IEEE SYSTEMS, MAN AND CYBERNETICS SOCIETY SECTION

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 November 2023, accepted 15 November 2023, date of publication 20 November 2023,
date of current version 13 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3335196

SURVEY

Deep Representation Learning: Fundamentals, Technologies, Applications, and Open Challenges

AMIRREZA PAYANDEH[1], **KOUROSH T. BAGHAEI**[1], **(Member, IEEE), POOYA FAYYAZSANAVI**[1], **SOMAYEH BAKHTIARI RAMEZANI**[2], **(Member, IEEE), ZHIQIAN CHEN**[2], **AND SHAHRAM RAHIMI**[2], **(Member, IEEE)**

[1]Department of Computer Science, George Mason University, Fairfax, VA 22030, USA
[2]Department of Computer Science, Mississippi State University, Mississippi State, MS 39762, USA

Corresponding author: Amirreza Payandeh (e-mail: apayande@gmu.edu).

ABSTRACT Machine learning algorithms have had a profound impact on the field of computer science over the past few decades. The performance of these algorithms heavily depends on the representations derived from the data during the learning process. Successful learning processes aim to produce concise, discrete, meaningful representations that can be effectively applied to various tasks. Recent advancements in deep learning models have proven to be highly effective in capturing high-dimensional, non-linear, and multi-modal characteristics. In this work, we provide a comprehensive overview of the current state-of-the-art in deep representation learning and the principles and developments made in the process of representation learning. Our study encompasses both supervised and unsupervised methods, including popular techniques such as autoencoders, self-supervised methods, and deep neural networks. Furthermore, we explore a wide range of applications, including image recognition and natural language processing. In addition, we discuss recent trends, key issues, and open challenges in the field. This survey endeavors to make a significant contribution to the field of deep representation learning, fostering its understanding and facilitating further advancements.

INDEX TERMS Representation learning, deep learning, feature extraction, transfer learning, natural language processing, computer vision.

I. INTRODUCTION

In recent years, machine learning [1], [2], [3], [4], [5], [6], [7], [8] has shown promising capabilities in various fields of study and application. Representation learning, as a core component in artificial intelligence is attracting more and more scientists every day. This interest is mirrored in an increasing number of papers, publications, and workshops on representation learning in international conferences and various influential journals.

Representation learning involves the detection, extraction, encoding, and decoding of features from raw data, which can then be used in learning tasks. Its objective is to abstract features that best represent data, and the algorithms developed for this purpose are collectively referred to as representation learning [9]. The performance of deep learning models relies heavily on the methods used to represent data. Consequently, the rapid growth of deep learning has been accompanied by significant advances in representation learning techniques. Deep learning owes its success to architectures composed of multi-layered non-linear modules, each transforming features into higher-level representations.

Learning representation aims to encode (embed) the raw input data into lower-dimensional real-valued vectors (embeddings), ideally disentangling the features that cause variation in the data distribution. Ideally, these representations should be robust to small differences or outliers in the input data, ensuring that temporally or spatially similar samples fall into close proximity in the representation space. Deep representation learning methods enable the hierarchical structuring of descriptive factors, where higher layers capture more abstract concepts. An ideal high-level representation

The associate editor coordinating the review of this manuscript and approving it for publication was Huiyan Zhang.

consists of simple and linearly correlated factors [10]. Owing to the nature of feature extraction in representation learning, representations can be shared and utilized across different tasks. Although achieving the characteristics mentioned above is challenging, the learned representation facilitates the discovery of latent patterns and trends in data for the learner, hence enhancing the learning of the multiple tasks [10]. Based on the application, the raw input data can be of any type, for instance, texts, images, audio, video, etc. Given a particular task, such as classification, segmentation, synthesis, and prediction, the main objective is to update the parameters of a neural network so that can represent the input data in a lower dimension.

In the domain of image processing, representation learning finds applications in visualization [11], regression [12], [13], [14], interpretation of predictions [15], [16], [17], generating synthetic data [18], finding and retrieving similar images [19], [20], image enhancement and denoising [21], [22], semantic segmentation, and object detection [23], [24], [25]. Challenges in 2D image processing also extend to volumetric image processing contexts, such as 3D MRI [26] and point cloud data captured by depth sensors [27].

In the analysis of sequential data, representation learning plays a crucial role in transferring representations across domains. This enables the generation of annotations and captions for images [28], [29], [30] and facilitates post-hoc interpretation in medical data analysis [31]. By leveraging learned representations, researchers can bridge the gap between different data modalities, allowing for more comprehensive and meaningful insights.

Natural language processing (NLP) leverages representation learning approaches across various domains, including text classification [32], question answering [33], machine translation [34], [35], [36], electronic health records [37], financial forecasting [38], chatbots [39], social media analysis [40], [41] and more. The field of NLP has witnessed an evolution from early rule-based methods to the application of statistical learning techniques, enabled by access to large amounts of data. However, the introduction of deep learning approaches to NLP in 2012 revolutionized the field, making neural network-based methods the dominant approaches [42]. In modern NLP, Word2Vec [43] and GloVe [44] have emerged as advanced, well-known approaches for representing words as vectors. Following a breakthrough in 2017 with attention-based models [45], advanced pre-trained models, particularly BERT [46], have garnered significant attention and generated excitement within the NLP community. These models have showcased exceptional performance and have become the focal point of current NLP research and applications.

Linear factor models, such as PCA and ICA, have been employed as early methods of feature extraction in representation learning. While these models can be extended to form more powerful representations, this article focuses primarily on deep models of representation. For a more comprehensive discussion on linear factor models, readers are encouraged to refer to [10] and [47]. The subsequent sections of this article delve into the prevalent approaches in deep representation learning, providing insights into their principles and techniques.

This survey provides a comprehensive overview of the current state-of-the-art methods and principles in deep representation learning. While representation learning has been reviewed in several previous surveys, this work offers a uniquely comprehensive and up-to-date treatment. Existing surveys have focused on specific approaches such as autoencoders [48], [49], generative adversarial networks [50], and foundation models [51]. Bengio et al. [10], in their 2013 publication, provided a perspective focused on disentangling factors of variation. LeCun et al. [9], in their 2015 work, reviewed representation learning, emphasizing deep learning breakthroughs. It's essential to consult the original paper for a detailed understanding of their coverage. More recent works, such as Zhou et al. [52] and Otter et al. [53], delivered insightful surveys on representation learning for computer vision and natural language processing, respectively. Zhou et al. discuss methods for various video segmentation tasks, while Otter et al. review developments in core NLP areas and related applications.

Our work encompasses a broader scope, including major techniques for both supervised and unsupervised feature learning. We discuss recent advancements spanning autoencoders, generative adversarial networks, graph neural networks, Bayesian deep learning, transformers, and other critical topics. Additionally, we explore applications across computer vision, natural language processing, healthcare, and other domains. This survey aims to connect key concepts in deep representation learning, tracing progress from foundational methods to cutting-edge techniques. By synthesizing a wide range of contemporary research into a single source, we hope to provide valuable insights into this rapidly evolving field and offer a comprehensive reference for representation learning distinct from previous works.

II. MULTI LAYER PERCEPTRON

A multi layer perceptron or feedforward neural network is a stack of multiple layers. Each layer, consists of one linear transformation and one non-linear activation function. Given an input vector $\vec{x} \in \mathbb{R}^n$ and weight matrix $W \in \mathbb{R}^{n\times m}$, transformed vector $\vec{y} \in \mathbb{R}^m$ can be calculated as:

$$\vec{y} = W^T\vec{x} \tag{1}$$

The weight matrix W in Eq. 1 consists of m rows $\vec{r}_i \in \mathbb{R}^m$ (where $1 \leq i \leq m$). As depicted in Fig. 1, each row $\vec{r}_i$ can be thought of as a vector perpendicular to a surface S_i in hyperspace that passes through the origin. Surface S_i divides the n-dimensional space into 3 sub-spaces: three sets of points residing on the surface and the two sides of it. Each y_i in vector $\vec{y} = (y_1, y_2, \ldots, y_m)$, is calculated by the dot product of row $\vec{r}_i$ and the input vector $\vec{x}$. Depending on the relative positions of the point $\vec{x}$ and surface S_i, the value of $\vec{y}$ along the i-th dimension may be positive, negative, or zero. A bias number b_i can also be employed to further control the value of $\vec{y}$. Essentially, the parameters of the weight ($\vec{r}_i$) and bias ($\vec{b}$) vectors decide on how the features of the input vector $\vec{x}$ affect $\vec{y}$ along the i-th dimension in the target space of m dimensions. The training process, updates these weights and biases so that they can fit the input data to their corresponding target values. Thus, the network learns how to distinguish or generate certain similarities and patterns among the features of the input data. Each of the parameters of $\vec{y}$ are passed to an activation function in order to add non-linearity to the output. In a similar way, an extra layer can be utilized to capture the patterns and similarities in the output vectors of the previous layer. Hence, extracting more complex characteristics in the data. Adding extra layers may increase the capability of a network in learning representations in exchange for its computational complexity.

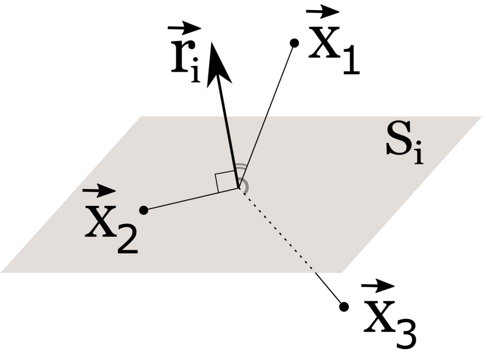


FIGURE 1. **An intuitive representation of how the weights of a linear layer transform from the input space to the output space. $\vec{r}_i$, the i-th row of the weight matrix of the linear layer may be considered as the normal vector of surface S_i, that may affect different input data points in different ways: a) Points above the surface: $\vec{r}_i \cdot \vec{x}_1 > 0$. b) Points residing on the surface: $\vec{r}_i \cdot \vec{x}_1 = 0$. c) Points that are posed below the surface: $\vec{r}_i \cdot \vec{x}_1 < 0$. The result of the product may also be passed through an activation function to add non-linearity.**

III. GENERATIVE MODELS

Generative models are unsupervised methods that aim to learn and approximate the distribution function from which the samples of a given unlabeled dataset are generated. By acquiring knowledge of this approximate generator function, models gain the ability to generate random samples that are not originally present in the dataset, yet possess resemblances to the existing data [54]. Generative models can be grouped into two categories: energy-based and function-based models [54]. Energy-based models include Boltzmann Machines (BM), Restricted Boltzmann Machines (RBM), and Deep Belief Networks (DBN) [55]. Energy-based models are probabilistic models that provide information about the probability density or mass function without explicitly determining the normalizing constant, resulting in un-normalized probabilities. These models exclusively define the energy function, which corresponds to the unnormalized negative log-probability [56]. On the other hand, function-based models, such as the Auto-Encoder [11], [57] and its variants and Generative Adversarial Networks (GANs) [58], learn the mapping function from input to output, enabling the generation of new samples based on this learned mapping.

A. BOLTZMANN MACHINES

1) BOLTZMANN MACHINE

The Boltzmann Machine is an energy-based model initially introduced for learning arbitrary probability distributions over binary vectors [47]. Later, continuous variations of Boltzmann Machines have been proposed [59].

Given a d-dimensional binary vector $x \in \{0, 1\}^d$ as input, the joint probability distribution is defined as:

$$P(x) = \frac{exp(-E(x))}{Z} \tag{2}$$

where Z is normalization parameter defined as:

$$Z = \sum_x exp(-E(x)) \tag{3}$$

ensuring that $P(x)$ forms a probability density. In Equation 2, $E(x)$ represents the energy function defined as:

$$E(x) = -(x^T Wx + b^T x) \tag{4}$$

The training process involves maximizing the likelihood and minimizing the energy function. Boltzmann Machines exhibit a learning procedure inspired by biological neurons, where the connection between two neurons strengthens if they are both excited together and weakens otherwise. This biologically inspired learning mechanism enhances the model's ability to capture dependencies and patterns within the data.

One popular training algorithm for Boltzmann Machines is Contrastive Divergence, which provides an efficient approximation to maximum likelihood training using Gibbs sampling [56], [60].

2) RESTRICTED BOLTZMANN MACHINE (RBM)

The Restricted Boltzmann Machine limits the connections among the nodes of a graph to only links between the visible and hidden neurons. Consequently, there are no connections among the hidden neurons or the visible ones. The vector of nodes, denoted as x, can be divided into two subsets: visible nodes v and hidden nodes h. The energy function for RBM is given by:

$$\mathrm{E}(v, h) = -b^T v - c^T h - v^T Wh \tag{5}$$

Here, b and c represent the bias weights, and the matrix W represents the connection weights.

The partition function for RBM, denoted as Z, is defined as:

$$Z = \sum_v \sum_h e^{-E(v,h)} \tag{6}$$

RBMs are probabilistic graphical models and serve as the fundamental building blocks of Deep Belief Networks (DBNs). However, due to the intractability of the partition function Z, training RBMs requires specialized methods such as Contrastive Divergence [61] and Score Matching [47].

3) DEEP BELIEF NETWORK (DBN)

A Deep Belief Network consists of several RBMs. When a DBN has only one hidden layer, it can be considered as an RBM. To train a DBN, an RBM is first trained using likelihood maximization or contrastive divergence [47]. Subsequently, another RBM is trained to model the distribution of the previous layers. By adding more layers, the variational lower bound of the log-likelihood of the data increases, enabling the DBN to capture complex patterns and dependencies.

The deepest layer of DBNs is characterized by undirected connections, setting them apart from other deep neural network architectures [55]. However, it is important to mention that the term "DBN" is sometimes incorrectly used to refer to any neural network, which may lead to confusion.

4) OTHER VARIANTS

There are other variants of Boltzmann machines proposed such as Deep Boltzmann Machines (DBM) [62], Spike and Slab Restricted Boltzmann Machines (ssRBM) [63], Convolutional Boltzmann Machines [64]. However, other generative models such as variational auto encoders and GANs have proved as viable substitutes for variations and derivations of Boltzmann machines [58].

B. AUTO-ENCODERS

Autoencoder-based models are considered to be some of the most robust unsupervised learning models for extracting effective and discriminating features from a large unlabeled dataset. The general architecture of an auto-encoder consists of two components: **Encoder:** Function f which aims to transform the inputs x to a latent variable h in lower dimensions. **Decoder:** Function g reconstructs the input $\hat{x}$, given the latent variable h. The training process involves updating the weights of the encoder and decoder networks according to the loss function of the reconstruction:

$$\mathcal{L}(x, \hat{x}) = \mathcal{L}\Big(x, g\big(f(x)\big)\Big) \tag{7}$$

Many variants of auto-encoders have been proposed in the literature; however, they can be categorized in four major groups [54].

1) UNDERCOMPLETE AUTOENCODER

In order to make the autoencoder learn the distributions from the data, the latent variables should have lower dimensions than the input data. Otherwise, the network would fail to learn any useful features from the data. This type of autoencoder is known as *undercomplete autoencoder* [47].

2) DENOISING AUTOENCODER (DAE)

Denoising Autoencoder corrupts the data by adding stochastic noise reconstructs it back into intact data. Hence, it is called *denoising autoencoder*. As depicted in Fig. 3, the added noise to the input is the only difference of this method to the traditional autoencoders. This approach results in better feature extraction and better generalization in classification tasks [65]. Also, Several DAEs can be trained locally by adding noise to their inputs and stacking consecutively to form a deep architecture called Stacked DAE, with higher representation capabilities.

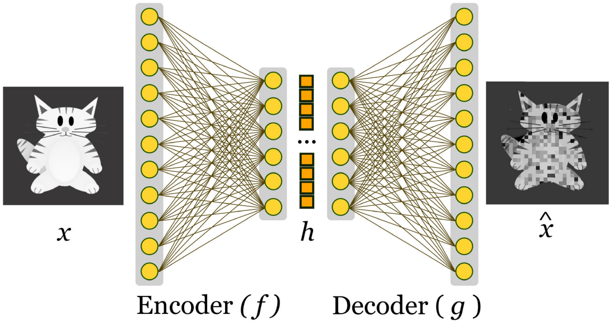


FIGURE 2. General architecture of an auto-encoder. The encoder transforms input x into the latent vector h: $h = f(x)$. The decoder reconstructs the input from h: $\hat{x} = g(h)$.

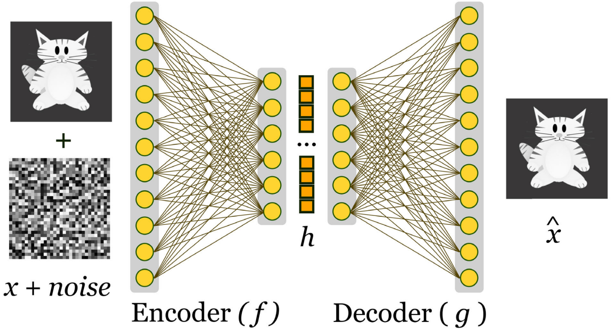


FIGURE 3. General architecture of a denoising auto-encoder (DAE). Adding noise to the input during the training process, results in more robust learning of the features. Hence, increasing the generalization ability.

3) SPARSE AUTOENCODERS (SAE)

Sparse representation refers to the technique of decomposing a data set into a set of overcomplete vectors where only a small subset of those vectors combine to describe the data. The overcompleteness of representation can lead to more expressive basis vectors which can capture complex structures more effectively. The sparsity puts an additional constraint on the number of basis vectors present for decomposing data to basis vectors. Sparse representation can be formulated as the disentangling of an input signal into a linear combination of its latent features [66]. The loss function of a sparse autoencoder includes an additional sparsity constraint ($\Omega(h)$) on the latent variables [47]:

$$\mathcal{L} = \mathcal{L}\big(x, g(f(x))\big) + \Omega(h) \tag{8}$$

Thus making the autoencoder to extract features from the data and represent them in sparse vectors and matrices [67].

4) VARIATIONAL AUTOENCODER (VAE)

Although this type of autoencoder has the same components as the traditional autoencoder (Fig. 2), its training process is based on variational inference [68]. Just as the traditional autoencoder, the encoder function f is trained to map the input data to the latent variables z and the decoder function g is trained to map the latent variables z to the input data. However, for this autoencoder to work, the latent variable z is assumed to be Guassian.[1] By choosing this representation, we gain significant control over how the latent distribution should be modeled, resulting in a smoother and more continuous latent space. And the loss function for this training consists of two parameters: First, *Kullback-Leibler(KL) divergence* [69] of the output of the encoder f and Guassian distribution; Thus forcing the encoder to map the input data to the Gaussian distribution in the latent space. Second, the reconstruction loss: [70]:

$$\mathcal{L} = \mathcal{D}\big(KL(f\|\mathcal{N}(0, I))\big) + \mathcal{L}\big(x, g(f(x))\big) \tag{9}$$

Variational inference is discussed in more details in section V-C.

