4EU+ International Workshop on Recent Advancements in Artificial Intelligence

Edited by Ruggero Donida Labati, Angelo Genovese, Vincenzo Piuri

Milano University Press

Contents

main.pdf 1

Pagina vuota 1

Pagina vuota 3

Title page 4

Colophon 5

Table of contents 5

Preface (English) 9

Prefazione (Italiano) 11

Chapter 1. Introduction 14

Chapter 2. Rigid Point Cloud Registration 16

Chapter 3. Deep Learning for PCG of 2D Video Game Levels 30

Chapter 4. Text Style Transfer: An Introductory Overview 48

Chapter 5. Simulation Study on Super-Resolution for Coded Aperture Gamma Imaging 68

Chapter 6. Deep Learning Applications to Particle Physics: A Review 84

4EU+ International Workshop on Recent Advancements in Artificial Intelligence

edited by Ruggero Donida Labati,
Angelo Genovese, Vincenzo Piuri

Milano University Press

4EU+ InternationalWorkshop on Recent Advancements in Artificial Intelligence/ edited by Ruggero Donida Labati, Angelo Genovese, Vincenzo Piuri. Milan: Milano University Press, 2026.

ISBN 979-12-5510-378-3 (print)
ISBN 979-12-5510-382-0 (PDF)
ISBN 979-12-5510-386-8 (EPUB)
DOI 10.54103/milanoup.282

For this volume, the Editorial Board of the press has authorised an internal or editorial review process in place of the usual external peer review. All published works are evaluated and approved by the Editorial Board, and must be compliant with the Review policy, the Open Access, Copyright and Licensing policy and the Publication Ethics and Complaint policy as reflected in MilanoUP publishing guidelines (Linee Guida per pubblicare su MilanoUP).

The present work is released under Creative Commons Attribution 4.0 - CC-BY, the full text of which is available at the URL:
https://creativecommons.org/licenses/by/4.0/

This and other volumes of Milano University Press are available in open access at:
https://libri.unimi.it/index.php/milanoup

© The Authors for the text 2026
© Milano University Press for this edition

Published by Milano University Press
Via Festa del Perdono 7 20122 Milano
Web Site: https://milanoup.unimi.it
e-mail: redazione.milanoup@unimi.it

The print edition of this volume can be ordered from bookstores, and is distributed by Ledizioni (www.ledizioni.it)

Table of contents

Preface (English) **7**
Prefazione (Italiano) **9**

1 Introduction **11**

2 Rigid Point Cloud Registration **13**
2.1 Abstract 14
2.2 Introduction 14
2.3 Related work 15
2.4 Problem Definition 16
2.5 Proposed Method 16
2.5.1 Feature Extraction 17
2.5.2 Corresponding point determination based on graph representation 18
2.5.3 Transformation matrix calculation 19
2.6 Results 20
2.6.1 Dataset 21
2.6.2 Experimental results 21
2.7 Discussion 24
2.8 Conclusion 24
Acknowledgment 25
Bibliography 25

3 Deep Learning for PCG of 2D Video Game Levels **27**
3.1 Abstract 27
3.2 Introduction 28
3.3 GANs in brief 29
3.4 2D Video Games Data Representation of Levels for GANs 30
3.4.1 VGDL representation 31
3.4.2 Maps representation 32

3.5 DOOM levels generation with GANs . . . 32
3.5.1 Data-set . . . 32
3.5.2 Deep Level Generation . . . 33
3.5.3 Generated Levels Evaluation . . . 33
3.6 Bootstrapping Conditional GAN for level generation of The Legend of Zelda 35
3.6.1 Data-set . . . 35
3.6.2 Deep Level Generation . . . 35
3.6.3 Generated Levels Evaluation . . . 37
3.7 Level Generation for Multiple Types of 2D Video Games with Common Latent Space with VGDL . . . 38
3.7.1 Data-set . . . 39
3.7.2 Deep Level Generation . . . 39
3.7.3 Generated Levels Evaluation . . . 40
3.8 Conclusion . . . 41
Bibliography . . . 42

4 Text Style Transfer: An Introductory Overview . . . 45
4.1 Abstract . . . 45
4.2 Introduction . . . 46
4.3 The Task . . . 47
4.3.1 Understanding Style and Content . . . 47
4.3.2 Problem Formulation . . . 48
4.4 Challenges . . . 48
4.5 Datasets and Benchmarks . . . 49
4.6 Text Style Transfer Approaches . . . 51
4.6.1 Supervised Training on Parallel Data . . . 51
4.6.2 Non-parallel Approaches . . . 52
4.6.3 Using Large Language Models . . . 53
4.7 Evaluation Measures . . . 53
4.7.1 Automatic Evaluation . . . 53
4.7.2 Human Evaluation . . . 54
4.8 Applications . . . 54
4.9 Ethical Concerns . . . 55
4.10 Conclusion . . . 56
Acknowledgment . . . 56
Bibliography . . . 56

5 Simulation Study on Super-Resolution for Coded Aperture Gamma Imaging . 65
5.1 Abstract . . . 66
5.2 Introduction . . . 66
5.3 Methods . . . 68
5.3.1 Simulating a coded aperture test image . . . 68

5.3.2 Measured data from the experimental gamma-camera 68
5.3.3 Generating low-resolution detector images 69
5.3.4 Analytical reconstruction methods 69
5.3.5 Reconstruction by Machine Learning 71
5.3.6 Nyquist-Shannon sampling theorem 72
5.4 Results 73
5.4.1 Critical super-resolution factors 73
5.4.2 Results on the test image 73
5.4.3 Results on the measurement data 76
5.5 Discussion 76
5.5.1 Simulated test image 76
5.5.2 Measured phantom data 76
5.6 Conclusion 77
Bibliography 77

6 Deep Learning Applications to Particle Physics: A Review 81
6.1 Abstract 81
6.2 Introduction and motivation 81
6.3 Jet physics 83
6.3.1 Tagging 84
6.3.2 Pileup mitigation 87
6.4 Non-collider physics (neutrino) 90
6.5 Tracking 93
6.6 Conclusion 96
Bibliography 96

Preface (English)

Artificial intelligence is increasingly pervasive in a broad variety of applications and in our daily life, from scientific applications to industrial manufacturing, from medical applications to biomedical systems, from ambient intelligence to consumer electronics, from entertainment to games, from communications to social networks, from finance to marketing, and many more. Addressing the growing needs of smart applications and expanding our knowledge on the foundations and the opportunities offered by artificial intelligence are essential to support for advancing science and technology as well as to provide the critical support for social and economic development.

The 4EU+ European University Alliance has been created and funded by the European Commission to create a comprehensive research-intensive European university, an integrated environment promoting and supporting research and higher education based on research. This Alliance aims to set a new standard of cooperation in education, research, innovation, and outreach by leveraging and expanding the research capacity and strong multi- and interdisciplinary profiles of the partner universities.

Within this vision, the AU4+ group of research and education experts on Transforming Science and Society: Advancing Information, Computation and Communication (Flagship 3 of the Alliance) organizes several activities for promoting research collaboration and education initiatives in a variety of topics related to information and communication technologies and their multi/inter-disciplinary foundations, including artificial intelligence. In 2022, the 4EU+ Summer School on Artificial Intelligence has been organized in Gargnano del Garda, Italy, to stimulate the interest of PhD students and young researchers in various fields of artificial intelligence, specifically in the scientific and technological foundations and the related applications.

Attendees have been suggested to present their ideas and research and contribute to research discussion in the 4EU+ International Workshop on Recent Advancements in Artificial Intelligence, organized in conjunction to the summer school, as an open forum for multi/inter-disciplinary discussion. Sharing and discussing advanced knowledge on the theory, techniques, methodologies, and applications of artificial intelligence has

been a stimulating experience, contributing to dive in advanced concepts, showcasing achievements, innovations, and opening new opportunities.

Participants submitted their manuscript for review and possible inclusion in the proceedings of the workshop. Only a limited number of papers have been accepted after a peer review process. This volume contains these papers, the most stimulating ideas and results that have been discussed in the workshop. We are sure that you will enjoy reading them.

Ruggero Donida Labati, Angelo Genovese, and Vincenzo Piuri
Editors

Prefazione (Italiano)

L'Intelligenza Artificiale è sempre più pervasiva in un'ampia varietà di applicazioni e nella nostra vita quotidiana: dalle applicazioni scientifiche alla produzione industriale, dalle applicazioni mediche ai sistemi biomedici, dall'intelligenza ambientale all'elettronica di consumo, dall'intrattenimento ai giochi, dalle comunicazioni ai social network, dalla finanza al marketing, e in molti altri ambiti. Rispondere ai crescenti bisogni di applicazioni intelligenti ed espandere la nostra conoscenza relativa ai fondamenti e alle opportunità offerte dall'intelligenza artificiale è essenziale per il progresso della scienza e della tecnologia, oltre che per fornire un supporto critico allo sviluppo sociale ed economico.

La 4EU+ European University Alliance è stata istituita con l'obiettivo di costituire un'università europea completa, ad alta intensità di ricerca, un ambiente integrato che promuova e sostenga la ricerca e l'istruzione superiore basata sulla ricerca. Questa Alleanza mira a stabilire un nuovo standard di cooperazione nell'ambito dell'istruzione, della ricerca, dell'innovazione e della divulgazione, valorizzando ed espandendo la capacità di ricerca e gli importanti profili eterogenei e interdisciplinari delle università partner.

In questa prospettiva, il gruppo AU4+ di esperti di ricerca e formazione Transforming Science and Society: Advancing Information, Computation and Communication (Flagship 3 dell'Alleanza) organizza diverse attività per promuovere la collaborazione nella ricerca e iniziative educative in vari ambiti legati alle tecnologie dell'informazione e della comunicazione e alle loro basi multi/inter-disciplinari, inclusa l'intelligenza artificiale. Nel 2022 è stata organizzata a Gargnano del Garda, Italia, la 4EU+ Summer School on Artificial Intelligence, con l'obiettivo di stimolare l'interesse di dottorandi e giovani ricercatori in diversi settori dell'intelligenza artificiale, in particolare nei fondamenti scientifici e tecnologici e nelle relative applicazioni.

Ai partecipanti è stato suggerito di presentare le proprie idee e ricerche e di contribuire alla discussione scientifica nel 4EU+ International Workshop on Recent Advancements in Artificial Intelligence, organizzato in concomitanza con la scuola estiva, come forum aperto per il confronto multi/inter-disciplinare. Condividere e discutere conoscenze avanzate sulla teoria, le tecniche, le metodologie e le applicazioni dell'intelligenza artificiale è stata

un'esperienza stimolante, che ha permesso di approfondire concetti avanzati, mostrare risultati e innovazioni e aprire nuove opportunità.

I partecipanti hanno inviato i propri manoscritti per la revisione e l'eventuale inclusione negli atti del workshop. Solo un numero limitato di articoli è stato accettato dopo un processo di revisione tra pari. Questo volume raccoglie tali articoli: le idee e i risultati più stimolanti discussi durante il workshop. Siamo certi che la loro lettura sarà di vostro interesse e gradimento.

Ruggero Donida Labati, Angelo Genovese e Vincenzo Piuri
Curatori

Chapter 1

Introduction

DOI: 10.54103/milanoup.282.c633

Artificial intelligence is increasingly pervasive in a broad variety of applications and in our daily life, from scientific applications to industrial manufacturing, from medical applications to biomedical systems, from ambient intelligence to consumer electronics, from entertainment to games, from communications to social networks, from finance to marketing, and many more. Addressing the growing needs of smart applications and expanding our knowledge on the foundations and the opportunities offered by artificial intelligence are essential to support for advancing science and technology as well as to provide the critical support for social and economic development.

The AU4+ group of research and education experts on Transforming Science and Society: Advancing Information, Computation and Communication (Flagship 3 of the Alliance) organizes several activities for promoting research collaboration and education initiatives in a variety of topics related to information and communication technologies and their multi/inter-disciplinary foundations, including artificial intelligence.

The 4EU+ Summer School on Artificial Intelligence has been organized in Gargnano del Garda, Italy, to stimulate the interest of PhD students and young researchers in various fields of artificial intelligence, specifically in the scientific and technological foundations and the related emerging applications.

Sharing and discussing advanced knowledge on the theory, techniques, methodologies, and applications of artificial intelligence has been a stimulating experience, contributing to dive in advanced concepts, showcasing achievements, discoveries, and innovations, and opening new opportunities.

This book includes the following contributions:

- S. Monji Azad, M. Kinz, M. Fotouhi, J. Hesser, “A Rigid Point Cloud Registration Method Based on Global Feature Learning and Graph Representation”;
- E. Chitti, “Deep Learning for PCG of 2D video game levels”;
- S. Mukherjee, Dušek, “Text Style Transfer: An Introductory Overview”;
- T. Meißner, W. Nahm, J. Hesser, M. Löw, “Simulation Study on Super-Resolution for Coded Aperture Gamma Imaging”;
- M. Rossi, “Deep Learning Applications to Particle Physics: A Review”.

Chapter 2

A Rigid Point Cloud Registration Method Based on Global Feature Learning and Graph Representation

Sara Monji Azad
Mannheim Institute for Intelligent Systems in Medicine
Heidelberg University, Germany
sara.monjiazad@medma.uni-heidelberg.de
ORCID: 0000-0002-2742-9961

Marvin Kinz
Mannheim Institute for Intelligent Systems in Medicine
Heidelberg University, Germany
marvin.kinz@std.uni-heidelberg.de
ORCID: 0009-0008-8226-5905

Mehran Fotouhi
Department of Computer Engineering
Sharif University of Technology, Iran
mehran.fotouhi@alum.sharif.edu

Jürgen Hesser
Mannheim Institute for Intelligent Systems in Medicine
Interdisciplinary Center for Scientific Computing (IWR)
Central Institute for Computer Engineering (ZITI)
CZS Heidelberg Center for Model-Based AI
Heidelberg University, Germany
juergen.hesser@medma.uni-heidelberg.de

DOI: 10.54103/milanoup.282.c634

2.1 Abstract

Point cloud registration involves estimating spatial transformations between point sets. Traditional registration approaches are categorized into rigid and non-rigid transformations. This paper introduces a rigid point cloud registration method leveraging the PointNet method to learn global shape features, focusing on graph representation during the registration step. Informative points are extracted as global features, represented as graphs, and then matched to compute a transformation matrix for registration. Experimental results demonstrate that the proposed algorithm significantly improves the mean square error (MSE) from 4.94×10^{-5} to 1.74×10^{-5} on average across 10 categories of the ModelNet10 dataset. Additionally, the method achieves superior accuracy for the chair category in the ShapeNet dataset compared to GP-Aligner and Norm-IP, while also reducing computational time through the graph intermediate representation.

2.2 Introduction

A point cloud is a set of data points for the representation of 2D or 3D shapes. Point sets find their use in various applications, namely 3D localization [1], 3D reconstruction [2], free-viewpoint generation [3], augmented/virtual reality [4], etc.

Point cloud registration is one of the most fundamental challenging areas in computer vision [5]. The primary goal of point cloud registration is to estimate the transformation between two or more corresponding point sets. Hence, the error between the transformed point cloud and the target one should be minimized. The registration approaches are categorized into rigid and non-rigid transformations. To be more accurate, in rigid registration the geometric extensions and shape of an object are invariant under affine transformations such as translation and rotation while for non-rigid registration the deformation field for the source point cloud should be determined [6].

A point cloud registration method based on rigid transformation is proposed in this paper. The proposed approach consists of three main steps, namely feature extraction, corresponding point determination based on a graph representation, and the transformation matrix calculation. The first step is based on PointNet [7] to extract global features which are the informative points of the input point clouds. In the second step, a graph representation of the extracted informative points is constructed. In the mentioned graph, each point represents an informative point. To this end, graph matching is applied to find the initial corresponding points. Then, the initial correspondences are used to estimate the transformation matrix. All steps of the proposed method are explained in Section 2.5 in more detail.

A summarization of the proposed method's novel parts can be discussed in several points:

- Considering the fact that PointNet [7] is able to extract critical points which can show the structure of an object, the proposed method extracts the efficient landmarks of an object by taking the advantage of PointNet.
- The proposed method constructs a star graph based on the extracted landmarks which are efficient to determine the initial corresponding points.
- In the proposed approach, the registration problem is solved using the graph-matching approach to increase the time efficiency as well as improving the accuracy of the matching step.
- The proposed method calculates the system equation to determine the affine transformation matrix. The equation systems are computed based on precise initial corresponding points which increasingly decrease the transformation matrix error.

This paper makes the following contribution:

- The proposed method calculates the accurate transformation by using the graph-matching approach.
- Experiments show the proposed method is robust to different affine transformations (rotation and translation).
- The proposed graph intermediate representation reduces the computational time.

The rest of this paper is organized as follows. Section 2.3 outlines the related work. Section 2.4 provides the problem definition of rigid point cloud registration. The proposed method is explained in Section 2.5. The experimental dataset and evaluation performance are described in Section 2.6. A discussion on the potential and limitations of the proposed method is presented in Section 2.7. Finally, Section 2.8 concludes the paper.

2.3 Related work

The registration problem is typically tackled by methods that are using either non-learning approaches or learning-based methods. Non-learning approaches, in particular, suffer huge computational effort. Hence, using non-learning approaches to solve the registration problem in a real-time manner is the most significant challenge. There are various approaches to solving the registration problem based on learning approaches. Some of them focus on generating an accurate feature descriptor to take local and global features into account. For the same purpose, some others concentrate on finding the best representation for the point cloud.

From another point of view, learning-based methods use different network architectures. Some architectures are more common like MLP, CNN, and GNN. In the following, some of the most cited papers are studied. However, the available studies are not limited to the mentioned categorizations.

Given that point clouds are unordered structures that, unlike images, do not have neighborhood information, there are some approaches to provide different representations of point clouds that are used in various registration studies. The intermediate representation is a common approach to overcoming the unordered problem. Voxels, graphs, and 3DMeshs are some samples of the intermediate representation, to mention a few. In this way, applying CNN architecture, using point clouds as the network's input, can be a challenge [8]. There are different approaches to overcoming these challenges. Deep closet point (DCP) [9], TIF-Reg [10], and the deep global registration (DGR) method [11] are some studies of rigid point cloud registration based on the CNN network.

GNN is another frequently used network to learn neighborhood information. Usually, each graph's node represents its local features which is one of the reasons for achieving good accuracy. GNN is not only used as a local feature representation, but also as a matching step. FIRE-Net is a rigid and interactive representation learning-based method [12]. G^2 Net [13] is another geometric guided network for both full and partial rigid point cloud registration problems. Also, a graph matching method (RGM) is proposed in [14].

In another categorization, point cloud registration methods are categorized into coarse and fine approaches. The coarse-based registration methods calculate an initial geometric alignment. However, estimating a transformation to solve registration problems as precisely as possible is called the fine-based registration method. Iterative Closest Point (ICP) [15] and Coherent Point Drift (CPD) [16] are two of the most well-known fine registration approaches.

2.4 Problem Definition

A point cloud P is a set of a fixed number n of 3D points $P = \{p_i \in P | i = 1, 2, ..., n\}$. The source point cloud (S) and the target point cloud (T) are defined as a set of $Data = \{(S, T) | S, T \in \mathbb{R}^{N \times n}\}$ where N is the dimension and n is the number of points.

A transformation is a parameterized mapping from the source to the target domain, in particular, the transform is defined by a rotation matrix $R(\theta)$, and a translation vector $\vec{t}$; we define $\alpha = \{R(\theta), \vec{t}\}$ as the free parameters of the transform $Trans(\alpha, S)$ where S is the source point cloud that should be transformed.

The metric defines the similarity of point clouds (T), $Trans(\alpha, S)$; whereby for our purpose, we use the Euclidean distance measure as the criterion of best fit. In particular, we assume that for point cloud registration, for each $s_i \in S$ there is a corresponding $t_i \in T$ as the corresponding point.

2.5 Proposed Method

In our approach, point cloud registration is based on identifying the corresponding points in the source and destination point clouds, and calculating the

transformation matrix, subsequently.

As shown in Figure 2.1 our method consists of three main steps, namely feature extraction, corresponding point determination based on a graph representation, and transformation matrix calculation. In the following, each step is discussed in detail.

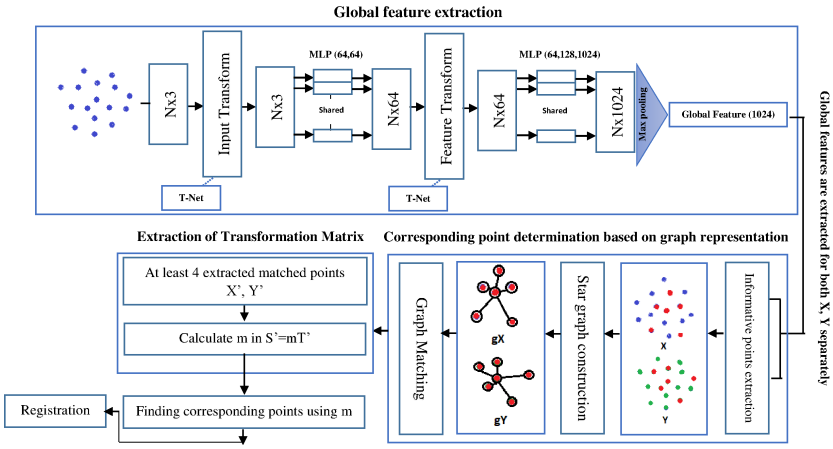


Figure 2.1 *The overview of the proposed method. The proposed method has three main steps: global feature extraction based on PointNet [7], corresponding point determination based on graph representation, and transformation matrix calculation.*

2.5.1 Feature Extraction

The learning-based feature descriptor is the first step of the proposed method. For the given point clouds, global features are extracted using a PointNet-based architecture. PointNet [7] is a very efficient method for feature extraction in learning-based registration methods.

The input is 3D point cloud $\{P_i | i = 1, ..., n\}$ where each point P_i is a vector $(x, y, z) \in \mathbb{R}^3$. The feature extraction step has two main modules, one is a T-net and the other one is a max pooling layer.

T-net [17] is a mini-network to predict an affine transformation matrix to map a point cloud to canonical space before processing. Then, the estimated transformation matrix is directly applied to the coordinates of input points. T-net is applied two times in our network architecture. For the first time, it is applied to $N \times 3$ coordinates, and the second time when feature vectors are transformed within the feature space with the dimension of $N \times 64$.

Max pooing as the second module is a function used to extract an order-invariant descriptor. It calculates the maximum value for patches of a feature map

and creates a downsampled or pooled feature map. In this paper, the output of the max pooling layer is called global features or informative points. Global features consist of a vector with 1024 elements which is calculated based on a $N \times 1024$ matrix from the previous layer, where N is the number of points in one point set. In Figure 2.2, the output of max pooling is shown.

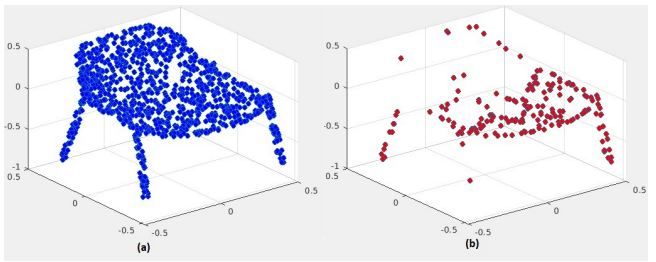


Figure 2.2 *Informative points representation to show the output of global shape features (a) Input point cloud (b) global feature representation*

2.5.2 Corresponding point determination based on graph representation

After extracting the global features from the first step of the proposed method, the extracted informative points are used to provide a new representation of the input point cloud. This representation is coded as a graph for source and target point clouds respectively. The informative points are hereby the vertices $v_i \in V$ of the graph $G = (V, E)$, and the edges $e_j \in E$ are the connectives between the central vertex to other vertices by defining a central node $v_c \in V$ and edges $e_j = (v_c, v_i), i \neq c, v_i \in V$. To reduce the computational time, instead of using a fully connected graph or a graph with a high number of edges, a star graph of informative points is constructed. Furthermore, using the median as a robust estimate for the point cloud center gives us the first certain corresponding point which is invariant under translation and rotation. The star graph is a type of graph in which the degree of $n-1$ vertices is 1 and the degree for one vertex is degree $n-1$. To be more precise, $n-1$ vertices are connected to a central vertex. A sample of the constructed star graph from the chair informative points is illustrated in Figure 2.3. For determining the corresponding points, star graphs for both source and target point clouds are constructed. Considering the fact that a rigid transformation (translation and rotation) does not change the distance between points, finding the edges v_c with unique weights can be used to estimate initial accurate corresponding points. Therefore, first, the unique edges (v_c, v_i) are extracted for both source and target point clouds, separately. Then, their vertices v_c can be considered as initial corresponding points.

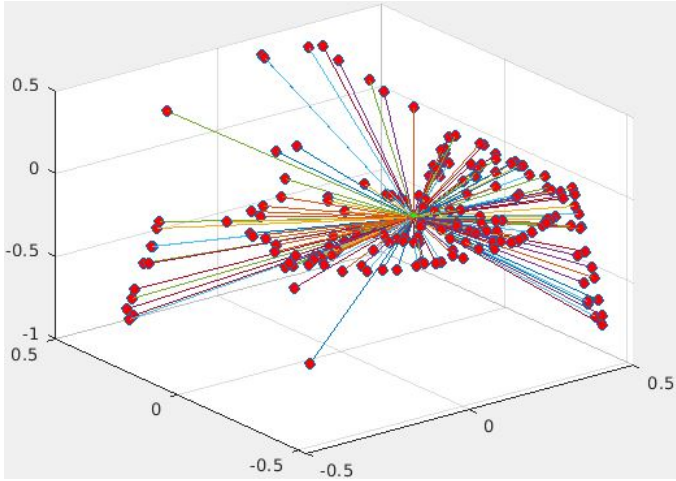


Figure 2.3 *A constructed star graph of the informative points of a chair*

2.5.3 Transformation matrix calculation

The transformation between the source and target point cloud is calculated through the initial corresponding points. Although at least four corresponding points are needed for calculating the transformation matrix, the number of unique edges is usually more than this value. Therefore, the limitation of the number of points is not a problem for the proposed method. To calculate the transformation matrix, the coordinates of the source and target point clouds are considered to be homogeneous. To be more precise, each point is shown as $(x, y, z, 1)$. Then, the transformation matrix m between the source and target point clouds is a 4 matrix which is a combination of various transformations.

$$m = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

By considering S as the source point (or a vector) and T as the target point (or a vector), the matrix m presents an affine transformation that transforms S from one coordinate system to T in another coordinate system. The m shown in Equ. 2.1, is combined of translation, rotations, scalings/reections, and shears. As discussed in Section 2.4, this paper is limited to calculating the translation and rotation. In this regard, the elements $[a11, a12, a13, a21, a22, a23, a31, a32, a33]$ represent the rotation transform and the elements of the last column $[a14, a24, a34]$ represent the translation transform. To calculate the transformation between the source point set (S) and the target one (T), matrix m should be computed. Matrix m is obtained by

calculating the system equation which is

$$T = mS \tag{2.2}$$

The transformation of the point x to point x' is thus written as $x' = mx$. This mapping can be shown as a matrix by

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.3}$$

For calculating the elements of matrix m, the system of matrix equation that is shown in Equ. 2.2 needs to be solved. The equations can be shown by

$$\begin{cases} x' = a11x + a12y + a13z + a14 \\ y' = a21x + a22y + a23z + a24 \\ z' = a31x + a32y + a33z + a34 \end{cases} \tag{2.4}$$

These are the equations that define the relationship between the matrices T, m, and S. To find the matrix m, we can solve these equations for the elements of m in terms of elements of T and S. It is worth mentioning that points and vectors are represented as mathematical column vectors in homogeneous coordinates with the difference that points have a 1 in the fourth position whereas vectors have a zero of that. For vectors, the value in row number 4 is 0 instead of 1 to remove the translation operation by multiplying the 4th vector of matrix m by $0(1 = 0x+0y+0z+1)$.

