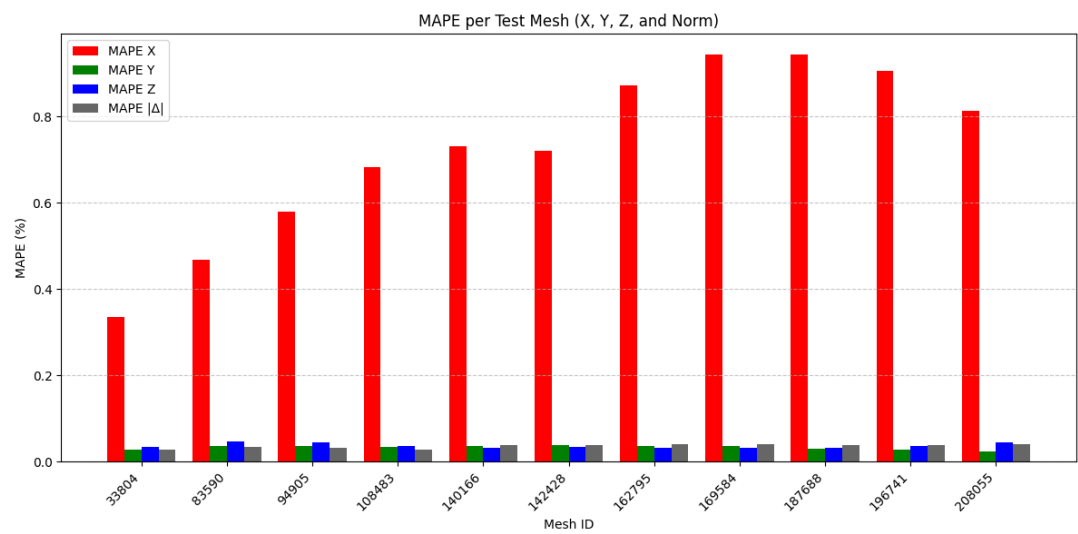
Table 6 presents the performance of each model at its best-performing epoch, using the total deformation as the variable for MAPE computation. For both vibration modes, MAMGN achieves lower error metrics than the GCN baseline when analyzing the best epoch. The performance gap is more pronounced for vibration mode 2, indicating superior modeling of long-range dependencies.

Figure 37 – MAPE over test set for GCN Mode 1.



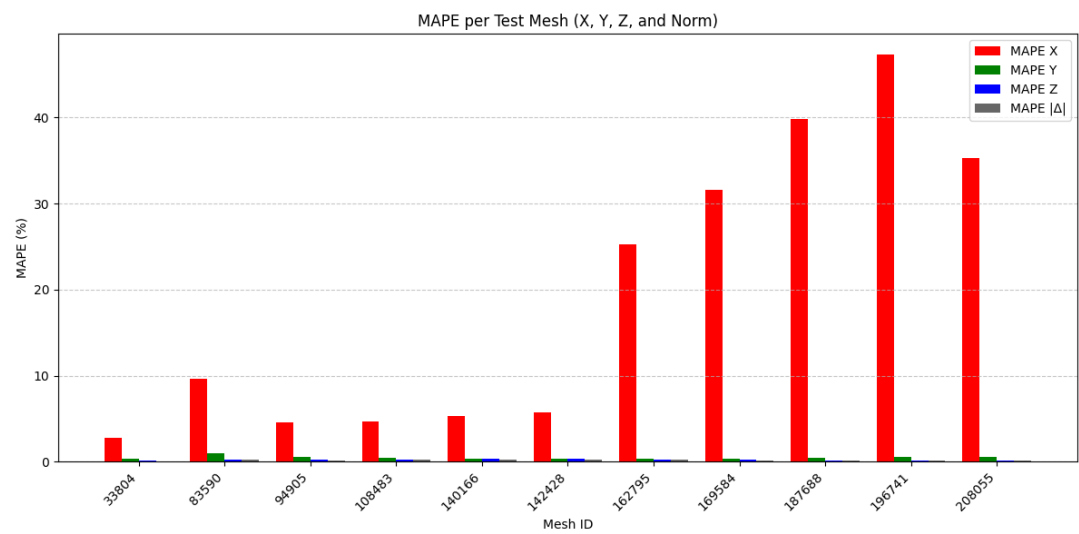
Source: Author.