5) CONTRACTIVE AUTOENCODER (CAE)

The main goal in proposing this variant of autoencoder was to make the features in the activation layer invariant with respect to small perturbations in the input [71]. The basic autoencoder may be converted to a contractive autoencoder by adding the following regularization to its loss function:

$$\|J_f(x)\|_F^2 = \sum_i \sum_j \left(\frac{\partial h_j(x)}{\partial x_i}\right)^2 \tag{10}$$

where $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a non-linear mapping function from input space $x \in \mathbb{R}^m$ to the hidden layer $h \in \mathbb{R}^n$. The regularization term is the squared value of the first-order partial derivatives of the hidden values with respect to the input values. By penalizing the first derivative of the encoding function, the derivative is forced to maintain lower values. In this way, the encoding function may learn a flatter representation. As a result, the encoding function may become more robust or invariant to small perturbations in the input.

The loss function of the contractive autoencoder may be written as:

$$\mathcal{L}_{CAE} = \sum_{x \in X} \left(\mathcal{L}_R\big(x, g(f(x))\big) + \lambda \|J_f(x)\|_F^2\right) \tag{11}$$

where X is the dataset of training samples, $\mathcal{L}_R$ denotes the reconstruction loss, and $\lambda \in \mathbb{R}$ controls the effect of contractive loss. The input points get closer in distance when mapped to the hidden state i.e. they are *contracted*. This contraction can be thought as the reason behind robustness in features.

C. GENERATIVE ADVERSARIAL NETWORKS

Although both Autoencoders and GANs are generative models, their learning mechanism is different. Autoencoders are trained to learn hidden representations, where GANs are designed to generate new data. The most prevalent generative model utilized in many applications is the GAN architecture [58]. As depicted in Fig. 4, it resembles a two-player minimax game where two functions known as the generator $\mathcal{G}$ and the discriminator $\mathcal{D}$ are trained as opponents. The $\mathcal{G}$ function tries to generate fake samples as similar as possible to the real input data from a noise variable z, and the $\mathcal{D}$ function aims to discriminate the fake and real data apart. The minimax game can be described with the following objective function:

$$\underset{\mathcal{G}}{min}\ \underset{\mathcal{D}}{max}\ V(\mathcal{D}, \mathcal{G}) = \mathbb{E}_{x \sim p_{data}(x)}[log\mathcal{D}(x)] + \mathbb{E}_{z \sim p_z(z)}[log\big(1 - \mathcal{D}(\mathcal{G}(z))\big)] \tag{12}$$

where $x \sim p_{data}$ denotes the real data sample x with its distribution p_{data}. And $\mathcal{D}(x)$ represents the class label that the discriminator $\mathcal{D}$ assigns to the input sample x. For the noise variable z a prior is assumed as $z \sim p_z(z)$.

The success of CNNs in image analysis and the capabilities that GANs provide, has made generative CNNs possible [72]. Numerous extensions to the original GAN have been proposed so far [73] such as interpretable representation learning by information maximizing (InfoGAN) that forces the model to disentangle and represent features of images in certain elements of the latent vector [74]. Or Cycle-Consistent GAN (CycleGAN) that learns characteristics of an image dataset and translates them into another image dataset without any dataset of paired images [75]. An inherent limitation of the original GAN is that it does not have any control over its output. Conditional Generative Adversarial Nets [76] incorporate auxiliary inputs such as class labels into their model to generate the desired output.

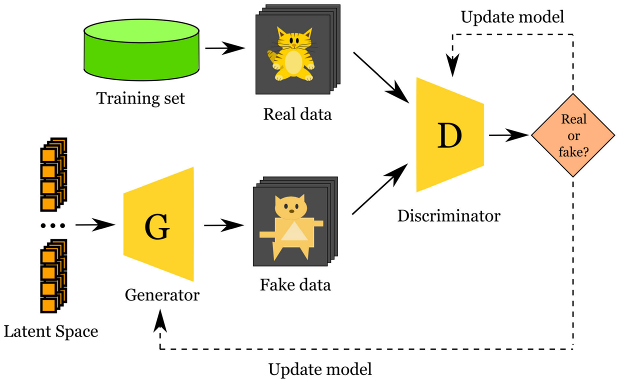


FIGURE 4. Generative adversarial network.

D. APPLICATIONS

Generative models provide a powerful framework for learning and approximating complex data distributions, allowing for the generation of realistic and novel samples. They have

[1] Depending on the type of data, this can also be Bernouli.

shown promise in a wide range of applications, contributing to advancements in various fields. These models have found applications in numerous domains, enabling the development of powerful deep architectures. In the field of NLP, generative models have been utilized for tasks such as text generation [77], [78] and machine translation [79]. Notably, the GPT-3.5 model has demonstrated remarkable performance in language generation tasks [39].

In image processing, generative models have demonstrated their effectiveness in various applications. They have been employed for tasks such as denoising 3D magnetic images [80], unsupervised image generation [81], image-to-image translation [75], [82], cross-modality synthesis [83], [84], data augmentation and anonymization [85], image segmentation [86], [87], super-resolution [73], [88], [89], [90], and video analysis [91].

Furthermore, generative neural networks and their derivatives have been utilized in combination with deep reinforcement learning algorithms for tasks such as object detection [92], [93]. They have also been applied in the analysis of graph data, contributing to advancements in areas like graph generation [94] and graph representation learning [95].

Overall, generative neural networks have proven to be versatile tools with applications spanning a wide range of disciplines, delivering state-of-the-art performance in various problem domains.

IV. GRAPH NEURAL NETWORKS

The widespread success of deep learning in a myriad of applications over the past decade is well-documented [35], [36], [59], [79], [96], [97]. In the evolving landscape of deep learning research, Graph Neural Networks (GNNs) stand out as a pivotal advancement for effective data analysis in non-Euclidean geometries. GNNs have found applications in diverse real-world contexts, including but not limited to, biological regulatory networks in genomics [98], [99], telecommunication infrastructures [100], social interaction frameworks [101], transportation systems [102], [103], [104], energy grids [105], [106], [107], electrical circuits [108], [109], epidemiological spread [110], and neural networks in the brain [111]. Traditional deep learning architectures like ConvNets struggle with the irregular, non-Euclidean structure of graphs, primarily because the varying neighborhood sizes of graph nodes are incompatible with ConvNets' fixed-size kernels. To address this, a plethora of GNN models have been proposed, leveraging the strengths of deep learning to capture the inherent complexities of non-Euclidean graphs [112], [113], [114]

A. BASICS OF GNN

Graph convolution originates from spectral graph theory which is the study of the properties of a graph in relationship to the eigenvalues, and eigenvectors of associated graph matrices [115], [116], [117]. The spectral convolution methods [112], [113], [114], [118] are the major algorithm designed as the graph convolution methods, and it is based on the graph Fourier transform [119], [120]. GCN focus processing graph signals defined on undirected graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, where $\mathcal{V}$ is a set of n vertexes, $\mathcal{E}$ represents edges and $\mathcal{W} = [w_{ij}] \in \{0,1\}^{n\times n}$ is an unweighted adjacency matrix. A signal $x : \mathcal{V} \rightarrow \mathbb{R}$ defined on the nodes may be regarded as a vector $x \in \mathbb{R}^n$. Combinatorial graph Laplacian [115] is defined as $\mathbf{L} = D - \mathcal{W} \in \mathbb{R}^{n\times n}$ where D is degree matrix. As $\mathbf{L}$ is a real symmetric positive semidefinite matrix, it has a complete set of orthonormal eigenvectors and their associated ordered real nonnegative eigenvalues identified as the frequencies of the graph. The Laplacian is diagonalized by the Fourier basis $\mathbf{U}^\intercal$: $\mathbf{L} = \mathbf{U}\,\Lambda\,\mathbf{U}^\intercal$ where Λ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues, i.e., $\Lambda_{ii} = \lambda_i$. The graph Fourier transform of a signal $x \in \mathbb{R}^n$ is defined as $\hat{x} = \mathbf{U}^\intercal x \in \mathbb{R}^n$ and its inverse as $x = \mathbf{U}\hat{x}$ [119], [120], [121]. To enable the formulation of fundamental operations such as filtering in the vertex domain, the convolution operator on graph is defined in the Fourier domain such that $f_1 * f_2 = \mathbf{U}\left[(\mathbf{U}^\intercal f_1) \odot (\mathbf{U}^\intercal f_2)\right]$, where $\odot$ is the element-wise product, and f_1/f_2 are two signals defined on vertex domain. It follows that a vertex signal $f_2 = x$ is filtered by spectral signal $\hat{f_1} = \mathbf{U}^\intercal f_1 = \mathbf{g}$ as:

$$\mathbf{g} * x = \mathbf{U}\left[\mathbf{g}(\Lambda) \odot \left(\mathbf{U}^\intercal f_2\right)\right] = \mathbf{U}\,\mathbf{g}(\Lambda)\,\mathbf{U}^\intercal x.$$

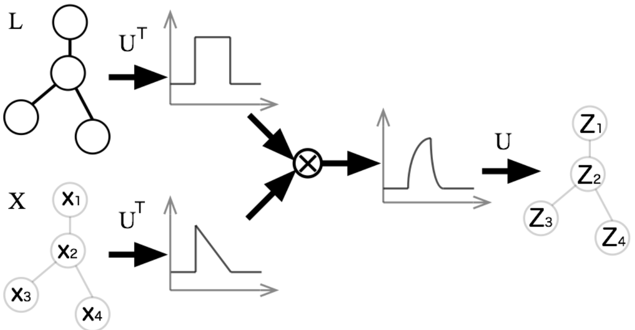


FIGURE 5. Illustration of graph convolution.

Note that a real symmetric matrix $\mathbf{L}$ can be decomposed as $\mathbf{L} = \mathbf{U}\,\Lambda\,\mathbf{U}^{-1} = \mathbf{U}\,\Lambda\,\mathbf{U}^\intercal$ since $\mathbf{U}^{-1} = \mathbf{U}^\intercal$. D. K. Hammond et al. and Defferrard et al. [114], [122] apply polynomial approximation on spectral filter $\mathbf{g}$ so that:

$$\begin{aligned}\mathbf{g} * x &= \mathbf{U}\,\mathbf{g}(\Lambda)\,\mathbf{U}^\intercal x \\ &\approx \mathbf{U}\sum_k \theta_k T_k(\tilde{\Lambda})\,\mathbf{U}^\intercal x \quad \left(\tilde{\Lambda} = \frac{2}{\lambda_{max}}\Lambda - \mathbf{I_N}\right) \\ &= \sum_k \theta_k T_k(\tilde{\mathbf{L}})x \quad \left(\mathbf{U}\,\Lambda^k\,\mathbf{U}^\intercal = (\mathbf{U}\,\Lambda\,\mathbf{U}^\intercal)^k\right)\end{aligned}$$

Kipf et al. [113] simplifies it by applying multiple tricks:

$$\begin{aligned}&\mathbf{g} * x \\ &\approx \theta_0\,\mathbf{I_N}\,x + \theta_1\tilde{\mathbf{L}}x \qquad \text{(expand to 1st order)}\end{aligned}$$

$$
\begin{aligned}
&= \theta_0\, \mathbf{I_N}\, x + \theta_1 \big(\frac{2}{\lambda_{max}}\, \mathbf{L} - \mathbf{I_N})\big) x && (\tilde{\mathbf{L}} {=} \tfrac{2}{\lambda_{max}}\, \mathbf{L} - \mathbf{I_N})) \\
&= \theta_0\, \mathbf{I_N}\, x + \theta_1 (\mathbf{L} - \mathbf{I_N}))x && (\lambda_{max}{=}2) \\
&= \theta_0\, \mathbf{I_N}\, x - \theta_1\, \mathbf{D}^{-\frac{1}{2}}\, \mathbf{A}\, \mathbf{D}^{-\frac{1}{2}}\, x && (\mathbf{L}{=}\mathbf{I_N} - \mathbf{D}^{-\frac{1}{2}}\, \mathbf{A}\, \mathbf{D}^{-\frac{1}{2}}) \\
&= \theta_0 (\mathbf{I_N} + \mathbf{D}^{-\frac{1}{2}}\, \mathbf{A}\, \mathbf{D}^{-\frac{1}{2}})x && (\theta_0{=}-\theta_1) \\
&= \theta_0 (\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}})x && (\text{renormalization:} \tilde{\mathbf{A}}{=}\mathbf{A} + \mathbf{I_N}, \\
& && \tilde{\mathbf{D}}_{ii}{=}\textstyle\sum_j \mathbf{A}_{ij}).
\end{aligned}
$$

Rewriting the above GCN in matrix form: $g_\theta * X \approx (\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}) X \Theta$, it leads to *symmetric normalized Laplacian* with raw feature. GCN has been analyzed in [123] using smoothing Laplacian [124], and the updated features (y) equals to the smoothing Laplacian, i.e., the weighted sum of itself (x_i) and its neighbors (x_j): $y = (1-\gamma)x_i + \gamma \sum_j \frac{\tilde{a}_{ij}}{d_i} x_j = x_i - \gamma(x_i - \sum_j \frac{\tilde{a}_{ij}}{d_i} x_j)$, where γ is a weight parameter between the current vertex x_i and the features of its neighbors x_j, d_i is degree of x_i, and y is the smoothed Laplacian. Rewriting in matrix form, the smoothing Laplacian is:

$$
\begin{aligned}
Y &= x - \gamma \tilde{\mathbf{D}}^{-1} \tilde{\mathbf{L}} x \\
&= (\mathbf{I_N} - \tilde{\mathbf{D}}^{-1} \tilde{\mathbf{L}}) x && (\gamma = 1) \\
&= (\mathbf{I_N} - \tilde{\mathbf{D}}^{-1} (\tilde{\mathbf{D}} - \tilde{\mathbf{A}}))x && (\tilde{\mathbf{L}} = \tilde{\mathbf{D}} - \tilde{\mathbf{A}}) \\
&= \tilde{\mathbf{D}}^{-1} \tilde{\mathbf{A}} x.
\end{aligned}
$$

The above formula is *random walk normalized Laplacian* as a counterpart of *symmetric normalized Laplacian*. Therefore, GCN can be treated as a first-order Laplacian smoothing which averages neighbors of each vertex.

B. TAXONOMY OF GNN

As many surveys on GNN state [118], [125], [126], [127], [128], [129], GCNs can be classified into two major categories based on the operation type. Therefore, we introduce a taxonomy of GNN in the following two perspectives.

C. SPECTRAL-BASED GNN

This group of GCN highly relies on spectral graph analysis and approximation theory. Spectral-based GNN models analyzes the weight-adjusting function (i.e., filter function) on eigenvalues of graph matrices, which corresponds to adjusting the weights assigned to frequency components (eigenvectors). Many of Spectral-based GNN models are equivalent to low-pass filters [94]. Based on the type of filter function, there are linear filtering [113], [130], [131], polynomial filtering [122], [132], [133], [134], [135], and rational filtering [94], [136], [137], [138]. Beyond that, [139] adaptively learns the center of spectral filter. Closely, [140] proposed a high-low-pass filter based on p-Laplacian. References [141], [142], and [143] revisit the spectral graph convolutional filter and make theoretical analyze. Optionally, one can choose graph wavelet to model spectrum of each node [144], [145], [146], [147].

D. SPATIAL-BASED GNN

Nowadays, there are more emerging GNNs using spatial operations. Based on the spatial operation, they can be categorized into three groups: local aggregation which only combine direct neighbors [130], [131], [148], [149], [150], higher order aggregation which involves second order or higher orders of neighbors [114], [122], [133], [134], [135], [151], and dual-directional aggregation that propagates information in both forward and backward directions [94], [136], [137], [138], [152], [153], [154].

E. APPLICATIONS

Graph neural networks have been applied in numerous domains such as physics, chemistry, biology, computer vision, NLP, intelligent transportation, social networks [118], [125], [126], [127], [128], [155]. To model physical objects, DeepMind [156] provides a toolkit to generalize the operations on graphs, including manipulating structured knowledge and producing structured behaviors, and [157] simulates fluids, rigid solids, and deformable materials. Treating chemical structure as a graph [158], [159], [160], [161] represent molecular structure, and [162], [163], [164] model protein interfaces. Further, [165] predict the chemical reaction and retrosynthesis. In computer vision, question-specific interactions are modeled as graphs in visual question answering [166], [167]. Similar to physics applications, human interaction with humans could be represented by their connections [168], [169], [170], [171]. Reference [172] model the relationship among word and document as a graph, while [173] and [174] characterize the syntactic relations as a dependency tree. Predicting traffic flow is a fundamental problem in urban computing, and transportation network can be modeled as a spatiotemporal graph [175], [176], [177], [178]. Functional MRI (fMRI) is a graph data where brain regions are connected by functional correlation [179], [180]. Reference [181] employs a graph convolutional network to localize eloquent cortex in brain tumor patients, [182] integrates structural and functional MRIs using Graph Convolutional Networks to do Autism Classification, and [183] applies graph convolutional networks to classify mental imagery states of healthy subjects by only using functional connectivity. To go beyond rs-fMRI and model both functional dependency among brain regions and the temporal dynamics of brain activity, spatio-temporal graph convolutional networks (ST-GCN) are applied to formulate functional connectivity networks in the format of spatio-temporal graphs, which can be also applied in physcial flows [102], [184], [185].

V. BAYESIAN DEEP LEARNING AND VARIATIONAL INFERENCE

Bayesian networks are statistical methodology that combines standard networks with Bayesian inference. Following the Bayes rule (Eq. 13), the random variables of a problem can be represented as a directed acyclic graph known as **Bayesian Network** or **belief network** [47].

Let $\mathbf{z} = \{z_1, z_2, \ldots, z_N\}$, and $\mathbf{x} = \{x_1, x_2, \ldots, x_M\}$ denote the latent variables and the observations respectively. The latent variables facilitate the representation of the observations' distribution. Given a prior distribution $p(\mathrm{z})$ over the latent variables, the Bayesian model maps the latent variables to the observations by the likelihood function $p(\mathrm{x|z})$. Thus producing the joint distribution of the latent variables and observations:

$$p(\mathrm{z,x}) = p(\mathrm{x|z})p(\mathrm{z}) = p(\mathrm{z|x})p(\mathrm{x}) \tag{13}$$

In Bayesian models, **inference** involves in calculating the **the posterior distribution** which is the conditional distribution of the latent variables given the observations:

$$p(\mathrm{z|x}) = \frac{p(\mathrm{z,x})}{p(\mathrm{x})} \tag{14}$$

The marginal density of the observations $p(\mathrm{x})$ is called **evidence** which is calculated by integration over latent variables:

$$p(\mathrm{x}) = \int p(\mathrm{x,z})dz \tag{15}$$

A. HIDDEN MARKOV MODEL

Hidden Markov Model (HMM) is a probabilistic Bayesian network architecture [186] that approximate the likelihood of distributions in a sequence of observations [187]. As opposed to Bayesian networks, these networks are undirected and can be cyclic. The family of HMMs, including the **Hidden Semi-Markov Model** (HSMM), are widely used to identify patterns in sequential data of time varying and non-time varying nature [188]. They are well suited for sequencing time series problems with a linear degree of growth over data patterns [189]. A generalized HMM is composed of a state model of Markov process z_t, linked to an observation model $P(x_t|z_t)$, which contains the observations x_t of the state model.