After calculating m based on the accurate initial corresponding points, all points can be transferred from the source point cloud to the target one based on the calculated m. Then, the distance between the moving points and the target ones is calculated to show how accurate the proposed method is which will be discussed in Section 2.6.

2.6 Results

To show the accuracy of the proposed method, the approach is evaluated on the ModelNet10 dataset and the results are compared with a rigid registration method proposed in [18]. In the following, the used dataset to evaluate the accuracy of the method will be studied. Afterward, the results of the proposed method will be presented.

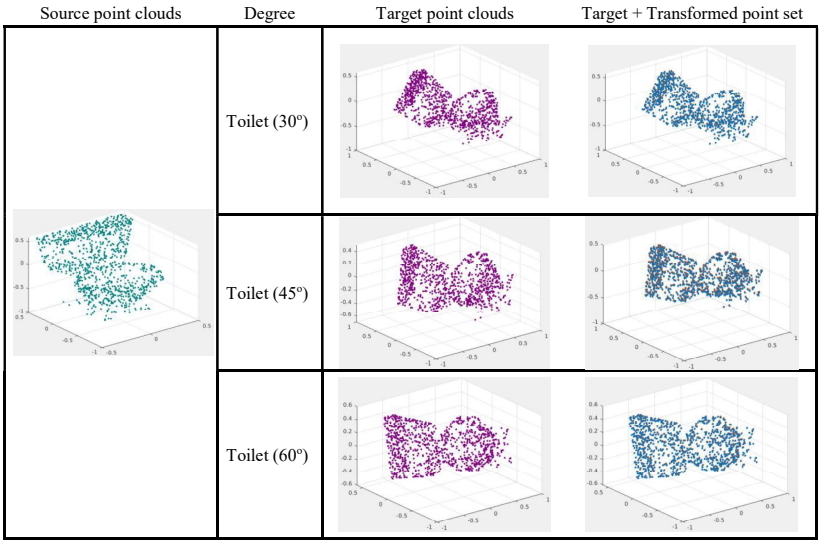


Figure 2.4 *The results of the proposed method on the toilet category. The results are shown for three different degrees of rotation, namely* $30°, 45°, 60°$. *The source point cloud is shown in the first column. Different degrees of target ones are shown in the second column. The third column shows the result after applying the transformation. The output of the proposed method is shown in the last column.*

2.6.1 Dataset

ModelNet [19] is one of the most used synthetic datasets which is provided in two subsets: ModelNet10 and ModelNet40. ModelNet40 contains 40 categories composed of 12311 CAD models while ModelNet10 contains 10 categories including 4899 CAD models in mesh format. The dataset is divided into train and test data, consisting of 3991 and 908 models, respectively. Furthermore, the Chair category of ShapeNet [20] is used to report the accuracy of the proposed method as well as the time efficiency.

In this paper, the test data from ModelNet10 and ShapeNet are considered as source point clouds. Then, to generate the target ones different degrees of rotation and random translation values for (x, y, z) are applied to the source point clouds. Moreover, to evaluate the method on the ShapeNet dataset, the objects are randomly selected from the armchair, folding chair, and swivel chair categorizations.

2.6.2 Experimental results

It is worth mentioning that to evaluate the proposed method, the data can be categorized into three parts, namely source point clouds, target point clouds, and

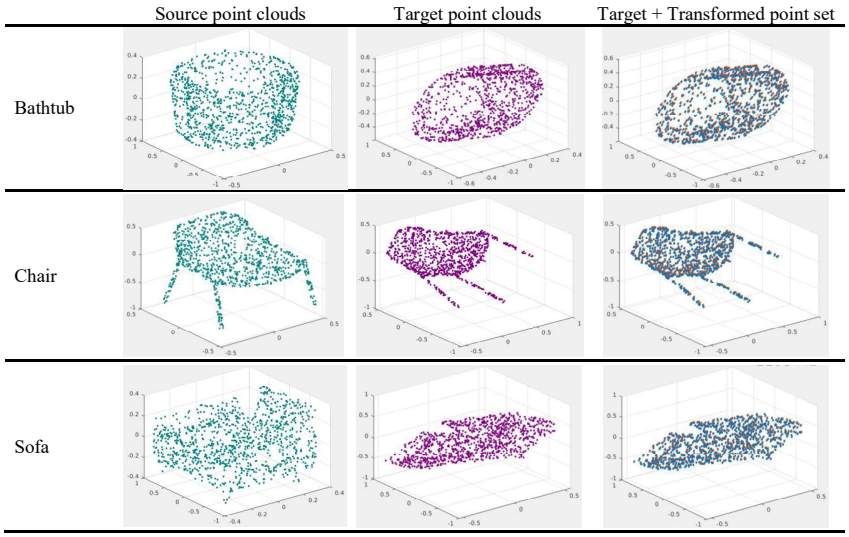


Figure 2.5 *The results of the proposed method. The results are shown for four categories: bathtub, chair, desk, and sofa. The source and target point clouds are shown in the second and third columns. The final results of the proposed registration method are shown in the last column which is called outputs.*

Table 2.1 *The accuracy of the proposed method is provided and compared with the method [18]. The results are shown as distance MSE for 10 categories of ModelNet10.*

Categories	Proposed Method	Method [18] (Distance MSE)	Method [18] (Rotation Error (deg.))
Bathtub	1.21×10^{-5}	4.92×10^{-5}	1.16
Bed	2.28×10^{-5}	4.41×10^{-5}	1.15
Chair	2.39×10^{-5}	5.03×10^{-5}	1.29
Desk	0.77×10^{-5}	5.26×10^{-5}	1.31
Dresser	1.82×10^{-5}	5.18×10^{-5}	1.04
Monitor	1.33×10^{-5}	3.83×10^{-5}	1.03
Night-stand	1.75×10^{-5}	5.04×10^{-5}	1.08
Sofa	1.09×10^{-5}	3.99×10^{-5}	1.10
Table	2.89×10^{-5}	6.36×10^{-5}	1.49
Toilet	1.89×10^{-5}	5.56×10^{-5}	1.15
Average	$\mathbf{1.74 \times 10^{-5}}$	$\mathbf{4.94 \times 10^{-5}}$	**1.18**

transformed point clouds. As source point clouds, the test data from ModelNet is considered. After applying different rigid transformations (different rotation and translation), the target point clouds are generated. Therefore, to study the proposed method's accuracy, a distance measurement is calculated between the target point clouds and the transformed ones. The Mean Square Error (MSE) is used to calculate

the distance error shown in Equ. 2.5.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2.5}$$

Furthermore, the results are evaluated in one more measurement called Chamfer distance (C. D.) which is shown in Equ. 2.6.

$$\begin{aligned} CD\,(S,T) = \frac{1}{|S|}\sum_{x\in S}\min_{x\in T}\|x-y\|_2^2 + \\ \frac{1}{|T|}\sum_{y\in T}\min_{x\in S}\|x-y\|_2^2 \end{aligned} \tag{2.6}$$

The performance of the presented method is analyzed by objective and subjective measures which will be discussed in the following.

Subjective measurement For subjective measurement evaluation, the results of the proposed method in different situations are demonstrated. As mentioned before, each target is generated in different degrees of rotation and random translation values for (x, y, z). Figure 2.4 shows an example of the proposed method output on the toilet category. In another evaluation, the final results for some categorizations are presented in Figure 2.5.

Objective measurement To demonstrate the accuracy of the calculated transformation matrix, the proposed method is implemented for all 10 categories. The MSE measurement is run on each category between the target point cloud and the output of the proposed method. The result should be minimized. Therefore, a value of zero corresponds to a registration accuracy of one hundred percent. The proposed method's accuracy is shown in Table 2.1. Furthermore, the results of method [18] are shown in this table to compare the accuracy of each category. In comparison with the other method, it can be concluded that the proposed method could achieve higher accuracy. Furthermore, results for the Chair category in the ShapeNet dataset in comparison with GP-Aligner [21] and Norm-IP [22] are presented in Table 2.2.

Table 2.2 *The accuracy of the proposed method in comparison with the methods [21] and [22]. The results are shown as Chamfer distance (C. D.) for the Chair category. The running- time of the methods is shown in the last column.*

Methods	C. D.	Time
Norm-IP [22]	$21.42 \times 10^{-}4$	4328s
GP-Aligner (Shape-Wise) [21]	$3.10 \times 10^{-}4$	16s
GP-Aligner (GroupWise) [21]	$2.20 \times 10^{-}4$	16s
Proposed Method	$2.159 \times 10^{-}5$	0.0024s

The amount of time an algorithm needs to execute is known as run-time efficiency. Various factors affect the execution speed such as the parallelism of the program, the hardware utilization, and the choice of programming language, to name a few. Regarding the mentioned factors, comparing the time efficiency of the approaches based on time can be unfair. All in all, in this paper, to provide an overview of the proposed method's running-time efficiency, the run-time in comparison with three other methods is presented. To study the time efficiency of the proposed method, the running time of the proposed method, GP-Aligner [21], and Norm-IP [22] are reported in Table 2.2.

It is worth mentioning that no parallelization has been done in the implementation of the proposed method and the presented time in Table 2.2 for all methods is for running the test dataset.

In the available studies beside reporting distance and loss function, the methods' robustness to noise also is reported. In this regard, two various dataset types are discussed in the literature, namely real-world datasets and synthetic ones. Real-world datasets which are commonly captured using LiDAR cameras or scanners suffer from different types of noise. This is not the case with synthetic datasets and it is usually added synthetically to the dataset to challenge proposed approaches [6]. In this paper, a synthetic dataset called ModelNet and ShapeNet is used and the noise challenge is not studied which can be considered as a future study for this paper.

2.7 Discussion

A rigid point cloud registration method based on global feature learning and graph representation is presented in this paper. The proposed method provides several contributions. One is the accurate transformation by using graph matching which is robust to different affine transformations (rotation and translation).

Considering the fact that the point cloud has an unordered structure, the proposed method takes the advantage of PointNet architecture, using max pooling, to overcome this challenge. Furthermore, PointNet architecture is used to extract global features as input for the next steps. Hence, PointNet makes the extracted features robust to different affine transformations. Additionally, the proposed corresponding point determination based on graph representation and graph matching makes the approach robust to different transformations. It is worth mentioning that regarding the start graph representation, the computation time is low in comparison with the complete graph.

2.8 Conclusion

In this paper, an efficient rigid point cloud registration method was proposed. The proposed method had three main steps, namely feature extraction, corresponding point determination based on graph representation, and transformation matrix

calculation. It is shown in the results that the proposed method has high accuracy in rigid registration and that it is robust to different degrees of rotation and various values of translation. In future work, it can be studied how the proposed method can be improved to estimate non-rigid transformations.

Acknowledgment

The authors gratefully acknowledge the data storage service SDS@hd supported by the Ministry of Science, Research and the Arts Baden-Württemberg (MWK) and the German Research Foundation (DFG) through grant INST 35/1314-1 FUGG and INST 35/1503-1 FUGG.

This work was partially funded by Zentrales Innovationsprogramm Mittelstand (ZIM) under grant KK5044704CS0.

Bibliography

[1] G. Elbaz, T. Avraham, and A. Fischer, "3D point cloud registration for localization using a deep neural network auto-encoder," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 4631–4640.

[2] R. Y. Takimoto, M. d. S. G. Tsuzuki, R. Vogelaar, T. de Castro Martins, A. K. Sato, Y. Iwao, T. Gotoh, and S. Kagei, "3D reconstruction and multiple point cloud registration using a low precision RGB-D sensor," *Mechatronics*, vol. 35, pp. 11–22, 2016.

[3] Z. Zhang, J. Sun, Y. Dai, D. Zhou, X. Song, and M. He, "A representation separation perspective to correspondences-free unsupervised 3D point cloud registration," *IEEE Geoscience and Remote Sensing Letters*, 2021.

[4] B. Mahmood and S. Han, "3D registration of indoor point clouds for augmented reality," in *Computing in Civil Engineering 2019: Visualization, Information Modeling, and Simulation*. American Society of Civil Engineers Reston, VA, 2019, pp. 1–8.

[5] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "Spinnet: Learning a general surface descriptor for 3D point cloud registration," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2021, pp. 11 753–11 762.

[6] S. Monji-Azad, J. Hesser, and N. Löw, "A review of non-rigid transformations and learning-based 3D point cloud registration methods," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 196, pp. 58–72, 2023.

[7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[8] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3D point clouds," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 9621–9630.

[9] Y. Wang, and J. Solomon, "Deep closest point: Learning representations for

point cloud registration," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*. IEEE, 2019, pp. 3522–3531.

[10] B. Chen, H. Chen, B. Song, and G. Gong, "TIF-Reg: Point cloud registration with transform-invariant features in se (3)," *Sensors*, vol. 21, no. 17, p. 5778, 2021.

[11] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. IEEE, 2020, pp. 2511–2520.

[12] B. Wu, J. Ma, G. Chen, and P. An, "Feature interactive representation for point cloud registration," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, 2021, pp. 5530–5539.

[13] T. Min, E. Kim, and I. Shim, "Geometry guided network for point cloud registration," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7270–7277, 2021.

[14] K. Fu, S. Liu, X. Luo, and M. Wang, "Robust point cloud registration framework based on deep graph matching," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2021, pp. 8893–8902.

[15] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[16] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

[17] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," *Advances in neural information processing systems*, vol. 28, 2015.

[18] V. Villena-Martinez, M. Saval-Calvo, J. Azorin-Lopez, A. Fuster-Guillo, and R. B. Fisher, "Local-global based deep registration neural network for rigid alignment," in *Proc. of the Int. Joint Conf. on Neural Networks*, 2021, pp. 1–8.

[19] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapenets: A deep representation for volumetric shapes," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[20] A. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3D model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[21] L. Wang, N. Zhou, H. Huang, J. Wang, X. Li, and Y. Fang, "GP-Aligner: Unsupervised groupwise nonrigid point set registration based on optimizable group latent descriptor," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[22] L. G. Sanchez Giraldo, E. Hasanbelliu, M. Rao, and J. C. Principe, "Groupwise point-set registration based on Renyi's second order entropy," *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 6693–6701, 2017.

Chapter 3

Deep Learning for PCG of 2D video game levels

Eleonora Chitti
Department of Computer Science
Università degli Studi di Milano, Italy
eleonora.chitti@unimi.it
ORCID: 0000-0003-4369-4615

DOI: 10.54103/milanoup.282.c635

3.1 Abstract

Content generation is one of the most expensive and time consuming tasks in video game development, and Procedural Content Generation (PCG) approaches have been proposed to reduce costs since 1980 in video games as Elite. In recent years, Generative Adversarial Network (GANs) have shown promising results in providing new possibilities for procedural generation of new content, including generation of 2D game levels. However, GAN research for PCG is still an open issue, since the new levels procedurally generated should respect constraints of the gameplay rules and they should be playable. In this paper we will review the state of the art and explore three interesting approaches in literature for procedural generation of game levels with GANs. First, we examine how the data of human-designed video game levels can be represented inside data-sets suitable for GANs; in particular, we focus on the representation of the game rules associated with each object present in the level. Second, we expose three different approaches of PCG with GANs, and for each approach we deepen the proposed adaptations on the GANs to generate video games levels, the data-set used, and the algorithm's performance. For each approach, we also analyze the proposed evaluation methods for the procedural generated levels, in particular taking into account the *traversability*, verifying that the areas in the

level are reachable, and the *playability*, verifying that the goal target of a game level is reachable without excessive difficulty.

3.2 Introduction

Video games are multimedia products made of different game elements, 3D or 2D assets, musics, sound effects, text, game story, game levels. These elements can be split into the two categories of cosmetic content and functional elements. The cosmetic content refers to assets and graphics from artistic point of view. The functional elements can be defines as the *game space* [1] or the game content that is directly related to game mechanics and rules, that are the core of a video game. According to [2] video games can be evaluated examining human actions, since the player experiences fun when they spend energy to reach the goal respecting the game mechanics and rules. To ensure playability, the rules, established by game designers, must be respected also during the development of a video game: level design heavily relies on human expertise and expensive playtesting of the levels created; for example in "Fig. 3.1" the ending target in a Super Mario Bros 2D level should be reachable and the obstacles should be positioned correctly, not too close nor too far to avoid the creation of a level too difficult or even impossible to play, that leaves the fun aside by replacing it with frustration. Therefore, one of the main costs of developing video game is content creation, that requires non-trivial human effort. To reduce costs and human labor, algorithmic approaches for automatic content creation following the game rules, or Procedural Content Generation (PCG), have been proposed from 1980, PCG can have different use cases, as autonomous generation, repair of non-playable levels, and data compression. In the Rogue dungeon-crawling video game (1980)[1], the developers exploited PCG to generate novel content (dungeon rooms and hallways) every time the game was started. In the Elite (1984)[2], a video game of space simulation, the developers exploited PCG for data compression, they stored the seeds for a pseudo-random number generator instead of the sequence of numbers to generate the universe represented in the game with the star systems. Nowadays PCG with seeds for pseudo-random generation is still used in commercial games as No Man's Sky[3].

In recent years deep learning allowed to produce different types of content as audio, images, or 3D objects, therefore, its application to video games is a natural consequence: new approaches exploiting *the Adversarial Networks* have been proposed also for PCG. For example, generative adversarial networks GANs have been applied to generate background terrain environments of games [3] or 2D game levels [4]. However GANs, that work well for generating content for other creative

[1] `https://www.mobygames.com/game/rogue`
[2] `https://www.mobygames.com/game/elite`
[3] `https://www.nomanssky.com`

Figure 3.1 *Example of Super Mario Bros level: Mario has to reach the pipe-goal of the level.*

purposes, are not always applicable to game generation. Procedural generation of levels, as well as PCG of sprites, should output content coherent with the game purposes and requires specific game-play skills: the game levels should be playable and the sprites should represent images of specific characters doing an action allowed by game rules (for example jumping or fighting).

We propose a focus on existing deep learning methods with GANs to generate new functional elements of 2D video games, and in particular, new game levels. We will analyze three different approaches for level generation that expose interesting results and that we have considered interesting for those wishing to approach PCG of video game levels with GANs. In the first section, we will analyze in brief what are the GANs and how this framework with two neural networks can generate new artificial samples starting from a real set of images. In the second section, we will analyze possible representations of the characteristics of a video game level that can be used to create new data-sets feasible as input for GANs. In the sections 4-6, we will deepen three different GANs implementations to generate game levels, analyzing promising results and limitations of each approach. Finally, in section 7, we draw the conclusions .

3.3 GANs in brief

Generative Adversarial Network (GAN) defines a machine learning framework that allows to generate new artificial data that are plausible, as new images or audio, from a real or hand-made training set. It comprises two neural networks, (i) the generator **G**, and (ii) the discriminator **D**, that compete in a zero sum game, where one agent's gain is another agent's loss.

- **G** generates new data as similar as possible to the training set X_{true}: it receives in input the data as N-dimensional uniform random variables (noise) as input

and it learns to map the input to distribution of interest.

- **D** distinguishes candidates produced by the generator X_{gen} from the true data distribution: it receives in input the true training set and the new data generated by **G** and, for each data, it outputs the probability that it is true or generated.

Through this technique new data are generated with the same statistics as the training set: **G** generates candidates while **D** evaluates them. The core idea of a GAN is based on the training of the generator through the discriminator, that can tell how much realistic are the data generated. This means that the **G** is not trained to minimize the distance to a specific data, as an image, but rather to fool the discriminator. Both networks are in fact updated dynamically: independent backpropagation procedures are applied so that **G** produces better samples, while **D** becomes more skilled at recognizing the fakes.

To summarize, the main goal of **G** is to learn how to successfully fool the discriminator, while the goal of **D** is to learn how to better recognize the true data from mocked ones [5].

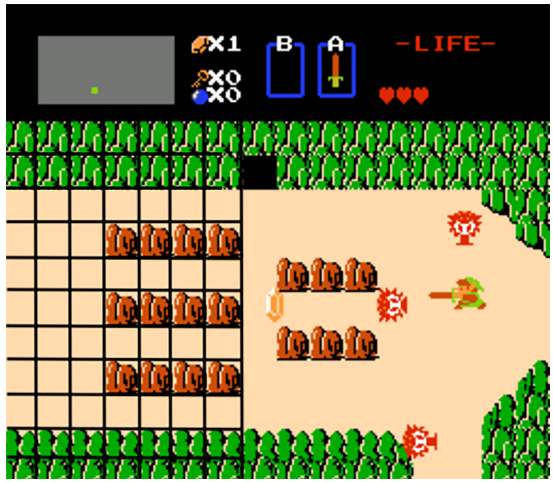


Figure 3.2 *Grid representation with tiles of Legend of Zelda. The map has walls and walkable areas, on the map are located tiles for the items (as rupees), obstacles (as stones) and enemies*

3.4 2D Video Games data representation of levels for GANs.

Before exploring the deep learning approaches, it is important to make an introduction regarding 2D video games data representation. Most of 2D games can be represented with a *tileset*, literally a set of tiles, or sprite images, each one of the same size. These tiles can be placed on a grid to compose larger images, to be used for example as game maps. Each tile can contain a wall, a door, a walkable area or

the floor, a non-walkable area, an item or an enemy, as in “Fig. 3.2”. Tiles can be overlapped in particular cases, as images of enemies or items on top of the floor sprite.

3.4.1 VGDL representation

Starting from the tileset representation of 2D games, the *General Video Game Artificial Intelligence* (GVGAI) framework provides, through the *Video Game Description Language* (VGDL) [6], a generic solution to represent 2D video games with same physics and game actions and similar rules (most of them with top-down or isometric perspectives), e.g., “Fig. 3.3”. VGDL represents 2D games with a grid-level of description of its custom properties, constraints and physics: each sprite image inside a level in VGDL brings with it additional information as the directional speed, interaction with other sprites, its movement inside the level, declaration of the sprite as a game termination condition (for ending targets), scoring related to this sprite, etc.

The GVGAI framework allows to run 2D games defined with VGDL standard, and it provides agents based on various heuristics that can play the games. These agents have been exploited by different studies as [4,7] to evaluate generated levels.

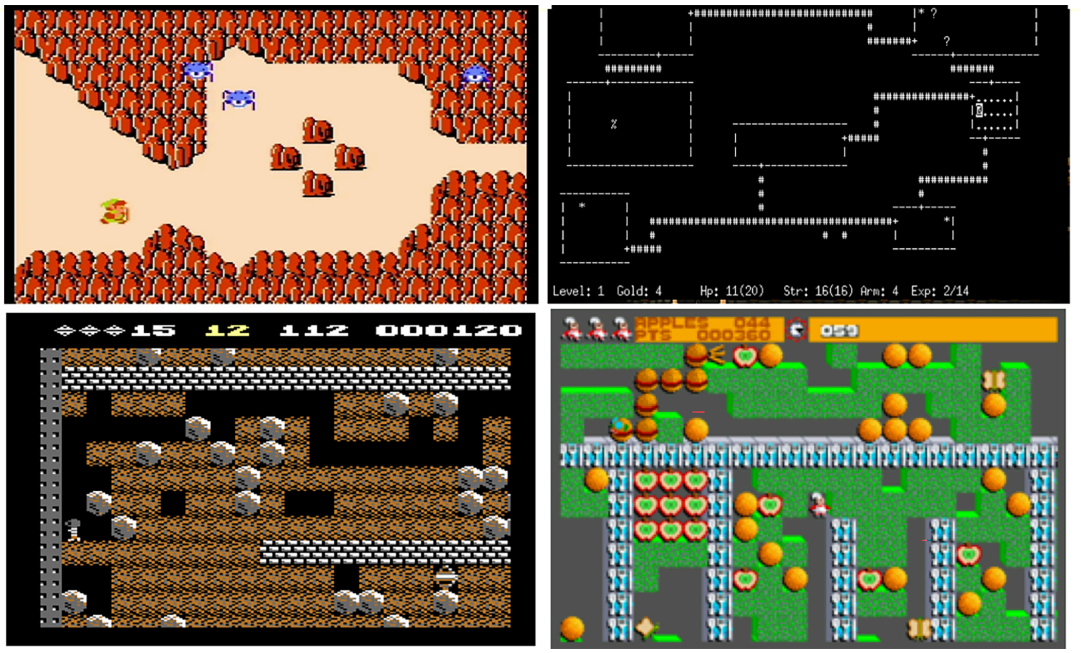


Figure 3.3 *Example set of four 2D games that can be represented with VGDL, from top left The Legend of Zelda, Rogue, Boulder-dash, Rockford.*

3.4.2 Maps representation

In the same year of the publication presenting the GVGAI, [8] proposed another approach defining a 2D game level with multiple overlapping images each one representing a feature of the level's map. This approach was explored for generating maps of first person games, in which the graphical perspective is rendered from the viewpoint of the player's character. The authors converted the walkable area of each level in top-down 2D maps, that can be used as a dataset.

The authors created 5 different images for each level's map:

- *Floor map*, representing floors.
- *Wall map*, representing walls.
- *Height map*, representing the height of floor of each room inside the level.
- *Things map*, representing all the objects included in the level that are not walls, floors or doors.
- *Trigger map*, representing all the triggers with different encoding, representing the type of trigger (door, teleport, lift or switch)

In addition they proposed a 6th image, called *Room map*, representing the map with a room segmentation approach similar to the one used to reconstruct maps of real life environments by robotic entities. These maps can be analyzed with the Simultaneous localization and mapping - SLAM algorithms, that can be used to evaluate the quality of the generated map.

3.5 DOOM levels generation with GANs

Authors in [8] propose and compare two different approaches to generate DOOM levels. In this study the authors applied GANs to train a model with existing DOOM levels and then they used the model to procedurally generate new levels.

3.5.1 Data-set

DOOM is a first person shooter video game developed in 1993[4], and its levels are widely represented in different textual data-sets as the Video Games Level Corpus by [9] and the *Idgames archive* [5]. [8] performed a preliminary conversion of data from text to images, they extracted topological features and converted data representing each level into 6 images, *(i) Floor map, (ii) Wall map, (iii) Height map, (iv) Things map, (v) Triggers map and (vi) Room map*, as described in the section 3.4.

The authors trained two different GANs with 1000 human-designed levels: the first GAN was trained with maps images, while the second one was trained using

[4] `https://it.wikipedia.org/wiki/Doom`
[5] `https://doomwiki.org/wiki/Idgames_archive`

both map images and a vector of features that were also extracted during the preliminary conversion of the data-sets. The vector of features include:

- *Metadata:* includes level name and number of downloads of the level
- *WAD*: the WAD is the name extension of files in the Idarchive data-set, these files include size of the level, number of lines and vertices, the same data are identified here with WAD term.
- *Image:* includes equivalent diameter of the level, fraction of area that can be traversed, perimeter of the level, vertical and horizontal size of the level
- *Graph:* graph representation extracted with the Room map image analyzing indoor environments with SLAM approach.