Figure 38 – MAPE over test set for MAMGN Mode 1.



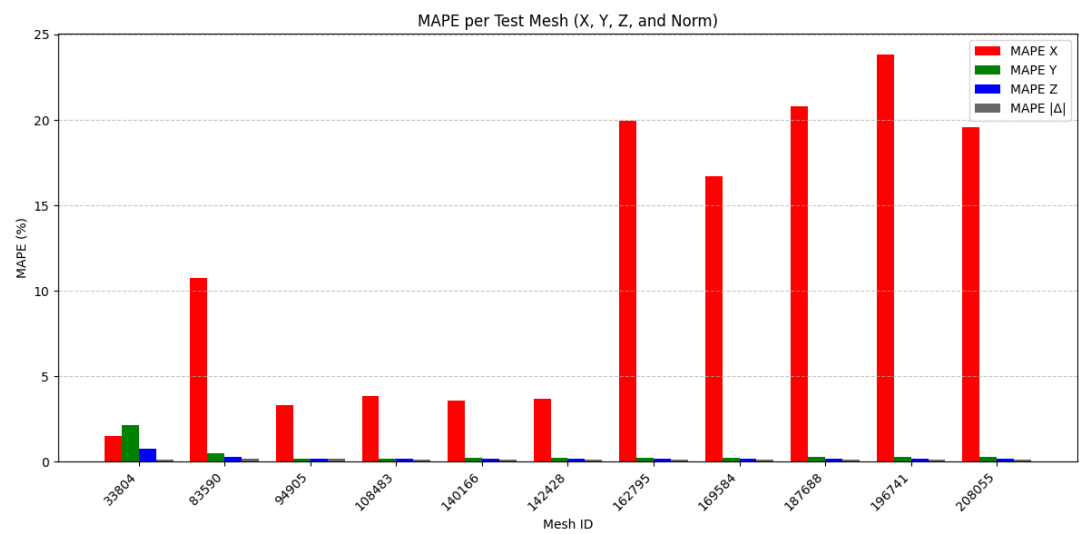
Source: Author.

Figure 39 – MAPE over test set for GCN Mode 2.



Source: Author.

Figure 40 – MAPE over test set for MAMGN Mode 2.



Source: Author.

Figures 37, 38, 39 and 40 shows the distribution of MAPE across test meshes, where the mesh IDs correspond to the toolpath steps, higher mesh IDs indicate that more material has been removed. Error increases with geometric complexity and material removal, particularly for mode 2. Across all cases, MAMGN demonstrates improved robustness, maintaining lower error levels.

Qualitative and Spatial Analysis

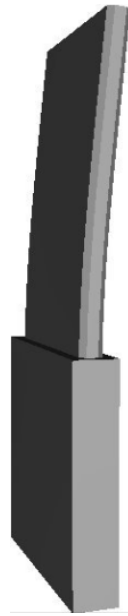
While quantitative metrics provide a global measure of predictive accuracy, they do not fully capture spatial error distributions or localized discrepancies in the deformation field. This subsection therefore presents a qualitative analysis of selected test cases, focusing on the spatial correspondence between predicted and reference deformation patterns.

Figures 41 and 44 show representative visualizations of the FEM ground truth alongside the predictions generated by the GCN (Figures 42 and 45) and MAMGN models (Figures 43 and 46). For vibration mode 1, both models reproduce the overall deformation shape and nodal displacement direction with high fidelity, indicating that dominant structural behaviour is well captured.

Differences become more evident when zooming in at the meshes, where localized deviations and amplitude attenuation can be observed, particularly in the middle of the blade or near boundary constraints. In these areas, both models tends to have noisy results, creating small spikes on the mesh, and since MAMGN has less error, the spikes are less prevalent, preserving the smooth surface. Error visualization further reinforces that predictions are able to capture where in the mesh the deformations are happening, being very accurate in producing an overall similar shape to the ground truth.

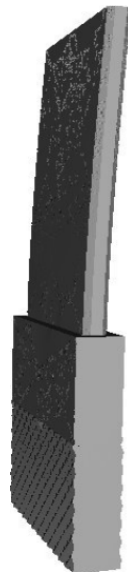
Overall, the qualitative analysis complements the quantitative evaluation by demonstrating that the proposed models not only minimize global error metrics but also capture physically meaningful deformation structures. These spatial insights are essential for assessing the practical applicability of surrogate models in vibration-aware process planning and digital twin integration.