While HMMs are considered agnostic of the duration of the states, the HSMMs can take the duration of each state into consideration [190], which makes HSMMs suitable for prognosis [191], [192]. Neither HMM nor HSMM can capture the inter-dependencies of observations in temporal data, which is a key factor in determining the state of the system. To overcome this shortcoming, one can use the **Auto-Regressive Hidden Markov Model** (ARHMM) which accounts for the inter-dependencies between consecutive observations to model longer time series [193], [194], [195]

HMMS can loose their efficiency when dealing with distributed state representations. The **Factorial HMM** (FHMM) is an extension of HMM that aims at addressing this problem by using several independent layers of state structure HMMs. These layers are free to evolve irrespective of the other layers, allowing observations at any given time to be dependent on the value of all states at that time [196].

Due to exponential time complexity of this integration, its computation is intractable. Thereupon, the posterior distribution cannot be calculated directly. Rather, it is approximated [68]. There are two major methods of posterior approximation:

- **Sampling based**: Markov Chain Monte Carlo (MCMC) methods are often able to approximate the true and unbiased posterior through sampling, although they are slow and computationally demanding on large and complex datasets with high dimensions.
- **Optimization based:** approaches for Variational Inference (VI) tend to converge much faster though they may provide over-simplified approximations.

The following subsections explain each of these approaches on these topics due to their importance.

B. MARKOV CHAIN MONTE CARLO

Monte Carlo estimation is a method for approximating the expectation of random variables where their expectation may involve intractable integrations as in Eq. 15. Markov Chain Monte Carlo, Metropolis-Hastings (MH) sampling, Gibbs sampling, and their parallel and scalable variations are instances of MCMC estimations [197].

Although the basic Monte Carlo algorithm requires the samples to be independent and identically distributed (i.i.d), obtaining such samples my be computationally intensive in practice. Nonetheless, the sample generation process can still be facilitated by satisfying some properties as described below [198]:

1) MARKOV PROPERTY

Given the past and present states, the probability of transition to the future states relies on the present state only. Mathematically speaking, a **Markov chain** is a sequence of random variables $\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n$ representing states, that hold the following property:

$$\begin{aligned} P(\mathrm{X}_{n+1} = x|\mathrm{X}_n = x_n, \mathrm{X}_{n-1} = x_{n-1}, \ldots, \mathrm{X}_1 = x_1) \\ = P(\mathrm{X}_{n+1} = x|\mathrm{X}_n = x_n) \end{aligned} \tag{16}$$

2) TIME-HOMOGENEITY

A stochastic process that the probability of transition is independent of the index n, is time-homogeneous.

3) STATIONARY DISTRIBUTION

A probability distribution of a Markov chain represented as a row vector π that is invariant by matrix of transition probabilities K.

$$\pi = \pi \mathrm{K} \tag{17}$$

4) IRREDUCIBLITY

A Markov chain is irreducible if in a discrete state space, it can go from any state x to any other state y in a finite number of transitions. In mathematical terms, given that:

$$\mathrm{K(x,y)} = P(\mathrm{X}_{n+1} = y|\mathrm{X}_n = x) \tag{18}$$

where K is a matrix, there exist an integer n such that $K^{n}_{(x,y)} > 0$.

A stationary distribution of a chain is unique if it has stationary distribution and it is irreducible. Considering a Markov chain with a unique stationary distribution π, according to the law of large numbers [198], the expectation value of a function $f(x)$ over π can be approximated by calculating the mean of the outputs from the Markov chain:

$$E_{\pi}[f(x)] = \int f(x)\pi(x)dx = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} f(x_i) \tag{19}$$

For a more detailed explanation on MCMCs, the readers may consult [197], [198].

C. VARIATIONAL INFERENCE (VI)

Variational Inference (VI) is of high importance in modern machine learning architectures. Regularization through variational dropout [199], [200], representing model uncertainty in classification tasks and reinforcement learning [201], are a few of scenarios in which variational inference is utilized. The core idea in VI is to find an approximate distribution function which is simpler than the true posterior and its Kullback-Liebler divergence [69] from the true posterior is as lowest as possible [202].

The problem changes to the search for a candidate density function q_c(x) among a specified family of distributions D such that it best resembles the true posterior function:

$$q_c(\mathrm{z}) = \underset{q(z)\in D}{\mathrm{argmin}}\ \mathrm{KL}(q(\mathrm{z})\|p(\mathrm{z}|\mathrm{x})) \tag{20}$$

The Eq. 20 may be optimized indirectly through maximization of the variational objective function ELBO(q):

$$\mathrm{ELBO(q)} = \mathbb{E}[\log p(\mathrm{x}|\mathrm{z})] - \mathrm{KL}\ (q(\mathrm{z})||p(\mathrm{z})) \tag{21}$$

where ELBO(q) is called *evidence lower bound* function. The KL(q(z)||p(z)) encourages the density function q(z) to get closer to the prior function. And the expected likelihood $\mathbb{E}[\log p(\mathrm{x}|\mathrm{z})]$ encourages preference of latent variable configurations that better explain the observed data. The Eq. 21 may be rewritten as follows:

$$\log p(\mathrm{x}) = \mathrm{KL}(q(\mathrm{z})||p(\mathrm{z}|\mathrm{x})) + \mathrm{ELBO(q)} \tag{22}$$

The value of the left hand side (log-evidence) is constant and the KL(.) $\geq$ 0. As a result, ELBO(q) is the lower-bound of evidence.

There are numerous extensions and proposed approaches for variational inference in the literature such as Expectation Propagation (EP) [203] and stochastic gradient optimization [197]. For a more detailed and comprehensive review, the readers are encouraged to consult [68], [202].

D. APPLICATIONS

Bayesian models and Variational Inference techniques have demonstrated their versatility and effectiveness in various domains. The Bayesian inference plays a crucial role in calculations across disciplines, including personalized advertising recommendation systems in healthcare applications [197], research in astronomy [204], and search engines [205].

In the fields of Physics and Chemistry, these models are utilized to simulate physical objects such as fluids, rigid solids, and deformable materials [157].

By leveraging Bayesian models, researchers have made significant strides in computer vision tasks, particularly in the field of semantic segmentation [206].

The impact of Bayesian models is also evident in the domain of robotics. Its application has been pivotal in tasks such as robot perception, enabling machines to understand and interpret their environment accurately. Additionally, it has facilitated advancements in motion planning, allowing robots to navigate complex and dynamic environments [207], [208], [209].

VI. CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks (CNN) are the prevalent approach in extracting features from image data. Though several variants of CNNs have been proposed, they all share pretty much the same basic components: convolution, pooling, and fully-connected layers.

1) CONVOLUTION LAYER

Extracts features from a given input layer and stores them on several feature maps which make up the higher layer. Each convolution layer has several feature extractors called kernels(filters) that each of them correspond to a single feature map. Every single neuron of the feature map corresponds to a group of neighboring neurons from the input layer referred to by neuron's receptive field. Each kernel is used to calculate the convolution over all of the possible receptive fields of the input layer. The convolution value is then passed through a non-linear activation function such as *tanh*(.) or *sigmoid* or *ReLU* [210] to add non-linearity to the representation.

The feature value $z^{l}_{i,j,k}$, at location (i,j) of the $k-$th feature map of layer l can be calculated as:

$$z^{l}_{i,j,k} = w^{k}_{l} \odot x^{l}_{i,j} + b^{l}_{k} \tag{23}$$

where $x^{l}_{i,j}$ represents the receptive field of neuron $z^{l}_{i,j,k}$ in the input layer. And the symbol $\odot$ represents the discrete convolution i.e. the sum of elements of Hadamard product(element-wise) of the two matrices. The activation of each feature can be obtained from the Eq. 24 [211].

$$a^{l}_{i,j,k} = f(z^{l}_{i,j,k}) \tag{24}$$

where f refers to the activation function.

2) POOLING LAYER

The next step after convolution is reducing the size of the shared feature map. Various pooling operations are proposed; though the average pooling and max pooling are typically used [212]. The pooling operation can be represented mathematically as:

$$y^l_{m,n,k} = pool(\{\forall a^l_k \in V_{m,n}\}) \tag{25}$$

The neuron $y^l_{m,n,k}$ at location (m,n) of the $k-$th pooled feature map of layer l would be calculated from a set of neighboring neurons $V_{m,n}$ on the convolution feature map passed through the pooling function *pool*. One of the main advantages of the convolutions over other architectures are having the shift-invariance. A small displacement(rotation, translation) of the input wouldn't change the output dramatically. The main characteristic comes from sharing the kernels and pooling layers.

3) FULLY-CONNECTED LAYER

After several convolutional and pooling layers, that perform as feature extractors, typically a few fully-connected layers (MLPs as discussed in section II) are added in order to perform high-level reasoning given the extracted features [213]. For classification tasks the fully-connected network takes in all the neurons from the previous layers as input and provide an output of classes followed by a *softmax* function. Given a dataset of N pairs of inputs and outputs $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, and the weights and biases of the whole network denoted by θ, the total classification error of the network can be calculated by the following loss function:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \ell(\theta; y_i, \hat{y}_i) \tag{26}$$

where $\hat{y}_i$ denotes the class label calculated by the network and y_i is the true value of the class label. The training process of the network would involve the global minimization of loss function. The model's parameters θ can be updated using Stochastic Gradient Descent (SGD) [214] that is a common method for training CNNs, although various other optimization methods and loss functions have been proposed. Additionally, the fully-connected layers and the final layer of CNN may be replaced by some other types of networks or models. Also, numerous variants of CNNs and additional components are proposed in the literature [215], [216], [217], [218]. The readers can consult [212] for a comprehensive introduction to the CNNs.

Fig 6 provides an abstract depiction of the structure and components of a CNN as described above.

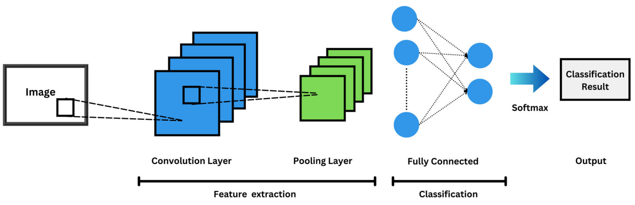


FIGURE 6. Abstract structure of convolutional neural networks.

A. APPLICATIONS

CNN and its extensions can be seen in almost all of the state of the art methods of deep representation learning. It has demonstrated competitive capabilities in numerous supervised and unsupervised tasks on 2D/3D images and point cloud data [27] such as image retrieval [219], segmentation [23], [220], [221], registration [24], object detection [222], [223], and data augmentation [85], [224]. It has also been applied to sequential data to extract longitudinal patterns of signals [225], [226], i.e. applicable to 1D data as well. CNNs, also empower reinforcement learning algorithms [227] and may be utilized to analyze graph data [127], [228] as will be discussed in section IV.

Fig 7 showcasing the evolution of CNN models over the years. The progression highlights key milestones and breakthrough models that have significantly impacted deep representation learning.

VII. WORD REPRESENTATION LEARNING

Representing words numerically is a crucial component of natural language processing, as it forms the foundation for employing Artificial Neural Networks (ANN) in NLP tasks. The simplest way to represent a word in a computer-readable format is through a one-hot vector, where each word is assigned a dimension in a vector equal to the size of the vocabulary [234]. The main flaw in this approach is that it neglects the semantic relatedness between words. Vector space model [235] is one of the first methods of representing words mathematically that made it possible to calculate similarity between documents in the field of Information Retrieval.

More recently, word embeddings have emerged as a method for learning low-dimensional vector representations of words from text corpora, capturing the semantic and contextual information of words [236], [237].

Word embeddings will not only present the semantic meanings of the word but also may show the word-context information. We can view language model as a tool which represents a sequence of words' probability distribution based on training data [238]. Language models, such as those based on neural networks, learn the joint probability function of word sequences in a corpus [238], [239]. One of the earliest neural language models proposed by Bengio et al. [239] aimed to address the challenge of learning the joint probability function for word sequences due to the curse of dimensionality. They introduced a method that estimates distributed representations for each word, allowing the model to learn about exponentially many semantically neighboring sentences. In their model, the learned distributed encoding of each word is fed into the last unit (softmax) to predict the probabilities for incoming words. Other research works have also explored word embeddings in prediction models [240], [241], [242]. Later on, as one of the most popular approaches, Word2vec [43] has proposed two methods: Continuous Bag of Words (CBOW) and Skip-Gram (SG) [44], [243].

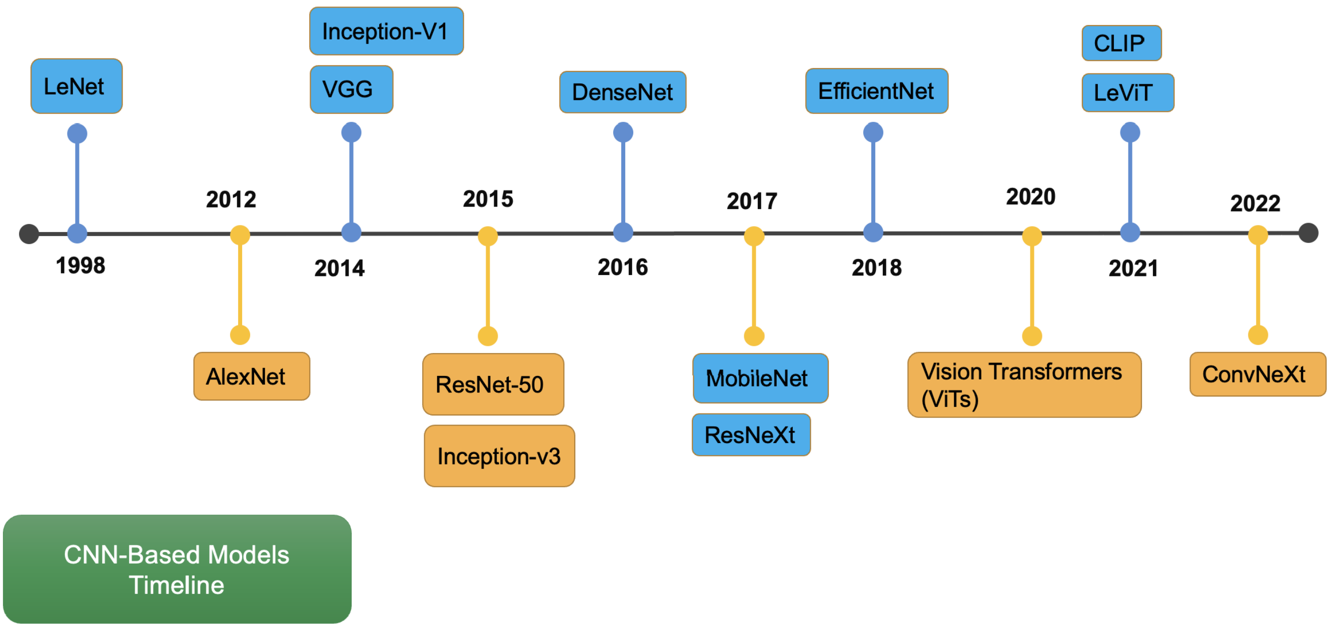


FIGURE 7. A general timeline for some of the most important CNN-Based models in history. [229], [230], [231], [232], [233].

In CBOW, the model predicts the middle word given the distributed representations of its context (or surrounding words), while SG predicts the context words given the center word. To address the computational burden of these methods, negative sampling was proposed [244]. Negative sampling involves using a random subsample of frequent words instead of the entire training set to calculate the denominator of the softmax equation.

The cross-entropy loss function is $H(p, q) = -\sum_{x \in X} p(x) log^{q(x)}$ in this model.

$$P(\text{ out } \mid \text{ center }) = \frac{\exp\left(u_{\text{out}}^{T} v_{\text{center}}\right)}{\sum_{w \in V} \exp\left(u_{w}^{T} v_{\text{center}}\right)} \tag{27}$$

As another influential approach Glove [44] captures the difference between a pair of words as ratio of co-occurence probabilities for target words with selected context word. Later on, FastText [245] built upon the "Glove" and "Word2Vec" to mitigate their shortcoming in handling out-of-vocabulary(OOV) [246], [247]. FastText builds word representations by considering subword information. It represents each word as a bag of character n-grams and utilizes these subword representations to generate word embeddings. This approach addresses the challenge of handling OOV words, as it can capture the meaning of unseen words based on their character compositions [245].

A. APPLICATIONS

In language modeling, word embeddings are used to capture the semantic meaning and contextual information of words. Using these embeddings, the model is capable of performing tasks such as machine translation [45], [248], sentiment analysis [249], [250], and named entity recognition [251], [252].

Word embeddings are being used in information retrieval applications, making it possible to calculate word similarity accurately and rank documents more effectively. Recent advances in this area include models such as SBERT (Sentence-BERT) [253] and CLIP (Contrastive Language-Image Pretraining) [254], due to their ability to enhance semantic understanding and cross-modal retrieval using contextual embeddings.

In question answering, word embeddings are a key component in models like GPT (Generative Pre-trained Transformer) [255] and T5 (Text-to-Text Transfer Transformer) [256], incorporating language generation capabilities and doing well on benchmark datasets.

Moreover, word embeddings are used in text classification tasks, including sentiment analysis, topic classification. Recent models such as ULMFiT (Universal Language Model Fine-tuning) [257] and RoBERTa (Robustly Optimized BERT Pretraining Approach) [258] show superior results by fine-tuning large pretrained language models.

In addition, word embeddings are employed in document summarization [259], [260], document clustering [261], [262], and text generation [263], [264].

VIII. SEQUENTIAL REPRESENTATION LEARNING

In many real-world applications, data often exhibit a sequential nature, where the order of elements in a sequence holds valuable information. Examples of such sequential data include sentences in NLP tasks [265] and medical records in healthcare research [266], [267]. In order to effectively capture and represent the underlying patterns in these sequences, it is crucial to employ architectures that can handle inputs of varying lengths and capture the dependencies between data points.