3.5.2 Deep Level Generation

The authors trained two different GANs I and II, one *unconditional* and one *conditional*. The authors employed the Wasserstein GAN with Gradient Penalty (WGAN-GP) developed by [10], they adapted the framework replacing the *tanh* activation function on the output layer, with a *sigmoid* function more suitable for grayscale images with few levels of gray.

The first GAN (I) is *unconditional*: for each level the generator G receives in input X_{true} **six images and a noise vector Z** sampled from a Gaussian distribution, while the second GAN (II) is *conditional*: the generator G gets in input the **the six images, a vector of features Y of the level and the noise vector Z**.

As a result, both the GANs generates six images that represent a generated level and can be used to produce a fully playable level. The G network is trained to generate an output X_{gen} that is as similar as possible to the original inputs X_{true}. In contrast, the D network is trained to distinguish between human-designed and generated levels, it receives in input either X_{true} or X_{gen} and it outputs the probability that X_{true} or X_{gen} are images of a generated level.

The authors trained the discriminator and generator networks following the approach proposed in [10], using Adam optimizer, they optimized the discriminator network five times for each update of the generator network. In each training iteration, they also applied a 90 degree clockwise rotation to the input images, so that the networks were trained using all the four possible level orientations. This transformation allows to exploit the rotation information in the representation of a DOOM level, since level's playability is not affected by its orientation in the space.

3.5.3 Generated Levels Evaluation

To evaluate the quality of samples generated by a GANs other studies proposed an approach based on human annotations or based on the score provided by an image classifier [11]. Instead in this paper [8], the authors designed a set of metrics inspired by the ones used to evaluate maps generated by the real world navigation

of robotic entities, that are very similar to the six maps that describe a DOOM level. The metrics proposed by [8] are:

- dE: It is the average absolute difference of entropy of the pixel distribution between human-designed and generated images. The authors selected the difference of entropy between true samples and generated samples to detect if the quantity of information encoded by the generated levels differs from the true samples.
- SSIM: average Structural Similarity index between the images of human-designed and generated levels, with values from 0 to 1, where value 1 represents a comparison of two images of the same map.
- Encoding Error - EE: measure of the average errors over the pixel values of the level images generated by the network (for each pixel there are few of meaningful color values, for example in the FloorMap each pixel should have either value black or white)
- CE: the average Corner Error between the images of human-designed levels and generated levels, that is a measure of how large is the difference between the average number of corners in the two sets of levels. This metric provides an estimated of the how close are human-designed and generated levels in terms of structural complexity. The author computed the CE's input value of number of corners through the Harris detector [12] for counting the corners contained in FloorMap and WallMap images.

The authors trained two GANs defined before as (I) *unconditional* and (II) *conditional*. In both networks I and II, the **discriminator loss** steadily decreases with training iterations and the loss achieved in the training set is close to the one achieved in the validation set, suggesting generalization capabilities.

For **dE of the FloorMap, WallMap and ThingsMap**, both networks I and II are able to generate images with dE values, not only close to 0 at the end of the training, but they are also able to generate images with entropy values very similar to the originals. For **dE of the HeightMap**, the generated images are more noisy, and none of the network is capable to reach values close to 0, although the II *conditional* network is capable to reach smaller dE that the I *unconditional*. For **SSIM and EE values** both GANs perform well generating images of good quality. In particular, the II obtains **SSIM** values of quality of images slightly better than I. For **CE values**, the levels generated by I do not improve over time, instead of CE values of levels generated by II slightly increase over time, suggesting that the levels generated by the *conditional* network are more complex and more similar to human-designed levels. The similarity can be examined also with visual comparison, since levels by the *conditional* network show a richer structure.

The authors proposed a novel approach to generate maps for the DOOM video games, the maps can be traversed since they have been examined using an evaluation approach similar to the one for SLAM algorithms. However, playability has

not been tested, and maybe the level are too easy or too difficult to play. Therefore, this paper has been an interesting starting point for future studies, as [7] that proposes an evaluation of the playability of generated levels. The approach in [7] will be analyzed below in this paper.

3.6 Bootstrapping Conditional GAN for level generation of The Legend of Zelda

[7] propose an approach with an adaptation of GAN called CESAGAN or Conditional Embedding Self-Attention GAN, to incorporate an embedding feature vector input to condition the training. The aim of the authors is to allow the network to model non-local dependencies between game objects that are fundamental for the playability of the level, for example a key that the player should pick up to pass a locked door. The authors then evaluate the playability of generated levels. In addition, not all the video game offer large data-sets as DOOM, so the authors propose a bootstrapping mechanism to overcome the lack of data, in this approach the new generated levels, that are marked as playable, are added to the training set. The authors observed that their approach does not only allow to generate a larger number of levels that are playable but also generates less duplicate levels compared to a standard GAN. The authors tested the framework developed generating new levels of The Legend of Zelda.

The Legend of Zelda (Nintendo, 1986) is an adventure video game in which the player needs to explore different dungeons, each dungeon is a level. The main goal of the player is to explore the dungeon, collect as many treasures and money as possible, find a key and reach the exit door of the dungeon without getting killed by the moving enemies. The player can kill the enemies using their sword to collect extra money.

3.6.1 Data-set

The GVGAI [6] is a framework built to run 2D arcade-like games written in VGDL language, described in section 3.4, and it offers a definition in VGDL of game rules pertaining each sprite present in The Legend of Zelda video game. The authors extended the VGDL definition of each sprite with the Identity information to be an additional input data for the GAN, as shown in Table. 3.1. The authors manually generated the data-set with 45 human-designed levels.

3.6.2 Deep Level Generation

According to the authors GAN-based models for level generation are built using convolutional layers, but it is not enough for level generation, since it is a local operation whose correlation depends on the spatial size of the kernel, and in an

Table 3.1 *VGDL encoding (Symbol) and encoding for GAN (Identity) of The Legend of Zelda elements*

Object	Symbol	Identity
Wall	w	0
Empty	.	1
Key	+	2
Exit	g	3
Enemy 1	1	4
Enemy 2	2	5
Enemy 3	3	6
Player	A	7

operation for level generation, it is not likely for an output on the top-left position to have any correlation to the output at bottom-right. So, to increase the search space, a deep convolution network with many layers would be required. In fact video games levels have correlations of items located far from each other, as the key and the door, and these correlations are fundamental to make the game playable.

Therefore the authors propose a new approach called CESAGAN, combining the methods of the self-attention GAN (SAGAN) proposed by [13] and the GAN adapted with the input vector proposed by [8].

The SAGAN allows to keep balance between efficiency and to capture long-range dependencies, and it is based on three different vectors: query (f), key (g) and value (h) which are three different mappings (output of a single perceptron neural network) of the input data (e.g. an image). The query and key undergo a matrix multiplication then pass through a softmax, which converts the resulting vector into a probability distribution attention map. This attention map determines the weight of each of the tiles and keep it in memory. Finally, the attention map is multiplied by the value to determine the relationship at a position in a sequence by attending to all positions within the same sequence.

The authors of [7] concatenate the feature embedding mapping and the self-attention feature map to combine SAGAN with the conditional vector representation u. This network is applied to both the generator (G) and discriminator (D). In addition the authors propose also a Bootstrapping mechanism to improve the CESAGAN (Fig. 3.4): after each epoch, a new set of levels is generated, and a playability analysis is carried out to identify the playable levels. The levels identified as playable are then added to the training set. Playability is evaluated with the following heuristics:

- There is only one player's avatar
- There is only one key
- There is only one door
- Enemies are less than 60% of the empty space

- The player can reach the key using an A* algorithm
- The player can reach the door using an A* algorithm
- The level has a border to prevent all the characters to go outside the level.

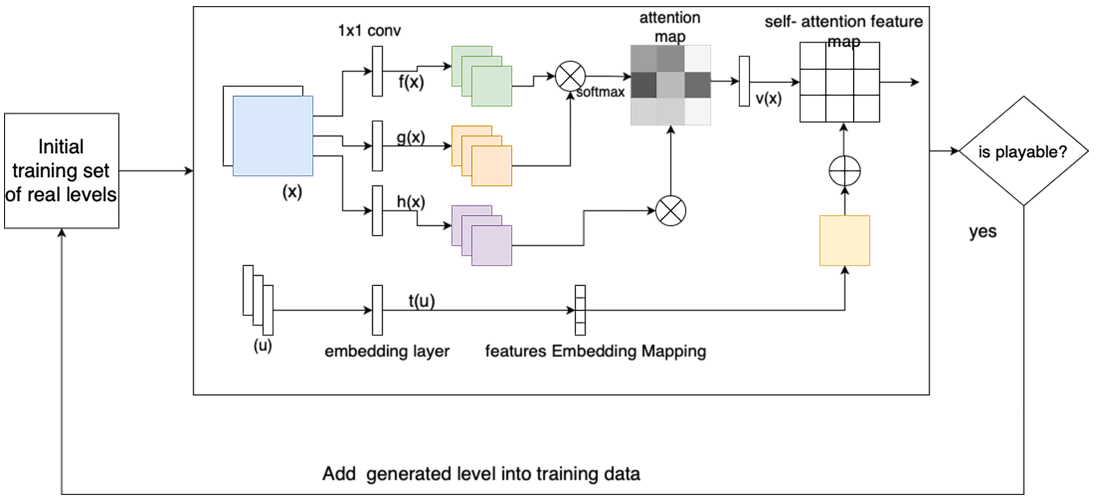


Figure 3.4 *Representation of the CESAGAN with bootstrapping mechanism.*

The authors suggest for future studies to substitute the proposed heuristics with an agent from the GVGAI framework to check for playability.

3.6.3 Generated Levels Evaluation

The authors tested their approach analyzing playability of generated levels and searching for duplicates of the generated samples. First, the authors tested the CESAGAN without the bootstrapping with the data-set with 45 levels and then compared the results with the state-of-the-art techniques. The CESAGAN performed better in generating playable levels, 58% against the 19.4% of the baseline, and it performed slightly better in not generating duplicates levels, 37.6% of duplicates against 39.4% of the baseline.

Second, the authors tested the CESAGAN with bootsrtapping with a data-set of only 5 human-designed levels. The authors motivated the decision of using a so small data-set by reporting that in to the state of the art data-sets for some type of 2D video games may contain few samples, and they want to propose an approach feasible for video games with few levels available. In any case, this second CESAGAN performed well with 47% against the 24.6% of the baseline of playable levels generated, and with 60.3% of duplicates against 98% of the baseline of duplicated levels generated.

3.7 Level generation for multiple types of 2D video games with common latent space with VGDL

[4] trained a GAN to generate game levels for four 2D video games with similar gameplay rules.

1. Boulder-dash
   Boulder Dash takes place in a series of caves, each one is a level designed as rectangular grid of blocks. The player can move freely to collect as many diamonds as possible while avoiding dangers, such as falling rocks.
2. Link
   Link is a strategy game with Roguelike elements (turn taking). The player controls a group of soldiers trying to survive on the hostile, arctic planet, crafting items, and collecting basic materials and advanced tools.
3. The Legend of Zelda
   The Legend of Zelda is an adventure game in which the player will explore different dungeons, each dungeon is full of monsters. The player is free to move and the main goal of the player is to explore the dungeon, avoiding or killing monsters, collecting treasures and money, while searching for the key to exit through a locked door.
4. Roguelike game
   In Rogue, players control a character as they explore several levels of a dungeon seeking the Amulet of Yendor located in the dungeon's lowest level. The player-character must fend off an array of monsters that roam the dungeons. Along the way, players can collect treasures that can help them offensively or defensively, such as weapons, armor, potions, scrolls, and other magical items. Rogue is turn-based, taking place on a square grid represented in ASCII or other fixed character set, allowing players to have time to determine the best move to survive. Rogue has been defined a masterpiece and it inspired the development of many games with similar mechanics, defined as Roguelike.

These four games have completely different gameplays, but they have in common the actions that the player performs to reach the goal, that are defined as the **sequences of actions** needed to complete successfully a level. For each game the same action leads to a different visual consequence but in all the games it permits to advance in the level: pick a bomb and break a wall is different from pick a key and open a locked door, but the same pick action leads to advance in the level. In fact the games are not equal considering the interaction of player with dynamic elements (as enemies, treasures...) that are different for each game.

Only *Boulder-dash* has falling boulders which obey gravity, while in *Roguelike* and *Zelda* have inside the map solid walls with locked doors or breakable walls that

require the player to first pick up a key or a bomb to pass. The ability of enemies, (soldiers, magicians, ...) are also unique for each game, and these differences result in a variety of sprite patterns.

3.7.1 Data-set

The VGDL language (section 3.4) defines for each sprite the particular properties, as scoring and movements allowed for that sprite, that can be useful for example to define enemies properties for each game. These four games have been already encoded in VGDL language and VGDL definitions is available for these games within the GVGAI framework [6].

The authors generated a new training data-set converting the **sequence of actions** along time in game levels, exploiting VGDL information on sprites. To generate the level from actions, the objective is to place obstacles that complement the actions at the right time and in the right location, such that the player action in the game timeline is necessary to move forward in a game.

A training sample is created by the authors through filling an empty grid: the starting point is on the top left quadrant and the goal point is chosen on the bottom right. A sequence of actions is selected, and each action will take the players avatar from one grid point to the next until the end. The action, as jump or break a wall, is carried out through the grid as sprite, as a ditch or a wall, that is placed in the way to match the corresponding action. The action sequences are varied by changing the order of actions or permuting the combinations of actions randomly. Multiple combinations of actions that take the players avatar from the start to the goal state are considered. The same action sequence is used in all the four games considered, but the specific game rules requires to place different types of obstacles to match the action. For example, in Boulder-dash one has to avoid falling boulders and in Roguelike one needs to first pick up a key before passing a locked gate. The approach is generic and can be used to generate levels for multiple games starting from a common action sequence and path through the grid. The training level generation algorithm is available at [4].

Training sets are generated explicitly with the same action sequence for all four game levels resulting in similar gameplay for all four levels. **Each level is represented as a multi-channel binary matrix**, with each channel representing one type of sprite in the game and each grid point being a binary representation of the presence of the corresponding sprite at that grid point.

3.7.2 Deep Level Generation

The authors propose a new approach of GAN to generate four different outputs with the same random seed, and each output corresponds to a level of the four games. This GAN has one Generator G and four Discriminators D, one for each

video game. Each discriminator is, in fact, tied to a single game and distinguishes between generated samples and original.

According to the authors this GAN will capture information that traditional GANs will not: the traditional GAN would not learn any common patterns as the only common element, the random input from the latent space, cannot be trained. In this new version of GAN the latent space and unbranched layers capture the commonality across all the four games while the branched D layers capture the differences, since D networks are trained independently one from the another.

Binary cross entropy loss from the discriminators is added to conditional loss from the generator. The generator loss is the sum of the binary cross entropy between the training sample and the generated image along with conditional loss if the number of sprites does not match the training level. The generators use batch normalization between convolution layers and LeakyReLU activation along with a final sigmoid activation to generate game level output. Each of the discriminators use a dropout of 30% to reduce overfitting. The generator generates a grid of size 16x16 for each game, from an initial input of 128 normally distributed random numbers. The training sample and generated game levels are represented as a tensor with nine channels, one each for each type of sprite (avatar, exit, floor, gold/health, key, lock, monster, wall and weapon). Unused channels are set to zero.

3.7.3 Generated Levels Evaluation

To evaluate the quality of samples generated by a GANs the authors exploited the agents from the GVGAI framework, each agent will solve the game ensuring that it is playable. The authors not evaluated only playability (or solvability), but also path similarity and novelty of the levels, defined as follows:

- **Solvability**: a level is considered as playable if a GVGAI agent can solve it at least once in 5 attempt.
- **Path Similarity**: this value defines if the GAN model has captured the similarity between the games, and it is calculated between game levels of distinct games generated together. If the GAN captures the gameplay similarity between distinct games, the distribution of path similarity distance should be the same between the training set and generated sets. A shortest path is calculated from the starting point to the goal. the shortest path do not ensure playability, since authors applied a Dijkstra shortest path algorithm, that takes into account stops on the grid cells with sprites of items that are necessary to step further in the level (as a key), that are considered as mandatory nodes to be traversed in the graph. The path similarity distance is the Manhattan distance between the grid locations in the path, and it is calculated between the shortest paths game levels generated together.

- **Novelty**: it is a measure of similarity within a game. A level is novel with respect to another if the path taken by the avatar, represented by the sequence of actions is different. To evaluate novelty, the authors used the Levenshtein distance (a string metric for measuring the difference between two sequences) between two action sequences. If the distance is large, the two levels are different: for example, a solution action sequence for a Zelda level is *(right, right, pick key, up, right, right)* and another level has actions *(right, up, pick key, up, right)* the distance between the levels is two: the number of changes inside one sequence to obtain the other.

The authors evaluate solvability of the generated levels by taking a set of 50 GAN's generated levels for each game and running an automated agent provided by the GVGAI framework. The Boulder-dash has the higher solvability of 70% over all the other games, probably because this game has the lowest number of obstacles (this game has not locked door or similar) so it does not have any dependency between actions. Probably because of the same reason the Link and the Roguelike are around 55%, and the Legend of Zelda has the lowest solvability with a score of 40%, since Zelda is the game with the highest number of obstacles, including locked doors, walls to broke, switches to activate and ditches to cross.

To evaluate the Path Similarity of gameplay across games for generated game-level sets, the authors plotted the average similarity distance between the ideal path for avatar to reach the end state from the start state, picking up the necessary items and ignoring dynamic items as monsters. The authors evaluated the distribution with the Wasserstein distance, reporting that the generated sets of four levels have path similarity distribution closer to the training set(Wasserstein distance value: 161) and further less that the baseline (Wasserstein distance value: 283), stating that the GAN has captured aspects of gameplay similarity across the four games.

Looking at the novelty values the authors reported that the generator has captured most of the complexity of the training set in its model. In fact, the variety of levels in the training set is captured by the GAN as the generated levels have a similar distribution of values for the Levenshtein distance.

The authors propose an innovative approach, that can be considered an interesting starting point to generate new levels procedurally. However, future studies should overcome the difficulty of achieving a better value of solvability of generated levels of games with more complex rules and obstacles as the Legend of Zelda.

3.8 Conclusion

We have shown three representative examples of Video Games Levels generation which use GANs to create procedurally new game levels that can be playable. In particular we described the problems of defining the levels with a representation feasible as input for GAN, and the two major approaches are: the representation of

a level wih 6 images each one representing a feature of the level's map, (as walls, triggers or collectable items) and the representation of a level with the VGDL language [6] that allows to encode for each image, that represents an element of the level (as a treasure), its features and its role within the game rules. We then analyzed use cases of the Video Games Levels generation, in particular in the area of 2D arcade-like video games. [8] compared two different approaches to generate DOOM levels, in the first approach the authors trained a GAN with a data-set of DOOM levels represented with the 6 maps images, and in the second one the authors adapted the GAN to take in input the same data-set and in addition a vector of level's features, however ths approach do not take into account the evaluation of playability of levels. [7] proposed an approach to procedurally generate levels of 2D video games for which large data-sets are not available, extending [8] approach with a [13] and a bootstrapping mechanism in which generated levels, that are marked as playable by an heuristic, are added to the training set. The results are interesting and future studies are needed to ensure the feasibility of using this framework with little data-sets. Authors in [4] propose the most recent approach, with the aim of developing levels of four different 2D games with a similar gameplay with only one Deep Learning Framework, that outputs four different generated levels (on for each game). Within this framework a Generator G generates all the 4 samples, that are then analyzed by 4 Discriminators D, each one taking in input only one sample relative to a type of game. Results are promising, however the performance of the network vary consistently between the four games, and the algorithms show lower performances in the generation of levels of the games which include more items in the gameplay, than the others games with less items, even if all the four games have similar rules.

Bibliography

[1] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, "Procedural content generation for games: A survey," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 9, no. 1, feb 2013.

[2] E. J. Aarseth, "Cybertext: Perspectives on ergodic literature," 1997.

[3] E. Panagiotou and E. Charou, "Procedural 3D terrain generation using generative adversarial networks," 2020. [Online]. Available: https://arxiv.org/abs/2010.06411

[4] V. Kumaran, B. Mott, and J. Lester, "Generating game levels for multiple distinct games with a common latent space," *Proc. of the AAAI Conf. on Artificial Intelligence and Interactive Digital Entertainment*, vol. 16, no. 1, pp. 109–115, Oct. 2020.

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014. [Online]. Available: https://arxiv.org/abs/1406.2661

[6] D. Perez-Liebana, J. Liu, A. Khalifa, R. D. Gaina, J. Togelius, and S. M. Lucas, "General video game ai: a multi-track framework for evaluating agents, games and content generation algorithms," 2018. [Online]. Available: https://arxiv.org/abs/1802.10363

[7] R. R. Torrado, A. Khalifa, M. C. Green, N. Justesen, S. Risi, and J. Togelius, "Bootstrapping conditional gans for video game level generation," 2019. [Online]. Available: https://arxiv.org/abs/1910.01603

[8] E. Giacomello, P. L. Lanzi, and D. Loiacono, "DOOM level generation using generative adversarial networks," 2018. [Online]. Available: https://arxiv.org/abs/1804.09154

[9] A. J. Summerville, S. Snodgrass, M. Mateas, and S. Ontanon, "The vglc: The video game level corpus," 2016. [Online]. Available: https://arxiv.org/abs/1606.07487

[10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," 2017. [Online]. Available: https://arxiv.org/abs/1704.00028

[11] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," 2016. [Online]. Available: https://arxiv.org/abs/1606.03498

[12] C. G. Harris and M. J. Stephens, "A combined corner and edge detector," in *Alvey Vision Conf.*, 1988.

[13] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," 2016. [Online]. Available: https://arxiv.org/abs/1601.06733

Chapter 4

Text Style Transfer: An Introductory Overview

Sourabrata Mukherjee
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Charles University, Czechia
mukherjee@ufal.mff.cuni.cz
ORCID: 0000-0002-1713-2769

Ondřej Dušek
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Charles University, Czechia
odusek@ufal.mff.cuni.cz
ORCID: 0000-0002-1415-1702

DOI: 10.54103/milanoup.282.c636

4.1 Abstract

Text Style Transfer (TST) is a pivotal task in natural language generation to manipulate text style attributes while preserving style-independent content. The attributes targeted in TST can vary widely, including politeness, authorship, mitigation of offensive language, modification of feelings, and adjustment of text formality. TST has become a widely researched topic with substantial advancements in recent years. This paper provides an introductory overview of TST, addressing its challenges, existing approaches, datasets, evaluation measures, subtasks, and applications. This fundamental overview improves understanding of the background and fundamentals of text style transfer.

4.2 Introduction

Natural Language Generation

Natural Language Generation (NLG) is the process of producing meaningful phrases and sentences in natural language. The main goal of NLG is to automatically produce narratives that describe, summarize, and explain the input data in a human-like manner. In other words, it generates fluent texts with minimal grammatical errors and retains the specific intended content.

Some of the popular NLG tasks include machine translation [1], dialogue systems [2], and text summarization [3]. Through these tasks, the generated text has shown to be more coherent, logical, and emotionally rich, especially with the latest approaches based on neural language models.

Controllable NLG

Most of the built NLG systems target text fluency and grammatical correctness, and do not consider any specific control over text style. This is a motivation for research on controllable text generation [4]. The aspects of text generation that are commonly controlled include topic [5–8], style [9–12], emotion [13–16], and user preferences [17–20]. Some of the applications of controllable text generation are context-based text generation [21], topic-aware text generation, [22], knowledge-enhanced text generation [23] and text style transfer [24].

Control can be applied at various stages of the neural generation process, such as input, hidden states, and decoding [25]. The Plug and Play language model (PPLM) that was proposed by [26] takes an external input, performs computations on hidden states, and then combines a pre-trained language model with one or more simple attribute classifiers that guide text generation toward the desired topic or sentiment. Another model by [27] describes a training method based on diverse ensembling that would lead models to learn distinct text styles. It can thus be inferred that end-to-end models can be equipped with the ability to control style and length. More details on how NLG can be controlled using various control strategies in the state-of-the-art models can be found in [4].

Style-Controlled Text Generation

In recent research, more attention has been paid to a subtask of controllable text generation dubbed *style-controlled text generation*, i.e., modeling and manipulating the style of the generated text [28]. The goal of this approach is to model the content of a text along with controlling its style. For example, the persona of a speaker in dialogue [29] or the sentiment of product reviews [30]. Understanding and dealing with style in text proves to be very complex [24], but recent advances in deep lear-

ning techniques are helping stylized text generation tasks in various ways [31]. For example, embedding learning techniques are used to represent style [13], and then adversarial learning is used to match content but to distinguish between different styles [30, 32, 33].

Text Style Transfer

In this paper, we will focus on Text Style Transfer (TST). TST is a task closely related to Style-Controlled Text Generation. *Style-Controlled Text Generation* aims to generate new text in a specific style. In contrast, *Text Style Transfer* is an existing text written in source style, aiming to change the text style, i.e. a text retaining most of the content but conforming to the target style. Our aim is to give a very basic introduction to the TST task. All of the sections are presented in a brief and simple manner with an illustrative number of examples. A more detailed overview can be found in [24, 25, 28, 31, 34].

The paper is organized as follows. After the introductory section, Section 4.3 provides an overview of text style transfer. Section 4.4 reflects on the challenges facing the TST task. The discussion of the existing data sets, approaches, evaluation measures and applications is presented in Sections 4.5-4.8. A short overview of the related ethical considerations is given in Section 4.9. Section 4.10 concludes the paper.

4.3 The Task

Text style transfer (TST) is an NLG task that aims to automatically control the style attributes of a text while preserving the style-independent content. Some of the attributes that TST aims to control are politeness, formality, sentiment, and many others. Table 4.1 shows some basic examples of TST. TST implies the need to understand the difference between the style and content of a text.

Table 4.1 *TST examples regarding sentiment, polarity, and formality.*

	Source Style	**Target Style**
Impolite → Polite:	Shut up! the video is starting!	Please be quiet, the video will begin shortly.
Negative → Positive:	The food is tasteless.	The food is delicious.
Informal → Formal:	The kid is freaking out.	That child is distressed.

4.3.1 Understanding Style and Content

[35] define style as a notion that refers to the manner in which semantics is expressed. Individualistic styles such as choice of words, sentence structures, metaphors, sentence arrangement, etc., vary from person to person. These variations are

shaped by the speakers's personality – everyone has a distinctive set of techniques for using the language to express and achieve their independent goals [36]. This individualistic nature also determines how a person perceives events, describes ideas, or provides additional information about them [24]. Style extends beyond individual sentences to the broader discourse level. This includes elements such as paragraph organization, theme progression, and use of cohesive devices that bind the text together. These stylistic features at the discourse level play a crucial role in ensuring that the text is coherent and engaging, thereby enhancing its ability to convey the intended message and intrigue the reader. Taking these aspects into account, the text can offer a richer and more nuanced understanding of its content.

Style has also been defined by [36] by its pragmatic aspects. Beyond these personal styles of expression, there are certain styles that are used as protocols to regularize the manner of communication. For example, in the case of academic writing, using formal expressions is the regularized protocol.

TST studies adopt a more data-driven approach to define text style in contrast to the theoretical definition used in linguistic studies [31]. We can define style in TST as the text style attributes or labels that are dependent on style-specific corpora [24]. For example, datasets are manually annotated with linguistic style definitions, such as formality [37] or sentiment [38–40]. Unfortunately, not all possible styles have very well-matched corpora, and many recent dataset collection works are looking for meta-information that would automatically link a corpus to a certain style. Some of the TST tasks are built upon the assumption that style is localized to certain tokens in a text, and a token has either content or style information, but not both [41].