Figure 41 – Deformations ground truth for Mode 1.



Source: Author.

Figure 42 – Deformations inferred by GCN for Mode 1.



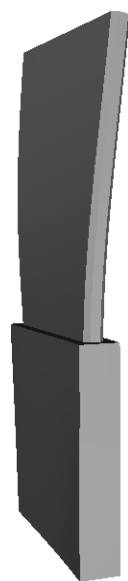
Source: Author.

Figure 43 – Deformations inferred by MAMGN for Mode 1.



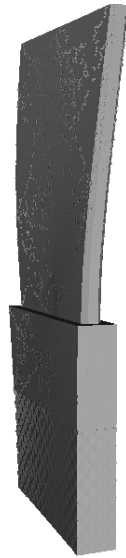
Source: Author.

Figure 44 – Deformations ground truth for Mode 2.



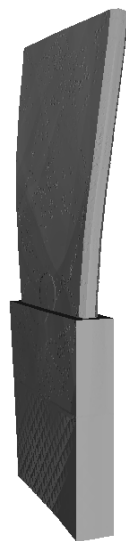
Source: Author.

Figure 45 – Deformations inferred by GCN Mode 2.



Source: Author.

Figure 46 – Deformations inferred by MAMGN Mode 2.



Source: Author.

In both modes, MAMGN more closely reproduces the FEM deformation patterns, having less noise at the blade part, presenting a better result overall.

5.2.2 Discussion and Result Synthesis

This section synthesizes the results presented in the previous subsections, linking quantitative metrics, training dynamics, and qualitative observations to the original objectives of this work. The experimental evaluation demonstrates that graph neural networks can effectively approximate FEM-based modal deformation fields when trained on mesh-derived graph representations.

Across all experiments, prediction accuracy is consistently higher for the first vibration mode, reflecting the smoother and more globally coherent nature of its deformation pattern. Higher-order modes introduce increased spatial complexity, which challenges models with limited message-passing depth and emphasizes the importance of iterative information propagation.

The comparison between GCN and MAMGN highlights a clear trade-off between architectural simplicity and expressive capacity. While the GCN offers faster convergence and lower computational overhead, MAMGN achieves superior accuracy and robustness, particularly for more complex deformation regimes.

Importantly, the results indicate that the surrogate models capture not only numerical patterns but also physically interpretable deformation behaviour, supporting their potential use in accelerated vibration analysis pipelines. However, residual errors in specific spatial regions and directions point to limitations related to dataset size, feature expressiveness, and boundary condition representation.

These findings collectively suggest that GNN-based surrogate modeling is a viable and promising approach for structural dynamics prediction in milling applications, while also identifying clear directions for future refinement and extension.

6 CONCLUSION

This chapter concludes the present work by summarizing the problem addressed, the proposed methodology, and the main results obtained. It reflects on what objectives were achieved, discusses the main limitations encountered, and suggests potential improvements and directions for future work.

6.1 SUMMARY OF THE WORK AND MAIN RESULTS

The objective of this project was to investigate the feasibility of using GNNs as surrogate models for predicting deformation fields in thin-walled workpieces subjected to machining operations. The problem is particularly challenging due to the high geometric complexity of finite element meshes, the strong influence of boundary conditions, and the dynamic nature of machining-induced vibrations.

To address this problem, a data generation pipeline based on harmonic finite element simulations in ANSYS was employed to obtain eigenfrequencies and associated nodal deformation fields. These simulation outputs were transformed into graph-structured data, where mesh nodes were mapped to graph nodes and mesh connectivity defined the graph topology. A dedicated preprocessing stage was developed to align STL-based mesh geometry with nodal deformation data and to enrich node features with geometric, boundary-condition, and process-related embeddings.

Two graph-based learning approaches were evaluated, a conventional graph convolutional network and a modified MeshGraphNet inspired architecture, referred to as MeshGraphNetLite. Both models were trained to predict nodal displacement vectors, and their performance was assessed in terms of spatial accuracy and qualitative deformation patterns. The results demonstrate that GNN-based surrogate models are capable of learning meaningful deformation behaviors directly from mesh-based data, with the MeshGraphNetLite architecture showing better performance.