Recurrent Neural Networks (RNNs) [57] have emerged as a popular choice for sequential representation learning due to their ability to address these requirements. RNNs are designed to process sequences by sharing parameters across different steps [47], allowing them to handle inputs with varying lengths. This characteristic enables RNNs to handle sequential data more effectively than traditional feedforward neural networks.

RNNs capture dependencies between data points. These models are capable of considering the historical context when processing each element in a sequence by maintaining an internal state or memory. The memory component of RNNs is essential for capturing the sequential patterns present in data, as it enables them to model relationships and dependencies between elements over time.

A. RECURRENT NEURAL NETWORK

The general architecture of a recurrent neural network is composed of cells with hidden states. In mathematical terms, hidden units h_t store the state of the model that depends on the state at previous time step h_{t-1} and the input of the current time step x_t:

$$h_{(t)} = f_a(Wh_{(t-1)}, Ux_{(t)} + b) \tag{28}$$

where the matrices U, W are weight matrices and b is bias vector, and f_a represents the activation function. The same set of model parameters is used for calculation of h_t for any of the elements in a sequence of inputs $(x_1, x_2, \ldots, x_n)$. In this way, the parameters are shared across the input elements. For supervised tasks such as classification, the hidden unit h_t is mapped to the output variables y_t via the weight matrix V:

$$\hat{y}_t = \text{softmax}(Vh_t + c) \tag{29}$$

c is the bias vector. Due to the recursive nature of Eq. (28), the unfolded computational graph for a given input sequence, can be displayed as a regular neural network.

RNNs can be trained by back-propagation [47] as if it is calculated for the unfolded computational graph. Two of the most important problems with RNNs are vanishing and exploding gradients. The longer the input sequence, the more gradient values are multiplied together, which may cause it to converge to zero, or exponentially gets large. Either way, the RNN fails to learn anything. Various methods have been proposed to facilitate training RNNs on longer sequences. For instance, skip connections [268] let the information flow from a farther past to the present state. Another method, is incorporation of leaky units [269] in order to keep track of the older states of hidden layers by linear self-connections. Nonetheless, the problem of learning long-term dependencies is yet to be resolved completely.

B. LONG SHORT-TERM MEMORY (LSTM)

The most prominent architecture for learning from sequential data, is Long Short-Term Memory that combines various strategies for handling longer dependencies [270]. In LSTM, tuning of the hyper-parameters is a part of the learning procedure. The general architecture of an LSTM cell contains separate gates that control the information flow across the time steps of sequences:

1) INPUT GATE

Controls whether the input is accumulated into the hidden state.

$$i_t = \sigma(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + b_i) \tag{30}$$

where:

i_t is the input gate at time step t,
x_t is the current input at time step t,
h_{t-1} is the previous hidden state at time step $t-1$,
W_{xi} is the weight matrix for the input connections,
W_{hi} is the weight matrix for the hidden state connections,
b_i is the bias term for the input gate,
σ is the sigmoid activation function.

The sigmoid activation function is defined as:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{31}$$

The input gate i_t determines the relevance of the current input x_t and its impact on updating the hidden state h_t. A value close to 0 for i_t indicates that the current input is ignored, while a value close to 1 indicates that the current input has a significant impact on the hidden state update.

By incorporating the input gate, LSTM networks can selectively accumulate relevant information from the current input and previous hidden state, enabling them to capture long-term dependencies and effectively learn from sequential data.

2) FORGET GATE

Controls the amount of effect that the previous state has on the current state. Whenever this gate lets information flow in completely, it acts as a skip-connection [268]. Otherwise, similar to leaky units, it keeps track of previous hidden states with a linear coefficient.

$$f_t = \sigma(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f) \tag{32}$$

where:

f_t is the forget gate at time step t,
x_t is the current input at time step t,
h_{t-1} is the previous hidden state at time step $t-1$,
W_{xf} is the weight matrix for the input connections,
W_{hf} is the weight matrix for the hidden state connections,

b_f is the bias term for the forget gate,

σ is the sigmoid activation function.

3) OUTPUT GATE

The output gate o_t controls whether the output of the LSTM cell should be stopped or allowed to propagate further. It regulates the flow of information from the hidden state to the output of the LSTM cell.

$$o_t = \sigma(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + b_o) \tag{33}$$

where:

o_t is the output gate at time step t,

x_t is the current input at time step t,

h_{t-1} is the previous hidden state at time step $t-1$,

W_{xo} is the weight matrix for the input connections,

W_{ho} is the weight matrix for the hidden state connections,

b_o is the bias term for the output gate,

σ is the sigmoid activation function.

C. GATED RECURRENT UNIT (GRU)

Another variant of recurrent neural network that addresses long-term dependencies through gating mechanisms is the Gate Recurrent Unit (GRU) [271]. The GRU architecture is simpler compared to LSTM, resulting in fewer parameters within a GRU cell. GRU cells consist of two types of gates:

1) UPDATE GATE

Controls the weights of interpolation of the current state and the candidate state in order to update the hidden state.

$$z_t = \sigma(W_{xz} \cdot x_t + W_{hz} \cdot h_{t-1} + b_z) \tag{34}$$

where:

z_t is the update gate at time step t,

x_t is the current input at time step t,

h_{t-1} is the previous hidden state at time step $t-1$,

W_{xz} is the weight matrix for the input connections,

W_{hz} is the weight matrix for the hidden state connections,

b_z is the bias term for the update gate,

σ is the sigmoid activation function.

The update gate z_t controls the weights used for interpolating between the current state and the candidate state in order to update the hidden state. A value close to 0 for z_t indicates that the current state is mostly updated based on the candidate state, while a value close to 1 indicates that the current state is mostly retained from the previous hidden state.

By incorporating the update gate, GRU networks can selectively update and retain relevant information from both the current input and the previous hidden state, enabling them to capture and utilize long-term dependencies effectively.

2) RESET GATE

Makes the hidden state forget the past dependencies.

$$r_t = \sigma(W_{xr} \cdot x_t + W_{hr} \cdot h_{t-1} + b_r) \tag{35}$$

where:

r_t is the reset gate at time step t,

x_t is the current input at time step t,

h_{t-1} is the previous hidden state at time step $t-1$,

W_{xr} is the weight matrix for the input connections,

W_{hr} is the weight matrix for the hidden state connections,

b_r is the bias term for the reset gate,

σ is the sigmoid activation function.

The reset gate r_t controls the degree to which the hidden state should forget past dependencies. By selectively resetting the hidden state based on the reset gate, the GRU can adjust the influence of previous states on the current state.

All the gates in variants of the gated recurrent neural networks, are controlled by linear neural networks. Training the recurrent neural network as a whole, also trains and updates the weights of the gate controller networks.

D. APPLICATIONS

There are myriad problems and use cases for the RNNs. Instances are: analysis and embedding of texts and medical reports to be combined with medical images [272], real-time denoising of medical video [273], classification of electroenephalogram (EEG) data [274], generating captions for images [28], [29], [30], [275], biomedical image segmentation [276], semantic segmentation of unstructured 3D point clouds [277], [278]. Other examples of RNN applications are predictive maintenance [279], prediction and classification of ICU outcomes [31], [280], [281].

In table 1, we have presented a summary of a few of the most important applications of RNNs, LSTMs, and GRUs with or without attention.

IX. ATTENTION-BASED MODELS

In deep learning, attention-based models have emerged as a powerful paradigm, providing breakthroughs to various domains by focusing selectively on relevant information. These models have gained significant popularity in NLP, where they have revolutionized tasks such as machine translation, sentiment analysis, and text summarization. By dynamically assigning different weights to different parts of the input sequence, attention mechanisms allow the models to capture dependencies and relationships effectively [45], [336]. As a result, not only are the predictions more accurate but they are also more interpretable since the important aspects of the input are highlighted [337]. Attention-based encoder-decoder models are innovated to solve the shortcomings of RNN, LSTM, and GRU, which were fairly known as the state-of-the-art approacehes.

TABLE 1. General usecases for RNN, LSTM, and GRU with or without attention in different areas.

Natural Language Processing	Information Extraction	Named Entity Recognition	[282] [283] [284] [285] [286]
		Relationship Extraction	[287] [288]
		Event Extraction	[289] [290] [291]
		Coreference resolution	[292] [293]
		Syntactic Analysis	[294] [295]
	Text Classification		[296] [297]
	Speech recognition		[298] [299]
	Information Retrieval		[300] [301] [302]
	Machine Translation		[303] [304] [305] [306]
	Question Generation		[307] [308]
	Text Summarization		[309] [310]
	Spell chek and correlation		[311]
Computer Vision	Video Captioning		[312] [313] [314]
	Image Captioning		[315] [316] [317] [318] [319]
	Image Classification		[320] [321] [322]
	Obejct Detection		[323] [324]
	Image generation		[325]
	Object tracking		[326] [327]
	Visual question answering		[328] [329]
Graph Based Sytems	Node Classification		[330]
	Graph Classification		[331]
	Graph representaion learning		[332]
	Graph to sequence		[333]
Robotics	Path Planning		[334]
	Motion Planning		[335]

A. LEARNING TO ALIGN AND TRANSLATE

The first Encode-decoder model with an attention mechanism was proposed by [34] in 2015 as a novel architecture to improve the performance of neural machine translation models. The key contribution of Bahdanau et al. [34] was the introduction of an attention mechanism in the decoder part, which involved calculating the weighted sum of the hidden states of the input. Unlike the basic encoder-decoder model that utilizes a single fixed-length vector, Bahdanau et al. extended this approach by encoding variable-length vectors. During the decoding process, the attention mechanism allows for selective focus on relevant parts of the input.

As illustrated in the Fig 8, the c_t is what is added to this model as attention. So, the $s_t = f(s_{t-1}, y_{t-1}, c_t)$ output of the decoder will be based on the $c_t = \sum \alpha_{tt'} h_t$ which is defined as weighted attention where

$$\alpha_{tt'} = \frac{exp(e_{tt'})}{\sum\limits_{T} exp(e_{tT})} \tag{36}$$

and $e_{tt'}$ is called alignment score. In other words, $\alpha_{tt'}$ is called amount of attention y^t (the output in time step t) pay to $x^{t'}$. There are different options to calculate the alignment score, and it is one of the parameters can be trained, but in general, it is based on $align(h_i, s_0)$.

The model proposed by Bahdanau et al. introduced a groundbreaking approach that served as a source of inspiration for subsequent state-of-the-art models. Nonetheless, it exhibited limitations inherent to conventional encoder-decoder recurrent models and did not possess parallel computing capabilities.

B. TRANSFORMERS

Despite the notable contribution made by Bahdanau et al. [34] in introducing attention for RNN-based models, this model is still challenging to train because of the long gradient path, specifically for long data sequences. The introduction of Transformers [45] brought about a significant breakthrough by making use of the power of attention within the context of the input sequence. Transformers overcome the two main drawbacks of LSTM-based models: 1. parallel computing capability and 2. the problem of long gradient paths. The model architecture is based on a stack of multiple encoder-decoder layers, each sharing the same structure. The input first undergoes an "input embedding" layer to transform one-hot token representations into word vectors. After positional encoding, the result is fed into the encoder. The core component of the encoder and decoder blocks is a multi-headed self-attention mechanism (Q, K, V), followed by point-wise feed-forward networks.

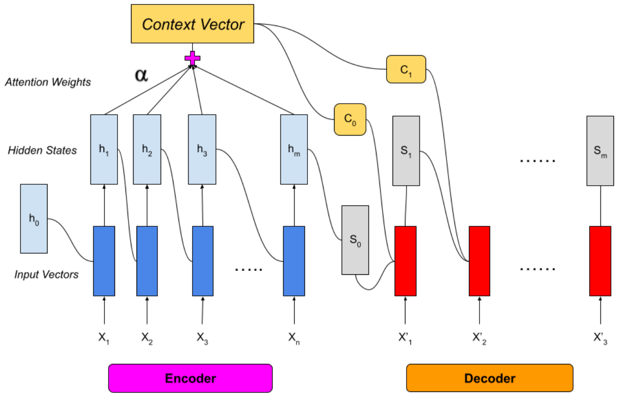


FIGURE 8. Based on the current target state h_t and all source states h_s, the model determines an alignment weight vector at time step t. A global context vector c_t is then computed as the weighted average over all the source states [34].

Self-attention structure:
To estimate the relevance of each element in a given series to all others, self-alignment is employed. The process involves the following steps:
Step 1: Randomly generate W_Q, W_K, W_V weights and calculate:
$Queries = X_k$ (embedding) $*W_Q$
$Key = X_k$ (embedding) $*W_K$
$Value = X_k$ (embedding) $*W_V$
(Where $X \in \mathbb{R}^{T\times d_m}$, $W^Q \in \mathbb{R}^{D\times D_Q}$,$W^K \in \mathbb{R}^{D\times D_K}$, $W^V \in \mathbb{R}^{D\times D_V}$)
Step 2: Calculate the z-score for each input by applying row-wise softmax on the scores obtained from the pairwise multiplication of queries and keys:

$$Z(Q, K, V) = Softmax(\frac{Q.K^T}{\sqrt{d_K}}).V \tag{37}$$

Next, concatenate the z-scores, initialize a new weight matrix, and multiply it by the z-scores. Finally, feed the result to a fully connected neural network (FCNN).

One common operation is to apply another set of feed-forward layers to the Z scores, often referred to as the ''point-wise feed-forward network'' (FFN). This step allows for additional nonlinear transformations and feature extraction.

After the FFN, the outputs can be passed to subsequent layers of the transformer, which may involve stacking multiple encoder-decoder layers or performing additional attention mechanisms. This hierarchical structure enables capturing complex dependencies and relationships among the input elements.

C. EXTRA LARGE TRANSFORMERS

Transformers have become indispensable to the modern deep learning stack by significantly impacting several fields. This has made it the center of focus and caused a overwhelming number of model variants proposing basic enhancements to mitigate a widely known concern with self-attention: its quadratic time and memory complexity [338]. These two drawbacks can pose significant challenges to model scalability in many settings. To overcome this limitation, researchers have explored various approaches, which can be categorized in several ways [339]. Some of these approaches include:

1) RECURRENCE

One of the most well-known extensions to the vanilla Transformer model is Transformer-XL [340]. It employs a segment-level recurrence mechanism that connects multiple adjacent blocks. This model introduces two key ideas. Firstly, by using segment-level recurrence, hidden states from the previous batch can be cached and reused. Secondly, it introduces a novel positional encoding scheme that enables temporal coherence.

As an extension to the block-wise approach, Transformer-XL splits the input into small non-overlapping subsequences known as blocks [341]. Although it exhibits impressive performance compared to the vanilla transformer, this model lacks the ability to maintain long-term dependencies and discards past activations as it progresses through the blocks. Specifically, Transformer-XL propagates gradients across the current segment, caches them, processes the second segment using the memory from the first segment (without gradients for the first segment), moves on to the third segment, and discards the gradient information from the first window. Consequently, this can be seen as a form of truncated back-propagation through time (BPTT).

The distinctive aspect of this model, which sets it apart from others, lies in its relative positional encoding scheme that ensures temporal coherence. The relative positional encoding encodes distances on edges rather than nodes. While previous work on relative positional encoding existed [342], Transformer-XL introduces two additional features: a global content and location bias, and the replacement of trainable positional embeddings with sinusoid embeddings. Their results demonstrate that Transformer-XL outperforms vanilla Transformers even without the use of a recurrence mechanism. Compressive Transformer [343] is another model which can be classified as a recurrence approach.

2) REDUCED DIMENSSIONS/ KERNELS/ LOW- RANK METHODS

The ''Transformers are RNNs'' [344] introduces the concept of utilizing a kernel function $\phi(X) = elu(x_i) + 1$ instead of softmax to map the attention matrix to its approximation. The function is applied on 'Keys' and 'Queries,' lowering their dimension. $Q_{(N\times D)}.k^T{}_{(D\times N)}$ and avoid computing the $N \times N$ matrix. Linformer [345] and Synthesizer [346] are other models based on this approach.

3) SPARSE ATTENTION

The Longformer model, introduced in the paper [347], achieves linear complexity of O(n) by employing a global memory technique and drawing analogies to convolutional neural networks (CNNs). To reduce dimensions, a combination of sliding window and global attention techniques is applied to each Query. Longformer, along with Bigbird [348], ETC [349], and SWIN Transformer [350], falls into the same category of models that utilize sparse attention techniques. On the other hand, Image Transformer [343] and Axial Transformer [351] are other examples of extended sparse attention works that primarily focus on vision data.

D. PRE-TRAINED MODELS

Pre-trained models are neural networks that have been trained on large-scale corpora and are designed to be capable of transfer and fine-tuning for various downstream tasks. Word embeddings as a base that enabled us to utilize machine learning for processing natural language can be viewed as pioneers of widely used pre-rained representations. Word2vec [43], and Glove [44], which we discussed earlier, are among the most famous models learning a constant embedding for each word in vector space. In what follows, we will try to pinpoint the most famous and important pre-trained models that retain contextual representations, which those mentioned earlier are incapable of.

Reference [352] from 2015 is one of the earliest instances of supervised sequence learning using LSTMs that pre-trained an entire language model for use in various classification tasks. ELMO [353], a deep contextualized word representation, is analogous to the earlier one; however, it is bidirectional. CoVE [354] is another recurrent model in this category that has demonstrated good performance. GPT1 [355] is a Transformer-based pre-trained model that was trained on a large book corpus dataset to learn a universal representation, enabling transfer with minimal adaptation. "Deep Bidirectional Transformers for Language Understanding " - BERT [46], a well-known turning point in the NLP area (perhaps also the entire ML stack), trains left context and right context at the same time rather than doing individually and concatenating at the end. Since BERT accesses information from both directions, it masks out K% of the input sequence to prevent the model from simply copying the input. XLNet [356], RoBERTa [357], ERNIE [358], and ELECTRA [359] are variations of the BERT model."

All the mentioned models have proven effective in NLP studies. These successful observations within the NLP space inspired researchers to apply a similar approach to other domains. [345] has shown that pre-trained models' success is not limited to transformer-based ones. They have demonstrated their pre-trained convolution seq2seq model can beat pre-trained Transformers in machine translation, language modeling, and abstractive summarization. Vision Transformer(ViT) [360] is one of all the foremost recent pre-trained models that helps transfer learning in image classification tasks. This model has shown outstanding results in training a pure transformer applied directly to sequences of image patches. ResNet50 [361], a pretrained CNN-based model which allows training networks with up to 1000 layers. ResNet50 consists of a succession of convolutional layers with different kernel settings. References [362], [363], [364], and [365] are all trained on the vast number of datasets for various image classification transfer learning usage categories.

While these models have shown excellent results, the range of pre-trained models is not restricted to the mentioned. One is to precisely study and examine different models and approaches to search out their dataset's best and most efficient model.