In opposition to style, content can be understood as the subject matter, theme, or topics the author writes about.

4.3.2 Problem Formulation

Given a text x, with an original style S, our goal is to rephrase x into a new text $\hat{x}$ with a target style S' ($S' \neq S$) while preserving its content that is independent of style.

Suppose that we have a dataset $X_S = x_1^{(S)}, \ldots, x_m^{(S)}$ representing texts in style S. The task is to transform texts in style S to the target style S' while maintaining the original meaning. We denote the output of this transformation by $X_{S \to S'} = \hat{x}1^{(S')}, \ldots, \hat{x}m^{(S')}$. Similarly, for the inverse transformation from style S' to style S, we denote the output as $X_{S' \to S} = \hat{x}1^{(S)}, \ldots, \hat{x}n^{(S)}$.

4.4 Challenges

Modeling the style of text comes with a lot of challenges in practice, which are discussed in this section.

No Parallel Data

TST models could be trained with respect to parallel text from a given style or on non-parallel corpora. Parallel datasets are those which consist of pairs of texts (i.e. sentences, paragraphs) where each text in the pair expresses the same meaning, but in a different style. Non-parallel datasets, on the other hand, have no paired examples to learn from, and simply exist as mono-style corpora. For parallel datasets, TST can be formulated in such a way that instead of translating between languages, one can translate between styles following machine translation. However, obtaining suitable, sufficient parallel data for each desired style attribute is the biggest challenge.

Style and content are hard to separate

Style transfer text generation implies the need to distinguish content from style. In some scenarios, the line between content and style can be blurry. This is since the subject on which an author is writing can also influence their choice of words and style. This interweaving of the style and semantics makes TST challenging.

No Standard Evaluation Measures

Evaluating the quality of the style-transferred text is hard. Human evaluation is regarded as the best indicator of quality, but unfortunately, it is expensive, slow, and hard to reproduce [42], making it an infeasible approach to use on a daily basis to validate model performance. For this reason, we often rely on automated evaluation metrics to serve as a cheap and quick proxy for human judgment.

In the case of automatic evaluation of TST, it has been noticed that when style transfer accuracy increases, the content preservation scores decrease, and vice versa [43]. The main reason behind this is the entanglement between the content and style (see above). This trade-off between style transfer accuracy and content preservation poses a very big challenge for evaluating TST tasks.

In order to effectively evaluate a TST output, one must pay attention to how semantically accurate the output text is and how fluent it is. The comprehensive TST evaluation also considers three criteria: transferred style accuracy, semantic preservation, and fluency, which often require human evaluation as automated metrics alone do not adequately identify these complex properties. Further discussion on evaluation measures is in Section 4.7.

4.5 Datasets and Benchmarks

To evaluate TST models, many datasets have been proposed over the years. We discuss a few popular datasets by individual subtasks as follows:

Politeness Transfer

Politeness transfer aims to control the politeness of a text [44, 45]. A compiled dataset with automatically labeled instances from the raw Enron e-mail corpus [46] was presented by [45]. This dataset mainly focuses on politeness in North American English.

Sentiment Transfer

Another common task in TST is sentiment transfer (transferring text's polarity from positive to negative or vice-versa) [43, 47]. There are three popular datasets proposed for this task.

- Yelp – This is a corpus consisting of restaurant reviews from Yelp collected by [38].
- Amazon – This is Amazon's product reviews that were collected by [39].
- IMDb – This is a movie review dataset constructed by [40].

Formality Transfer

Formality transfer is yet another task in TST which is not only complex but also involves multiple attributes that affect text formality. Grammarlys Yahoo Answers Formality Corpus (GYAFC) is the largest human-labeled parallel dataset that was proposed for formality transfer tasks by [37]. The authors extracted informal sentences from the Entertainment & Music and Family & Relationship domains of the Yahoo Answers L6 corpus for preparing the dataset.

Author's Style Re-writing

The task of paraphrasing a sentence to match a specific author's style is called author imitation. To tackle such tasks, [48] collected a parallel dataset that captured line-by-line modern interpretations of 16 Shakespeare's plays, with the help of the educational site Sparknotes.[1] The objective behind collecting the dataset was to imitate Shakespeares text style by transferring modern English sentences into Shakespearean-style sentences. This dataset has been used in other TST studies as well [49, 50].

Image Captions Transfer

The task of transferring image captions from factual formal ones to romantic and humorous styles was proposed by [9]. Following this, a caption dataset was

[1] https://www.sparknotes.com

collected by the authors where each sentence was labeled as factual, romantic, or humorous.

Text Simplification

Another important use of TST is to lower the language barrier for readers, which includes tasks like converting general English into Simple English, based on a dataset collected from Wikipedia [51]. Another task is to simplify medical descriptions to patient-friendly text [52].

Political-slant Transfer

Political slant transfer is a task that modifies a writer's political affiliation writing style while preserving the content. Comments from Facebook posts from 412 members of the United States Senate and House who have public Facebook pages were collected by [11] and further annotated with each congresspersons political party affiliation, i.e., Democrat or Republican.

Fixing offensive texts

Correcting offensive and abusive language [53] is another important task of TST which is a major problem in today's world, due to the prevalence of abusive comments on social media. Posts from Twitter and Reddit were collected by [54] and then classified into *offensive* and *non-offensive* classes using a classifier pre-trained on an annotated offensive language dataset.

4.6 Text Style Transfer Approaches

Standard data-driven TST approaches can be classified based on the data used for training (parallel vs. non-parallel). Recently, new approaches using large language models (LLMs) emerged that do not specifically need in-domain training data.

4.6.1 Supervised Training on Parallel Data

For situations where style-parallel data is available, like most supervised methods, a standard sequence-to-sequence model [47, 55, 56] with the encoder-decoder structure is typically used [24]. This process is similar to machine translation and text summarization. The encoder-decoder architecture can be implemented by either LSTM [57] or the Transformer [58] architecture. For example, [49] trained a sequence-to-sequence model on a parallel corpus and then applied the model to translate modern English phrases to Shakespearean English. However, the

application of basic sequence-to-sequence approaches is quite limited due to the lack of parallel data (see Section 4.4).

4.6.2 Non-parallel Approaches

Methods applicable to non-parallel data can broadly be divided into three unsupervised approaches:

Prototype Editing

This process works by deleting only the parts of the sentences which represent the source style and replacing them with words with the target style while making sure that the resulting text is still fluent. The advantage of this approach is its simplicity and explainability. For example, [9, 59] found that parts of a text that are associated with the original style can be replaced with new phrases associated with the target style. The text was then fed into a sequence-to-sequence model to generate a fluent text sequence in the target style. However, these approaches are not suitable for TST applications where simple phrase replacement is not enough or a correct way to transfer style. The style marker retrieval might not work if the datasets have confounded style and contents. This is because they may lead to the incorrect extraction of style markers, affecting some content words.

Disentanglement

This approach aims at disentangling the text into its content and style in an embedding latent space, then applies generative modeling. TST models first learn the latent representations of the content and style of the given text. The latent representation of the original content is then combined with the latent representation of the desired target style to generate text in the target style. Techniques such as back-translation [11, 43, 60] and adversarial learning [13, 38, 61] have been proposed to disentangle latent representations into content and style. In general, total disentanglement is impossible without inductive biases or some other forms of supervision [62].

Pseudo-Parallel Corpus Creation

This process is used to train the model in a supervised way by generating pseudo-parallel data. One way of constructing pseudo-parallel data is through retrieval, i.e., extracting aligned sentence pairs from two mono-style corpora. [63] constructed pseudo-parallel corpora by matching sentence pairs in two style-specific corpora according to cosine similarity over pre-trained sentence embeddings. The

constructed pseudo-parallel corpora must reach a certain level of quality to be useful for TST.

4.6.3 Using Large Language Models

LLMs have revolutionized the field of natural language processing by generating coherent and contextually relevant text e.g., [64,65]. By learning from vast amounts of text data, LLMs capture various linguistic styles and nuances. This capability is particularly beneficial for TST tasks.

A distinctive feature of LLMs is their ability to perform valuable tasks without fine-tuning, showcasing zero- and few-shot capabilities [66]. Style transfer has been framed as a sentence rewriting task, enhancing LLMs' zero-shot performance for arbitrary TST by using task-related exemplars [67]. A reranking method has been proposed to select high-quality outputs from multiple candidates generated by the LLM, thereby improving performance [68]. Additionally, dynamic prompt generation has been introduced to guide the language model in producing text in the desired style [69].

While prompt engineering is the prevalent approach [70, 71], LLMs are highly sensitive to prompts [72, 73] and may not always guarantee optimal performance [69]. Despite good results for prompting, finetuning the LLMs still leads to significant performance improvements [74].

4.7 Evaluation Measures

A successful style transfer output is one that portrays the correct target style along with preserving the original semantics of the text and maintaining natural language fluency.

4.7.1 Automatic Evaluation

Automatic evaluation metrics provide an economic, reproducible, and scalable way to assess the quality of generation results. There are several automated evaluation metrics that have been proposed to measure the effectiveness of TST models [75–78]. They can be divided into three different categories based on the aspect of TST they focus on:

Style Transfer Strength

The ability to transfer the text style or the transfer strength of a TST model is measured using Style Transfer Accuracy [13, 30, 38, 79, 80]. Mostly, a binary style classifier [81] is pre-trained separately to predict the style label of the input sentence and

is then used to estimate the style transfer accuracy of the transferred style sentence. This is done by considering the target style as the ground truth.

Content Preservation

In order to measure the amount of original content preserved after the style transfer procedure, some automated evaluation metrics from other NLG tasks have been adopted for TST. For instance, the BLEU word-n-gram-overlap metric [82] is computed similarly as with machine translation. Match against a target-style sentence can be computed when parallel TST datasets or target-style human references are available. Since most of the TST tasks assume a non-parallel setting and matching references of style transferred sentences are not always a feasible option, evaluation using *source-BLEU (sBLEU)* is adopted. In this method, a transferred sentence is compared to its source. The overlap with the source is considered a proxy for content preservation. Cosine Similarity [83] can also be calculated between the original sentence embeddings and the transfer sentence embeddings [13]. This methodology follows the idea that the embeddings of the two sentences should be close if most of the semantics are preserved.

Fluency

One of the most common goals for all NLG tasks is producing fluent outputs. A common approach to measuring the fluency of a sentence is using a language model [84]: A pre-trained language model is used to compute the perplexity scores of the style-transferred sentences to evaluate the sentences' fluency.

4.7.2 Human Evaluation

Human evaluation stands out from automatic evaluation due to its flexibility and comprehensiveness. However, this evaluation approach is very challenging since the interpretation of text style can be subjective and vary from individual to individual [75, 76, 78]. In spite of this shortcoming, human evaluations still offer valuable insights into how well the TST algorithms can transfer style and generate sentences that are acceptable according to human standards.

In terms of evaluation types, there is point-wise scoring, wherein humans are asked to provide absolute scores of the model outputs (e.g. on a 1-5 Likert scale), and pairwise comparison, wherein they are asked to judge which of two outputs is better, or by providing a ranking for multiple outputs.

4.8 Applications

TST has a wide range of downstream applications in various NLP fields that include stylized chatbots [85], stylized writing assistants, automatic text simplification,

debiasing online text and even fighting against offensive language. A few very popular examples are discussed below.

[86] carried out a study that showcased the impact of chatbot's conversational style on users. [87] encoded personas of individuals in contextualized embeddings that helped in capturing the background information and style to maintain consistency in the generated responses. [88] focused on generating polite personalized dialog responses in agreement with the user's profile and consistent with their conversational history.

Another important application of TST is enhancing the human writing experience [89–91]. This application aids in people restyling their content to appeal to a variety of audiences, i.e., making a text polite, humorous, professional, or even Shakespearean.

Another inspiring application of TST is automatically simplifying content for better communication between experts and non-expert individuals in certain knowledge domains, thus lowering language barriers. For example, complicated legal, medical, or technical jargon is transferred into simple terms that a layman can comprehend [92].

TST can also offer a means to neutralize subjective attitudes for certain texts where objectivity is strongly needed. For example, in the domains of news, encyclopedia, and textbooks. Such applications can help in reshaping gender roles that are portrayed in writing [93]. TST can also help in transforming hateful sentences into non-hateful ones. For instance, [94] propose an extension of a basic encoder-decoder architecture by including a collaborative classifier to deal with abusive language redaction.

4.9 Ethical Concerns

An essential part of research is to consider the ethical implications of the project through its potential benefits and risks.

For example, TST has the potential to reduce toxicity, hate speech, sexist and racist language, aggression, harassment, trolling, and cyberbullying [95]. This task is beneficial for modeling non-offensive text to help reduce toxicity on social media platforms [54,96]. It can also be used on social chatbots to make sure there is no bad content in the generated text [97]. TST is also able to neutralize subjective-toned language, which can be helpful for certain types of publications such as textbooks [98].

However, the same technology can also be misused to purposely generate the opposite attribute, i.e., generating hateful, offensive text, that counters any intended social benefit [99]. Furthermore, as TST is now generally performed using trained language models, these inherit all the potential risks associated with this technology in general, such as reflecting unjust, toxic, or oppressive speech present in the training data [100].

The goal of a discussion on ethics is to take into account various concerns like how a system should be built, who it is intended for, and how to assess its societal impact [99] [101]. Instead of abandoning the whole idea of building such tools, one must explore the concerns and find ways to deal with them [102]. This should be viewed as an opportunity to increase transparency by surfacing the risks and finding the best ways to its strategy into practice.

4.10 Conclusion

The main goal of this work is to offer an introductory overview of Text Style Transfer (TST), highlighting key components such as subtasks, datasets, evaluation methods, and the challenges inherent to TST. Additionally, we discussed the ethical considerations surrounding this area of research. We aim for this overview to serve as a useful guide for those new to the field.

Acknowledgment

This research was funded by the European Union (ERC, NG-NLG, 101039303) and by Charles University projects GAUK 392221 and SVV 260698.

Bibliography

[1] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[2] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," *arXiv preprint arXiv:1503.02364*, 2015.

[3] A. M. Rush, S. Harvard, S. Chopra, and J. Weston, "A neural attention model for sentence summarization," in *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, 2017.

[4] Y. Len, F. Portet, C. Labbé, and R. Qader, "Controllable neural natural language generation: comparison of state-of-the-art control strategies," in *Proc. of the 3rd Workshop on Natural Language Generation from the Semantic Web*, 2020.

[5] N. Dziri, E. Kamalloo, K. Mathewson, and O. Zaiane, "Augmenting neural response generation with context-aware topical attention," in *Proc. of the First Workshop on NLP for Conversational AI*. Association for Computational Linguistics, Aug. 2019, pp. 18–31.

[6] X. Feng, M. Liu, J. Liu, B. Qin, Y. Sun, and T. Liu, "Topic-to-essay generation with neural networks." in *IJCAI*, 2018, pp. 4078–4084.

[7] L. Wang, J. Yao, Y. Tao, L. Zhong, W. Liu, and Q. Du, "A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization," in *Proc. of the 27 Int. Joint Conf. on Artificial Intelligence*, 7 2018, pp. 4453–4460.

[8] C. Xing, W. Wu, Y. Wu, J. Liu, Y. Huang, M. Zhou, and W.-Y. Ma, "Topic aware neural response generation," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 31, no. 1, 2017.

[9] J. Li, R. Jia, H. He, and P. Liang, "Delete, retrieve, generate: a simple approach to sentiment and style transfer," in *Proc. of the 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1865–1874.

[10] A. Sudhakar, B. Upadhyay, and A. Maheswaran, "transforming delete, retrieve, generate approach for controlled text style transfer," in *Proc. of the Conf. on Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3260–3270.

[11] S. Prabhumoye, Y. Tsvetkov, R. Salakhutdinov, and A. W. Black, "Style transfer through back-translation," in *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 866–876.

[12] L. Chen, S. Dai, C. Tao, H. Zhang, Z. Gan, D. Shen, Y. Zhang, G. Wang, R. Zhang, and L. Carin, "Adversarial text generation via feature-mover's distance," in *Advances in Neural Information Processing Systems*, 2018, pp. 4666–4677.

[13] Z. Fu, X. Tan, N. Peng, D. Zhao, and R. Yan, "Style transfer in text: Exploration and evaluation," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 32, no. 1, 2018.

[14] X. Kong, B. Li, G. Neubig, E. Hovy, and Y. Yang, "An adversarial approach to high-quality, sentiment-controlled neural dialogue generation," *arXiv preprint arXiv:1901.07129*, 2019.

[15] X. Sun, J. Li, X. Wei, C. Li, and J. Tao, "Emotional editing constraint conversation content generation based on reinforcement learning," *Inf. Fusion*, vol. 56, pp. 70–80, 2020.

[16] H. Zhou, M. Huang, T. Zhang, X. Zhu, and B. Liu, "Emotional chatting machine: Emotional conversation generation with internal and external memory," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 32, no. 1, 2018.

[17] J. Li, M. Galley, C. Brockett, G. Spithourakis, J. Gao, and B. Dolan, "A persona-based neural conversation model," in *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Aug.

2016, pp. 994–1003.

[18] Y. Luan, C. Brockett, B. Dolan, J. Gao, and M. Galley, "Multi-task learning for speaker-role adaptation in neural conversation models," in *Proc. of the 8 Int. Joint Conf. on Natural Language Processing (Volume 1: Long Papers)*, Nov. 2017, pp. 605–614.

[19] M. Yang, Q. Qu, K. Lei, J. Zhu, Z. Zhao, X. Chen, and J. Z. Huang, "Investigating deep reinforcement learning techniques in personalized dialogue generation," in *Proc. of the SIAM Int. Conf. on Data Mining*, 2018, pp. 630–638.

[20] M. Yang, Z. Zhao, W. Zhao, X. Chen, J. Zhu, L. Zhou, and Z. Cao, "Personalized response generation via domain adaptation," in *Proc. of the 40th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2017, pp. 1021–1024.

[21] A. Jaech, and M. Ostendorf, "Low-rank RNN adaptation for context-aware language modeling," *Trans. of the Association for Computational Linguistics*, vol. 6, pp. 497–510, 2018.

[22] L. Wang, J. Yao, Y. Tao, L. Zhong, W. Liu, and Q. Du, "A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization," *arXiv preprint arXiv:1805.03616*, 2018.

[23] T. Young, E. Cambria, I. Chaturvedi, H. Zhou and S. Biswas, and M. Huang, "Augmenting end-to-end dialogue systems with commonsense knowledge," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 32, no. 1, 2018.

[24] Z. Hu, R. K.-W. Lee, C. C. Aggarwal, and A. Zhang, "Text style transfer: A review and experimental evaluation," *ACM SIGKDD Explorations Newsletter*, vol. 24, no. 1, pp. 14–45, 2022.

[25] S. Prabhumoye, A. W. Black, and R. Salakhutdinov, "Exploring controllable text generation techniques," *arXiv preprint arXiv:2005.01822*, 2020.

[26] S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu, "Plug and play language models: A simple approach to controlled text generation," *arXiv preprint arXiv:1912.02164*, 2019.

[27] S. Gehrmann, F. Z. Dai, H. Elder, and A. M. Rush, "End-to-end content and plan selection for data-to-text generation," *arXiv preprint arXiv:1810.04700*, 2018.

[28] L. Mou and O. Vechtomova, "Stylized text generation: Approaches and applications," in *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, 2020, pp. 19–22.

[29] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and B. Dolan, "A persona-based neural conversation model," *arXiv preprint arXiv:1603.06155*, 2016.

[30] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, "Toward controlled generation of text," in *Proc. of the Int. Conf. on Machine Learning*. PMLR, 2017, pp. 1587–1596.

[31] D. Jin, Z. Jin, Z. Hu, O. Vechtomova, and R. Mihalcea, "Deep learning for text style transfer: A survey," *Computational Linguistics*, vol. 48, no. 1, pp. 155–205, 2022.

[32] J. Xu, X. Sun, Q. Zeng, X. Ren, X. Zhang, H. Wang, and W. Li, "Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach," *arXiv preprint arXiv:1805.05181*, 2018.

[33] V. John, L. Mou, H. Bahuleyan, and O. Vechtomova, "Disentangled representation learning for non-parallel text style transfer," *arXiv preprint arXiv:1808.04339*, 2018.

[34] M. Toshevska and S. Gievska, "A review of text style transfer using deep learning," *IEEE Trans. on Artificial Intelligence*, 2021.

[35] D. D. McDonald and J. Pustejovsky, "A computational theory of prose style for natural language generation," in *Proc. of the Conf. of the European Chapter of the Association for Computational Linguistics*, 1985.

[36] E. Hovy, "Generating natural language under pragmatic constraints," *Journal of Pragmatics*, vol. 11, no. 6, pp. 689–719, 1987.

[37] S. Rao and J. Tetreault, "Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer," *arXiv preprint arXiv:1803.06535*, 2018.

[38] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola, "Style transfer from non-parallel text by cross-alignment," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[39] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proc. of the 25th Int. Conf. on World Wide Web*, 2016, pp. 507–517.

[40] N. Dai, J. Liang, X. Qiu, and X.-J. Huang, "Style transformer: Unpaired text style transfer without disentangled latent representation," in *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5997–6007.

[41] D. Lee, Z. Tian, L. Xue, and N. L. Zhang, "Enhancing content preservation in text style transfer using reverse attention and conditional layer normalization," *arXiv preprint arXiv:2108.00449*, 2021.

[42] A. Belz, S. Agarwal, E. Reiter, and A. Shimorina, "Reprogen: Proposal for a shared task on reproducibility of human evaluations in NLG," 2020.

[43] S. Mukherjee, Z. Kasner, and O. Dušek, "Balancing the style-content trade-off in sentiment transfer using polarity-aware denoising," in *Proc. of the Int. Conf. on Text, Speech, and Dialogue*, 2022, pp. 172–186.

[44] S. Mukherjee, V. Hudeček, and O. Dušek, "Polite chatbot: A text style transfer application," in *Proc. of the 17th Conf. of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, May 2023, pp. 87–93.

[45] A. Madaan, A. Setlur, T. Parekh, B. Poczos, G. Neubig, Y. Yang, R. Salakhutdinov, A. W. Black, and S. Prabhumoye, "Politeness transfer: A tag and generate approach," *arXiv preprint arXiv:2004.14257*, 2020.

[46] J. Shetty and J. Adibi, "The enron email dataset database schema and brief statistical report," *Information sciences institute technical report, University of Southern California*, vol. 4, no. 1, pp. 120–128, 2004.

[47] S. Mukherjee, A. Bansal, P. Majumdar, A. K. Ojha, and O. Dušek, "Low-resource text style transfer for Bangla: Data & models," in *Proc. of the First Workshop on Bangla Language Processing*, Dec. 2023, pp. 34–47.

[48] W. Xu, A. Ritter, B. Dolan, R. Grishman, and C. Cherry, "Paraphrasing for style," in *Proc. of COLING 2012*, 2012, pp. 2899–2914.

[49] H. Jhamtani, V. Gangal, E. Hovy, and E. Nyberg, "Shakespearizing modern language using copy-enriched sequence to sequence models," in *Proc. of the Workshop on Stylistic Variation*, 2017, pp. 10–19.

[50] J. He, X. Wang, G. Neubig, and T. Berg-Kirkpatrick, "A probabilistic formulation of unsupervised text style transfer," in *Proc. of the Int. Conf. on Learning Representations*, 2020.

[51] Z. Zhu, D. Bernhard, and I. Gurevych, "A monolingual tree-based translation model for sentence simplification," in *Proc. of the 23rd Int. Conf. on Computational Linguistics*, 2010, pp. 1353–1361.

[52] L. Van den Bercken, R.-J. Sips, and C. Lofi, "Evaluating neural text simplification in the medical domain," in *Proc. of the World Wide Web Conf.*, 2019, pp. 3286–3292.

[53] M. Sourabrata, B. Akanksha, K. O. Atul, P. M. John, and D. Ondrej, "Text detoxification as style transfer in English and Hindi," in *Proc. of the 20th Int. Conf. on Natural Language Processing*, Dec. 2023, pp. 133–144.

[54] C. dos Santos, I. Melnyk, and I. Padhi, "Fighting offensive language on social media with unsupervised text style transfer," in *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 189–194.

[55] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[56] S. Mukherjee, A. K. Ojha, A. Bansal, D. Alok, J. P. McCrae, and O. Dušek, "Multilingual text style transfer: Datasets & models for Indian languages," *arXiv preprint arXiv:2405.20805*, 2024.

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

[58] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[59] S. Mukherjee, A. Bansal, P. Majumdar, A. K. Ojha, and O. Dušek, "Low-resource text style transfer for bangla: Data & models," in *Proc. of the First Workshop on Bangla Language Processing*, 2023, pp. 34–47.

[60] Z. Zhang, S. Ren, S. Liu, J. Wang, P. Chen, M. Li, M. Zhou, and E. Chen, "Style transfer as unsupervised machine translation," *CoRR*, vol. abs/1808.07894, 2018.

[61] J. Zhao, Y. Kim, K. Zhang, A. M. Rush, and Y. LeCun, "Adversarially regularized autoencoders," in *Proc. of the 35th Int. Conf. on Machine Learning*, 2018, pp. 9405–9420.

[62] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *international Conf. on machine learning*, 2019, pp. 4114–4124.

[63] Z. Jin, D. Jin, J. Mueller, N. Matthews, and E. Santus, "IMaT: Unsupervised text attribute transfer via iterative matching and translation," in *Proc. of the Conf. on Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. on Natural Language Processing*, 2019, pp. 3088–3100.

[64] H. Touvron, T. Lavril, G. Izacard, *et al.*, "LLaMA: Open and Efficient Foundation Language Models, CoRR, vol. abs/2302.13971, 2023.

[65] H. Touvron, L. Martin, K. Stone, *et al.*, "Llama 2: Open Foundation and Fine-tuned Chat Models, CoRR, vol. abs/2307.09288, 2023.

[66] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Comput. Surv.*, vol. 55, no. 9, pp. 195:1–195:35, 2023.

[67] E. Reif, D. Ippolito, A. Yuan, A. Coenen, C. Callison-Burch, and J. Wei, "A recipe for arbitrary text style transfer with large language models," *arXiv preprint arXiv:2109.03910*, 2021.

[68] M. Suzgun, L. Melas-Kyriazi, and D. Jurafsky, "Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models," *arXiv preprint arXiv:2205.11503*, 2022.

[69] Q. Liu, J. Qin, W. Ye, H. Mou, Y. He, and K. Wang, "Adaptive prompt routing for arbitrary text style transfer with pre-trained language models," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 18 689–18 697.

[70] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[71] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig, "How can we know what language models know?" *Trans. of the Association for Computational Linguistics*, vol. 8, pp. 423–438, 2020.

[72] S. Mishra, D. Khashabi, C. Baral, Y. Choi, and H. Hajishirzi, "Reframing instructional prompts to gptk's language," *arXiv preprint arXiv:2109.07830*, 2021.

[73] K. Zhu, J. Wang, J. Zhou, Z. Wang, H. Chen, Y. Wang, L. Yang, W. Ye, N. Z. Gong, Y. Zhang *et al.*, "Promptbench: Towards evaluating the robustness of large language models on adversarial prompts," *arXiv preprint arXiv:2306.04528*, 2023.