6.1.1 Expectations, Objectives, and Limitations

The expectations for this project were high, as it represents an experimental and new approach within the department and addresses a topic that is still being explored in the literature. While machine learning has been increasingly applied to scientific discovery in the recent years, its use as a surrogate for finite element simulations in machining contexts is still limited.

The primary objectives of the project that were achieved include the development of a complete data generation, preprocessing, and training pipeline, and it was demonstrated that graph-based models are suitable to the task of learning deformation patterns on complex meshes. However, some goals were not achieved, as this study

was not able to provide with a method to properly measure the scalability and speed to, failing to propose a comparison to the traditional FEM method. The dataset size was constrained by the computational cost of FEM simulations and the necessity to manually design new processes, limiting the acquisition and diversity of training samples. Additionally, only a subset of vibration modes was considered, and the prediction task was restricted to deformation fields, excluding direct frequency estimation. Some unsatisfactory results, indicate that further feature engineering or architectural refinements are required.

6.2 EVALUATION OF FUNCTIONAL REQUIREMENTS

6.2.1 Data Processing and Graph Construction

The first requirement concerned the acquisition and transformation of mesh and FEM simulation data into graph-based representations suitable for machine learning. This objective was successfully achieved through the development of a structured preprocessing pipeline that converts FEM meshes into graph structures with explicitly defined nodes, edges, and node-level features.

Node features were extracted from geometric and physical quantities, and graph connectivity was derived directly from the mesh topology. The results in Chapter 5 confirm that the generated graph representations preserve the spatial relationships required for surrogate modeling. The stability of the training process and the spatial consistency of the predictions demonstrate that the graph construction pipeline retained the essential physical information embedded in the FEM data.

6.2.2 Graph-Based Machine Learning Implementation

The implementation of multiple GNN architectures fulfilled the requirement of exploring graph-based surrogate modeling approaches. The comparative experiments presented show clear differences in convergence behavior, prediction accuracy, and generalization capability between the tested models. The results indicate that graph-based architectures are capable of learning the spatially distributed deformation behavior of the workpiece directly from mesh-structured input data. The observed reduction in prediction error over epochs confirms that the models successfully captured relevant geometric and mechanical patterns.

6.2.3 Model Evaluation Against FEM Results

A central requirement was the quantitative comparison of GNN predictions with reference Ansys FEM simulations. This was achieved through a comparison using appropriate error metrics and spatial visualization of deviations. The results demonstrate that the trained models approximate FEM outputs within an acceptable range.

While minor localized discrepancies remain, the overall prediction trends align with the simulation data ground truth.

6.2.4 Scalability Testing

The scalability requirement was not addressed in this work, only one dataset was created and used, having the purpose to validate and conclude that the surrogate model is capable of solving the proposed problem.

6.2.5 Result Visualization

The visualization requirement was fulfilled through spatial plots of predicted and reference displacement fields, convergence curves and quantitative error diagrams, as well as STL models and point cloud visualizations. These visualizations enabled qualitative assessment of spatial prediction patterns and provide interpretability for domain experts. The integration of visualization into the evaluation workflow strengthened the transparency and communicability of the results.

6.3 EVALUATION OF NON-FUNCTIONAL REQUIREMENTS

6.3.1 Computational Efficiency

A key non-functional requirement was that the surrogate model must offer significantly faster prediction times than traditional FEM simulations. While the training phase itself is computationally intensive, inference is highly efficient. But a proper comparison was not presented in this work.

6.3.2 Accuracy

Accuracy was defined as maintaining prediction errors within an acceptable. Based on the quantitative metrics presented, the best-performing models for the first mode are able to output good results. Although the results for the second mode could be better, the overall deformations are reproduced reliably. This level of approximation is sufficient for a first experiment with surrogate modeling purposes.

6.3.3 Scalability

Since scalability testing was not implemented, this requirement cannot be calculated and therefore, was not shown in this work.

6.3.4 Impact of the Work

From a technological perspective, this work demonstrates the potential of graph-based learning methods to act as fast surrogate models for simulation-driven engi-

neering problems. Such models can significantly reduce the time required to evaluate deformation responses compared to full FEM analyses, enabling faster design iterations and decision-making processes.

For industrial and research contexts, the proposed approach opens the possibility of integrating learned simulators into existing digital manufacturing pipelines. This could lead to improved productivity, reduced computational costs, and enhanced insight into process interactions.