Language models (LMs) are computational models with the capacity to comprehend and generate human language. Language models have the impressive ability to calculate the likelihood of word sequences or generate new text based on given input [366]. Researchers find that scaling pretrained-language models such as BERT can lead to an improved model capacity [367]. Recent years have witnessed incredible progress in pre-training of large language models (LLMs) like GPT-4 [368], PaLM2 [369], LLaMA 2 [370], which have proven extremely effective for transfer learning in NLP. While concerns remain around bias, safety, and environmental impact [371], [372], the application [373], [374] of LLMs continues to rapidly advance. Though the eventual impacts remain speculative, LLMs have already catalyzed a revolution in representation learning.

E. RECURRENT CELL TO RESCUE

With the availability of GPUs as a powerful computation tool in the machine learning toolkit, LSTM (Long Short-Term Memory) emerged as a practical approach in numerous sequence-based machine learning models. With the introduction of word embeddings in 2013, LSTM and other RNN-based models have been widely dominant in sequence learning problems. After presenting transformers with their *All-to-all comparison* mechanism and their performance on transfer learning tasks, they became the SOTA model and dominated the deep learning space.

1) RNN VS. TRANSFORMER

While transformers can grasp the context and be used more efficiently for transferring knowledge to tasks with limited supervision by pre-trained models, these benefits come with quadratic memory and time complexity of $O(N^2)$ [344]. Most of the current pre-trained transformer-based models do only accept 512 numbers of the input sequence. IndRNN model [375] has shown the ability to process sequences over 5000 time steps. The *Legendre Memory Unit* [376] is based on recurrent architecture and can be implemented by a spiking neural network [377], which can maintain the dependencies across 100,000 time steps. Apart from computation cost, [378] by Facebook shows that the accuracy gap between Bert-based [46] pre-trained models versus vanilla LSTM for a massive corpus of data is less than 1%. Henceforth, a competitive accuracy result is achievable by training a simple LSTM when many training examples are available. They also show that reusing the pre-trained token embeddings learned in BERT can significantly improve the LSTM model's accuracy. Reference [379] shows that standard transformers are not as efficient as RNN-based models for reinforcement learning tasks. [A-13] has investigated the performance of Transformer and RNN in speech application and shows both have the same performance in text-to-speech tasks and slightly better performance by Transformer in the automatic speech recognition task.

On the other hand, [379] shows that their attention-based model can outperform the state-of-the-art in terms of precision, time, and memory requirements for satellite image time series. Reference [380] has compared LSTM performance with transformers in their proposed Frozen Pretrained Transformer model as part of their paper. They evaluate a diverse set of classification tasks to investigate the ability to learn representations for predictive learning across various modalities and show that transformers perform better. Reference [381] has proposed an improved-Transformer-based comment generation method that extracts both the text and structure information from the program code. They show that their model outperforms the regular Transformer and classical recurrent models. Reference [382] is a transformer-based transcoder network for end-to-end speech-to-speech translation that surpasses all the SOTA models in natural speech-to-speech translation tasks. Reference [383] has introduced an AttentiveConvolutional Transformer which takes advantage of Transformer and CNN for text classification tasks. Their experiment reveals that ACT can outperform RNN-based models evaluated on three different datasets.

2) COMBINING RECURRENT AND ATTENTION

R-Transformer [384] inherits the Transformers' architecture and is adding what they call "Local RNN" to capture sequential information in data. The main improvement proposed is defining a sequence window to capture the sequential information and sliding the Local RNN over the whole time series to get the global sequential information [385]. This approach is similar to 1-D CNN; however, CNN ignores the sequential information of positions. Also, the Transformer's positional embedding that mitigates this problem is limited to a specific sequence length. Henceforth, they have proposed a 'Local RNN' model that can efficiently do parallel computation of several short sequences to capture the local structure's global long-term dependency by applying a multi-head attention mechanism. This model has replaced the Transformers' position embeddings with multiple local RNNs, which can outperform the simple recurrent approaches such as GRU, LSTM, convolutional [386], and regular Transformer. Reference [381] has proposed a modified LSTM cell to mitigate the similarity between hidden representations learned by LSTM across different time steps in which attention weights cannot carry much meaning. They propose two approaches: first, by orthogonalizing the hidden state at time t with the mean of previous states, they ensure low conicity between hidden states. The second is a loss function in which a joint probability for the ground truth class and input sentences is used and also minimizes the conicity between the hidden states. These mutations provide a more precise ranking of hidden states, are better indicative of words important for the model's predictions, and correlate better with gradient-based attribution methods.

While we have mentioned works in which LSTM outperforms Transformers and vice versa, one should study the proper approach based on the dataset, accessible computation resources, and so forth.

F. APPLICATIONS

A wide range of transformer models and variants have been applied in various domains, demonstrating their versatility and effectiveness. We discuss some of these models' notable applications.

In machine translation, models like Transformer [45]have surpassed traditional recurrent neural network-based models, achieving state-of-the-art performance. Several models have demonstrated the ability to understand context and generate accurate answers for question answering tasks, including BERT [387] and GPT4 [368]. Various transformer models have demonstrated excellent performance in classifying sentiment in text, such as BERT and XLNet. Moreover, transformer-based models, such as BART [388] and T5 [256], have been successfully applied to the summarization of lengthy documents and articles.

In the field of computer vision, transformers have made significant contributions. In image classification, Vision Transformer (ViT) [389] applies transformers and achieves competitive performance against convolutional neural networks (CNNs) on benchmark datasets. For object detection, DETR (DEtection TRansformer) [390] is a transformer-based model that directly predicts object bounding boxes and class labels. The use of transformer models has also been applied to image generation tasks, such as the VQ-VAE-2 model [391], which combines transformers with vector quantization to generate high-quality images. Additionally, transformers have been used in generative models such as DALL-E [392], which enables the generation of images from textual descriptions.

In robotics, transformers allow capturing long-range dependencies and global context, leading to improved perception capabilities [393], [394]. In robot planning [395], [396], transformers have been utilized for motion planning and task planning, leveraging their ability to capture complex spatial and temporal dependencies [397], [398]. Transformers have also been employed in robot control, learning policies and generating appropriate actions [399], [400], [401].

As illustrated in Fig 9, we have tried to show some of the best and most famous sequence to sequence models, including the recent transformer-based models with different applications.

X. TRANSFER LEARNING

Many of the advancements in machine learning techniques make a huge improvement over the existing benchmarks. There are, however, some assumptions and challenges that make it difficult to apply the methods to real-world situations. In many cases, the assumption is that the trained model will be tested on the same feature distribution as the training stage. This assumption usually does not hold as the environment changes. In addition, many promising results are obtained by training models with large datasets. These pre-requisites makes it very challenging to adapt to many different tasks. For many applications, acquiring large amounts of data can be costly, time-consuming or even impossible. The absence of data for specific tasks may not be the only challenge; Massive data collection poses a huge privacy problem in many healthcare and medical applications [403]. In other cases, annotating the data would require an expert and could be expensive, such as low-resource languages [404].

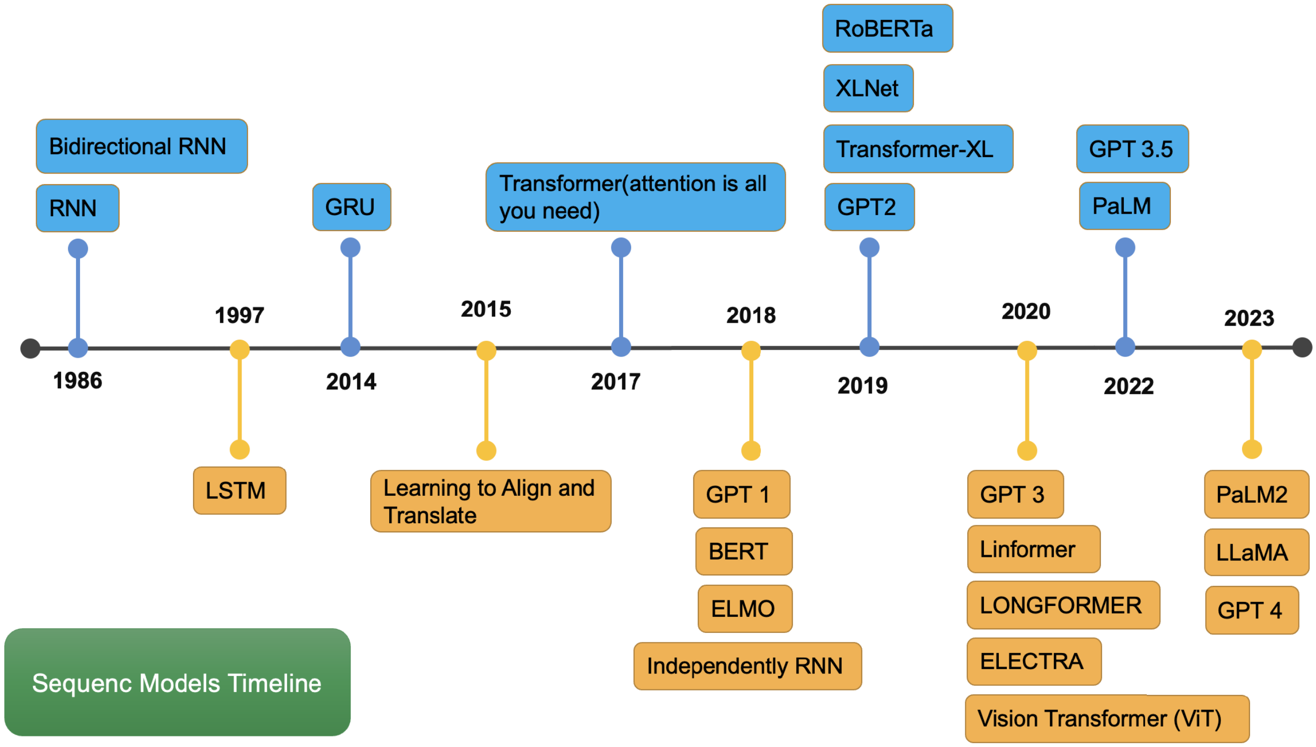


FIGURE 9. A general timeline for some of the most important sequence models in history, including pretrained ones. [402].

Transfer learning aims to alleviate the mentioned problems. Generally, transfer learning refers to when a learner wants to improve the performance on the target domain by transferring knowledge from the source domain. It derives from the human intuitive ability to share knowledge across different domains and tasks. For example, learning a language might help you learn the second one if there's some relation in between. The term itself is very general and there have been many extensions to it in recent years.

Transfer learning enables machine learning models to be retrained and reuse their previously learned knowledge. A general definition of the problem is divided into two components: *Domain* and *Task* [405].

The **Domain** is defined as $D = \{\chi, P(X)\}$, with χ representing the feature space, and $P(X)$ for each $X = \{x_1, \ldots, x_n\} \in \chi$ denoting the marginal probability over the feature space. In cases where different domains are encountered, the source domain D_S and the target domain D_T can assume different feature space or marginal probability distributions [405].

Given a specific domain $D = \{\chi, P(X)\}$, **Task** is represented by $T = \{y, f(x)\}$, where y is the feature space and $f(x) = P(Y|X)$ denotes the function that can be learned from the training data to predict the target, in a supervised manner from the labeled data x_i, y_i, where $x_i \in X$ and $y_i \in Y$. In cases where no labels are considered for the data, as is the case for unsupervised algorithms, y can be a latent variable such as the cluster number, or a variable that is produced by an unsupervised algorithm (e.g., the reduced dimensions of the original data) [405]. In light of both the domain D and the task T being defined as tuples, four transfer learning scenarios can be arise.

The first scenario is when the source and target domain are different $X_s \neq Y_t$. A good example of this is, in the computer vision community, where the source task is an image of a humans, but the target task is an image of an objects. A similar example can be found in NLP when it comes to cross-lingual adaptation.

The second scenario happens when $P(X_s) \neq P(X_t)$, the marginal probability distributions of source and target domain are different. This scenario is generally known as domain adaptation. An example could a detection problem where the source and target has different kind of cars.

The third occurs when $Y_s \neq Y_t$, the label spaces between the two tasks are different. For example, consider a detection problem where the source task considers the detection of cars, while the target task considers animals.

The last is when $P(Y_s|X_s) \neq P(Y_t|X_t)$, the conditional probability distributions of the source and target tasks are different. The imbalancness of data between source and target tasks is a very common example.

A number of surveys have been conducted to categorize the available methods [405], [406], [407]. In [406] the available methods are categorized into three different sections, transductive, inductive, and unsupervised transfer learning. In [407] the available methods are categorized in more detail based on the data or model perspectives. Although this categorization could give some insights, there are many newer methods that cannot fit in those categories or belong to more than one, zero-shot transfer learning [408], reinforcement transfer learning [409], and online transfer learning [410] are among these methods.

Several sub-categories of transfer learning can be considered for each of these main categories based on the nature of the knowledge transfer. In the following subsection, the most prominent sub-categories of transfer learning will be discussed.

1) INSTANCE-BASED:

Despite differences in the source and the target domains, an instance based transfer learning, such as TrAdaBoost [411] or Bi-weighting Domain Adaptation (BIW) [412], adjusts the weights used for a subset of the source instances that are similar to the target domain, to predict the target instances. Since similarity of the selected source instance to those of the target domain play a crucial role in instance based transfer learning, a filter is used to remove dissimilar instances that would otherwise mislead the algorithm [405], [413], [414], [415], [416], [417].

2) FEATURE-BASED

Determining the common denominator between related tasks would allow for defining a representative feature that would apply to all domains and reduce differences between them. In this case, the common feature attempts to identify some partial overlap between the defined tasks. Having a representative feature among different tasks would also allow for a reduction in the overall error [405], [418], [419], [420]. While the source and target domains may have differences between them in their original data space, it is likely that the two would exhibit similarities in a transformed data space. Mapping-based deep transfer learning techniques, such as Transfer Component Analysis [421], create a union between the source and target domain instances by applying a mapping between the two and transforming them into a new data space based on their similarity so that they can be used for deep nets [417], [422].

3) NETWORK-BASED

Different models derived from related tasks can have many similarities and differences. Similar models often have knowledge about the model parameters or the behavior of hyperparameters shared between the individual models. In such cases, it is possible to create a learning algorithm that infers the model parameters and the distributions of its hyperparameters by examining the prior distributions of several other tasks [405], [423], [424]. Similar to the learning and inference process followed by the human brain, where the trained brain cells can ad-hoc to other brain cells in related tasks, the network-based approach aims at using an already trained neural network as part of a much more extensive deep neural network. This approach trains the subnet on its relevant domain data, and the resulting pre-trained network is transferred to a larger deep net [423], [425], [426]. A few examples of network-based deep transfer learning approaches include ResNet, VGG, Inception, and LeNet, which can extract a versatile set of features in the network's front layers [417].

4) RELATIONAL KNOWLEDGE-BASED

There are several instances, such as the social network data, where the data are not independent and identically distributed (IID). Relational domains allow for the handling of this scenario. In a relational domain, each entry is represented by multiple relations, not just a single identifier [427]. Unlike other methods discussed before, the cross-domain relational knowledge transfer algorithms, such as TAMAR, use the Markov Logic Networks (MLNs) to transfer the relational knowledge without requiring each data point to be IID [405], [428], [429].

5) ADVERSARIAL-BASED

Built on the strong foundation of the GANs, the adversarial-based approaches to transfer learning use a generator challenged by a discriminator to identify the transferable representations. A representation is considered transferable when it discriminates between the different components of the main learning task but does not discriminate the source domain from the target domain [417]. Most approaches use a single domain discriminator to align the source and target distributions, or use multiple discriminators to align subdomains [430], [431], [432], [433].

There is no unified approach where one can use Transfer learning. A very common transfer learning approach is when your target domain does not have sufficient training data. The model first pre-trains on the source data and then fine-tunes on the target data. Many well-known architectures in different communities are being used for related downstream tasks. In NLP, Bert [387], Word2vec [434], and ERNIE [435] are the famous models where a lot of downstream tasks can learn their specific related task with the shared knowledge backbone. Similarly, in the Vision community, Resnet [436], Vision Transformers [437] and ConvNeXt [438] could be used. Also in Speech, Wav2Vec [439], DeepSpeech [440] and HuBERT [441] are among famous models. It is important to note that there are different levels of fine-tuning. With enough data in the target domain, fine-tuning can also alter the entire backbone representation. In many applications, however, this can be done partially (the last few layers) or just for the task-specific heads without changing the backbone representation. The mentioned models these are expected to be general enough so that they can be used for many downstream processes. For instance, Resnet trained to classify images into 1000 different categories. If the model knows to classify cars, the knowledge can be used to detect airplanes or even a different task like semantic segmentation [442].

Other common ways of transfer are when the task is the same, but the domain changes. A helpful example might be applying the knowledge gained from the simulation data to real-world data [443]. In many applications like robotics, and computer vision, acquiring simulation data is very easy and straightforward.

The goal of transfer learning is to adapt the knowledge learned in one domain to another but closely related one. Many recent papers [444], [445] suggested that using pre-trained models and fine-tuning might not be the optimal approach. It is therefore important to know why and what to transfer. Another issue with pre-training solutions is the accumulation of parameters in each sub-task. These networks can have millions or even billions of parameters [445], it is then very impractical to fine-tune the backbone representation for every downstream task. Consider an application where the model should use the representation to do sentiment analysis along with entity recognition. If one wants to fine-tune a separate backbone for every downstream task, it would be very memory inefficient. Multitask Learning [446] aims to learn a shared representation for multiple related tasks which can be generalized across all tasks. As opposed to creating an instance of the backbone for each task, the representation is being shared across multiple tasks to improve efficiency [447].

There are also instances where transfer learning occurs in the feature space; instead of transferring the representation to the new task, a related but fixed representation can be used. In this case, the main representation remains intact and a small network learns the representation specific to the target task. Having common latent features acts as a bridge for knowledge transfer. In [448] the authors trained a lightweight CNN module on top of a generic representation called mid-level representation. In comparison to training a complex CNN module which also learns the representations, they achieved superior performance in terms of accuracy, efficiency, and generalization with the method.

A. APPLICATIONS

As mentioned, the use of transfer learning does not follow any conventional approach. Therefore, one should precisely study examples of how researchers can use transfer learning in their problems.

When it comes to medical applications, both privacy and expert labeling are key issues that make data availability difficult. In [449], [450], [451], and [452] the authors try to transfer the knowledge learned from the pre-trained models, Resnet [453] or ALexNet [454] trained on ImageNet, and transfer it for different tasks like Brain Tumor Segmentation, 3d medical image analysis and Alzheimer. Reference [455] found that due to the mismatch in learned features between the natural image, e.g., ImageNet, and medical images the transferring is ineffective and they propose an in-domain transferring approach to alleviate the issue.