[74] S. Mukherjee, A. K. Ojha, and O. Dušek, "Are large language models actually good at text style transfer?" *arXiv preprint arXiv:2406.05885*, 2024.

[75] R. Y. Pang, "Towards actual (not operational) textual style transfer auto-evaluation," in *Proc. of the 5th Workshop on Noisy User-generated Text*, 2019, pp. 444–445.

[76] ——, "The daunting task of real-world textual style transfer auto-evaluation," *CoRR*, vol. abs/1910.03747, 2019.

[77] R. Y. Pang and K. Gimpel, "Unsupervised evaluation metrics and learning criteria for non-parallel textual transfer," in *Proc. of the 3rd Workshop on Neural Generation and Translation*, 2019, pp. 138–147.

[78] R. Mir, B. Felbo, N. Obradovich, and I. Rahwan, "Evaluating style transfer for text," in *Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies)*, 2019, pp. 495–504.

[79] F. Luo, P. Li, J. Zhou, P. Yang, B. Chang, X. Sun, and Z. Sui, "A dual reinforcement learning framework for unsupervised text style transfer," in *Proc. of the 28th Int. Joint Conf. on Artificial Intelligence*. AAAI Press, 2019, pp. 5116–5122.

[80] V. John, L. Mou, H. Bahuleyan, and O. Vechtomova, "Disentangled representation learning for non-parallel text style transfer," in *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 424–434.

[81] A. Moschitti, B. Pang, and W. Daelemans, Eds., *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, 2014.

[82] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. of the 40th Annual Meeting on Association for Computational Linguistics*, 2002, pp. 311–318.

[83] F. Rahutomo, T. Kitasuka, and M. Aritsugi, "Semantic cosine similarity," in *The 7th Int. Student Conf. on Advanced Science and Technology*, vol. 4, no. 1, 2012, p. 1.

[84] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, 1995, pp. 181–184 vol.1.

[85] S. Mukherjee, "Stylized dialog response generation," in *Proc. of the 19th Annual Meeting of the Young Reseachers' Roundtable on Spoken Dialogue*

Systems, 2023, pp. 44–46.

[86] S. Kim, J. Lee, and G. Gweon, "Comparing data from chatbot and web surveys: Effects of platform and conversational style on survey response quality," in *Proc. of the CHI Conf. on Human Factors in Computing Systems*, 2019, pp. 1–12.

[87] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and W. B. Dolan, "A persona-based neural conversation model," in *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[88] M. Firdaus, A. Shandilya, A. Ekbal, and P. Bhattacharyya, "Being polite: Modeling politeness variation in a personalized dialog agent," *IEEE Trans. on Computational Social Systems*, 2022.

[89] F. Can and J. M. Patton, "Change of writing style with time," *Computers and the Humanities*, vol. 38, no. 1, pp. 61–82, 2004.

[90] B. Johnstone, "Stance, style, and the linguistic individual," *Stance: Sociolinguistic Perspectives*, pp. 29–52, 2009.

[91] V. G. Ashok, S. Feng, and Y. Choi, "Success with style: Using writing style to predict the success of novels," in *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, 2013, pp. 1753–1764.

[92] Y. Cao, R. Shui, L. Pan, M.-Y. Kan, Z. Liu, and T.-S. Chua, "Expertise style transfer: A new task towards better communication between experts and laymen," in *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[93] E. Clark, A. S. Ross, C. Tan, Y. Ji, and N. A. Smith, "Creative writing with a machine in the loop: Case studies on slogans and stories," in *Proc. of the Int. Conf. on Intelligent User Interfaces*, 2018, pp. 329–340.

[94] C. N. d. Santos, I. Melnyk, and I. Padhi, "Fighting offensive language on social media with unsupervised text style transfer," *arXiv preprint arXiv:1805.07685*, 2018.

[95] Z. Waseem, T. Davidson, D. Warmsley, and I. Weber, "Understanding abuse: A typology of abusive language detection subtasks," *arXiv preprint arXiv:1705.09899*, 2017.

[96] S. Harrison. (2019) Twitter and Instagram unveil new ways to combat hate–again. [Online]. Available: https://www.wired.com/story/twitter-instagram-unveil-new-ways-combat-hate-again/

[97] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, E. M. Smith, Y. Boureau, and J. Weston, "Recipes for building an open-domain chatbot," in *Proc. of the 16th Conf. of the European Chapter of the Association for Computational Linguistics*, 2021, pp. 300–325.

[98] R. Pryzant, R. D. Martinez, N. Dass, S. Kurohashi, D. Jurafsky, and D. Yang, "Automatically neutralizing subjective bias in text," in *Proc. of the aaai Conf. on artificial intelligence*, vol. 34, no. 01, 2020, pp. 480–489.

[99] D. Hovy and S. L. Spruit, "The social impact of natural language processing," in *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 591–598.

[100] L. Weidinger, J. Uesato, M. Rauh, C. Griffin, P.-S. Huang, J. Mellor, A. Glaese, M. Cheng, B. Balle, A. Kasirzadeh *et al.*, "Taxonomy of risks posed by language models," in *Proc. of the ACM Conf. on Fairness, Accountability, and Transparency*, 2022, pp. 214–229.

[101] T. L. Beauchamp and J. F. Childress, "Respect for autonomy," *Principles of biomedical ethics*, vol. 5, pp. 57–112, 2001.

[102] J. L. Leidner and V. Plachouras, "Ethical by design: Ethics best practices for natural language processing," in *Proc. of the First ACL Workshop on Ethics in Natural Language Processing*, 2017, pp. 30–40.

Chapter 5

Simulation Study on Super-Resolution for Coded Aperture Gamma Imaging

Tobias Meißner
Mannheim Institute for Intelligent Systems in Medicine
Heidelberg University, Germany
tobias.meissner@medma.uni-heidelberg.de
ORCID: 0000-0002-9680-7153

Werner Nahm
Institute of Biomedical Engineering
Karlsruhe Institute of Technology (KIT), Germany
werner.nahm@kit.edu
ORCID: 0000-0001-8095-5090

Jürgen Hesser
Mannheim Institute for Intelligent Systems in Medicine
Interdisciplinary Center for Scientific Computing (IWR)
Central Institute for Computer Engineering (ZITI)
CZS Heidelberg Center for Model-Based AI
Heidelberg University, Germany
juergen.hesser@medma.uni-heidelberg.de

Nikolas Löw
Mannheim Institute for Intelligent Systems in Medicine
Heidelberg University, Germany
nikolas.loew@medma.uni-heidelberg.de

DOI: 10.54103/milanoup.282.c637

5.1 Abstract

Coded Aperture Imaging (CAI) has been proposed as an alternative collimation technique in nuclear imaging. To maximize spatial resolution small pinholes in the coded aperture mask are required. However, a high-resolution detector is needed to correctly sample the point spread function (PSF) to keep the Nyquist-Shannon sampling theorem satisfied. The disadvantage of smaller pixels, though, is the resulting higher Poisson noise. Thus, the aim of this paper was to investigate if sufficiently accurate CAI reconstruction is achievable with a detector which undersamples the PSF. With the Monte Carlo simulation framework TOPAS a test image with multiple spheres of different diameter was simulated based on the setup of an experimental gamma camera from previous work. Additionally, measured phantom data were acquired. The captured detector images were converted to low-resolution images of different pixel sizes according to the super-resolution factor k. Multiple analytical reconstruction methods and a Machine Learning approach were compared based on the contrast-to-noise ratio (CNR). We show, that all reconstruction methods are able to reconstruct both the test image and the measured phantom data for $k \leq 7$. With a synthetic high-resolution PSF and upsampling the simulated low-resolution detector image by bilinear interpolation the CNR can be kept approximately constant. Results of this simulation study and additional validation on measured phantom data indicate that an undersampling detector can be combined with small aperture holes. However, further experiments need to be conducted.

5.2 Introduction

Accurate localization and visualization of radioactive sources is an essential task in nuclear medicine [1], high-energy astrophysics [2] and in monitoring of nuclear

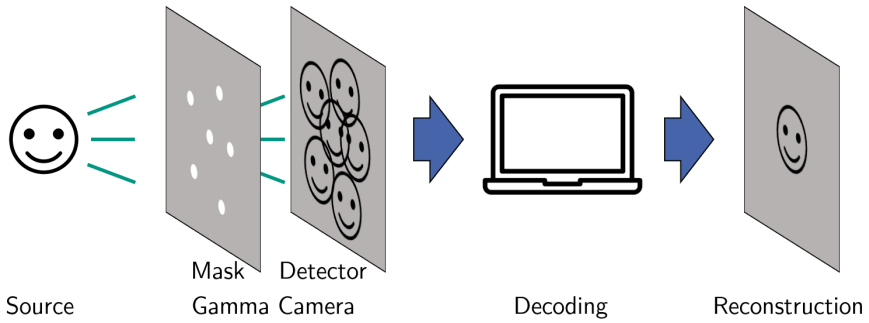


Figure 5.1 *The basic principle of planar Coded Aperture Imaging: A mask with pinholes projects the source image onto the detector, where multiple overlapping projections emerge. Decoding or image reconstruction is necessary to obtain the original source image.*

Figure 5.2 *Super-resolution is referred to the process to combine multiple low quality images to reconstruct an image of the underlying high-resolution scene.*

waste [3, 4]. Recently, small handheld gamma cameras for localizing sentinel lymph nodes in breast cancer patients are under investigation [5–8]. Due to the high-energy photons involved refractive lenses cannot be used for producing an image of the scene and instead parallel or pinhole collimators are employed to capture the necessary spatial information [1]. However, the size of the opening is usually subject to a balanced trade-off between the number of captured photons (photon efficiency) and the spatial resolution as the former increases and the latter decreases with the size of the pinhole. A high photon efficiency is desired since the guiding principle in the medical domain is to reduce the exposed radiation to As Low As Reasonable Achievable (ALARA). Thus, photon flux is limited and achieving high quantum yield is of major importance.

To improve the mentioned trade-off Coded Aperture Imaging (CAI) has been introduced [9, 10]: A mask between object and detector consisting of a radiopaque material with pinholes encodes the directional information of incoming gamma rays. As Figure 5.1 shows, each pinhole in the mask generates a projection of the source image on the detector resulting in a multitude of overlapping projections. Therefore, image reconstruction (also referred to as decoding) becomes necessary. If the distance between source and collimator is large and the extension in depth is small relative to the distance, CAI can be considered as an image-to-image mapping and is denoted as planar Coded Aperture Imaging [11]. In this paper, the captured detector image is denoted as $p(x, y)$, the original source image and its reconstruction as $f(x, y)$ and $\hat{f}(x, y)$, respectively.

The term super-resolution refers to the process of combining several low resolution, noisy, slightly shifted observations [12] to reconstruct an image of the underlying high resolution scene, as Figure 5.2 illustrates.

Because the spatial resolution in CAI is mainly influenced by the mask's pinhole diameter [13], increasing the MURA rank and thus the amount of pinholes while reducing its diameter would increase the spatial resolution. So far, the pinhole diameter has been chosen such that the utilized detector can properly sample the resulting PSF [7, 8].

To the best of the authors' knowledge, no research group has investigated the combination of small pinholes and a low-resolution detector. Therefore, the investigated hypothesis is as follows: Existing CAI reconstruction methods are capable of reconstructing point sources from an undersampling detector, and thus achieving super-resolution, at reasonable quality even though the detector cannot resolve the higher spatial resolution of the aperture. This is due to the shifted but overlapping projections caused by the coded aperture.

5.3 Methods

5.3.1 Simulating a coded aperture test image

For simulating a test image the Monte Carlo simulation toolkit TOPAS [14], a wrapper library around Geant4 [15], is deployed. Unlike ray-casting simulations, TOPAS accounts for photon-mass interactions like scattering and mask penetration, and is therefore considered to be the gold standard in gamma imaging [16].

The geometrical components and dimensions were simulated according to the experimental gamma camera from Rozhkov et al. [16]. The main characteristics can be summarized as follows: A 2×2 mosaicked, 1 mm thick Tungsten not-two-holes-touching (NTHT) MURA mask of rank 31 with pinholes of 0.34 mm in diameter (denoted as d) was placed 42 mm (a) in front of a 2 mm thick 256×256 pixelated CdTe semiconductor detector coupled to a Timepix© readout circuit. The detector has a side length of 14.1 mm and hence a single pixel size $s = 0.0551$ mm. The virtual object plane is 172 mm (b) in front of the mask plane, resulting in a field of view (FoV) of 57.75×57.75 mm.

The test image consists of three spherical sources with diameters d_1, d_2 and d_3 of 1, 2 and 3 mm distributed within the FoV as Figure 5.8 shows. 10^9 gamma photons with a photon energy of 140.5 keV (corresponding to the photon peak of ^{99m}Tc the most commonly used radiotracer in nuclear medicine [1]) were distributed to the three sources according to their area. Every photon hitting the detector was collected and stored in a so called *phase space file*. In addition to the coded aperture a single pinhole collimator with the same diameter was simulated to serve as a reference for the reconstructed images. The captured pinhole image was smoothed by Gaussian blurring with a σ of 2 pixels.

The ground truth image was generated from the geometrical model and remains binary: 1 for where a source is located and 0 everywhere else.

5.3.2 Measured data from the experimental gamma-camera

Captured images from an experimental gamma-camera also used in previous work [16, 17] was used to validate the effect of super-resolution on real-world measurement data. The camera set-up was the same as for the simulation of the test

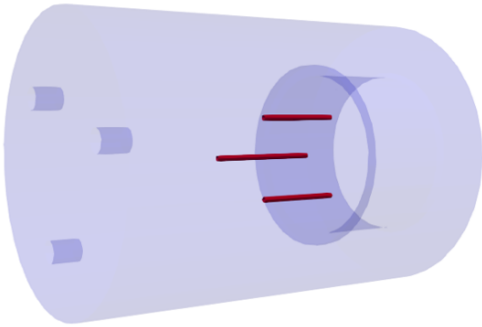


Figure 5.3 *The utilized phantom with its three tubes (red) filled with ^{99m}Tc of which 120 images where captured.*

image. The phantom has the basic form of a cylinder with a height of 80 mm and 50 mm in diameter, where tubes along the vertical axis were filled with ^{99m}Tc. These three tubes have a diameter of 1.1 mm, and two of them are 15 mm long while the central one is 20 mm long. The total activity at the beginning of the measurements was 83 MBq. A depiction of the geometric computer model can be seen in Figure 5.3. The phantom was exposed to the gamma-camera for 2 min and afterwards rotated by 3 degrees. This way, a total of 120 images were captured. Outlier replacement as described in [17] was applied afterwards.

5.3.3 Generating low-resolution detector images

To analyze the effect of different pixel sizes low-resolution images of different resolutions were produced as follows: The captured photons from the phase space file and from the measured phantom data respectively were binned into images of different low resolution. The actual detector served as reference with a resolution of 256×256 pixels, which corresponds to the resolution of the final reconstructed image. Therefore, the super-resolution factor k is introduced. It means that $k \times k$ high-resolution pixels are reconstructed from a single low-resolution pixel. Note that the absolute detector size remains the same: 14.1×14.1 mm. The single pixel size s changes proportional to k: $s = k \cdot 14.1\,\text{mm}/256$. Finally, all low-resolution images were upsampled by bilinear interpolation to 256×256 pixels in order to fit the synthetic high-resolution PSF and to fit into the CED-IN respectively. The process of generating low-resolution images is shown in Figure 5.4 for $k = 8$.

5.3.4 Analytical reconstruction methods

Three different methods for super-resolution reconstruction are analyzed and compared in this paper: MURA decoding [18], a convolutional Maximum Likelihood Expectation Maximization algorithm (MLEM) [19] and a convolutional

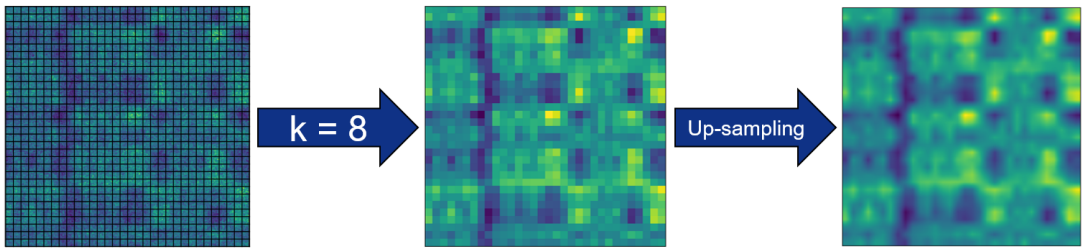


Figure 5.4 *Pixels of the high-resolution detector image from the TOPAS simulation are accumulated (here with $k = 8$ into 32×32 pixels) to form the low-resolution detector image. Afterwards this image is upsampled by bilinear interpolation to the high-resolution of 256×256 pixels.*

encoder-decoder network (CED) from previous work [17]. MURA Decoding is the most commonly used reconstruction method. It consists of a single circular convolution of the detector image $p(x, y)$ with the decoding pattern $g(x, y)$:

$$\hat{f}(x, y) = p(x, y) \circledast g(x, y) \tag{5.1}$$

with $\circledast$ denoting the circular convolution operator. All circular operations in this paper are carried out by periodically padding the second operand to twice its size, i.e. to 512×512 pixels, and cropping the result to its central 256×256 pixels.

The decoding pattern $g(x, y)$ is based on $h(x, y)$ and its definition can be found in [20]: It is equivalent to changing all 0 to -1 and adding a positive pixel to the center of the PSF [20].

The MLEM algorithm works in iterations and is derived from a random Poisson process. It consists of a combination of forward and backward projections, where ten iterations were deployed in this paper:

$$\hat{f}^{k+1}(x, y) = \hat{f}^{k}(x, y) \odot \left[\frac{p(x, y)}{\hat{f}^{k}(x, y) \circledast h(x, y)} \otimes h(x, y) \right] \tag{5.2}$$

where $\odot$ denotes point-wise multiplication and $\otimes$ is circular cross-correlation.

Instead of the real measured PSF with round pinhole projections, the two-holes-touching (THT) version of the PSF without gaps between neighboring pinholes is used for reconstruction, since it suppresses periodical noise [17]. Both the THT-PSF and its corresponding decoding pattern are of rectangular structure and a square of 8 bright pixels represents the position and bounding box of each projected pinhole. They also define the reconstructed resolution of 256×256 pixels. Figure 5.5 depicts the measured PSF, the THT-PSF and the decoding pattern.

Additionally, MURA Decoding with low-resolution THT-PSF was implemented, where the synthetic high-resolution THT-PSF was not used, but the down-sampled

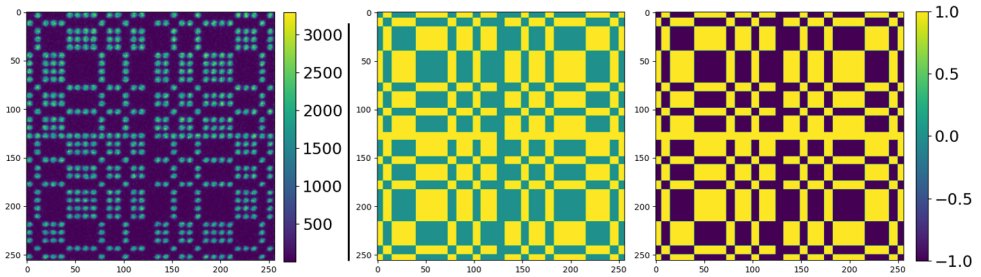


Figure 5.5 *Left: The measured point spread function (PSF) of the experimental gamma camera, where the pixel intensity represents photon counts. Center: The two-holes-touching (THT) version of the used rank 31 MURA mask $h(x, y)$ fundamental for MURA Decoding and MLEM. Right: The respective decoding pattern $g(x, y)$ for MURA Decoding. Note the additional positive square at the center of $g(x, y)$ [20]. Both patterns were resized to 256×256 pixels by nearest neighbor interpolation to maintain the original shape.*

THT-PSFs emulating a low-resolution detector. Since the reconstructions come in low-resolution, the reconstructed images were upsampled to 256×256 pixels by bilinear interpolation.

5.3.5 Reconstruction by Machine Learning

A Convolutional Encoder-Decoder (CED) is a widely used form of Convolutional Neural Networks (CNN). A CED consists of trainable parameters that transform an input into an output image. However, this transformation is not derived from a mathematical description but by providing a sufficient amount of paired training images. First experiments were conducted on the application of CNNs to CAI reconstruction, but its validation exclusively relied on simulated and low-resolution images [21] or were only visually compared on few images [13]. While the two analytical reconstruction methods solely rely on the PSF of the gamma camera, which acts as a linear approximation of the imaging system, the CED-IN is in theory capable of more complex mappings [22]. Recent advances in Machine Learning in the field of image reconstruction [23–25] underline the potential of CEDs for CAI reconstruction. The CED used in this paper is denoted as CED-IN because it was trained with a convolutional simulation based on natural photographs from the ImageNet database [26]. Its architecture is presented in Figure 5.6 and for more information on the training process and data simulation the reader is referred to [17].

After reconstructing all images, the contrast-to-noise ratio (CNR) is calculated based on the reconstructed and the ground truth image. The binary ground truth image enables a separation of the reconstruction into the signal part S and

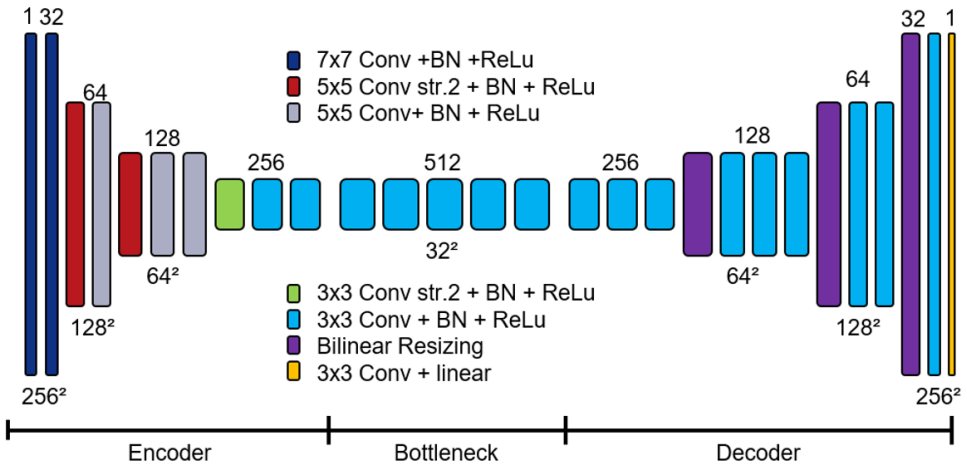


Figure 5.6 *The convolutional encoder-decoder network architecture deployed in this paper. The top row represents the number of filters per layer and the bottom row the feature map size in pixels.*

background part B. The following definition of CNR is employed [27]:

$$\mathrm{CNR} = \frac{\left|\bar{S} - \bar{B}\right|}{\sigma_B}, \tag{5.3}$$

where $\bar{S}$ denotes the mean intensity of the signal, $\bar{B}$ the mean intensity and σ_B the standard deviation of the background.

5.3.6 Nyquist-Shannon sampling theorem

The Nyquist-Shannon sampling theorem states that the sampling frequency of a pixelated representation must be larger than twice the maximum frequency of the periodic image [28]. Thus, when the smallest occurring structure is sampled by two pixels or less, an image is not represented unambiguously which leads to aliasing and hence signal degradation [28]. Since the aforementioned analytical reconstruction methods MURA Decoding and MLEM consist of one or more convolutions of two discretized signals, a reconstruction without aliasing artefacts is only possible when both images were sampled by enough pixels. Thus, critical super-resolution factors $\tilde{k}$ were determined both for the coded aperture test image and the THT-PSF $h(x, y)$. The smallest point source of the test image is 1 mm wide and therefore much larger than the pinhole diameter: $d_1 \gg d$. Hence, the smallest structure on the detector caused by the small point source can be approximated by $t = d_1 \cdot m = 1.244$ mm. For $h(x, y)$ the smallest structure t is 8 pixels wide, i.e. $t = 8 \cdot s$ (Figure 5.5). For the given gamma camera with its magnification factor $m = (1 + a/b) = 1.244$, the smallest

depicted structure t and the single pixel side length of $s = 0.0551\,\text{mm}$ $\tilde{k}$ can be defined as follows:

$$\tilde{k} = \left\lfloor \frac{1}{2} \cdot \frac{t}{s} \right\rfloor \tag{5.4}$$

where $\lfloor \cdot \rfloor$ denotes rounding off to the next smallest integer value.

5.4 Results

5.4.1 Critical super-resolution factors

The following critical super-resolution factors $\tilde{k}$ were obtained from the Nyquist-Shannon sampling theorem: The THT-PSF $h(x, y)$ must be sampled by at least 64×64 pixels leading to the following critical super-resolution factor: $\tilde{k}_{\text{THT-PSF}} = 4$. This means that the synthetic high-resolution THT-PSF in this paper is 4-times oversampled. However, the test image with the 1 mm point source results in a higher critical super-resolution factor of $\tilde{k}_1 = \lfloor 11.29 \rfloor = 11$. The tubes of the phantom captured by the experimental gamma-camera have a diameter of 1.1 mm, that are magnified to approximately 1.37 mm and thus 24.83 pixels. This results in a critical super-resolution factor for the measured data of $\tilde{k}_{\text{measurement}} = \lfloor 12.42 \rfloor = 12$.

5.4.2 Results on the test image

Figure 5.7 shows the CNR of the four reconstruction methods over the super-resolution factor k. The red dotted line at CNR = 7.63 denotes the smoothed image captured with a pinhole collimator where no reconstruction was required. It is depicted central on the right-hand side together with the ground truth image and the coded aperture test image in Figure 5.8. The left-hand side shows exemplary reconstructions for $k = 1, 3, 6, 11$ and 16.

Clearly visible, the CNRs of the reconstruction methods with the synthetic high-resolution THT-PSF increase until $k = 3$ and steadily decline afterwards. The CED-IN is an exception, where the CNR increases further until falling below its baseline at $k = 13$. For all reconstruction methods using the synthetic high-resolution THT-PSF the smallest point source starts to disappear for $k > 11$ and is hardly visible for $k = 16$.

MURA Decoding with the respective low-resolution THT-PSF does not exceed its baseline CNR and falls beneath the pinhole reference at $k = 5$ and again for all $k > 9$. The reconstructions for $k \geq 11$ fail entirely and resemble no similarity to the ground truth anymore.