6.3.5 Future Works

Several directions for future work can be cited. First, the model outputs could be extended to predict both nodal deformations and their corresponding natural frequencies, enabling a more complete representation of the vibrational behavior. Noise reduction strategies and improved feature normalization may further enhance prediction accuracy, particularly for challenging deformation components.

From a modeling perspective, the dataset could be adapted to more closely follow the original MeshGraphNets model by incorporating edge features that encode geometric or physical relationships. Increasing the size and variability of the dataset would likely improve generalization and robustness. Finally, full integration of the developed models into the Datagen pipeline would enable automated end-to-end surrogate simulation, allowing the approach to be applied to other machining processes and structural configurations and establish ground for comparisons between the two different approaches.

REFERENCES

- AKESSON, Henrik. **Analysis of Structural Dynamic Properties and Active Vibration Control Concerning Machine Tools and a Turbine Application**. 2009. PhD thesis – Blekinge Institute of Technology, Karlskrona. ISBN 978-91-7295-172-3.
- ALTINTAS, Yusuf. **Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design**. [S. l.]: Cambridge University Press, 2012.
- ALTINTAS, Yusuf; KERSTING, Patrick; BIERMANN, Dirk; BUDAK, Erhan; DENKENA, Berend; LAZOGLU, Ismail. Virtual process systems for part machining operations. **CIRP Annals - Manufacturing Technology**, v. 63, n. 2, p. 585–605, 2014.
- ALTINTAS, Yusuf; LEE, P. A General Mechanics and Dynamics Model for Helical End Mills. **CIRP Annals**, 1996.
- BATTAGLIA, Peter W. *et al.* **Relational inductive biases, deep learning, and graph networks**. [S. l.: s. n.], 2018. arXiv: 1806.01261 [cs.LG]. Available from: <https://arxiv.org/abs/1806.01261>.
- BAUR, Michael. **Aktives Dämpfungssystem zur Ratterunterdrückung an spanenden Werkzeugmaschinen**. 2013. PhD thesis – Technische Universität.
- BAYRAM, Eda. **Propagation on Multi-relational Graphs for Node Regression**. [S. l.: s. n.], 2021. arXiv: 2110.08185 [cs.LG]. Available from: <https://arxiv.org/abs/2110.08185>.
- BISHOP, Christopher M. **Pattern Recognition and Machine Learning**. [S. l.]: Springer, 2006.
- BRAÑA, Luis F. *et al.* Quiet Transformers: Design Issues. **IEEE Transactions on Power Delivery**, 2013.
- BRANDSTETTER, Johannes; WORRALL, Daniel; WELLING, Max. **Message Passing Neural PDE Solvers**. [S. l.: s. n.], 2023. arXiv: 2202.03376 [cs.LG]. Available from: <https://arxiv.org/abs/2202.03376>.
- BRECHER, Christian; WECK, Manfred. **Werkzeugmaschinen Fertigungssysteme 1: Maschinenarten und Anwendungsbereiche**. Berlin: Springer Vieweg, 2019. (VDI-Buch). ISBN 978-3-662-46564-6.
- BRONSTEIN, Michael M.; BRUNA, Joan; LECUN, Yann; SZLAM, Arthur; VANDERGHEYNST, Pierre. Geometric Deep Learning: Going beyond Euclidean data. **IEEE Signal Processing Magazine**, Institute of Electrical and Electronics Engineers (IEEE), v. 34, n. 4, p. 18–42, July 2017. ISSN 1558-0792. DOI:

10.1109/msp.2017.2693418. Available from:
<http://dx.doi.org/10.1109/MSP.2017.2693418>.

BUDAK, Erhan; ALTINTAS, Yusuf. Modeling and avoidance of static form errors in peripheral milling of plates. **International Journal of Machine Tools and Manufacture**, v. 33, n. 3, p. 459–476, 1993.

CABRAL, Gustavo Francisco. **Modelling and Simulation of Tool Engagement and Prediction of Process Forces in Milling**. 2015. PhD thesis – RWTH Aachen.

CLAUSEN, Bjorn. **Zerspankraftprognose und -simulation für Dreh- und Fräsprozesse**. 2005. PhD thesis – Universität Hannover.