There was a great deal of success with transfer learning in the field of NLP. Several reasons exist for this, but largely it is because it is easy to access the large corpus of texts. It is inherent for pre-trained models to generalize across many domains due to the millions or billions of text data that they are trained on [46], [456], [457], [458], [459]. These representations can be transferred in different areas such as sentiment analysis [460], [461], [462], Question Answering [463], [464], [465], and Cross-lingual knowledge transfer [460], [466], [467].

There are many speech recognition applications that are similar to NLP because of the nature of language. These applications are discussed in [468], [469], [470], and [471]

The progress of transfer learning in various domains have motivated researchers to adapt explored approaches for time series datasets [472]. For time-series tasks, transfer learning applications range from classification [473], anomaly detection [474], [475] to forecasting [476], [477].

Transfer learning has also been applied to various fields, ranging from text classification [478], [479], [480], spam email and intrusion detection [481], [482], [483], [484], recommendation systems [485], [486], [487], [488], [489], [490], [491], [492], [493], [494], biology and gene expression modeling [495], [496], to image and video concept classification [497], [498], [499], [500], human activity recognition [501], [502], [503]. While these fields are vastly different, they all benefit from the core functionalities of transfer learning, in applying the knowledge gained under controlled settings or similar domains, to new areas that may otherwise lack this knowledge.

B. CHALLENGES

Despite many successes in the area of transfer learning, some challenges still remain. This section discusses current challenges and possible improvements.

1) NEGATIVE TRANSFER

One of the earliest challenges discovered in transfer learning is called negative transfer learning. The term describes when the transfer results in a reduction in performance. One of the reasons could be the interference with previous knowledge [504] or the dissimilarity between the domains [444], [445] could be one of the reasons. There might be some cases where the transfer does not degrade, but doesn't make full use of its potential to obtain a representative feature. In [504] has been shown that contrastive pre-training on the same domain may be more effective than attempting to transfer knowledge from another domain. Similarly, in [505] the study was conducted to explore which tasks will gain from sharing knowledge and which will suffer from negative transfer and should be learned in a separate model. In [506] the authors proposes a formal definition of negative transfer and analyzes three key aspects, as well as a model for filtering out unrelated source data.

2) MEASURING KNOWLEDGE GAIN

The concept of transfer learning enables remarkable gains in learning new tasks. However, it's difficult to quantify how much knowledge is transferred. A mechanism for quantifying transfer in transfer learning is essential for understanding the quality of transfer and its viability. In addition to the available evaluation metrics, we need to assess the generalizability/robustness of the models, especially in situations where class sets are different between problems [507]. There was an attempt in [506], [508], and [509] to formulate the problem so that transfer learning related gains could be quantified.

3) SCALABILITY AND INTERPRETABILITY

Although many works demonstrate the ability of tasks to be transferred and their effectiveness, there is no guideline on how and what should be transferred. It has been shown that transfer learning can be effective only when there is a direct relationship between source and target; however, there have been many instances where transfer learning has failed despite the assumption of reletivity. Furthermore, as pre-trained models are becoming more widespread, with millions or billions of papameters, it would not be feasible to try all of the available methods to see which transfer could be helpful. Moreover, this requires a tremendous amount of computation, resulting in a large carbon footprint [510], [511]. It is critical that models are interpretable not only for their task, but also in terms of their ability to be transferred to other tasks. This work [512] defines the interpretable features that will be able to explain the relationship between the source and target domain in a transfer learning task.

4) CROSS-MODAL TRANSFER

In general, transfer learning is used when the source and target domains have the same modalities or input sizes. However, in many scenarios, this assumption could present a problem in adopting knowledge. Our ability to transfer knowledge from different modalities is crucial, since many tasks in our daily lives require information from multiple sources (perception and text or speech). One of the most recent studies, Bert [513] and ViLBert [514], attempts to transfer knowledge between text and image data. Additionally, we should be able to transfer knowledge regardless of the difference between input sizes in the source and target domain. An example could be transferring knowledge from 2D to 3D datasets [515], [516].

5) HOW TO BUILD TRANSFERABLE MODELS

The development of neural networks and deep learning models often requires significant architecture engineering. In addition, these models are engineered to outperform the existing models on the target dataset. As a result of the performance gain, the model's ability to generalize is usually degraded. We should be able to build models that enable transferability and reduce the dataset bias. As shown in [517], deep features eventually transition from general to specific along the network, which make the feature transferability drops significantly in higher layers. Works in [509], [517], [518], [519], [520], and [521] try to build the model with the focus of the transferability across domains.

XI. NEURAL RADIANCE FIELDS

A. DEFINITION AND APPLICATIONS

Several contributions in computer graphics have had a major impact on deep learning techniques to represent scenes and shapes with neural networks. A particular aim of the computer vision community is to represent objects and scenes in a photo-realistic manner using novel views. It enables a wide range of applications including cinema-graph [522], [523], video enhancement [524], [525], virtual reality [526], video stabilization [527], [528] and to name a few.

The task involves the collection of multiple images from different viewpoints of a real world scene, and the objective is to generate a photo-realistic image of such a novel view in the same scene. Many advancements have been made, one of the most common is to predict a 3D discrete volume representation using a neural network [529] and then render novel views using this representation. Usually these models take in the images and pass them through a 3D CNN model [530], then the model outputs the RGBA 3D volume [531], [532], [533]. Even though these models are very effective for rendering, they don't scale since each scene requires a lot of storage. A new approach to scene representation has emerged in recent years, in which the neural network represents the scene itself. In this case, the model takes in the X, Y, Z location and outputs the shape representation [534], [535], [536], [537]. The output of these models could be distance to the surface [534], occupancy [535], or a combination of color, and distance [536], [537]. As the shape itself is a neural network model, it is difficult to optimize it for different renderings. However, the key advantage is the shapes are compressed by the neural network which makes it very efficient in terms of memory. Nerf [538] combines these ideas into a single architecture. Given the spatial location X, Y, Z and viewing direction θ, ϕ, a simple fully connected outputs the color r, g, b and opacity σ of the specified input location and direction.

A very high level explanation of Nerfs could be think of as function that can map the the 3D location(x) and ray direction(d) to the color(r, g, b) and volume density(σ).

$$\mathrm{F}(\underline{\mathbf{x}}, \underline{\mathbf{d}}) = (\mathrm{r}, \mathrm{g}, \mathrm{b}, \boldsymbol{\sigma})$$

During the training stage, given a set of image from different views(well-known camera poses) an MLP is trained to optimize it weights.

In order to generate a realistic photo, we have to hypothetically place the camera(having the position) and point it to a specific direction.

Consider from the camera we shoot a ray and we want to sample from the NeRF along the way. There might be a lot of free space, but eventually the ray should collide with the surface of the object. The summation along the ray should represent the pixel's color at the specific location and viewing direction. In other words, the pixel value in image space is the weighted combination of these output values as below.

$$C \approx \sum_{i=1}^{N} T_i \alpha_i c_i \tag{38}$$

where T_i, can be think of as weights, is the accumulated product of all of the values behind it:

$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{39}$$

where α_i is:

$$\alpha_i = 1 - e^{-\sigma_i \delta t_i} \tag{40}$$

In the end, we can put all the pixels together to generate the image. The whole process, including the ray shooting, is fully differentiable, and can be trained using the total squared error:

$$\min_{\theta} \sum_{i} \|\text{render}_i (F_\theta) - I_i\|^2 \tag{41}$$

where i representing the ray and the loss minimizing the error between the rendered value from the network, F_θ, and the I is the actual pixel value.

B. CHALLENGES

In spite of the many improvements and astounding quality of rendering, the original Nerf paper left out many aspects.

One of the main assumptions in the original Nerf paper was static scenes. For many applications, including AR/VR, video game renderings, objects in the scenes are not static. The ability to render objects with respect to time along with the novel views are essential in my applications. There are some works attempting to solve the problem and change the original formulation for dynamic scenes and non-rigid objects [539], [540], [541], [542].

The other limitation is slow training and rendering. During the training phase, the model needs to qeury every pixel in the image. That results about 150 to 200 million queries for a one megapixel image [538], also, inference takes around 30 sec/frame. In order to solve the training issue, [543] proposes to use the depth data, which makes the network to need less number of views during training. Other network properties and optimizations can be change to speed-up the training issue [544], [545]. Inference also needs to be real-time for many rendering applications. Many works try to address the issue in different aspects; changing the scene representation to voxel base [546], [547], separate models for foreground and background [548] or other network improvements [541], [549], [550], [551].

A key feature of the representation for real-world scenarios is the ability to generalize across many cases. In contrast, the original Nerf trained an MLP for every scene. Every time a new scene is added, the MLP should be retrained from scratch. Several works have explored the possibility of generalizing and sharing the representation across multiple categories or at least within the category [552], [553], [554], [555].

For the scope to be widened to other possible applications, we need control over the renderings in different scenarios. The control over the camera position and direction was examined in the original Nerf paper. Some works attempted to control, edit, and condition it in terms of materials [556], [557], color [554], [558], object placement [559] and [560], facial attributes [561], [562] or text-guided editing [563].

XII. THE CHALLENGES OF REPRESENTATION LEARNING

Numerous challenges shall be addressed while learning representations from the data. The following section will provide a brief discussion on the most prominent challenges faced in the deep representation learning.

A. INTERPRETABILITY

There is a fine distinction between explainability and interpretability of a system. An explanation can be defined as any piece of information that helps the user understand the model's behavior and the process that it goes through to make the decision. Explanations can give insights on the role of each attribute in the overall performance of the system, or rules that determine the expected outcome, i.e., when a condition is met [564]. Interpretability, however, is considered as a human's ability to predict what the model result would be, based on the decision flow that the model follows [565]. A highly interpretable ML model is an easily comprehensible one, but the deep neural networks miss this aspect. Despite the promising performance of deep neural networks in various applications, the inherent lack of transparency in the process by which a deep neural network provides an output is still a major challenge. This black-box nature may render them useless in several applications, such as in situations where high degree of safety [566], security [567], fairness and ethicality [568], or reliability [17], [281], [569], [570], [571] are critical.

Therefore, design and implementation of problem-specific methods of interpretability and explainability is necessary [572]. Although the conventional methods of learning from data, such as decision trees, linear models, or self-organization maps [573], may provide visual explainability [15], deep neural network require post-hoc methods of interpretation. From a trained model, the underlying representation of the input data, may be extracted and presented in understandable formats for the end-users. Examples of post-hoc approaches are [15]: sentences generated as explanations [15], visualizations [574], explanation by examples [575]. Granted that, the post-hoc approaches provide another representation of the captured features, they do not directly reveal the exact causal connections and correlations at the model parameters level [15]. Nonetheless, it increases the reliability of the deep models.

B. SCALABILITY

Scalability is an essential and challenging aspect of many representation learning models, partly because getting models to maintain the quality and scale up to real-world applications relies on several different factors, including high-performance computing, optimized workloads distribution, managing a large distributed infrastructure, and Generalization of the algorithm [576], [577], [578]. Reference [579] has classified big-data machine learning approaches based on distributed or non-distributed fashion. In general, the scalability of representation learning models faces multiple dimensions and significant technical challenges: 1) availability of large amount of data 2) scaling the model size 3) scaling the number of models and/or computing machines 4)computing resources that can support the computational demands [577], [578], [580].

The huge amount of data can be accessed from a variety of sources, including internet clicks, user-generated content, business transactions, social media, sensor networks, etc [581]. Despite the growing pervasiveness level of big data, there are still challenges to accessing a high-quality training set. Data sharing agreements, violation of privacy [582], [583], noise problem [584], [585], poor data quality(fit for purpose) [586], imbalance of data [587], and lack of annotated datasets are number of challenges businesses face seeking raw data. Oversampling, undersampling, dynamic sampling [588] for imbalanced data, Surrogate Loss, Data Cleaning, finding distribution in solving the problem of learning from noisy labels for noisy data sets, and active learning [589] for lack of annotated data are a number of methods have been proposed to alleviate these problems.

Model scalability is one of the other concerns in which tasks may exhibit very high dimensionality. To efficiently handle this requirement, different approaches are proposed that cover the last two significant technical challenges mentioned earlier: using multiple machines in a cluster to improve the computing power (scaling out) [590] or using more powerful graphics processing units. Another crucial challenge is managing a large distributed infrastructure that hosts several deep learning models trained with a large amount of data. Over the last decade, there have been several types of research done in the area of high-performance computing to alleviate open research problems in infrastructure and hardware, Parallelization Methods, Optimizations for Data Parallelism, Scheduling and Elasticity, Data Management [576], [591], [592], [593], [594], [595]. While building large clusters of computing nodes may face several problems, such as communication bottlenecks, on the other hand, attempts to accelerate the performance of GPUs capable of implementing energy-efficient DL execution run across several major hurdles [595], [596]. Though we are able to train extremely large neural networks, they may optimize for a single outcome, and several challenges still remain

In addition, model pruning techniques [597] can help improve scalability by reducing model size and computational requirements. Pruning removes redundant or non-critical connections in neural networks to obtain a smaller, efficient model that maintains accuracy. This helps address hardware constraints and improves inference speed. Some of the most important pruning techniques include: Structured Pruning [598], which focuses on removing entire structured sections like layers or channels, producing more regular, hardware-friendly architectures; Unstructured Pruning [599], which removes individual weights from the network, leaving the overall architecture unchanged but with sparser connections; and Magnitude-based Pruning [600], a method where weights below a specified magnitude threshold are pruned, offering an optimal balance between simplicity and efficacy.

C. SECURITY, ROBUSTNESS, ADVERSARIAL ATTACKS

Machine learning is becoming more widely used, resulting in security and reliability concerns. Running these AI workflows for real-world applications may be vulnerable to adversarial attacks. AI models are developed under carefully controlled conditions for optimal performance. However, these conditions are rarely maintained in real-world scenarios. These changes could be both incidental or intentional adversity, both could result in a wrong prediction. Efficacy in detecting and detecting adversarial threats is referred to as adversarial robustness. A major challenge in robustness is the non-interpretability of many advanced models' representations. In [601] the authors show that there's a positive connections between model interpretability and adversarial robustness. In some cases [602], [603], [604], researchers attempt to interpret the results, but they usually pick examples and show the correlation between the representations and semantic concepts. However, such a relationship may not exist in general [605], [606]. The discontinuity of the representation first introduced in [607], where deep neural networks can be misclassified by adding imperceptible, non-random noise to inputs. For a more detailed discussion of different types of attacks, readers can refer to [608] and [609]. It is worth mentioning that Research on adversarial perturbations and attack techniques is primarily carried out in image classification [610], [611], [612], the same behavior is also seen in NLP [613], [614], speech recognition [615], [616], [617], and time-series analysis [618], [619]. For the systems based on biometrics verification [620], [621], [622], an adversarial attack could compromise its security. The use of biometrics in establishing a person's identity has become increasingly common in legal and administrative tasks [623]. The goal of representation learning is to find a (non-linear) representation of features $f : \mathcal{X} \rightarrow \mathcal{Z}$ fro from input space $\mathcal{X}$ to feature space $\mathcal{Z}$ so that f retains relevant information regarding the target task $\mathcal{Y}$ while hiding sensitive attributes [624]. Despite all the proposed defenses, deep learning algorithms still remain vulnerable to security attacks, as proposed defenses are only able to defend against the attacks they were designed to defend against [625]. In addition to the lack of universally robust algorithms, there is no unified metric by which to evaluate the robustness and resilience of the algorithms.

XIII. CONCLUSION

In this survey, we have explored the importance of deep representation learning in achieving competitive performances in state-of-the-art architectures. The methods of representing data serve as the foundation for the proposed techniques, making it crucial to understand the major approaches for learning representations. Since many of the state-of-the-art architectures rely on variants of neural networks to achieve competitive performances, the methods of representing data can be considered as the building blocks of the proposed methods. To achieve competitive performance in deep neural network architectures, it is essential to understand the major methods for learning representations. Our objective was to present each topic in a concise manner, while also providing detailed references and real-world applications to facilitate a deeper understanding for interested readers. As deep representation learning continues to be an active area of research, it holds great potential for impacting a wide range of applications. It is worth noting that the field of deep representation learning is dynamic and constantly evolving. As new advancements are made, further research may uncover more efficient and effective methods for learning representations from data.

ACKNOWLEDGMENT

(Amirreza Payandeh and Kourosh T. Baghaei are co-first authors.)

REFERENCES

[1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 2015.

[2] I. El Naqa and M. Murphy, "What is machine learning?" Jan. 2015, pp. 3–11, doi: 10.1007/978-3-319-18305-3_1.

[3] E. Alpaydin, *Machine Learning*. Cambridge, MA, USA: MIT Press, 2021.

[4] E. Alpaydin, *Machine Learning: The New AI*. Cambridge, MA, USA: MIT Press, 2016.

[5] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2018.

[6] K. P. Murphy, *Probabilistic Machine Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2022.

[7] K. P. Murphy, *Probabilistic Machine Learning: Advanced Topics*. Cambridge, MA, USA: MIT Press, 2022.

[8] S. Rafatirad, H. Homayoun, Z. Chen, and S. M. P. Dinakarrao, *Machine Learning for Computer Scientists and Data Analysts: From an Applied Perspective*. Cham, Switzerland: Springer, 2022.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: https://www.nature.com/articles/nature14539

[10] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," Apr. 2014, *arXiv:1206.5538*.

[11] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006. [Online]. Available: https://science.sciencemag.org/content/313/5786/504

[12] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression pointnet for 3D hand pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 475–491.

[13] E. Mosadegh and A. W. Nolin, "Estimating Arctic sea ice surface roughness by using back propagation neural network," in *Proc. AGU Fall Meeting Abstr.*, vol. 2020, 2020, Paper C014–0005.

[14] E. Mosadegh and A. W. Nolin, "A new data processing system for generating sea ice surface roughness products from the multi-angle imaging spectroradiometer (MISR) imagery," *Remote Sens.*, vol. 14, no. 19, p. 4979, Oct. 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/19/4979

[15] Z. C. Lipton, "The mythos of model interpretability," Mar. 2017, *arXiv:1606.03490*.

[16] P. Savadjiev, J. Chong, A. Dohan, M. Vakalopoulou, C. Reinhold, N. Paragios, and B. Gallix, "Demystification of AI-driven medical image interpretation: Past, present and future," *Eur. Radiol.*, vol. 29, no. 3, pp. 1616–1624, Mar. 2019.

[17] C. Chen, C. Qin, H. Qiu, G. Tarroni, J. Duan, W. Bai, and D. Rueckert, "Deep learning for cardiac image segmentation: A review," Nov. 2019, *arXiv:1911.03723*.