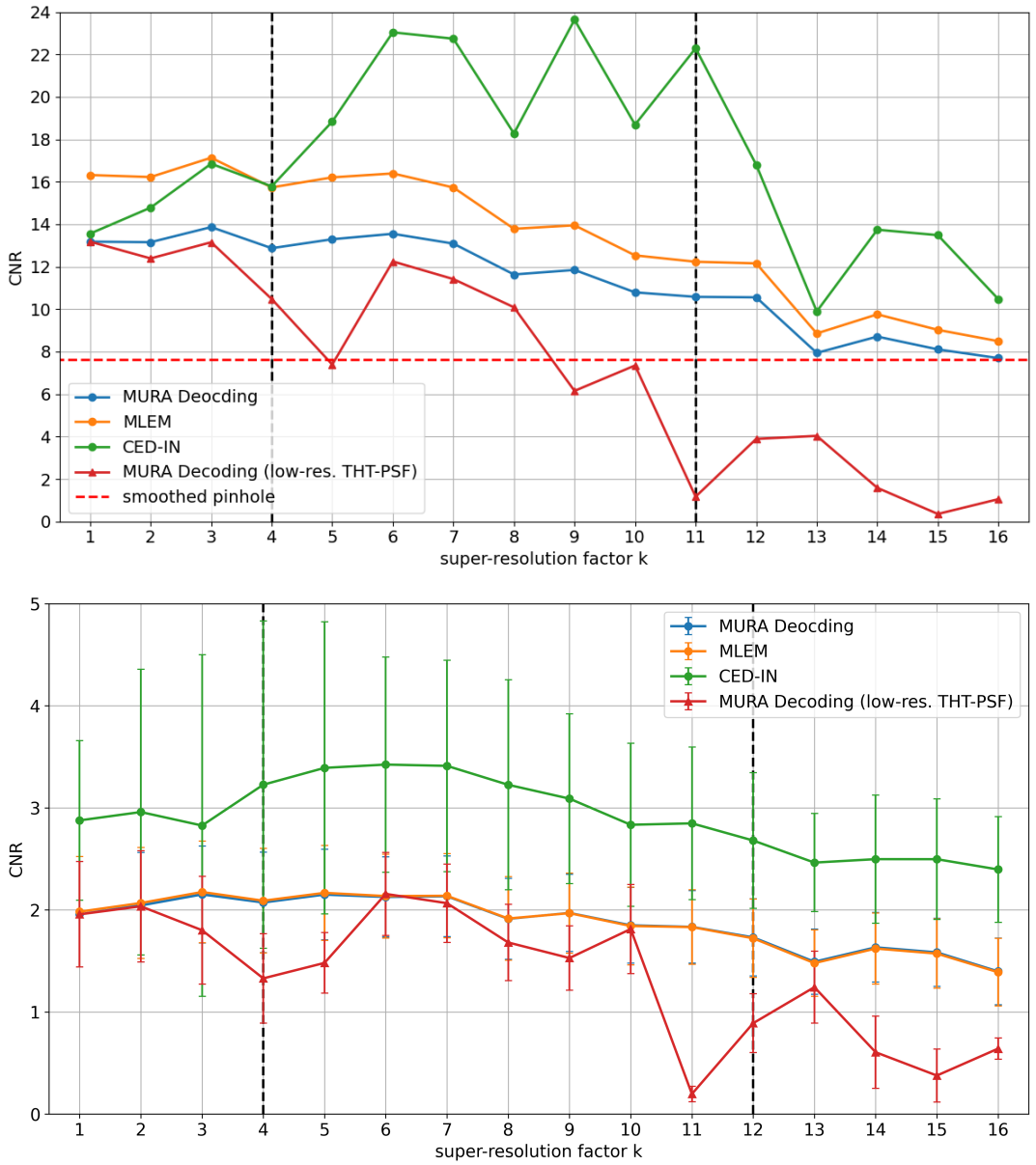


Figure 5.7 *Measured phantom data: Error bars represent the standard deviation at each data point. The vertical dotted lines denote the critical super-resolution factors* $\tilde{k}_{THT\text{-}PSF} = 4$ *and* $\tilde{k}_{measurement} = 12$. *Top: Simulated test image: The black dotted vertical lines mark the critical super-resolution factors* $\tilde{k}_{THT\text{-}PSF} = 4$ *and* $\tilde{k}_1 = 11$.*The red dotted line represents the CNR of the smoothed image captured by a pinhole collimator and serves as reference. Bottom: The contrast-to-noise ratio (CNR) for reconstructions of different reconstruction methods depending on the super-resolution factor* k.

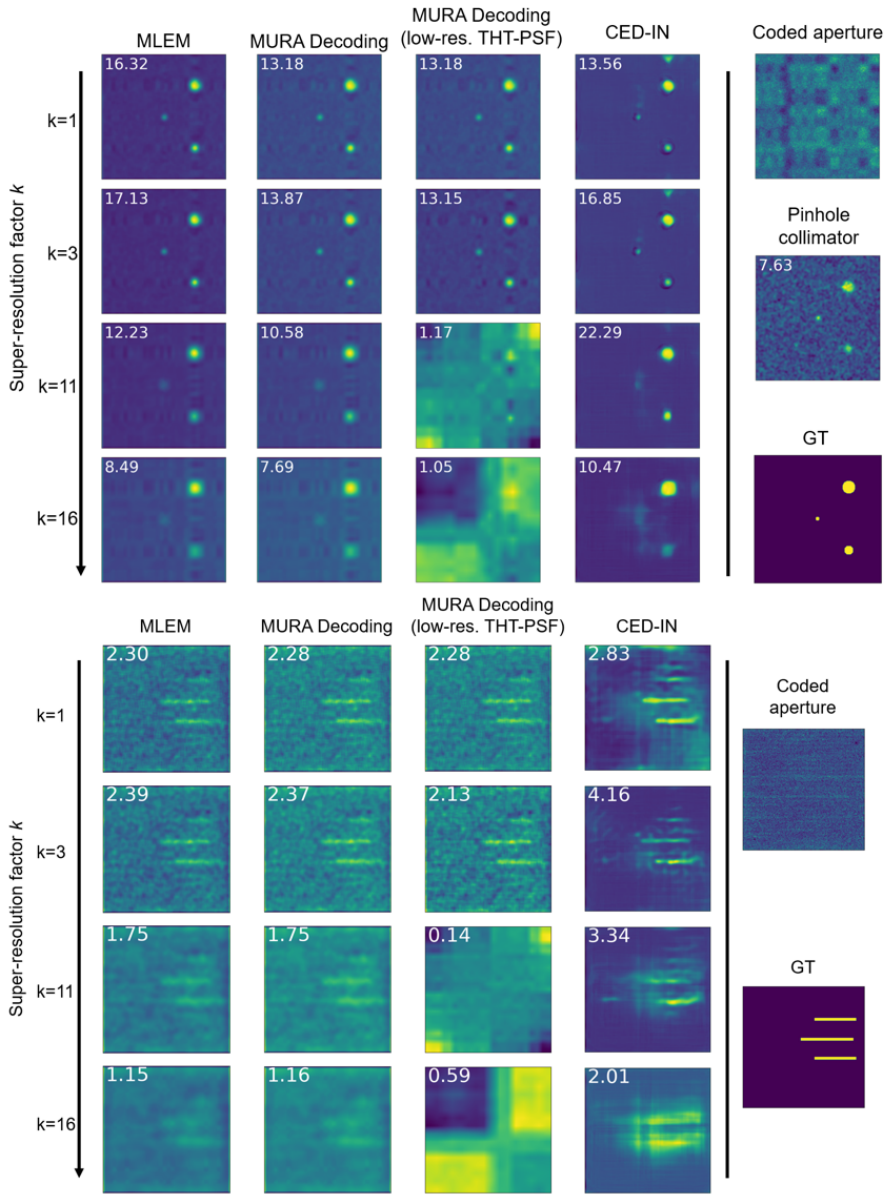


Figure 5.8 *On the right-hand side the coded aperture detector and the ground truth (GT) images are shown The CNR are printed in the top right corner of each reconstruction. Top: Exemplary reconstructions of the test image generated by MC simulation at different super-resolution factors k. For reference, the coded aperture simulation in* 256×256 *pixels, the smoothed pinhole collimator simulation in the same resolution and the ground truth is shown on the right-hand side. Bottom: super-resolution evaluated on the SRP data captured with our experimental gamma camera.*

5.4.3 Results on the measurement data

Analogously to the test image Figure 5.7 shows the CNR for the presented reconstruction method at different super-resolution factors k. However, because 120 reconstructions were analyzed the marker represents the average CNR and additional error bars represent the standard deviation for each reconstruction method and super-resolution factor k.

In general, compared to the test image, lower CNRs can be observed. Similarly, all methods except for the MURA Decoding with low-resolution THT-PSF, slightly rise for small k and then fall after approximately $k = 7$. This behavior can also be seen in the exemplary images in Figure 5.8. The maximum median CNR is reached by the CED-IN with 3.42 at super-resolution factor $k = 6$. Visually, a higher background noise is present compared to the test image and the three line sources are prominent until they start to disappear for larger k.

5.5 Discussion

5.5.1 Simulated test image

For the given setup the Nyquist-Shannon sampling theorem states, that for super-resolution factors above $k = 4$ the PSF is not sufficiently sampled anymore. If the THT-PSF is undersampled it loses its characteristic to properly function for CAI reconstruction. The simulation study of this paper shows this behavior where CNRs of MURA Decoding with low-resolution THT-PSF drop notably for $k \geq \tilde{k}_{\text{THT-PSF}}$ and the reconstruction of $k = 11$ show major artefacts rendering the three sources unrecognizably. But, when upsampling the low-resolution detector image to a high-resolution of 256×256 pixels and using a synthetic high-resolution THT-PSF for reconstructions, the CNRs drop slower, as MURA Decoding and MLEM demonstrate. For $k \leq 6$ the CNRs even stay approximately constant.

Additionally, the reconstructed images are visually closer to the expected output. For $k > \tilde{k}_1$, as predicted by the Nyquist-Shannon sampling theorem, the smallest point source cannot be reconstructed properly and starts to dissolve.

In this simulation study super-resolution with small k and upsampling by bilinear interpolation even had a positive impact on the CNR and the reconstructions look smoother.

5.5.2 Measured phantom data

Similar behavior was observed for the measured phantom data. For all reconstruction methods, the CNR does not decrease until approximately $k = 7$, even though undersampling of the THT-PSF starts at $k = 4$. However, the gain in CNR for the measured phantom data is far less for small k compared to the simulated test image.

Interestingly, the CED-IN, even though not trained on processing upsampled low-resolution images, performs better than all other reconstruction methods for super-resolution factors of $k \geq 4$. For the measured phantom data it is the best reconstruction method for all k. This indicates that the CED-IN generalized from the training domain of natural photographs to discrete sources on a dark background, even when the low-resolution input image was upsampled by bilinear interpolation. Especially the background is reconstructed more uniform compared to all other methods.

This implies that the CED-IN was taught to compress the inputted detector image into a robust representation of the image, suppressing noise and dead pixels, like in the top right corner of the coded aperture image in Figure 5.8.

The test image generated by Monte Carlo simulation shows in theory that super-resolution in CAI is possible and the simulated low-resolution detector images based on phantom data captured by an experimental gamma-camera strengthen this hypothesis. However, the question remains as to how a real low-resolution detector would affect the reconstruction. Erroneous pixels on a low-resolution detector will have a higher impact since it is supposed to capture a larger fraction of the coded aperture pattern. Another point not investigated in this paper are other types of gamma sources. Especially extended sources are known to cause problems in CAI [29].

5.6 Conclusion

The conducted simulation study indicates that super-resolution reconstruction for planar CAI is possible even if the detector is not capable of sampling the PSF with a sufficient amount of pixels. Instead, a synthetic high-resolution THT-PSF is combined with upsampling the captured low-resolution detector image by bilinear interpolation. This way, established reconstruction methods were able to reconstruct the simulated test image. However, for large super-resolution factors, the smallest point source could not be reconstructed as the Nyquist-Shannon sampling theorem predicted. Applying the same technique to simulated low-resolution detector images from data of a hot-rod phantom captured with an experimental gamma-camera strengthen these findings. For future research, though, further experiments with a more realistic undersampling detector including erroneous pixels are needed.

Bibliography

[1] T. E. Peterson and L. R. Furenlid, “SPECT detectors: The Anger Camera and beyond,” *Physics in Medicine and Biology*, vol. 56, no. 17, 2011.

[2] J. Braga, “Coded Aperture Imaging in High-energy Astrophysics,” *Publica-*

tions of the Astronomical Society of the Pacific, vol. 132, no. 1007, p. 12001, 2020.

[3] K. Amgarou, V. Paradiso, A. Patoz, F. Bonnet, J. Handley, P. Couturier, F. Becker, and N. Menaa, "A comprehensive experimental characterization of the iPIX gamma imager," *Journal of Instrumentation*, vol. 11, no. 8, 2016.

[4] S. Sun, Y. Liu, and X. Ouyang, "Near-field high-resolution coded aperture gamma-ray imaging with separable masks," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 951, no. October 2019, p. 163001, 2020.

[5] M. Tsuchimochi and K. Hayama, "Intraoperative gamma cameras for radioguided surgery: Technical characteristics, performance parameters, and clinical applications," *Physica Medica*, vol. 29, no. 2, pp. 126–138, 2013.

[6] A. K. Kogler, A. M. Polemi, S. Nair, S. Majewski, L. T. Dengel, C. L. Slingluff, B. Kross, S. J. Lee, J. E. McKisson, J. McKisson, A. G. Weisenberger, B. L. Welch, T. Wendler, P. Matthies, J. Traub, M. Witt, and M. B. Williams, "Evaluation of camera-based freehand SPECT in preoperative sentinel lymph node mapping for melanoma patients," *EJNMMI Research*, vol. 10, no. 1, 2020.

[7] P. Russo, F. Di Lillo, V. Corvino, P. M. Frallicciardi, A. Sarno, and G. Mettivier, "CdTe compact gamma camera for coded aperture imaging in radioguided surgery," *Physica Medica*, vol. 69, no. June 2019, pp. 223–232, 2020.

[8] I. Kaissas, C. Papadimitropoulos, A. Clouvas, C. Potiriadis, and C. P. Lambropoulos, "Signal to Noise Ratio optimization for extended sources with a new kind of MURA masks," *Journal of Instrumentation*, vol. 15, no. 1, 2020.

[9] J. G. Ables, "Fourier Transform Photography: A New Method for X-Ray Astronomy," *Publications of the Astronomical Society of Australia*, vol. 1, no. 4, pp. 172–173, dec 1968.

[10] R. H. Dicke, "Scatter-hole cameras for x-rays and gamma rays," *The astrophysical journal*, vol. 153, p. L101, 1968.

[11] R. Accorsi and R. C. Lanza, "Near-field artifact reduction in planar coded aperture imaging," *Applied Optics*, vol. 40, no. 26, p. 4697, sep 2001.

[12] R. F. Marcia and R. M. Willett, "Compressive Coded Aperture Superresolution Image Reconstruction," *Proc. of the IEEE Int. Conf.e on Acoustics, Speech and Signal Processing*, pp. 833–836, 2008.

[13] A. Kulow, A. G. Buzanich, U. Reinholz, F. Emmerling, S. Hampel, U. E. A. Fittschen, C. Streli, and M. Radtke, "Comparison of three reconstruction methods based on deconvolution, iterative algorithm, and neural network for X-ray fluorescence spectroscopy with coded apertures," *Journal of Analytical Atomic Spectrometry*, 2020.

[14] J. Perl, J. Shin, J. Schümann, B. Faddegon, and H. Paganetti, "TOPAS: An innovative proton Monte Carlo platform for research and clinical applications," *Medical Physics*, vol. 39, no. 11, pp. 6818–6837, 2012.

[15] J. Allison, K. Amako, J. Apostolakis, P. Arce, M. Asai, T. Aso, E. Bagli, A. Ba-

gulya, S. Banerjee, G. Barrand, B. Beck, A. Bogdanov, D. Brandt, J. Brown, H. Burkhardt, P. Canal, D. Cano-Ott, S. Chauvie, K. Cho, G. Cirrone, G. Cooperman, M. Cortès-Giraldo, G. Cosmo, G. Cuttone, G. Depaola, L. Desorgher, X. Dong, A. Dotti, V. Elvira, G. Folger, Z. Francis, A. Galoyan, L. Garnier, M. Gayer, K. Genser, V. Grichine, S. Guatelli, P. Guéye, P. Gumplinger, A. Howard, I. Hřivnàčovà, S. Hwang, S. Incerti, A. Ivanchenko, V. Ivanchenko, F. Jones, S. Jun, P. Kaitaniemi, N. Karakatsanis, M. Karamitros, M. Kelsey, A. Kimura, T. Koi, H. Kurashige, A. Lechner, S. Lee, F. Longo, M. Maire, D. Mancusi, A. Mantero, E. Mendoza, B. Morgan, K. Murakami, T. Nikitina, L. Pandola, P. Paprocki, J. Perl, I. Petrovič, M. Pia, W. Pokorski, J. Quesada, M. Raine, M. Reis, A. Ribon, A. Ristič Fira, F. Romano, G. Russo, G. Santin, T. Sasaki, D. Sawkey, J. Shin, I. Strakovsky, A. Taborda, S. Tanaka, B. Tomè, T. Toshito, H. Tran, P. Truscott, L. Urban, V. Uzhinsky, J. Verbeke, M. Verderi, B. Wendt, H. Wenzel, D. Wright, D. Wright, T. Yamashita, J. Yarba, and H. Yoshida, "Recent developments in Geant4," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 835, pp. 186–225, 2016.

[16] V. Rozhkov, G. Chelkov, I. Hernández, O. Ivanov, D. Kozhevnikov, A. Leyva, A. Perera, D. Rastorguev, P. Smolyanskiy, L. Torres, and A. Zhemchugov, "Visualization of radiotracers for SPECT imaging using a Timepix detector with a coded aperture," *Journal of Instrumentation*, vol. 15, no. 06, pp. P06 028–P06 028, 2020.

[17] T. Meißner, V. Rozhkov, J. Hesser, W. Nahm, and N. Loew, "Quantitative comparison of planar coded aperture imaging reconstruction methods," *Journal of Instrumentation*, vol. 18, no. 01, p. P01006, jan 2023.

[18] E. E. Fenimore, "Coded aperture imaging: predicted performance of uniformly redundant arrays," *Applied Optics*, vol. 17, no. 22, p. 3562, 1978.

[19] Z. Mu and Yi Hwa–Liu, "Aperture collimation correction and maximum–likelihood image reconstruction for near–field coded aperture imaging of single photon emission computerized tomography," *IEEE Trans. on Medical Imaging*, vol. 25, no. 6, pp. 701–711, jun 2006.

[20] M. J. Cieślak, K. A. Gamage, and R. Glover, "Coded-aperture imaging systems: Past, present and future development A review," *Radiation Measurements*, vol. 92, pp. 59–71, sep 2016.

[21] R. Zhang, P. Gong, X. Tang, P. Wang, C. Zhou, X. Zhu, L. Gao, D. Liang, and Z. Wang, "Reconstruction method for gamma-ray coded-aperture imaging based on convolutional neural network," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 934, pp. 41–51, aug 2019.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, Massachusetts ; London, England: The MIT Press, 2016.

[23] I. Häggström, C. R. Schmidtlein, G. Campanella, and T. J. Fuchs, "DeepPET:

A deep encoderdecoder network for directly solving the PET image reconstruction inverse problem," *Medical Image Analysis*, vol. 54, pp. 253–262, 2019.

[24] C. Belthangady and L. A. Royer, "Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction," *Nature Methods*, vol. 16, no. 12, pp. 1215–1225, dec 2019.

[25] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, "Image reconstruction by domain-transform manifold learning," *Nature*, vol. 555, no. 7697, pp. 487–492, mar 2018.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[27] R. Zhang, X. Tang, P. Gong, P. Wang, C. Zhou, X. Zhu, D. Liang, and Z. Wang, "Low-noise reconstruction method for coded-aperture gamma camera based on multi-layer perceptron," *Nuclear Engineering and Technology*, vol. 52, no. 10, pp. 2250–2261, oct 2020.

[28] J. Beyerer, F. P. León, and C. Frese, *Machine vision: Automated visual inspection: Theory, practice and applications*. Springer, 2015.

[29] Z. Mu, L. W. Dobrucki, and Y. H. Liu, "SPECT Imaging of 2-D and 3-D Distributed Sources with Near-Field Coded Aperture Collimation: Computer Simulation and Real Data Validation," *Journal of Medical and Biological Engineering*, vol. 36, no. 1, pp. 32–43, feb 2016.

Chapter 6

Deep Learning Applications to Particle Physics: A Review

Marco Rossi
Department of Physiscs
Università degli Studi di Milano, Italy
marco.rossi5@unimi.it

DOI: 10.54103/milanoup.282.c638

6.1 Abstract

Particle physics experiments like those at the LHC produce massive, high-dimensional datasets, making machine learning essential for efficient data processing and analysis. Traditional methods, based on engineered features and machine learning techniques, often discard valuable information. The rise of deep learning and accessible high-performance hardware has enabled models to learn directly from raw detector data, improving performance across various tasks. With support from labeled Monte Carlo simulations, deep learning has become increasingly prominent in particle physics. This article reviews key applications in collider physics, jet physics, tracking, fast simulation, anomaly detection, and recent advances in neutrino physics.

6.2 Introduction and motivation

Particle physics produces huge datasets. For example, the Large Hadron Collider (LHC) collects data from protons, organized in bunches colliding at ~40 MHz frequency, with $\mathcal{O}10^8$ sensors. Each collision generates a large number of particles, whose properties must be measured and stored. Gathering this enormous amount of data might give physicists enough statistics to study interesting rare events.

These facts highlight that not only the quantity of collected data is immense, but also its dimensionality. Therefore, machine learning is a set of techniques of paramount importance in this scenario, providing automation in data processing and dimensionality reduction of such information.

For years, physicists in the High-Energy Physics (HEP) domain investigated machine learning techniques like neural networks, support vector machines, genetic algorithms and predominantly boosted decision trees (BDTs) implemented in the TMVA framework [1]. This approach was based on the idea of engineering high-level low-dimensional quantities from raw detector data to be fed as multiple inputs to multivariate analysis (MVA) and provided important boosts in many data analysis tasks. However, it was clear that reducing the input dimensionality consisted in discarding a large part of potentially interesting information, leading to inherently limited algorithms. As a consequence, these tools often struggled to provide competitive performance in applications where the dimensionality gap between raw data and extracted features grew large.

Starting in 2012, the computer science community achieved important results in training big neural networks [2–4], converging to models able to provide outperforming solutions against traditional approaches. These publications set the stage for further investigation of deep learning techniques in many other research fields, including particle physics. Moreover, this explosion of research activity was helped by the recent technical improvements in hardware accelerators and their spread as consumer-grade products, granting high-quality computational power at affordable prices. In HEP, this wave mostly translated into the idea that engineered features, designed at cost of time and great intellectual effort, could have been replaced by high-dimensional low-level raw information if processed by deep enough models.

Besides producing large datasets, the particle physics field is especially suited for the proliferation of deep learning applications thanks to the availability of labeled datasets from Monte Carlo event generators. These programs aim to simulate the physics world employing probabilistic laws, accurately describing particle interactions hierarchically from the sub-atomic scales, all the way up to include the macroscopic long-range effects of physics theories. [5–9] represent modern examples of Monte Carlo event generators. The role of the artificial intelligence tools in this picture is often to grasp the probability laws of nature from sets of observations (like particle momenta and charge) and estimate the corresponding Monte Carlo truths (such as the type of a particle or even an interaction between particles in the event).

The article is dedicated to an overview of the main results in particle physics obtained with deep learning models. We split the plethora of models proposed in the literature by their sector of application. Among physics at colliders, we identify four main areas: jet physics, tracking, fast simulation and anomaly detection. Conversely, in dealing with non-collider physics, we restrict our attention to the advancements in deep learning tools for neutrino physics only, given its prominent role in the present thesis work.

Regarding physics at colliders, we develop a detailed discussion on jet physics and tracking, while mentioning the AI applications to fast simulation and anomaly detection only briefly in this section. We remark that the fast simulation of detector data is mainly achieved with the implementation of Generative Adversarial Networks (GAN) [10]: the model generates the specific detector response with a fast inference pass of the GAN generator, producing physical distributions from synthetic random numbers.

The anomaly detection applications, instead, are mainly devoted to searches of new physics, which is not described by the current theories, i.e. the Standard Model (SM) [11, 12]. Physicists acknowledge that SM is not the ultimate theory of nature and contains many problems and inconsistencies [13, 14]. Therefore, the search for evidence of the existence of Beyond Standard Model (BSM) theories is currently an active field of research. In this picture, machine learning algorithms aim to identify rare events or tensions between data and theoretical predictions that signal the presence of some new physics mechanism. In this area, model-dependent searches aim to identify new kinds of particles or interactions through classification, such as in [15–18], while model independent approaches [19, 20] design ad-hoc strategies to look for new physics with a model agnostic approach.

The remainder of the article is organized as follows. First, in Section 6.3, we describe the application of machine learning and deep learning to jet physics from a historical point of view. Then, Section 6.4 illustrates the methods developed within the neutrino oscillation research field. In particular, we categorize the reviewed approaches by the specific neutrino experiment which proposed each solution. Section 6.5 is devoted to the problem of tracking at colliders. Finally, we present our conclusions in Section 6.6.

6.3 Jet physics

In this section, we deal with deep learning applications to jet physics focusing on how the different ways of encoding the available data dramatically changed the physicists' approach to the problem: new kinds of data encoding opened the possibility to implement several models to try to solve the problems of jet tagging and pileup mitigation, two central issues in this research field.

Events at HEP colliders are associated with a hierarchical picture of subsequent particle splittings called parton showering. This mechanism consists in recursive branchings, where each particle in the event involved in the main interaction undergoes multiple subsequent splittings in a $1 \to 2$ fashion, resulting in a tree structure. At this stage, a large number of particles is created and eventually, their momenta directions are mostly focused in a collimated region around the particle initiating the shower, called a jet. These concepts are key ingredients for Monte Carlo event generators, which are tools that manage to link the predictions of physics theories with the outcomes of the measuring experiments at colliders.

Machine learning applications to jet physics mainly involve classification algorithms and include flavor tagging, jet substructure tagging, quark-gluon tagging and pileup removal. All the tagging tasks are related to the identification of the shower initiating particle from the knowledge of the properties of either the final particles representing the tree leaves of the jet or the whole tree nodes itself. Flavor tagging classifies the jet among heavy (c, b, t) or light (u, d, s) quarks, gluons or W/Z/H bosons. Jet substructure tagging, instead, discriminates between W/Z/H and t jets. Finally, quark-gluon tagging is the ability to distinguish between the two kinds of particles contributing to the main source of background in HEP events, sometimes named QCD background.

Pileup is a concept that roots in the design of accelerators machines: in order to increase the probability to produce interactions, at colliders, bunches of protons tightly packed together are smashed against each other, rather than individually. The luminosity $\mathscr{L}$ is a quantity measuring such compactness: the the higher the luminosity, the more many protons are squeezed together in the bunch and increase the expected number of collisions. As a consequence, it is likely that for each beam crossing, more than one couple of protons scatters, emitting low-energy radiation at wide angles named pileup (PU), as opposed to the interesting high-energy interaction often referred as the leading vertex (LV). Pileup is extensively studied at colliders and depends on the machine operative setup.

The data collected by the two major experiments at LHC, Atlas and CMS, estimated a number of pileup collisions per event n_{PU}~20 during the LHC run II from 2015 to 2018, while for run III (2021–2023) and the High Luminosity LHC phase (HL-LHC, from 2026), this quantity is expected to increase to n_{PU}~80 and n_{PU}~200, respectively.

Pileup interactions modify the shape of the quantities measured in the events, affecting jet properties like its overall momentum and mass, rather than jet multiplicity in the event. Being able to design automatic tools to mitigate those effects is expected to be one of the biggest data analysis challenges during the forthcoming LHC phases. These considerations justify the importance given to pileup mitigation strategies at colliders.

The next paragraphs will review the proposed techniques in the literature concerning the described two areas of jet tagging and pileup removal. Both of them try to present the advancements in the research activity as it evolved during the last decades. from the point of view of the different data representations of jet objects used as inputs of the several neural network architectures proposed.