DATA CAMP. **A Comprehensive Introduction to Graph Neural Networks (GNNs)**. [S. l.: s. n.], 2021. <https://www.datacamp.com/tutorial/comprehensive-introduction-graph-neural-networks-gnns-tutorial>. Accessed: 2025-12-15.

DEFFERRARD, Michaël; BRESSON, Xavier; VANDERGHEYNST, Pierre. **Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering**. [S. l.: s. n.], 2017. arXiv: 1606.09375 [cs.LG]. Available from:
<https://arxiv.org/abs/1606.09375>.

DEUTSCHES INSTITUT FÜR NORMUNG. **DIN 1311-1:2000-02 Schwingungen und schwingungsfähige Systeme – Teil 1: Grundbegriffe, Einteilung**. [S. l.: s. n.], 2000.

DIESTEL, Reinhard. **Graph Theory**. 5th. Berlin, Heidelberg: Springer-Verlag, 2017. v. 173. (Graduate Texts in Mathematics). ISBN 978-3-662-53621-6.

FENG, J. *et al.* Mechanism of Process Damping in Milling of Thin-Walled Workpiece. **International Journal of Machine Tools and Manufacture**, 2018.

FEY, Matthias; LENSSEN, Jan Eric. **PyTorch Geometric: Data Handling Cheat Sheet**. [S. l.: s. n.], 2023. <https://pytorch-geometric.readthedocs.io>. Accessed: 2026.

GANSER, Philipp; VENEK, Tommy; RUDEL, Viktor; BERGS, Thomas. dPart - A Digital Twin Framework for the Machining Domain. **MM Science Journal**, p. 5134–5141, 2021. DOI: 10.17973/MMSJ.2021_11_2021168.

GANSER, Phillip. **Adaptive Milling of Free-Form Surfaces with Circle Segment End Mills**. 2025. PhD thesis – RWTH Aachen.

GAO, Han; Zahr, Matthew J.; WANG, Jian-Xun. Physics-informed graph neural Galerkin networks: A unified framework for solving PDE-governed forward and inverse problems. **Computer Methods in Applied Mechanics and Engineering**, Elsevier BV,

v. 390, p. 114502, Feb. 2022. ISSN 0045-7825. DOI: 10.1016/j.cma.2021.114502. Available from: <http://dx.doi.org/10.1016/j.cma.2021.114502>.

GESELLSCHAFT, Fraunhofer. **About Fraunhofer Gesellschaft**. [S. l.], 2025. Available from: <https://www.fraunhofer.de/en/about-fraunhofer.html>. Visited on: 19 Sept. 2025.

GEY, Christoph. **Prozessauslegung für das Flankenfräsen von Titan, volume Nr.625 of Berichte aus dem Institut für Fertigungstechnik und Werkzeugmaschinen**. 2003. PhD thesis – Universität Hannover.

GILMER, Justin; SCHOENHOLZ, Samuel S.; RILEY, Patrick F.; VINYALS, Oriol; DAHL, George E. **Neural Message Passing for Quantum Chemistry**. [S. l.: s. n.], 2017. arXiv: 1704.01212 [cs.LG]. Available from: <https://arxiv.org/abs/1704.01212>.

HIGH PERFORMANCE CUTTING DEPARTMENT. **Dpart presentation**. Aachen: [s. n.], 2025. Internal Document. Not Published.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 3002-3: Basic Quantities in Cutting and Grinding**. [S. l.: s. n.], 1984.

IPT, Fraunhofer. **dPart A Digital Twin Framework for the Machining Domain**. [S. l.], 2025a. Available from: <https://www.ipt.fraunhofer.de/en/projects/dpart.html>. Visited on: 19 Sept. 2025.

IPT, Fraunhofer. **High performance cutting Modern machining in the digital age**. [S. l.], 2025b. Available from: <https://www.ipt.fraunhofer.de/en/Profile/departments/high-performance-cutting.html>. Visited on: 19 Sept. 2025.

IPT, Fraunhofer. **The Fraunhofer IPT - a profile**. [S. l.], 2025c. Available from: <https://www.ipt.fraunhofer.de/en/Profile.html>. Visited on: 19 Sept. 2025.

KIENAST, Pascal. **Datenbasierte Untersuchung unregelmäßig auftretender Schwingungen im Fräsprozess bei Serienfertigungen von Blisks**. 2021. MA thesis – RWTH Aachen.