[18] D. Nie, R. Trullo, J. Lian, L. Wang, C. Petitjean, S. Ruan, Q. Wang, and D. Shen, "Medical image synthesis with deep convolutional adversarial networks," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 12, pp. 2720–2730, Dec. 2018.

[19] N. A. M. Zin, R. Yusof, S. A. Lashari, A. Mustapha, N. Senan, and R. Ibrahim, "Content-based image retrieval in medical domain: A review," *J. Phys., Conf. Ser.*, vol. 1019, Jun. 2018, Art. no. 012044, doi: 10.1088%2F1742-6596%2F1019%2F1%2F012044.

[20] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proc. 22nd ACM Int. Conf. Multimedia (MM)*. New York, NY, USA: ACM, 2014, pp. 157–166. [Online]. Available: http://doi.acm.org/10.1145/2647868.2654948

[21] J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Išgum, "Generative adversarial networks for noise reduction in low-dose CT," *IEEE Trans. Med. Imag.*, vol. 36, no. 12, pp. 2536–2545, Dec. 2017.

[22] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Anal.*, vol. 42, pp. 60–88, Dec. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361841517301135

[23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[24] S. Miao, Z. J. Wang, and R. Liao, "A CNN regression approach for real-time 2D/3D registration," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1352–1363, May 2016.

[25] S. Amirian, Z. Wang, T. R. Taha, and H. R. Arabnia, "Dissection of deep learning with applications in image recognition," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2018, pp. 1142–1148.

[26] M. A. Mazurowski, M. Buda, A. Saha, and M. R. Bashir, "Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI," *J. Magn. Reson. Imag.*, vol. 49, no. 4, pp. 939–954, Apr. 2019.

[27] W. Liu, J. Sun, W. Li, T. Hu, and P. Wang, "Deep learning on point clouds and its application: A survey," *Sensors*, vol. 19, no. 19, p. 4188, Sep. 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6806315/

[28] H.-C. Shin, K. Roberts, L. Lu, D. Demner-Fushman, J. Yao, and R. M. Summers, "Learning to read chest X-rays: Recurrent neural cascade model for automated image annotation," Mar. 2016, *arXiv:1603.08486*.

[29] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–36, Feb. 2019. [Online]. Available: http://doi.acm.org/10.1145/3295748

[30] S. Amirian, K. Rasheed, T. R. Taha, and H. R. Arabnia, "A short review on image caption generation with deep learning," in *Proc. Int. Conf. Image Process., Comput. Vis., Pattern Recognit. (IPCV)*, 2019, p. 1.

[31] B. C. Kwon, M.-J. Choi, J. T. Kim, E. Choi, Y. B. Kim, S. Kwon, J. Sun, and J. Choo, "RetainVis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records," 2018, *arXiv:1805.10724*.

[32] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, Apr. 2019.

[33] Y. Lan, G. He, J. Jiang, J. Jiang, W. X. Zhao, and J.-R. Wen, "Complex knowledge base question answering: A survey," 2021, *arXiv:2108.06688*.

[34] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[35] Y. Wu et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*.

[36] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, *arXiv:1508.04025*.

[37] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare," *Nature Med.*, vol. 25, no. 1, pp. 24–29, Jan. 2019.

[38] F. Z. Xing, E. Cambria, and R. E. Welsch, "Natural language based financial forecasting: A survey," *Artif. Intell. Rev.*, vol. 50, no. 1, pp. 49–73, Jun. 2018.

[39] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2022, pp. 27730–27744.

[40] N. Kalantari, D. Liao, and V. G. Motti, "Characterizing the online discourse in Twitter: Users' reaction to misinformation around COVID-19 in Twitter," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 4371–4380.

[41] V. G. Motti, N. Kalantari, and V. Neris, "Understanding how social media imagery empowers caregivers: An analysis of microcephaly in Latin America," *Pers. Ubiquitous Comput.*, vol. 25, no. 2, pp. 321–336, Apr. 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:220462314

[42] M. Zhou, N. Duan, S. Liu, and H.-Y. Shum, "Progress in neural NLP: Modeling, learning, and reasoning," *Engineering*, vol. 6, no. 3, pp. 275–290, Mar. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2095809919304928

[43] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[44] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1532–1543. [Online]. Available: https://aclanthology.org/D14-1162

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[46] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[47] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[48] P. Li, Y. Pei, and J. Li, "A comprehensive survey on design and application of autoencoder in deep learning," *Appl. Soft Comput.*, vol. 138, May 2023, Art. no. 110176. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494623001941

[49] D. Bank, N. Koenigstein, and R. Giryes, *Autoencoders*. Cham, Switzerland: Springer, 2023, pp. 353–374, doi: 10.1007/978-3-031-24628-9_16.

[50] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3313–3332, Apr. 2023.

[51] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He, H. Peng, J. Li, J. Wu, Z. Liu, P. Xie, C. Xiong, J. Pei, P. S. Yu, and L. Sun, "A comprehensive survey on pretrained foundation models: A history from BERT to ChatGPT," 2023, *arXiv:2302.09419*.

[52] T. Zhou, F. Porikli, D. Crandall, L. Van Gool, and W. Wang, "A survey on deep learning technique for video segmentation," 2021, *arXiv:2107.01153*.

[53] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 604–624, Feb. 2021.

[54] A. Oussidi and A. Elhassouny, "Deep generative models: Survey," in *Proc. Int. Conf. Intell. Syst. Comput. Vis. (ISCV)*, Apr. 2018, pp. 1–8.

[55] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006, doi: 10.1162/neco.2006.18.7.1527.

[56] Y. Song and D. P. Kingma, "How to train your energy-based models," 2021, *arXiv:2101.03288*.

[57] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986. [Online]. Available: https://www.nature.com/articles/323533a0

[58] I. Goodfellow, A. Courville, and Y. Bengio, *Generative Models*. Cambridge, MA, USA: MIT Press, 2014.

[59] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[60] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002.

[61] M. Á. Carreira-Perpiñán and G. E. Hinton, "On contrastive divergence learning," in *Proc. AISTATS*, 2005, pp. 33–40.

[62] R. Salakhutdinov and G. Hinton, "Deep Boltzmann machines," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, in Proceedings of Machine Learning Research, vol. 5, D. van Dyk and M. Welling, Eds. Clearwater Beach, FL, USA. Hilton Clearwater, Apr. 2009, pp. 448–455. [Online]. Available: http://proceedings.mlr.press/v5/salakhutdinov09a.html

[63] A. Courville, J. Bergstra, and Y. Bengio, "A spike and slab restricted Boltzmann machine," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, in Proceedings of Machine Learning Research, vol. 15, G. Gordon, D. Dunson, and M. Dudãk, Eds. Fort Lauderdale, FL, USA, Apr. 2011, pp. 233–241. [Online]. Available: http://proceedings.mlr.press/v15/courville11a.html

[64] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. Int. Conf. Mach. Learn.* New York, NY, USA: Association for Computing Machinery, Jun. 2009, pp. 609–616, doi: 10.1145/1553374.1553453.

[65] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Mar. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1953039

[66] Y. Bengio, *Learning Deep Architectures for AI*. Boston, MA, USA: Now, 2009. [Online]. Available: https://ieeexplore.ieee.org/document/8187120

[67] J. M. Giron-Sierra, "Sparse representations," in *Digital Signal Processing With MATLAB Examples: Model-Based Actions and Sparse Representation* (Signals and Communication Technology), vol. 3, J. M. Giron-Sierra, Ed. Singapore: Springer, 2017, pp. 151–261, doi: 10.1007/978-981-10-2540-2_2.

[68] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, Apr. 2017.

[69] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, Mar. 1951, doi: 10.1214/aoms/1177729694.

[70] D. P Kingma and M. Welling, "Auto-encoding variational Bayes," May 2014, *arXiv:1312.6114*.

[71] S. Rifai, P. Vincent, X. Müller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*. Madison, WI, USA: Omnipress, 2011, pp. 833–840.

[72] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," Jan. 2016, *arXiv:1511.06434*.

[73] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.

[74] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," 2016, *arXiv:1606.03657*.

[75] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.

[76] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[77] X. Chen, Y. Li, P. Jin, J. Zhang, X. Dai, J. Chen, and G. Song, "Adversarial sub-sequence for text generation," 2019, *arXiv:1905.12835*.

[78] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio, "Generating sentences from a continuous space," 2015, *arXiv:1511.06349*.

[79] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[80] M. Ran, J. Hu, Y. Chen, H. Chen, H. Sun, J. Zhou, and Y. Zhang, "Denoising of 3D magnetic resonance images using a residual encoder–decoder Wasserstein generative adversarial network," *Med. Image Anal.*, vol. 55, pp. 165–180, Jul. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361841518306534

[81] K. Munir, H. Elahi, A. Ayub, F. Frezza, and A. Rizzi, "Cancer diagnosis using deep learning: A bibliographic review," *Cancers*, vol. 11, no. 9, p. 1235, Aug. 2019. [Online]. Available: [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6770116/

[82] Z. Zhang, L. Yang, and Y. Zheng, "Translating and segmenting multimodal medical volumes with cycle- and shape-consistency generative adversarial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9242–9251.

[83] Y. Li, W. Li, P. He, J. Xiong, J. Xia, and Y. Xie, "CT synthesis from MRI images based on deep learning methods for MRI-only radiotherapy," in *Proc. Int. Conf. Med. Imag. Phys. Eng. (ICMIPE)*, Nov. 2019, pp. 1–6.

[84] C.-B. Jin, H. Kim, M. Liu, W. Jung, S. Joo, E. Park, Y. Ahn, I. Han, J. Lee, and X. Cui, "Deep CT to MR synthesis using paired and unpaired data," *Sensors*, vol. 19, no. 10, p. 2361, May 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6566351/

[85] H.-C. Shin, N. A. Tenenholtz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andriole, and M. Michalski, "Medical image synthesis for data augmentation and anonymization using generative adversarial networks," in *Simulation and Synthesis in Medical Imaging* (Lecture Notes in Computer Science), A. Gooya, O. Goksel, I. Oguz, and N. Burgos, Eds. Cham, Switzerland: Springer, 2018, pp. 1–11.

[86] K. Kamnitsas, C. Ledig, V. F. J. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Med. Image Anal.*, vol. 36, pp. 61–78, Feb. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361841516301839

[87] A. Alansary, K. Kamnitsas, A. E. Davidson, R. Khlebnikov, M. Rajchl, C. Malamateniou, M. A. Rutherford, J. V. Hajnal, B. Glocker, D. Rueckert, and B. Kainz, "Fast fully automatic segmentation of the human placenta from motion corrupted MRI," in *Proc. MICCAI*, 2016, pp. 589–597.

[88] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8798–8807.

[89] K. Bahrami, F. Shi, X. Zong, H. W. Shin, H. An, and D. Shen, "Reconstruction of 7T-like images from 3T MRI," *IEEE Trans. Med. Imag.*, vol. 35, no. 9, pp. 2085–2097, Sep. 2016.

[90] L. Qu, Y. Zhang, S. Wang, P.-T. Yap, and D. Shen, "Synthesized 7T MRI from 3T MRI via deep learning in spatial and wavelet domains," *Med. Image Anal.*, vol. 62, May 2020, Art. no. 101663. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361841520300293

[91] H. Huang, P. S. Yu, and C. Wang, "An introduction to image synthesis with generative adversarial nets," Nov. 2018, *arXiv:1803.04469*.

[92] Z. Chen, B. Subagdja, and A.-H. Tan, "End-to-end deep reinforcement learning for multi-agent collaborative exploration," in *Proc. IEEE Int. Conf. Agents (ICA)*, Oct. 2019, pp. 99–102.

[93] Z. Wu, N. M. Khan, L. Gao, and L. Guan, "Deep reinforcement learning with parameterized action space for object detection," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2018, pp. 101–104.

[94] Q. Li, X.-M. Wu, H. Liu, X. Zhang, and Z. Guan, "Label efficient semi-supervised learning via graph filtering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9574–9583.

[95] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. 32nd AAAI Conf. Artif. Intell.* AAAI Press, 2018, pp. 3538–3545.

[96] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[97] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[98] S. Jin, X. Zeng, F. Xia, W. Huang, and X. Liu, "Application of deep learning methods in biological networks," *Briefings Bioinf.*, vol. 22, no. 2, pp. 1902–1917, Mar. 2021.

[99] E. Davidson et al., "A genomic regulatory network for development," *Science*, vol. 295, no. 5560, pp. 1669–1678, Apr. 2002.

[100] J. H. Drew and H. Liu, "Diagnosing fault patterns in telecommunication networks," U.S. Patent 7 428 300, Sep. 23, 2008.

[101] D. Lazer et al., "Life in the network: The coming age of computational social science," *Science*, vol. 323, no. 5915, p. 721, 2009.

[102] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *Expert Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 117921.

[103] K.-H.-N. Bui, J. Cho, and H. Yi, "Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues," *Int. J. Speech Technol.*, vol. 52, no. 3, pp. 2763–2774, Feb. 2022.

[104] F. Chen, Z. Chen, S. Biswas, S. Lei, N. Ramakrishnan, and C.-T. Lu, "Graph convolutional networks with Kalman filtering for traffic prediction," in *Proc. 28th Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2020, pp. 135–138.

[105] W. Liao, B. Bak-Jensen, J. R. Pillai, Y. Wang, and Y. Wang, "A review of graph neural networks and their applications in power systems," *J. Mod. Power Syst. Clean Energy*, vol. 10, no. 2, pp. 345–360, Mar. 2022.

[106] B. Donon, B. Donnot, I. Guyon, and A. Marot, "Graph neural solver for power systems," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[107] D. Owerko, F. Gama, and A. Ribeiro, "Optimal power flow using graph neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 5930–5934.

[108] G. Zhang, H. He, and D. Katabi, "Circuit-GNN: Graph neural networks for distributed circuit design," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7364–7373.

[109] Z. Chen, G. Kolhe, S. Rafatirad, C.-T. Lu, S. Manoj P. D., H. Homayoun, and L. Zhao, "Estimating the circuit de-obfuscation runtime based on graph deep learning," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2020, pp. 358–363.

[110] M. E. J. Newman, "Spread of epidemic disease on networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 66, no. 1, Jul. 2002, Art. no. 016128.

[111] V. Marx, "High-throughput anatomy: Charting the brain's networks," *Nature*, vol. 490, no. 7419, p. 293, 2012.

[112] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2013.

[113] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017. [Online]. Available: https://mathweb.ucsd.edu/~fan/research/revised.html

[114] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[115] F. R. Chung, *Spectral Graph Theory*, no. 92. Providence, RI, USA: American Mathematical Society, 1997.

[116] R. Grone, R. Merris, and V. S. Sunder, "The Laplacian spectrum of a graph," *SIAM J. Matrix Anal. Appl.*, vol. 11, no. 2, pp. 218–238, 1990.

[117] K. C. Das, "The Laplacian spectrum of a graph," *Comput. Math. With Appl.*, vol. 48, nos. 5–6, pp. 715–724, Sep. 2004.

[118] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[119] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[120] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Appl. Comput. Harmon. Anal.*, vol. 40, no. 2, pp. 260–291, Mar. 2016.

[121] X. Zhu and M. Rabbat, "Approximating signals supported on graphs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 3921–3924.

[122] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.

[123] Q. Li, Z. Han, and X. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. AAAI*, 2018, pp. 3538–3545.

[124] G. Taubin, "A signal processing approach to fair surface design," in *Proc. 22nd Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 1995, pp. 351–358.

[125] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," 2018, *arXiv:1812.08434*.

[126] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," 2018, *arXiv:1812.04202*.

[127] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," 2019, *arXiv:1901.00596*.

[128] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," 2017, *arXiv:1709.05584*.

[129] Z. Chen, F. Chen, L. Zhang, T. Ji, K. Fu, L. Zhao, F. Chen, L. Wu, C. Aggarwal, and C.-T. Lu, "Bridging the gap between spatial and spectral domains: A survey on graph neural networks," 2020, *arXiv:2002.11867*.

[130] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[131] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" 2019, *arXiv:1810.00826*.

[132] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.

[133] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1993–2001.

[134] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[135] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6861–6871.

[136] F. M. Bianchi, D. Grattarola, L. Livi, and C. Alippi, "Graph neural networks with convolutional ARMA filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3496–3507, Jul. 2022.

[137] Z. Chen, F. Chen, R. Lai, X. Zhang, and C.-T. Lu, "Rational neural networks for approximating graph convolution operator on jump discontinuities," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 59–68.

[138] J. Gasteiger, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized PageRank," 2018, *arXiv:1810.05997*.

[139] M. Li, X. Guo, Y. Wang, Y. Wang, and Z. Lin, "G2CN: Graph Gaussian convolution networks with concentrated graph filters," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2022, pp. 12782–12796.

[140] G. Fu, P. Zhao, and Y. Bian, "p-Laplacian based graph neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 6878–6917.

[141] M. Yang, Y. Shen, R. Li, H. Qi, Q. Zhang, and B. Yin, "A new perspective on the effects of spectrum in graph neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2022, p. 25261–25279.

[142] H. Kenlay, D. Thanou, and X. Dong, "Interpretable stability bounds for spectral graph filters," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5388–5397.

[143] X. Wang and M. Zhang, "How powerful are spectral graph neural networks," 2022, *arXiv:2205.11172*.

[144] G. Meng, Q. Jiang, K. Fu, B. Lin, C.-T. Lu, and Z. Chen, "Early forecast of traffic accident impact based on a single-snapshot observation (student abstract)," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 11, pp. 13015–13016.

[145] G. Meng, Q. Jiang, K. Fu, B. Lin, C.-T. Lu, and Z. Chen, "Early forecasting of the impact of traffic accidents using a single shot observation," in *Proc. SIAM Int. Conf. Data Mining (SDM)*. Philadelphia, PA, USA: SIAM, 2022, pp. 100–108.

[146] J. Tang, J. Li, Z. Gao, and J. Li, "Rethinking graph neural networks for anomaly detection," 2022, *arXiv:2205.15508*.

[147] X. Zheng, B. Zhou, J. Gao, Y. G. Wang, P. Lió, M. Li, and G. Montúfar, "How framelets enhance graph neural networks," 2021, *arXiv:2102.06986*.

[148] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" 2018, *arXiv:1810.00826*.

[149] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1263–1272.

[150] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[151] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 1067–1077.

[152] A. Loukas, A. Simonetto, and G. Leus, "Distributed autoregressive moving average graph filters," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1931–1935, Nov. 2015.

[153] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 274–288, Jan. 2017.

[154] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "CayleyNets: Graph convolutional neural networks with complex rational spectral filters," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 97–109, Jan. 2019.