6.3.1 Tagging

Jet tagging mainly concerns classification algorithms. Since the early ′90s, shallow artificial neural networks have been used to detect the type of jet-initiating particles. In these initial years, the predominant approach was to feed neural

networks, comprised of just a few layers, with event features tailor-made for the specific task or, sometimes, by packing jet information into small vectors of fixed size, containing the most representative characteristics of the object.

Following this idea, [21] exploited a neural network with 3 fully connected hidden layers with 6 neurons each, to process the 4−momenta of the four leading particles within a jet, to discriminate between quarks and gluons. [22], instead, implemented a neural network to distinguish between b and c jets al LEP, with the help of the Fortran77 JETNET 3.0 library [23], which represented the de facto standard for machine learning in HEP physics during those early years.

The advent of deep learning and the improvements in hardware accelerator technologies paved the way for new strategies to solve the jet tagging problem. [24] processed for the first time entire events through neural networks: their insight was to encode calorimeter information as an image. The image consisted of a two-dimensional regular grid in the pseudorapidity η and azimuthal angle ϕ coordinates, where each pixel value described the energy deposited by the particle, or, equivalently, its transverse momentum p_T.

The pixels of the image contained raw event information that could be used to compute discriminative quantities. The authors implemented a recipe to compute the Fisher linear discriminant [25] after some physics-inspired preprocessing of the images. The algorithm was tested for W boson tagging against QCD background (gluon tagged jets), providing performance improvements against the traditional discrimination method based on the N-subjettiness (τ_2/τ_1) quantity [26, 27].

Raw inputs-based neural network tools started being investigated extensively from that point onward. Examples can be found in top quark tagging tasks [28] and jet substructure classification (namely, understanding if the considered jet is due to a showering of a low-mass single particle or a massive particle decaying into multiple fast-moving lighter objects producing overlapping jets in the calorimeter, like for the $W \to qq$ process) [29]. Another application of this framework has been presented by [30], who studied the dependency of trained models on the Monte Carlo truths labels in the training datasets. The key observation pointed out that the supervised learning algorithm might bias the model predictions following the QCD approximations employed by the specific generator used to collect the dataset, rather than focusing on learning the underlying true laws of nature. The work raised the problem of the interpretability of neural networks in the jet physics research field for the first time, finding large discrepancies when testing the models on datasets produced by different generators. The authors' final assertion underlined the need to deeply understand how the input information is exploited to extract the output and what assumptions a trained architecture relies on.

The calorimeter tower representation of [24] has then proven to be a powerful representation of jets events, mainly thanks to the success of Convolutional Neural Networks (CNNs) [3]. Indeed, [31] exploited CNNs to inspect the (η, ϕ) plane deposited energy encoding of jet events. They proposed a network to identify

highly boosted W bosons against the quark-gluon QCD background. The inputs were initially cast to grayscale images (one channel only), however further developments considered also multi-channel input images. In particular, [32] proposed to build a three-channel RGB image tensor stacking information from charged and neutral particles' transverse momenta, plus the number of charged particles measured within each pixel area.

The standard calorimeter tower images were not the only image-like encoding that has been studied in the literature: an alternative strategy has been given by the Lund Jet Plane [33]. It considers kinematic variables arising while rewinding backward the Cambridge Aachen clustering algorithm [34, 35], attempting to reconstruct a de-clustering history of a jet. The output of this procedure is an ordered set of variables that characterizes a jet object and can be seen as an image tensor. According to the authors, this description should provide greater output interpretability as well as discrimination power when employed in classification tasks.

Although the image based successfully tackled the jet classification problem multiple times, CNNs rely on the assumption that pixels form a perfect grid, while it is known that actual detectors' geometry is not perfectly regular. Moreover, jet images often contain sparse features which lead to inefficient processing by convolutional kernels. Hence, different data representation strategies have been investigated. A jet object is the result of a clustering algorithm[1], which generates a list of jet constituent particles. As a consequence, it can be represented as a sequence of tracks and vertices, forming an acyclic-directed graph or, equivalently, a tree. The complication arising from adopting this encoding scheme is mostly given by the variable length size of the sequences, which cannot be handled by standard Feed Forward Neural Networks.

[37] overcame this difficulty by proposing a Recursive Neural Network architecture comprised of Long Short Term Memory (LSTMs) [38] cells, able to deal with variable-size inputs. The work takes into account other solutions involving Feed Forward Neural Networks supported by input truncation and zero padding. The authors presented a comparison of the different strategies applying them to the problem of light (u, d, s, c) versus heavy quark (b) jet flavor classification, achieving similar performance for the different models. Since then, several algorithms based on RNNs have been proposed to become part of the Atlas [39, 40] and CMS [41, 42] software stack and many more have been published to exploit variable size inputs [43, 44].

The introduction of RNNs allowed for the treatment of the jet as lists and trees of particles. However, even if some natural ordering is obtained by clustering like in the k_t−algorithm, this is just an approximation. Imposing an ordering often means

[1] A modern C++ implementation of jet definitions and clustering algorithms is given by the FASTJET 3.0 [36] library.

Table 6.1 *Summary of the proposed architectures for jet classification. The table is inspired from [51].*

	quark/gluon	W/Z	H	b/c	t
Image	[32, 52, 53]	[29, 31]	[54]		[55, 56]
Sequences	[53]			[39]	
Tree	[44]	[57, 58]			[56]
Graph			[45]		
Unordered set	[47]				
Point Cloud	[48]				[48, 56]

establishing a spatio-temporal relationship between particles to be identified as a history producing a specific final state. However, quantum mechanics principles break down the causality concepts of space and time relying on probabilistic laws. Therefore, the most natural way to represent a jet object would be to decouple from this artificial ordering and process it like an unordered set of particles described by their 4−momenta and quantum numbers. Designing an architecture with the ability to deal with unordered sets of particles would then be desirable. Graph neural networks, deep sets and point clouds networks achieve this objective.

[45] implemented a RelNet [46] to accomplish W jet tagging against QCD background: particles are regarded as graph nodes and the adjacency matrix is learned to aggregate information between nodes through a message-passing operation. [47] constrained a network architecture acting on deep sets, to build infra-red and collinear (IRC) safe observables: the information in each particle observable is then aggregated with a global permutation-invariant operation. The authors implement two different networks called EnergyFlow and ParticleFlow, which consider IRC-safe and non-IRC-safe quantities, respectively. [48] proposed to to use the EdgeConv operation [49] on the k-nearest neighbors points of each particle in a point cloud. The point cloud jet representation encodes an event as a matrix where each row represents a vector of properties associated with each particle.

Figura 6.1 shows how the neural network input representations for jet physics evolved through time. Table 6.1, instead, summarizes all the relevant applications of machine learning and deep learning to jet physics.

6.3.2 Pileup mitigation

Pileup contribution from charged particles can be removed almost completely thanks to the excellent vertex resolution at the ATLAS and CMS detectors [59–61]. These particles are identified and removed from the event with the charged-hadron subtraction (CHS) procedure [62]. The challenge comes from pileup radiation due to neutral particles, which must be taken into account with specialized algorithms. The rich literature on traditional methods can be categorized on the level of

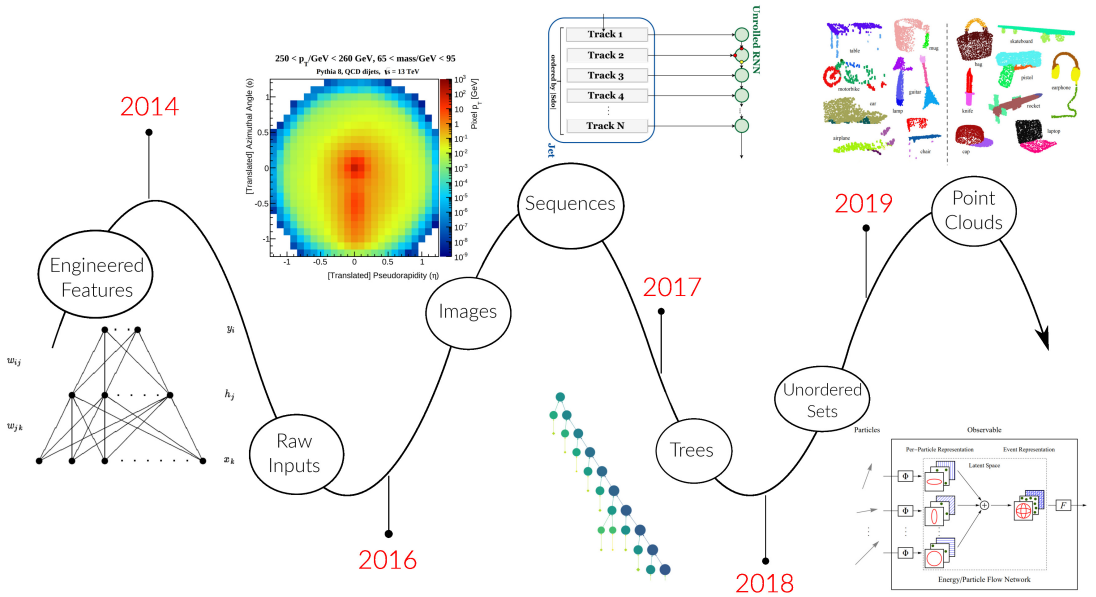


Figure 6.1 *The data structure timeline of physics jets: only engineered features were used as input to neural networks before 2014; after [24], several encoding structures have been investigated to efficiently represent jets. The descriptive pictures in the chart are taken, in order of appearance, from: [23, 31, 39, 44, 47, 50].*

detail these tools act on. A first technique, known as jet areas subtraction [63], relied on calibrating jet level information, scaling its 4−momentum by a relevant factor. However, this procedure did not manage to mitigate the pileup contribution effectively for the computation of several subjet observables.

Therefore, other algorithms have been proposed to act on the internal jet structure, namely at the subjet level. Examples of such tools are usually classified as jet constituents pre-processing, jet or event grooming, subjet corrections and constituent corrections. Grooming, in particular, progressively removes jet constituents contaminated by pileup, cutting the tree description of a jet arising from clustering algorithms through filtering [64], pruning [65,66] and trimming [67]. SoftKiller [68], instead, is a popular event-level grooming algorithm that equally divides the (η, ϕ) plane in patches of a certain area and imposes a cut-off p_T^{cut} on the transverse momentum cumulated on the patches, such that half of the patches are radiation free. This tool has been used by several works as a benchmark to test the goodness of the proposed models.

Finally, the most advanced pileup mitigating algorithms act at the deepest level, working on a particle-by-particle basis [69–71]. Among those, an excellent example is PUPPI, which evaluates a scaling factor for each particle 4−momentum in the

event, by computing a local shape variable α, which collects information about each particle neighborhood. The α distribution for charged particles, for which pileup information is known thanks to the CHS method, can be exploited to extract the scaling weight for each neutral particle. The net effect is to correct jet and subjet observables of interest for physics analysis as if the pileup effects have been switched off.

Machine learning applications for pileup removal mainly act at the particle level, since, as already discussed in this chapter, they can extract useful information from the low-level description of events. The first application of this framework was PUMML [72], a Convolutional Neural Network to inspect RGB images in the (η, ϕ) plane. The three RGB channels convey information about the transverse momenta of all neutral particles, all charged pileup particles and all charged leading vertex particles, respectively. The architecture is trained in a supervised way to output the missing p_T of the neutral leading vertex particles. Performance comparisons were presented against SoftKiller and PUPPI algorithms for the reconstruction of mass and transverse momentum distributions of the LV jets.

Other models have been proposed in subsequent years, trying to take advantage of the different technologies developed in the computer vision research field: [73] introduced PUPPIML, a network working on a graph representation of the event. After subtracting the charged pileup particles, the remaining ones are arranged in a graph where all pairs of particles closer than a fixed radius R_1 in the (η, ϕ) plane (default value is $R_1 = 0.3$) are connected by an edge. This graph is processed by several Gated Recurrent Units (GRU) [74] and outputs a binary score for each particle to discriminate between leading vertex and pileup. The authors claimed performance improvements up to ~30% of PUPPIML against PUPPI on the resolution of jet-related quantities and even higher ones with respect to SoftKiller.

PUMA [75] exploits the attention mechanism [76] to tackle the pileup mitigation task in realistic detector scenarios, corresponding to extreme setups with n_{PU}~200. The performance was tested against classical benchmarks, like CHS and PUPPI, showing large improvements in the key reconstructed jet variable distributions. The authors judged this work as an important achievement in showing the usefulness of statistically-learned algorithms during the HL-LHC phase.

Beyond the supervised algorithms presented in the paragraphs above, some alternative approaches have been proposed. [77] implemented a grooming procedure within a reinforcement learning (RL) framework: a jet is represented as a binary tree graph where each node i is described by a Lund plane derived variable $\mathcal{T}^{(i)}$, containing the state vector observed by the RL agent as well as a pointer to the the parent node and the two child ones. The algorithm concerns applying recursively a policy function π_g to all the nodes in the graph. The policy function outputs the probability to groom or not a node in the tree, which determines the action of the RL agent on the environment. The agent is trained through a smooth reward function carefully designed to optimize the resolution of kinematic

variables both at the graph and node level, such as the mass of the resulting jet or the fact that a node contributes to the wide-angle soft radiation (PU) rather than to the hard-collinear emission (LV), respectively.

A semi-supervised learning approach for Graph Neural Networks, named Graph SSL, has been investigated by [78]. The main advantage introduced by this technique is the possibility to train directly on real detector data, without the need of Monte Carlo truth labels. The algorithm is based on supervised training to learn charged particles' properties, while inference is done on neutral particles, which represents the main challenge in the identification of pileup. A careful masking procedure is required to train effectively on charged particles as if they were neutral ones. This method allows for avoiding the complex issues regarding the dependence of the models on Monte Carlo datasets and the high costs in terms of simulation time to reproduce physics processes with Monte Carlo generators. The authors benchmarked Graph SSL against PUPPI and observed performance improvements both for the accuracy in the LV-PU identification at the particle level and regarding the resolution of the reconstructed jet quantities.

6.4 Non-collider physics (neutrino)

This section presents the applications of deep learning in neutrino physics. For sake of brevity, we restrict our attention to experiments focusing on neutrino oscillations only: in fact, a large number of experiments are concentrating their efforts to improve the understanding of such mechanism. In the field of neutrino physics, deep learning is mainly investigated as a tool for event classification and automated reconstruction algorithms. The former task is well established in the physics community since the end of the 20th century as a robust strategy to select signal and reject background events, while the latter is still an open issue and many techniques are currently being inspected.

The first neural network application to neutrino event classification is given by [79] in the context of the SNO experiment. The network contains a modest $\mathcal{O}(700)$ number of trainable parameters employing a shallow feed-forward neural network with $\mathcal{O}(30)$ inputs engineered on detector hit patterns and count a single hidden layer with 20 neurons, to distinguish between four classes of neutrino interactions:

- Charge current (CC): $\nu_e + {}^2\mathrm{H} \rightarrow p + p + e^-$;
- Electron scattering (ES): $\nu_x + e^- \rightarrow \nu_x + e^-$;
- Chlorine neutral current (NC): $n + {}^{35}\mathrm{Cl} \rightarrow {}^{36}\mathrm{Cl} + \gamma$;
- Deuteron neutral current (ND): $n + {}^2\mathrm{H} \rightarrow {}^3\mathrm{H} + \gamma$.

The investigation of artificial neural networks eventually spread among neutrino physicists. The main cause of this success can be found in the detector data format: most of the detectors built for detecting neutrinos produce image-like data, which can be processed with the help of modern computer vision and

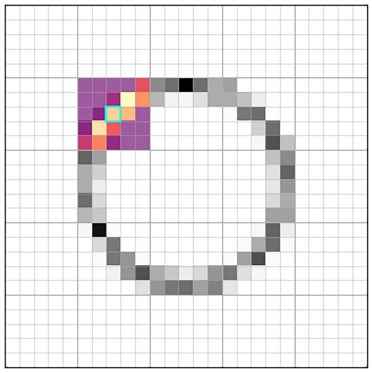
convolutional neural network techniques. As a consequence, two decades after the SNO paper, the NOνA collaboration [80] proposed to build a CNN to identify neutrino background interactions [81]. The network, named Convolutional Visual Network (CVN), is comprised of two GoogLeNet [82] separate branches inspecting (x, y) and (y, z) hit projections, respectively. The two resulting feature embeddings are concatenated and fed into a classifier to extract the desired multi-class score. It is interesting to notice that the two views are not concatenated along the channel axis like in RGB images: the authors recognize that each coordinate pixel in 2D projection would overlap unrelated features, as they do not refer to the same (x, y, z) spatial 3D point. The network is trained to compute the ν_e appearance and ν_μ disappearance rates. This work marked a milestone in the field since it became the first neural network-based analysis whose results were included in a physics publication [83].

The GoogLeNet architecture has also been exploited by [84] to search for neutrino-less double beta decay $0\nu\beta\beta$ [85] process at the NEXT experiment. In this application, as opposed to the NOνA one, three 2D projections of event images are concatenated like RGB images. The authors highlight an improvement against the traditional blob discrimination method described in [84].

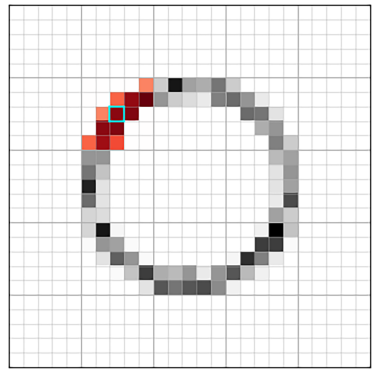
The techniques just reviewed try to process images with CNNs, achieving better performance than baseline methods. In recent years, several other articles and many experimental collaborations showed interest in developing CNN-based classifiers, showing the performance superiority of this approach compared to the traditional methods of event classification [81, 86–90].

CNNs have also proved useful in several tasks of reconstruction. First, they have been used to tackle regression problems, namely to predict the interacting neutrino energy value [91] or its direction in the frame of reference of the detector [88, 92]. Then, they helped in identifying non-empty activity regions, drawing bounding boxes around interactions to discard uninteresting parts of the input images: this technique is called Faster-Region Convolutional Neural Network (Faster-CNN). Alternatively, [93] exploit a model inspired by the U-Net architecture [94] to precisely locate track end-points and shower vertices. Finally, [95–97] implemented CNNs aiming at segmenting the input images to assign each pixel to a type of particle drifting in the detector, detecting Region Of Interest (ROI) coordinates in raw data in 2-dimensional planes and 1-dimensional channels, respectively.

Although Convolutional Neural Network models are the de-facto standard in image processing, neutrino detectors often collect data with special features that cause these techniques to be inefficient. The majority of the events recorded by these experiments contain sparse long 1-dimensional tracks with locally dense features. The result is that large portions of such images are empty, leading to a waste of computational resources when inspected with convolutional filters: those filters, indeed, transform equally both the empty spaces and signal regions. Additionally, the huge quantity of sensors in these detectors gathers information into



(a) Normal 2-dimensional convolution: single kernel transformation.



(b) Sparse 2-dimensional convolution: single kernel transformation.

Figure 6.2 *The cyan pixel highlights the current convolution pixel. The normal convolution kernel operates on all pixels within the kernel window, while the sparse one acts on non-zero neighboring pixels only.*

high-resolution images with $\mathcal{O}(10^6)$ of pixels, that barely fit the memory constraints of modern hardware accelerating devices.

During the last few years, then, the neutrino community has dedicated a great effort to design better encodings and experiment novel techniques to analyze such data. In this picture, Sparse Convolutional Neural Networks (Sparse CNNs) [98] and Graph Neural Networks (GNNs) [99] have been investigated. Two architectures based on Sparse CNNs, acting with convolutional filters on non-zero pixels only, have been implemented by [100, 101]. The operation, depicted by Figura 6.2b, allows to store the event data in an efficient sparse format and dramatically decrease the number of operations required by each convolutional layer forward pass.

On the other hand, GNNs provided performance improvements in processing data from detectors with irregular geometry like IceCube [102] and JUNO [103]. Besides this success, even if the data graph encoding is not always a natural choice when dealing with either image or point cloud data, several works [104–106] showed good results implementing these architectures for event classification and other reconstruction tasks.

The use of GNNs is subject to two major issues. First, there is no standard choice of encoding neutrino data into a graph. Luckily, the sparsity of neutrino images allows identifying detector hits as graph nodes, resulting in graphs of manageable sizes (usually up to a few thousand nodes). Node connectivity, instead, is use-case dependent. The majority of the authors reviewed in this paragraph use similar approaches with small modifications: they rely on some distance metric, computed between each pair of nodes i and j in the graph, and a pre-defined cut-off value d_{cut} above which no edge between the corresponding nodes is drawn. Alternatively,

they propose to weight each edge with a normalized version of the distance metric value itself, such that distant nodes have a suppressed information flow in the network. The second problem related to this approach is the additional overhead represented by the graph construction operation: this is often done through dedicated algorithms, like in [103, 104], and must be repeated for all events inevitably increasing the pre-processing wall-time.

In table 6.2 we collect the main deep learning applications to neutrino oscillation experiments found in the literature. The table groups the works published by several collaborations into five task categories:

- event classification, which encompasses event topology, interaction classification and background rejection;
- regression, grouping neutrino energy reconstruction and neutrino direction reconstruction;
- object detection, that collects interaction localization (vertex reconstruction, bounding box drawing around pixel activity), track end-point localization and shower starting-point localization;
- graph operations, that include background rejection (graph classification), clustering (node classification), 3D reconstruction (graph cleaning through node classification) and primary particle classification (edge classification);
- segmentation, receiving contribution from pixel-level particle identification, instance segmentation and region of interest (ROI) finding.

6.5 Tracking

Tracking is a central process of reconstruction at colliders, it consists in grouping detector hits within an event produced by a charged particle interacting in the inner detector region and moving inside a static magnetic field. The traditional approach is based on four different phases: hit clustering, track seed finding, track building and track fitting. The present discussion gives a brief overview of the traditional method employed to solve the tracking problem and it is inspired by specialized reviews on tracking strategies at LHC, [113–115].

The tracking process consists in sequentially reducing with clustering algorithms the number of data from $\mathcal{O}(10^8)$ detector readout channels, to $\mathcal{O}(10^4)$ hits containing energy depositions and finally to $\mathcal{O}(10^3)$ tracks per event. The hierarchical approach starts with hit clustering, which consists in finding the 3−dimensional locations of hits and the corresponding deposited energies from the pixel-level raw data readouts.

After this first stage, the two most computationally expensive steps take place. First, the hits in the inner detector are processed to identify triplets, which consist of the minimum number of points to estimate two important track parameters, namely the curvature and the perigee with respect to the center of the interaction

Table 6.2 *Review of the deep learning for neutrino physics publications. The first column identifies which detector the publication focuses on. PilarNet [107] is a generalpurpose open dataset for LArTPCs data. Note: [100] was published before the PilarNet [107] dataset but deals with similar data and objectives. The citations are colorcoded based on the neural network type implemented in the relative work: FFNNs, CNNs, GNNs, Hexagonal CNNs, Sparse CNNs, Quantum CNNs.*

Detector	Event classification	Regression	Object detection	Graph	Segmentation
SNO	[79]	–	–	–	–
NEXT	[108]	–	–	–	–
Daya Bay	[86]	–	–	–	–
NOνA	[81]	[91]	–	–	–
MicroBooNE	[87, 90]	–	[87]	–	[95] [109]
KM3NeT/ORCA	[88]	[88]	–	–	–
DUNE/pDUNE	[89] [110]	[92]	–	[106]	[96]
JUNO	[111]	[103]	[103]	[103]	–
SuperFGD (T2K)	–	–	–	[105]	–
IceCube	[102]	[112]	–	[102]	–
ArgoNeuT	–	–	–	–	[97]
PilarNet	–	–	[93]	[104]	[100, 101]

region. The three hits in each triplet form a seed for the final track. Therefore, this step fixes the final track multiplicity.

Second, once the seeds are selected, the proper track construction process starts: the trajectory is sequentially extrapolated from the triplet from the inner to the outer layers of the detector. Many pattern recognition techniques have been designed to tackle this problem, ranging from global methods, such as conformal mapping and Hough transform [116], to local ones, like the track road methods. However, the most efficient algorithm in use is the Kalman filter [117–119]. A more refined version of the original algorithm, the Combinatorial Kalman Filter [120], is leveraged to build tracks from seeds, including the possibility to keep track of branching when multiple candidate points are identified within the same layer and eventually, discard the fake tracks with high efficiency.

The final stage of the tracking problem, namely track fitting, requires estimating the track parameters for each reconstructed trajectory. These include the location

of the interaction vertex, the direction of the track along with its curvature and the momentum associated with the interacting particle. Moreover, tests to remove outliers that do not belong to the track are performed in this final phase to further refine the output. This technique achieves almost perfect performance, meaning that the investigation of new methods is devoted to optimizing the existing software implementation and trying to reduce CPU usage time.

However, the next generation High Luminosity LHC (HL-LHC) phase [121], starting from 2026, will see an increase in the current luminosity setup of the Large Hadron Collider by a factor of 10, putting these low-level reconstruction tools under enormous stress. It is expected, in this collider configuration, a great improvement in the hit detector occupancy and the particle tracking software should be able to manage charged particles at a rate of $\mathcal{O}(50\,\text{MHz})$. The traditional approach does not scale at such regimes. A naive solution would be to limit the reconstruction to detector regions around specific calorimetry depositions compatible with rare signatures like leptons or jets with high p_T. However, this approach will completely neglect other phenomena that might hide in discarded regions, like low p_T ones. Hence, alternative methods are currently under investigation.

In this context, the HEP.TrkX project [122] aims to study deep learning solutions to the particle tracking issue. The main outcome has been a model [123] combining CNNs and Recurrent Neural Networks (RNNs), mainly employing Long Short Term Memories (LSTMs) cells, to reconstruct tracks within a simplified detector simulation. The generated data involve straight-line tracks and neglect all other kinds of physical complexities, such as track curvature, material effects and detection inefficiencies. The model is trained to solve two tasks in particular: a 2−dimensional single-track reconstruction starting from seeded hits and an end-to-end estimation of the track parameters without any seeding.

Detector data are projected onto two axes representing the detector layer and the channel within each specific layer. The tool opens for the possibility of encoding irregular layer geometries of varying size with two strategies: either zero padding the input to retrieve a regular rectangular grid, or through an autoencoder-like architecture that embeds each layer input into a fixed-size vector representation with the help of a dense network, followed at the end of the pipeline by another fully connected layer that projects back the output into the original layer dimensionality. The data encoding based on bi-dimensional images has also been exploited by [124], the key idea is again to model the recursive track-following approach of the Kalman filter through an LSTM, showing promising results in a semi-realistic detector simulation.

Other approaches based on Graph Neural Networks (GNNs) have been introduced by [125] driven by the observation that the image-like representation of the data would not be able to manage realistic use cases matching the HL-LHC conditions. Indeed, the collider and detector updates will provide high-dimensional and sparse data due to the increased number of detector layers built with irregular

geometries, which would probably cause inefficiencies in the standard approaches with CNNs. The authors advocate the investigation of methods acting on the space-point representation of data, instead, involving variable amounts of hits per event and exploiting the full detector resolution.

In 2018, the TrackML competition [126] took off within the HEP community with the intent of finding the best candidate for the future particle tracking algorithm. The desired feature of such a tool would be to achieve the best performance score across several metrics reflecting the need to target high reconstruction efficiencies with the fastest algorithm in terms of inference time. Event examples from the TrackML dataset highlight a large number of tracks to be reconstructed and the complex detector design.