KIPF, Thomas N.; WELLING, Max. **Semi-Supervised Classification with Graph Convolutional Networks**. [S. l.: s. n.], 2017. arXiv: 1609.02907 [cs.LG]. Available from: <https://arxiv.org/abs/1609.02907>.

KLOCKE, Fritz. **Fertigungsverfahren 1 - Zerspanung mit geometrisch bestimmter Schneide**. [S. l.]: Springer, 2018.

KOIKE, Yusuke; MATSUBARA, Atsushi; YAMAJI, Iwao. Design method of material removal process for minimizing workpiece displacement at cutting point. **CIRP Annals - Manufacturing Technology**, v. 62, n. 1, p. 419–422, 2013.

KUTTNER, Thomas. **Praxiswissen Schwingungsmesstechnik**. Wiesbaden: Springer Vieweg, 2015.

LADJ, Asma; WANG, Zhiqiang; MESKI, Oussama; BELKADI, Farouk; RITOU, Mathieu; CUNHA, Catherine Da. A knowledge-based Digital Shadow for machining industry in a Digital Twin perspective. **Journal of Manufacturing Systems**, v. 58, p. 168–179, 2020.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **Nature**, v. 521, n. 7553, p. 436–444, 2015.

LI, Qimai; HAN, Zhichao; WU, Xiao-Ming. **Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning**. [S. l.: s. n.], 2018. arXiv: 1801.07606 [cs.LG]. Available from: <https://arxiv.org/abs/1801.07606>.

LING, Xiang; WU, Lingfei; WANG, Saizhuo; MA, Tengfei; XU, Fangli; LIU, Alex X.; WU, Chunming; JI, Shouling. Multilevel Graph Matching Networks for Deep Graph Similarity Learning. **IEEE Transactions on Neural Networks and Learning Systems**, v. 34, n. 2, p. 799–813, 2023. DOI: 10.1109/TNNLS.2021.3102234.

LITWINSKI, Karl M. **Ratterschwingungen an Werkzeugmaschinen**. 2010. PhD thesis – Institut für Fertigungstechnik und Werkzeugmaschinen (IFW).

LIU, Dongsheng; LUO, Ming; URBKAIN, Gorka; TREJO, Daniel Olvera; ZHANG, Dinghua. Position-oriented process monitoring in milling of thin-walled parts. **Journal of Manufacturing Systems**, v. 60, p. 360–372, 2021.

LO, Wai Weng; LAYEGHY, Siamak; SARHAN, Mohanad; GALLAGHER, Marcus; PORTMANN, Marius. E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT. *In*: NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium. [S. l.]: IEEE, Apr. 2022. p. 1–9. DOI: 10.1109/noms54207.2022.9789878. Available from: <http://dx.doi.org/10.1109/NOMS54207.2022.9789878>.

MARKWORTH, A. J. **Fundamentals of Machining**. [S. l.]: ASM International, 2005.

MARTELLOTTI, M. E. An Analysis of the Milling Process. Part II—Down Milling. **Transactions of the ASME**, v. 67, p. 233–251, 1945.

MASLO, Semir. **Simulation of the Dynamic Vibration Behaviour and Spindle Speed Optimization During the Milling Process of Turbomachinery Components**. 2022. PhD thesis – RWTH Aachen University.

MASLO, Semir; MENEZES, Bruno; KIENAST, Pascal; GANSER, Philipp; BERGS, Thomas. Improving Dynamic Process Stability in Milling of Thin-Walled Workpieces by Optimization of Spindle Speed Based on a Linear Parameter-Varying Model. **Procedia CIRP**, v. 126, p. 372–377, 2020.

MEINECKE, Matthias. **Prozessauslegung zum fünfachsigem zirkularen Schrappfräsen von Titanlegierungen**. 2009. PhD thesis – RWTH Aachen University.

OONO, Kenta; SUZUKI, Taiji. **Graph Neural Networks Exponentially Lose Expressive Power for Node Classification**. [S. l.: s. n.], 2021. arXiv: 1905.10947 [cs.LG]. Available from: <https://arxiv.org/abs/1905.10947>.

PFAFF, Tobias; FORTUNATO, Meire; SANCHEZ-GONZALEZ, Alvaro; BATTAGLIA, Peter W. **Learning Mesh-Based Simulation with Graph Networks**. [S. l.: s. n.], 2021. arXiv: 2010.03409 [cs.LG]. Available from: <https://arxiv.org/abs/2010.03409>.

RATCHEV, Svetan; LIU, S.; BECKER, A. A. Error compensation strategy in milling flexible thin-wall parts. **Journal of Materials Processing Technology**, v. 162–163, p. 673–681, 2005.

REULEAUX, Franz. **Die praktischen Beziehungen der Kinematik zu Geometrie und Mechanik**. [S. l.]: Friedrich Vieweg und Sohn, Braunschweig, 1900.

RUDEL, Viktor; KIENAST, Pascal; VINOGRADOV, Georg; GANSER, Philipp; BERGS, Thomas. Cloud-based process design in a digital twin framework with integrated and coupled technology models for blisk milling. **Frontiers in Manufacturing Technology**, v. 2, 2022. DOI: 10.3389/fmtec.2022.1021029.

SANCHEZ-GONZALEZ, Alvaro; GODWIN, Jonathan; PFAFF, Tobias; YING, Rex; LESKOVEC, Jure; BATTAGLIA, Peter W. **Learning to Simulate Complex Physics with Graph Networks**. [S. l.: s. n.], 2020. arXiv: 2002.09405 [cs.LG]. Available from: <https://arxiv.org/abs/2002.09405>.

SCHMITZ, Tony L.; SMITH, Kevin S. **Machining Dynamics: Frequency Response to Improved Productivity**. Boston, MA: Springer, 2009. ISBN 978-0-387-09644-5.

SI, Hao; WANG, Liping. Error compensation in the five-axis flank milling of thin-walled workpieces. **Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture**, v. 233, n. 3, p. 459–476, 2019.

STABLER, G. V. The chip flow law and its consequences. In: TOBIAS, S. A.; KOENIGSBERGER, F. (eds.). **Advances in Machine Tool Design and Research**. Oxford, UK: Pergamon Press, 1964. p. 243–251.

SUTTON, Richard S.; BARTO, Andrew G. **Reinforcement Learning: An Introduction**. [S. l.]: MIT Press, 2018.

VELICKOVIC, Petar; CUCURULL, Guillem; CASANOVA, Arantxa; ROMERO, Adriana; LIO, Pietro; BENGIO, Yoshua. **Graph Attention Networks**. [S. l.: s. n.], 2018. arXiv: 1710.10903 [stat.ML]. Available from: <https://arxiv.org/abs/1710.10903>.

WAIBEL, Michael. **Aktive Zusatzsysteme zur Schwingungsreduktion an Werkzeugmaschinen**. 2013. PhD thesis – Technische Universität München.

WECK, Manfred. **Werkzeugmaschinen**. Berlin: Springer, 2006a.

WECK, Manfred. **Werkzeugmaschinen**. Berlin: Springer, 2006b. (Studium und Praxis). ISBN 3-540-22502-1.

XU, Keyulu; HU, Weihua; LESKOVEC, Jure; JEGELKA, Stefanie. **How Powerful are Graph Neural Networks?** [S. l.: s. n.], 2019. arXiv: 1810.00826 [cs.LG]. Available from: <https://arxiv.org/abs/1810.00826>.

YANG, Zhenyu *et al.* **State of the Art and Potentialities of Graph-level Learning**. [S. l.: s. n.], 2023. arXiv: 2301.05860 [cs.LG]. Available from: <https://arxiv.org/abs/2301.05860>.

YING, Rex; BOURGEOIS, Dylan; YOU, Jiaxuan; ZITNIK, Marinka; LESKOVEC, Jure. **GNNExplainer: Generating Explanations for Graph Neural Networks**. [S. l.: s. n.], 2019. arXiv: 1903.03894 [cs.LG]. Available from: <https://arxiv.org/abs/1903.03894>.

ZAHEER, Manzil; KOTTUR, Satwik; RAVANBAKSH, Siamak; POCZOS, Barnabas; SALAKHUTDINOV, Ruslan; SMOLA, Alexander. **Deep Sets**. [S. l.: s. n.], 2018. arXiv: 1703.06114 [cs.LG]. Available from: <https://arxiv.org/abs/1703.06114>.

ZHANG, Y.; SIMS, N. D. Milling Workpiece Chatter Avoidance Using Piezoelectric Active Damping: A Feasibility Study. **Smart Materials and Structures**, v. 14, n65–n70, 2005.