Following this competition, HEP.TrkX evolved into the Exa.TrkX project [127] which investigated a wide variety of models to solve the task, mainly through GNNs [128, 129]. The project finally published an article [130] summarizing the GNN pipeline on the TrackML dataset, towards a first validation on ATLAS and CMS real detector data. The potential of GNNs has also been exploited on implementations for specific hardware acceleration, mainly provided by Field Programmable Gate Arrays (FPGAs) [131].

6.6 Conclusion

In this article, we reviewed the most important contributions to the literature concerning the applications of deep learning in the high-energy and neutrino physics fields. We presented an overview of different sectors of active research, including both collider and non-collider physics. This work aims to show the huge success of deep learning models when applied to problems in the physics domain. Although the first implementations of machine learning techniques date back to some decades ago, the popularity of such methods is constantly increasing. The main challenge for the next decade is to demonstrate that such models provide a robust generalization to unseen data, so that they can become the standard approach to solve the complex problems we introduced in this work and many others. The interpretability of their predictions is going to be another crucial point in the affirmation of these solutions and will certainly be a hot topic in the research of the following years.

Bibliography

[1] H. Voss, A. Höcker, J. Stelzer, and F. Tegenfeldt, “TMVA, the Toolkit for Multivariate Data Analysis with ROOT,” in *Proc.of XI Int. Workshop on Advanced Computing and Analysis Techniques in Physics Research*, vol. 050, 2009.

[2] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with

deep convolutional neural networks," in *Proc.of the 25th Int. Conf. on Neural Information Processing Systems*, 2012, pp. 1097–1105.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "deep learning," *Nature*, vol. 521, pp. 436–444, may 2015.

[4] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[5] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis *et al.*, "Geant4a simulation toolkit," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 506, no. 3, pp. 250–303, 2003.

[6] T. Sjöstrand, S. Mrenna, and P. Skands, "PYTHIA 6.4 physics and manual," *Journal of High Energy Physics*, vol. 2006, no. 05, pp. 026–026, may 2006.

[7] M. Bähr, S. Gieseke, M. Gigg, D. Grellscheid, K. Hamilton, Latunde-Da., S. Plätzer, P. Richardson, M. Seymour, A. Sherstnev, and B. Webber, "Herwig++ physics and manual," *The European Physical Journal C*, vol. 58, pp. 639–707, dec 2008.

[8] T. Gleisberg, S. Höche, F. Krauss, M. Schönherr, S. Schumann, F. Siegert, and J. Winter, "Event generation with SHERPA 1.1," *Journal of High Energy Physics*, vol. 2009, no. 02, pp. 007–007, feb 2009.

[9] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.-S. Shao, T. Stelzer, P. Torrielli, and M. Zaro, "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations," *Journal of High Energy Physics*, vol. 2014, jul 2014.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014.

[11] S. F. Novaes, "Standard model: An Introduction," in *10th Jorge Andre Swieca Summer School: Particle and Fields*, 1 1999, pp. 5–102.

[12] "Standard Model." [Online]. Available: https://www.physik.uzh.ch/groups/serra/StandardModel.html

[13] J. M. Butterworth, "The standard model: how far can it go and how can we tell?" *Philosophical Trans. of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2075, 2016.

[14] E. R. Paudel, "Problems of standard model, review," *BMC Journal of Scientific Research*, vol. 4, no. 1, pp. 65–73, Dec. 2021.

[15] P. Baldi, P. Sadowski, and D. Whiteson, "Searching for Exotic Particles in High-Energy Physics with Deep Learning," *Nature Commun.*, vol. 5, 2014.

[16] O. Cerri, T. Q. Nguyen, M. Pierini, M. Spiropulu, and J. R. Vlimant, "Variational Autoencoders for New Physics Mining at the Large Hadron Collider,"

JHEP, vol. 05, p. 036, 2019.

[17] E. Bernreuther, T. Finke, F. Kahlhoefer, M. Krämer, and A. Mück, "Casting a graph net to catch dark showers," *SciPost Phys.*, vol. 10, no. 2, p. 046, 2021.

[18] D. Cogollo, F. F. Freitas, C. A. de S. Pires, Y. M. Oviedo-Torres, and P. Vasconcelos, "Deep learning analysis of the inverse seesaw in a 3-3-1 model at the LHC," *Phys. Lett. B*, vol. 811, p. 135931, 2020.

[19] R. T. D'Agnolo and A. Wulzer, "Learning New Physics from a Machine," *Phys. Rev. D*, vol. 99, no. 1, p. 015014, 2019.

[20] R. T. d'Agnolo, G. Grosso, M. Pierini, A. Wulzer, and M. Zanetti, "Learning new physics from an imperfect machine," *Eur. Phys. J. C*, vol. 82, no. 3, p. 275, 2022.

[21] L. Lönnblad, C. Peterson, and T. Rögnvaldsson, "Finding gluon jets with a neural trigger," *Phys. Rev. Lett.*, vol. 65, pp. 1321–1324, Sep 1990.

[22] T. Behnke and D. Charlton, "Electroweak measurements using heavy quarks at lep," *Physica Scripta*, vol. 52, no. 2, pp. 133–157, aug 1995.

[23] C. Peterson, T. Rögnvaldsson, and L. Lönnblad, "JETNET 3.0A versatile artificial neural network package," *Computer Physics Communications*, vol. 81, no. 1, pp. 185–220, 1994.

[24] J. Cogan, M. Kagan, E. Strauss, and A. Schwarztman, "Jet-images: Computer vision inspired techniques for jet tagging," *JHEP*, vol. 02, p. 118, 2015.

[25] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[26] J. Thaler and K. Van Tilburg, "Identifying Boosted Objects with N-subjettiness," *JHEP*, vol. 03, p. 015, 2011.

[27] ——, "Maximizing Boosted Top Identification by Minimizing N-subjettiness," *JHEP*, vol. 02, p. 093, 2012.

[28] L. G. Almeida, M. Backović, M. Cliche, S. J. Lee, and M. Perelstein, "Playing Tag with ANN: Boosted Top Identification with Pattern Recognition," *JHEP*, vol. 07, p. 086, 2015.

[29] P. Baldi, K. Bauer, C. Eng, P. Sadowski, and D. Whiteson, "Jet Substructure Classification in High-Energy Physics with Deep Neural Networks," *Phys. Rev. D*, vol. 93, no. 9, 2016.

[30] J. Barnard, E. N. Dawe, M. J. Dolan, and N. Rajcic, "Parton Shower Uncertainties in Jet Substructure Analyses with Deep Neural Networks," *Phys. Rev. D*, vol. 95, no. 1, 2017.

[31] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, and A. Schwartzman, "Jet-images — deep learning edition," *JHEP*, vol. 07, p. 069, 2016.

[32] P. T. Komiske, E. M. Metodiev, and M. D. Schwartz, "Deep learning in color: towards automated quark/gluon jet discrimination," *JHEP*, vol. 01, p. 110, 2017.

[33] F. A. Dreyer, G. P. Salam, and G. Soyez, "The Lund Jet Plane," *JHEP*, vol. 12, p. 064, 2018.

[34] Y. L. Dokshitzer, G. D. Leder, S. Moretti, and B. R. Webber, "Better jet clustering algorithms," *JHEP*, vol. 08, 1997.

[35] M. Wobisch and T. Wengler, "Hadronization corrections to jet cross-sections in deep inelastic scattering," in *Proc. of the Workshop on Monte Carlo Generators for HERA Physics*, 4 1998, pp. 270–279.

[36] M. Cacciari, G. P. Salam, and G. Soyez, "FastJet User Manual," *Eur. Phys. J. C*, vol. 72, p. 1896, 2012.

[37] D. Guest, J. Collado, P. Baldi, S. C. Hsu, G. Urban, and D. Whiteson, "Jet flavor classification in high-energy physics with deep neural networks," *Phys. Rev. D*, vol. 94, Dec 2016.

[38] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.

[39] ATLAS Collaboration, "Identification of Jets Containing *b*-Hadrons with Recurrent Neural Networks at the ATLAS Experiment," 3 2017. [Online]. Available: https://inspirehep.net/literature/1795312

[40] ——, "Optimisation and performance studies of the ATLAS *b*-tagging algorithms for the 2017-18 LHC run," 7 2017. [Online]. Available: https://inspirehep.net/literature/1795306

[41] CMS Collaboration, "Heavy flavor identification at CMS with deep neural networks," 2017. [Online]. Available: https://cds.cern.ch/record/2255736

[42] ——, "Performance of heavy flavour identification algorithms in proton-proton collisions at 13 TeV at the CMS experiment," 2017. [Online]. Available: https://cds.cern.ch/record/2263801

[43] S. Egan, W. Fedorko, A. Lister, J. Pearkes, and C. Gay, "Long short-term memory (lstm) networks with jet constituents for boosted top tagging at the lhc," 2017. [Online]. Available: https://arxiv.org/abs/1711.09059

[44] T. Cheng, "Recursive Neural Networks in Quark/Gluon Tagging," *Comput. Softw. Big Sci.*, vol. 2, no. 1, p. 3, 2018.

[45] I. Henrion, J. Brehmer, J. Bruna, K. Cho, K. Cranmer, G. Louppe, and G. Rochette, "Neural message passing for jet physics," in *Workshop on Deep Learning for Physical Sciences of the 31st Annual Conf. on Neural Information Processing Systems (NIPS17)*, 2017.

[46] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, "A simple neural network module for relational reasoning," 2017. [Online]. Available: https://arxiv.org/abs/1706.01427

[47] P. T. Komiske, E. M. Metodiev, and J. Thaler, "Energy Flow Networks: Deep Sets for Particle Jets," *JHEP*, vol. 01, p. 121, 2019.

[48] H. Qu and L. Gouskos, "ParticleNet: Jet Tagging via Particle Clouds," *Phys. Rev. D*, vol. 101, no. 5, 2020.

[49] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, oct 2019.

[50] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2016. [Online]. Available: https://arxiv.org/abs/1612.00593

[51] A. J. Larkoski, I. Moult, and B. Nachman, "Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning," *Phys. Rept.*, vol. 841, pp. 1–63, 2020.

[52] ATLAS Collaboration, "Quark versus Gluon Jet Tagging Using Jet Images with the ATLAS Detector," 7 2017. [Online]. Available: https://inspirehep.net/literature/1795319

[53] CMS Collaboration, "New Developments for Jet Substructure Reconstruction in CMS," 2017. [Online]. Available: https://cds.cern.ch/record/2275226

[54] J. Lin, M. Freytsis, I. Moult, and B. Nachman, "Boosting $H \to b\bar{b}$ with Machine Learning," *JHEP*, vol. 10, p. 101, 2018.

[55] G. Kasieczka, T. Plehn, M. Russell, and T. Schell, "Deep-learning Top Taggers or The End of QCD?" *JHEP*, vol. 05, p. 006, 2017.

[56] A. Butter *et al.*, "The Machine Learning landscape of top taggers," *SciPost Phys.*, vol. 7, p. 014, 2019.

[57] G. Louppe, K. Cho, C. Becot, and K. Cranmer, "QCD-Aware Recursive Neural Networks for Jet Physics," *JHEP*, vol. 01, p. 057, 2019.

[58] A. Andreassen, I. Feige, C. Frye, and M. D. Schwartz, "Junipr: a framework for unsupervised machine learning in particle physics," *Eur. Phys. J. C*, vol. 79, no. 2, p. 102, 2019.

[59] S. Chatrchyan *et al.*, "Description and performance of track and primary-vertex reconstruction with the CMS tracker," *JINST*, vol. 9, no. 10, p. P10009, 2014.

[60] ATLAS Collaboration, "Characterization of interaction-point beam parameters using the pp event-vertex distribution reconstructed in the atlas detector at the lhc," 5 2010. [Online]. Available: https://inspirehep.net/literature/1203962

[61] ——, "Performance of primary vertex reconstruction in proton-proton collisions at $\sqrt{s}$ =7 tev in the atlas experiment," 7 2010. [Online]. Available: https://inspirehep.net/literature/1204019

[62] CMS Collaboration, "Pileup Removal Algorithms," 2014. [Online]. Available: https://inspirehep.net/literature/1311934

[63] M. Cacciari and G. P. Salam, "Pileup subtraction using jet areas," *Phys. Lett. B*, vol. 659, pp. 119–126, 2008.

[64] J. M. Butterworth, A. R. Davison, M. Rubin, and G. P. Salam, "Jet substructure as a new higgs-search channel at the large hadron collider," *Phys. Rev. Lett.*, vol. 100, p. 242001, Jun 2008.

[65] S. D. Ellis, C. K. Vermilion, and J. R. Walsh, "Techniques for improved heavy particle searches with jet substructure," *Phys. Rev. D*, vol. 80, p. 051501, Sep

2009.

[66] ——, "Recombination algorithms and jet substructure: Pruning as a tool for heavy particle searches," *Phys. Rev. D*, vol. 81, p. 094023, May 2010.

[67] D. Krohn, J. Thaler, and L. T. Wang, "Jet Trimming," *JHEP*, vol. 02, p. 084, 2010.

[68] M. Cacciari, G. P. Salam, and G. Soyez, "SoftKiller, a particle-level pileup removal method," *Eur. Phys. J. C*, vol. 75, no. 2, p. 59, 2015.

[69] D. Bertolini, P. Harris, M. Low, and N. Tran, "Pileup Per Particle Identification," *JHEP*, vol. 10, p. 059, 2014.

[70] P. Berta, M. Spousta, D. W. Miller, and R. Leitner, "Particle-level pileup subtraction for jets and jet shapes," *JHEP*, vol. 06, p. 092, 2014.

[71] ATLAS Collaboration, "Constituent-level pile-up mitigation techniques in ATLAS," 8 2017. [Online]. Available: https://inspirehep.net/literature/1620091

[72] P. T. Komiske, E. M. Metodiev, B. Nachman, and M. D. Schwartz, "Pileup Mitigation with Machine Learning (PUMML)," *JHEP*, vol. 12, p. 051, 2017.

[73] J. Arjona Martínez, O. Cerri, M. Pierini, M. Spiropulu, and J. R. Vlimant, "Pileup mitigation at the Large Hadron Collider with graph neural networks," *Eur. Phys. J. Plus*, vol. 134, no. 7, p. 333, 2019.

[74] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014. [Online]. Available: https://arxiv.org/abs/1409.1259

[75] B. Maier, S. M. Narayanan, G. de Castro, M. Goncharov, C. Paus, and M. Schott, "Pile-up mitigation using attention," *Mach. Learn. Sci. Tech.*, vol. 3, no. 2, p. 025012, 2022.

[76] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[77] S. Carrazza and F. A. Dreyer, "Jet Grooming through Reinforcement Learning," *J. Phys. Conf. Ser.*, vol. 1525, p. 012111, 2020.

[78] T. Li, S. Liu, Y. Feng, G. Paspalaki, N. Tran, M. Liu, and P. Li, "Semi-supervised Graph Neural Networks for Pileup Noise Removal," 3 2022. [Online]. Available: https://arxiv.org/abs/2203.15823

[79] S. J. Brice, "The results of a neural network statistical event class analysis," 1996. [Online]. Available: https://sno.phy.queensu.ca/str/SNO-STR-96-001.pdf

[80] D. S. Ayres *et al.*, "The NOvA Technical Design Report," 10 2007. [Online]. Available: https://inspirehep.net/files/1e897a237c85bae0087a7f644e9ad832

[81] A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M. D. Messier, E. Niner, G. Pa-

wloski, F. Psihas, A. Sousa, and P. Vahle, "A Convolutional Neural Network Neutrino Event Classifier," *JINST*, vol. 11, no. 09, p. P09001, 2016.

[82] C. Szegedy, L. Wei, J. Yangqing *et al.*, "Going deeper with convolutions," in *2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[83] P. Adamson, L. Aliaga, D. Ambrose *et al.*, "Constraints on oscillation parameters from ν_e appearance and ν_μ disappearance in nova," *Phys. Rev. Lett.*, vol. 118, p. 231801, Jun 2017.

[84] J. Renner *et al.*, "Background rejection in NEXT using deep neural networks," *JINST*, vol. 12, no. 01, p. T01004, 2017.

[85] J. Schechter and J. W. F. Valle, "Neutrinoless Double beta Decay in SU(2) x U(1) Theories," *Phys. Rev. D*, vol. 25, p. 2951, 1982.

[86] E. Racah, S. Ko, P. Sadowski, W. Bhimji, C. Tull, S. Oh, P. Baldi, and Prabhat, "Revealing Fundamental Physics from the Daya Bay Neutrino Experiment using Deep Neural Networks," 1 2016. [Online]. Available: https://arxiv.org/abs/1601.07621

[87] R. Acciarri *et al.*, "Convolutional Neural Networks Applied to Neutrino Events in a Liquid Argon Time Projection Chamber," *JINST*, vol. 12, no. 03, p. P03011, 2017.

[88] S. Aiello *et al.*, "Event reconstruction for KM3NeT/ORCA using convolutional neural networks," *JINST*, vol. 15, no. 10, p. P10005, 2020.

[89] B. Abi *et al.*, "Neutrino interaction classification with a convolutional neural network in the DUNE far detector," *Phys. Rev. D*, vol. 102, no. 9, p. 092003, 2020.

[90] P. Abratenko *et al.*, "Convolutional neural network for multiple particle identification in the MicroBooNE liquid argon time projection chamber," *Phys. Rev. D*, vol. 103, no. 9, p. 092003, 2021.

[91] L. Hertel, L. Li, P. Baldi, and J. Bian, "Convolutional neural networks for electron neutrino and electron shower energy reconstruction in the nova detectors," in *Proc. of the Workshop on Deep Learning for Physical Sciences of the 31st Annual Conf. on Neural Information Processing Systems*, 2017.

[92] J. Liu, J. Ott, J. Collado, B. Jargowsky, W. Wu, J. Bian, and P. Baldi, "Deep-Learning-Based Kinematic Reconstruction for DUNE," 12 2020. [Online]. Available: https://arxiv.org/abs/2012.06181

[93] L. Dominé, P. C. de Soux, F. Drielsma, D. H. Koh, R. Itay, Q. Lin, K. Terao, K. V. Tsang, and T. L. Usher, "Point proposal network for reconstructing 3D particle endpoints with subpixel precision in liquid argon time projection chambers," *Phys. Rev. D*, vol. 104, no. 3, p. 032004, 2021.

[94] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[95] C. Adams *et al.*, "Deep neural network for pixel-level electromagnetic particle identification in the MicroBooNE liquid argon time projection

chamber," *Phys. Rev. D*, vol. 99, no. 9, 2019.

[96] H. Yu *et al.*, "Augmented signal processing in Liquid Argon Time Projection Chambers with a deep neural network," *JINST*, vol. 16, no. 01, p. P01036, 2021.

[97] R. Acciarri *et al.*, "A deep-learning based raw waveform region-of-interest finder for the liquid argon time projection chamber," *JINST*, vol. 17, no. 01, p. P01018, 2022.

[98] B. Graham, M. Engelcke, and L. van der Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 9224–9232.

[99] F. Scarselli, M. Gori *et al.*, "The graph neural network model," *IEEE Trans. on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[100] L. Dominé and K. Terao, "Scalable deep convolutional neural networks for sparse, locally dense liquid argon time projection chamber data," *Phys. Rev. D*, vol. 102, no. 1, p. 012005, 2020.

[101] D. H. Koh, P. Côte De Soux, L. Dominé, F. Drielsma, R. Itay, Q. Lin, K. Terao, K. V. Tsang, and T. L. Usher, "Scalable, Proposal-free Instance Segmentation Network for 3D Pixel Clustering and Particle Trajectory Reconstruction in Liquid Argon Time Projection Chambers," 7 2020. [Online]. Available: https://arxiv.org/abs/2007.03083

[102] N. Choma, F. Monti, L. Gerhardt *et al.*, "Graph neural networks for icecube signal classification," in *Proc. of the IEEE Int. Conf. on Machine Learning and Applications*, 2018, pp. 386–391.

[103] Z. Qian *et al.*, "Vertex and energy reconstruction in JUNO with machine learning methods," *Nucl. Instrum. Meth. A*, vol. 1010, p. 165527, 2021.

[104] F. Drielsma, Q. Lin, P. C. de Soux, L. Dominé, R. Itay, D. H. Koh, B. J. Nelson, K. Terao, K. V. Tsang, and T. L. Usher, "Clustering of electromagnetic showers and particle interactions with graph neural networks in liquid argon time projection chambers," *Phys. Rev. D*, vol. 104, no. 7, p. 072004, 2021.

[105] S. Alonso-Monsalve, D. Douqa, C. Jesús-Valls, T. Lux, S. Pina-Otey, F. Sánchez, D. Sgalaberna, and L. H. Whitehead, "Graph neural network for 3D classification of ambiguities and optical crosstalk in scintillator-based neutrino detectors," *Phys. Rev. D*, vol. 103, no. 3, p. 032005, 2021.

[106] J. Hewes *et al.*, "Graph Neural Network for Object Reconstruction in Liquid Argon Time Projection Chambers," *Proc. of the EPJ Web Conf.*, vol. 251, p. 03054, 2021.

[107] C. Adams, K. Terao, and T. Wongjirad, "PILArNet: Public Dataset for Particle Imaging Liquid Argon Detectors in High Energy Physics," 6 2020. [Online]. Available: https://arxiv.org/abs/2006.01993

[108] J. Martín-Albo *et al.*, "Sensitivity of NEXT-100 to Neutrinoless Double Beta Decay," *JHEP*, vol. 05, p. 159, 2016.

[109] P. Abratenko *et al.*, "Semantic segmentation with a sparse convolutional

neural network for event reconstruction in MicroBooNE," *Phys. Rev. D*, vol. 103, no. 5, p. 052012, 2021.

[110] S. Y.-C. Chen, T.-C. Wei, C. Zhang, H. Yu, and S. Yoo, "Quantum convolutional neural networks for high energy physics data analysis," *Phys. Rev. Res.*, vol. 4, no. 1, p. 013231, 2022.

[111] B. Clerbaux, M. C. Molla, P. A. Petitjean, Y. Xu, and Y. Yang, "Study of Using Machine Learning for Level 1 Trigger Decision in JUNO Experiment," *IEEE Trans. Nucl. Sci.*, vol. 68, no. 8, pp. 2187–2193, 2021.

[112] R. Abbasi *et al.*, "A Convolutional Neural Network based Cascade Reconstruction for the IceCube Neutrino Observatory," *JINST*, vol. 16, 2021.

[113] R. Frühwirth and R. K. Bock, *Data analysis techniques for high-energy physics experiments*, H. Grote, D. Notz, and M. Regler, Eds. Cambridge University Press, 2000, vol. 11.

[114] F. Ragusa and L. Rolandi, "Tracking at LHC," *New J. Phys.*, vol. 9, p. 336, 2007.

[115] R. Frühwirth and A. Strandlie, "Pattern recognition and reconstruction: Datasheet from Landolt-Börnstein - Group I elementary particles, nuclei and atoms." [Online]. Available: https://materials.springer.com/lb/docs/sm_lbs_978-3-642-03606-4_13

[116] H. Kälviäinen, P. Hirvonen, L. Xu, and E. Oja, "Probabilistic and non-probabilistic hough transforms: overview and comparisons," *Image Vis. Comput.*, vol. 13, no. 4, pp. 239–252, 1995.

[117] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 03 1960.

[118] D. E. Catlin, "The discrete Kalman filter," in *Estimation, Control, and the Discrete Kalman Filter*. New York, NY: Springer New York, 1989, pp. 133–163.

[119] R. Fruhwirth, "Application of Kalman filtering to track and vertex fitting," *Nucl. Instrum. Meth. A*, vol. 262, pp. 444–450, 1987.

[120] R. Mankel, "A Concurrent track evolution algorithm for pattern recognition in the HERA-B main tracking system," *Nucl. Instrum. Meth. A*, vol. 395, pp. 169–184, 1997.

[121] B. Schmidt, "The High-Luminosity upgrade of the LHC: Physics and Technology Challenges for the Accelerator and the Experiments," *J. Phys. Conf. Ser.*, vol. 706, no. 2, p. 022002, 2016.

[122] The HEP.TrkX Collaboration, "Hep advanced tracking algorithms with cross-cutting applications," 2016. [Online]. Available: https://heptrkx.github.io

[123] S. Farrell, D. Anderson, P. Calafiura, G. Cerati, L. Gray, J. Kowalkowski, M. Mudigonda, Prabhat, P. Spentzouris, M. Spiropoulou, A. Tsaris, J. R. Vlimant, and S. Zheng, "The hep.trkx project: deep neural networks for hl-lhc

online and offline tracking," *Proc. of the EPJ Web Conf.*, vol. 150, p. 00003, 2017.

[124] A. Tsaris *et al.*, "The hep.trkx project: Deep learning for particle tracking," *Journal of Physics: Conf. Series*, vol. 1085, no. 4, p. 042023, sep 2018.

[125] S. Farrell, P. Calafiura, M. Mudigonda, Prabhat, D. Anderson, J.-R. Vlimant, S. Zheng, J. Bendavid, M. Spiropulu, G. Cerati, L. Gray, J. Kowalkowski, P. Spentzouris, and A. Tsaris, "Novel deep learning methods for track reconstruction," in *Proc. of the Int. Workshop Connecting the Dots*, 10 2018.

[126] S. e. o. Amrouche, "The tracking machine learning challenge: Accuracy phase," in *The NeurIPS '18 Competition*, 2020, pp. 231–264.

[127] The Exa.TrkX Collaboration, "Hep advanced tracking algorithms at the exascale," 2019. [Online]. Available: https://exatrkx.github.io

[128] X. Ju *et al.*, "Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors," in *Proc. of the Annual Conf. on Neural Information Processing Systems*, 3 2020.

[129] N. Choma *et al.*, "Track Seeding and Labelling with Embedded-space Graph Neural Networks," in *Proc.of the Connecting the Dots Workshop*, 6 2020.

[130] X. Ju *et al.*, "Performance of a geometric deep learning pipeline for HL-LHC particle tracking," *Eur. Phys. J. C*, vol. 81, no. 10, p. 876, 2021.

[131] A. Elabd *et al.*, "Graph Neural Networks for Charged Particle Tracking on FPGAs," *Front. Big Data*, vol. 5, 2022.

4EU+ International Workshop on Recent Advancements in Artificial Intelligence

Edited by Ruggero Donida Labati, Angelo Genovese, Vincenzo Piuri

Artificial intelligence is increasingly pervasive in a broad variety of applications and in our daily life, from scientific applications to industrial manufacturing, from medical applications to biomedical systems, from ambient intelligence to consumer electronics, from entertainment to games, from communications to social networks, from finance to marketing, and many more. Addressing the growing needs of smart applications and expanding our knowledge on the foundations and the opportunities offered by artificial intelligence are essential to advancing science and technology as well as to provide the critical support for social and economic development.

The 4EU+ group of research and education experts on "Transforming Science and Society: Advancing Information, Computation and Communication" (Flagship 3 of the Alliance) organizes several activities for promoting research collaboration and education initiatives in a variety of topics related to information and communication technologies and their multi/interdisciplinary foundations, including artificial intelligence.

ISBN 979-12-5510-378-3 (print)
ISBN 979-12-5510-382-0 (PDF)
ISBN 979-12-5510-386-8 (EPUB)
DOI 10.54103/milanoup.282