[155] K. Fu, Z. Chen, and C.-T. Lu, "StreetNet: Preference learning with convolutional neural network on urban crime perception," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, Nov. 2018, pp. 269–278.

[156] P. W. Battaglia et al., "Relational inductive biases, deep learning, and graph networks," 2018, *arXiv:1806.01261*.

[157] A. Sanchez-Gonzalez, M. W. Hoffman, I. Demir, B. Lakshminarayanan, E. V. Bonilla, and Y. W. Teh, "Learning structured dynamics models with variational inference," 2020, *arXiv:2002.08791*.

[158] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2224–2232.

[159] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, "Molecular graph convolutions: Moving beyond fingerprints," *J. Comput.-Aided Mol. Des.*, vol. 30, no. 8, pp. 595–608, Aug. 2016.

[160] S. Wang, X. Guo, and L. Zhao, "Deep generative model for periodic graphs," 2022, *arXiv:2201.11932*.

[161] P. Reiser, M. Neubert, A. Eberhard, L. Torresi, C. Zhou, C. Shao, H. Metni, C. van Hoesel, H. Schopmans, T. Sommer, and P. Friederich, "Graph neural networks for materials science and chemistry," *Commun. Mater.*, vol. 3, no. 1, p. 93, Nov. 2022.

[162] M. Zitnik, M. Agrawal, and J. Leskovec, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinformatics*, vol. 34, no. 13, pp. i457–i466, Jul. 2018.

[163] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur, "Protein interface prediction using graph convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6530–6539.

[164] X.-M. Zhang, L. Liang, L. Liu, and M.-J. Tang, "Graph neural networks and their current applications in bioinformatics," *Frontiers Genet.*, vol. 12, Jul. 2021, Art. no. 690049.

[165] H. Dai, C. Li, C. Coley, B. Dai, and L. Song, "Retrosynthesis prediction with conditional graph logic network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8872–8882.

[166] W. Norcliffe-Brown, S. Vafeias, and S. Parisot, "Learning conditioned graph structures for interpretable visual question answering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8334–8343.

[167] M. Narasimhan, S. Lazebnik, and A. Schwing, "Out of the box: Reasoning with graph convolution nets for factual visual question answering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2654–2665.

[168] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu, "Learning human-object interactions by graph parsing neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 401–417.

[169] H. Xu, C. Jiang, X. Liang, and Z. Li, "Spatial-aware graph relation network for large-scale object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9290–9299.

[170] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3588–3597.

[171] J. Gu, H. Hu, L. Wang, Y. Wei, and J. Dai, "Learning region features for object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 381–395.

[172] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 7370–7377.

[173] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," 2018, *arXiv:1809.10185*.

[174] S. Vashishth, M. Bhandari, P. Yadav, P. Rai, C. Bhattacharyya, and P. Talukdar, "Incorporating syntactic and semantic information in word embeddings using graph convolutional networks," 2018, *arXiv:1809.04283*.

[175] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial–temporal network for taxi demand prediction," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2588–2595.

[176] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "GaAN: Gated attention networks for learning on large and spatiotemporal graphs," 2018, *arXiv:1803.07294*.

[177] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*.

[178] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*.

[179] J. Wang, X. Zuo, and Y. He, "Graph-based network analysis of resting-state functional MRI," *Frontiers Syst. Neurosci.*, vol. 4, p. 16, Jun. 2010. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnsys.2010.00016

[180] S. Wang, L. He, B. Cao, C.-T. Lu, P. S. Yu, and A. B. Ragin, "Structural deep brain network mining," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 475–484.

[181] N. Nandakumar, K. Manzoor, J. J. Pillai, S. K. Gujar, H. I. Sair, and A. Venkataraman, "A novel graph neural network to localize eloquent cortex in brain tumor patients from resting-state fMRI connectivity," in *Proc. Int. Workshop Connectomics Neuroimaging*. Cham, Switzerland: Springer, 2019, pp. 10–20.

[182] D. Arya, R. Olij, D. K. Gupta, A. El Gazzar, G. van Wingen, M. Worring, and R. M. Thomas, "Fusing structural and functional MRIs using graph convolutional networks for autism classification," in *Proc. 3rd Conf. Med. Imag. Deep Learn.* (Proceedings of Machine Learning Research), vol. 121, T. Arbel, I. Ben Ayed, M. de Bruijne, M. Descoteaux, H. Lombaert, and C. Pal, Eds. PMLR, Jul. 2020, pp. 44–61. [Online]. Available: https://proceedings.mlr.press/v121/arya20a.html

[183] M. Craig, R. Adapa, I. Pappas, D. Menon, and E. Stamatakis, "Deep graph convolutional neural networks identify frontoparietal control and default mode network contributions to mental imagery," in *Proc. Conf. Cognit. Comput. Neurosci.*, Philadelphia, PA, USA, 2018, pp. 1–8.

[184] S. Gadgil, Q. Zhao, A. Pfefferbaum, E. V. Sullivan, E. Adeli, and K. M. Pohl, "Spatio-temporal graph convolution for resting-state fMRI analysis," 2020, *arXiv:2003.10613*.

[185] Z. Zhang, Z. Zhang, and Z. Chen, "XFlow: Benchmarking flow behaviors over graphs," 2023, *arXiv:2308.03819*.

[186] P. Smyth, D. Heckerman, and M. I. Jordan, "Probabilistic independence networks for hidden Markov probability models," *Neural Comput.*, vol. 9, no. 2, pp. 227–269, Feb. 1997.

[187] Z. Ghahramani, "An introduction to hidden Markov models and Bayesian networks," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 15, no. 1, pp. 9–42, Feb. 2001. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/S0218001401000836

[188] A. M. Selva, M. S. Yahaya, N. Azis, M. Z. A. A. Kadir, J. Jasni, and Y. Z. Y. Ghazali, "Estimation of transformers health index based on condition parameter factor and hidden Markov model," in *Proc. IEEE 7th Int. Conf. Power Energy (PECon)*, Dec. 2018, pp. 288–292.

[189] S. Z. Yu, "Hidden semi-Markov models," *Artif. Intell.*, vol. 174, pp. 215–243, Feb. 2010.

[190] K. P. Murphy, "Hidden semi-Markov Models," in *Introduction to Hidden Semi-Markov Models*. Cambridge, U.K.: Cambridge Univ. Press, Nov. 2002, pp. 110–124. [Online]. Available: https://www.cambridge.org/core/product/identifier/CBO9781108377423A053/type/book_part

[191] M. Dong and D. He, "A segmental hidden semi-Markov model (HSMM)-based diagnostics and prognostics framework and methodology," *Mech. Syst. Signal Process.*, vol. 21, no. 5, pp. 2248–2266, Jul. 2007.

[192] S. B. Ramezani, B. Killen, L. Cummins, S. Rahimi, A. Amirlatifi, and M. Seale, "A survey of HMM-based algorithms in machinery fault prediction," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2021, pp. 1–9.

[193] M. Dong, "A novel approach to equipment health management based on auto-regressive hidden semi-Markov model (AR-HSMM)," *Sci. China F, Inf. Sci.*, vol. 51, no. 9, pp. 1291–1304, Sep. 2008.

[194] X. Guan, R. Raich, and W. K. Wong, "Efficient multi-instance learning for activity recognition from time series data using an auto-regressive hidden Markov model," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, vol. 5, 2016, pp. 3452–3473.

[195] B. Mor, S. Garhwal, and A. Kumar, "A systematic review of hidden Markov models and their applications," *Arch. Comput. Methods Eng.*, vol. 28, no. 3, pp. 1429–1448, May 2021, doi: 10.1007/s11831-020-09422-4.

[196] Z. Ghahramani and M. I. Jordan, "Factorial hidden Markov models," *Mach. Learn.*, vol. 29, nos. 2–3, pp. 1–31, 1997.

[197] E. Angelino, M. J. Johnson, and R. P. Adams, "Patterns of scalable Bayesian inference," 2016, *arXiv:1602.05221*.

[198] S. Sharma, "Markov chain Monte Carlo methods for Bayesian data analysis in astronomy," *Annu. Rev. Astron. Astrophys.*, vol. 55, no. 1, pp. 213–259, Aug. 2017, doi: 10.1146/annurev-astro-082214-122339.

[199] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Advances in Neural Information Processing Systems*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2015, pp. 2575–2583. [Online]. Available: http://papers.nips.cc/paper/5666-variational-dropout-and-the-local-reparameterization-trick.pdf

[200] D. Krueger, C.-W. Huang, R. Islam, R. Turner, A. Lacoste, and A. Courville, "Bayesian hypernetworks," Apr. 2018, *arXiv:1710.04759*.

[201] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," Oct. 2016, *arXiv:1506.02142*.

[202] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt, "Advances in variational inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 2008–2026, Aug. 2019.

[203] M. W. Seeger, "Expectation propagation for exponential families," 2005. [Online]. Available: https://api.semanticscholar.org/CorpusID:1139278

[204] J. Regier, A. Miller, J. McAuliffe, R. Adams, M. Hoffman, D. Lang, D. Schlegel, and P. Prabhat, "Celeste: Variational inference for a generative model of astronomical images," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, vol. 37, 2015, pp. 2095–2103.

[205] T. Graepel, J. Q. n. Candela, T. Borchert, and R. Herbrich, "Web-scale Bayesian click-through rate prediction for sponsored search advertising in microsofts bing search engine," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*. Madison, WI, USA: Omnipress, 2010, pp. 13–20.

[206] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5574–5584.

[207] M. P. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 465–472.

[208] Y. Gal, R. T. McAllister, and C. E. Rasmussen, "Improving PILCO with Bayesian neural network dynamics models," in *Proc. Conf. Robot Learn. (CoRL)*, 2016, pp. 25–32.

[209] X. Tang, K. Yang, H. Wang, J. Wu, Y. Qin, W. Yu, and D. Cao, "Prediction-uncertainty-aware decision-making for autonomous vehicles," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 4, pp. 849–862, 2022, doi: 10.1109/TIV.2022.3188662.

[210] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*. Madison, WI, USA: Omnipress, 2010, pp. 807–814. [Online]. Available: http://dl.acm.org/citation.cfm?id=3104322.3104425

[211] E. Million, "The Hadamard product," Tech. Rep., 2007.

[212] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320317304120

[213] M. D Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," Nov. 2013, *arXiv:1311.2901*.

[214] R. G. J. Wijnhoven and P. H. N. de With, "Fast training of object detection using stochastic gradient descent," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 424–427.

[215] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2528–2535.

[216] Q. V. Le, J. Ngiam, Z. Chen, D. Chia, P. W. Koh, and A. Y. Ng, "Tiled convolutional neural networks," in *Proc. 23rd Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 1. Red Hook, NY, USA: Curran Associates, 2010, pp. 1279–1287. [Online]. Available: http://dl.acm.org/citation.cfm?id=2997189.2997332

[217] M. Lin, Q. Chen, and S. Yan, "Network in network," Mar. 2014, *arXiv:1312.4400*.

[218] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*.

[219] Y. Cai, Y. Li, C. Qiu, J. Ma, and X. Gao, "Medical image retrieval based on convolutional neural network and supervised hashing," *IEEE Access*, vol. 7, pp. 51877–51885, 2019.

[220] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," May 2015, *arXiv:1505.04597*.

[221] Y. Chen, J. Chen, D. Wei, Y. Li, and Y. Zheng, "OctopusNet: A deep learning segmentation network for multi-modal medical images," Aug. 2019, *arXiv:1906.02031*.

[222] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[223] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," 2017, *arXiv:1711.08488*.

[224] X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: A review," *Med. Image Anal.*, vol. 58, Dec. 2019, Art. no. 101552. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361841518308430

[225] W.-H. Weng and P. Szolovits, "Representation learning for electronic health records," Sep. 2019, *arXiv:1909.09248*.

[226] S. Parvaneh, J. Rubin, S. Babaeizadeh, and M. Xu-Wilson, "Cardiac arrhythmia detection using deep learning: A review," *J. Electrocardiology*, vol. 57, pp. S70–S74, Nov. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022073619303784

[227] Y. Li, "Deep reinforcement learning," 2018, *arXiv:1810.06339*.

[228] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," 2015, *arXiv:1506.05163*.

[229] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," 2019, *arXiv:1905.11946*.

[230] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018, *arXiv:1608.06993*.

[231] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015, *arXiv:1512.00567*.

[232] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," 2016, *arXiv:1611.05431*.

[233] B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jégou, and M. Douze, "LeViT: A vision transformer in ConvNet's clothing for faster inference," 2021, *arXiv:2104.01136*.

[234] *One-Hot Encoding Definition*. Accessed: Aug. 15, 2022. [Online]. Available: https://www.investopedia.com/terms/o/one-hot-encoding.asp

[235] G. Salton, "A vector space model for information retrieval," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[236] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Word embeddings: A survey," 2020, *arXiv:1901.09069*.

[237] M. Bansal and V. K. Singh, "Word embedding and its applications in natural language processing," *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 3711–3742, 2020.

[238] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Nov. 1985.

[239] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 1137–1155, Mar. 2003.

[240] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," in *Proc. 24th Int. Conf. Mach. Learn.*, Jun. 2007, pp. 641–648.

[241] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*. New York, NY, USA: Association for Computing Machinery, 2008, pp. 160–167, doi: 10.1145/1390156.1390177.

[242] J. Camacho-Collados and M. T. Pilehvar, "From word to sense embeddings: A survey on vector representations of meaning," *J. Artif. Intell. Res.*, vol. 63, pp. 743–788, Dec. 2018.

[243] T. Mikolov, W.-T. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2013, pp. 746–751.

[244] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2. Red Hook, NY, USA: Curran Associates, 2013, pp. 3111–3119.

[245] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2016, *arXiv:1607.04606*.

[246] F. K. Khattak, S. Jeblee, C. Pou-Prom, M. Abdalla, C. Meaney, and F. Rudzicz, "A survey of word embeddings for clinical text," *J. Biomed. Informat.*, vol. 100, Jan. 2019, Art. no. 100057. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590177X19300563

[247] B. Athiwaratkun, A. G. Wilson, and A. Anandkumar, "Probabilistic FastText for multi-sense word embeddings," 2018, *arXiv:1806.02901*.

[248] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[249] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for Twitter sentiment classification," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2014, pp. 1555–1565.

[250] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489.

[251] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1723–1732.

[252] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 260–270.

[253] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[254] A. Radford, J. Wook Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021, *arXiv:2103.00020*.

[255] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:49313245

[256] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2019, *arXiv:1910.10683*.

[257] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," 2018, *arXiv:1801.06146*.

[258] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[259] L. Dong and M. Lapata, "Language to logical form with neural attention," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 33–43.

[260] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 372–382.

[261] J. Xu, A. Neelakantan, M. Jones, K. Chai, D. Joglekar, S. Chandar, C.-J. Chen, and M. I. Jordan, "Deep clustering with convolutional autoencoders," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6620–6629.

[262] Z. Yang, G. Xue, Y. Yang, and Y. Wu, "Topic clustering using word embeddings," 2019, *arXiv:1911.11683*.

[263] M. Tao, H. Tang, F. Wu, X.-Y. Jing, B.-K. Bao, and C. Xu, "DF-GAN: A simple and effective baseline for text-to-image synthesis," 2022, *arXiv:2008.05865*.

[264] H. Guo, K. Tang, and J. Ye, "Deep clustering with convolutional autoencoders," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 6154–6163.

[265] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Deep learning for time series classification: A review," *Data Mining Knowl. Discovery*, vol. 33, no. 4, pp. 917–963, Jul. 2019, doi: 10.1007/s10618-019-00619-1.

[266] T. Ching et al., "Opportunities and obstacles for deep learning in biology and medicine," *J. Roy. Soc. Interface*, vol. 15, no. 141, Apr. 2018, Art. no. 20170387. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5938574/

[267] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-W.-H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Sci. Data*, vol. 3, no. 1, May 2016, Art. no. 160035, doi: 10.1038/sdata.2016.35.

[268] T. Lin, B. G. Horne, P. Tino, and C. L. Giles, "Learning long-term dependencies in NARX recurrent neural networks," *IEEE Trans. Neural Netw.*, vol. 7, no. 6, pp. 1329–1338, Nov. 1996.

[269] S. El Hihi and Y. Bengio, "Hierarchical recurrent neural networks for long-term dependencies," in *Proc. 8th Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Cambridge, MA, USA: MIT Press, 1995, pp. 493–499.

[270] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[271] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[272] T.-M. H. Hsu, W.-H. Weng, W. Boag, M. McDermott, and P. Szolovits, "Unsupervised multimodal representation learning across medical images and reports," Nov. 2018, *arXiv:1811.08615*.

[273] P. Sadda and T. Qarni, "Real-time medical video denoising with deep learning: Application to angiography," *Int. J. Appl. Inf. Syst.*, vol. 12, no. 13, pp. 22–28, May 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5985814/

[274] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: A review," *J. Neural Eng.*, vol. 16, no. 3, Apr. 2019, Art. no. 031001, doi: 10.1088/1741-2552/ab0ab5.

[275] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," Apr. 2015, *arXiv:1411.4555*.

[276] J. Chen, L. Yang, Y. Zhang, M. Alber, and D. Z. Chen, "Combining fully convolutional and recurrent neural networks for 3D biomedical image segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 3044–3052.

[277] X. Ye, J. Li, H. Huang, L. Du, and X. Zhang, "3D recurrent neural networks with context fusion for point cloud semantic segmentation," in *Computer Vision—ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 415–430.

[278] Q. Huang, W. Wang, and U. Neumann, "Recurrent slice networks for 3D segmentation of point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2626–2635.

[279] M. S. K. Kopuru, S. Rahimi, and K. T. Baghaei, "Recent approaches in prognostics: State of the art," in *Proc. Int. Conf. Artif. Intell. (ICAI)*, 2020, pp. 358–365.

[280] S. Barbieri, J. Kemp, O. Perez-Concha, S. Kotwal, M. Gallagher, A. Ritchie, and L. Jorm, "Benchmarking deep learning architectures for predicting readmission to the ICU and describing patients-at-risk," *Sci. Rep.*, vol. 10, no. 1, Jan. 2020, Art. no. 1111, doi: 10.1038/s41598-020-58053-z.

[281] K. T. Baghaei and S. Rahimi, "Sepsis prediction: An attention-based interpretable approach," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jun. 2019, pp. 1–6.

[282] J. P. C. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Trans. Assoc Comput. Linguistics*, vol. 4, pp. 357–370, Dec. 2016.

[283] N. Bölücü, D. Akgöl, and S. Tuc, "Bidirectional LSTM-CNNs with extended features for named entity recognition," in *Proc. Sci. Meeting Elect.-Electron. Biomed. Eng. Comput. Sci. (EBBT)*, Apr. 2019, pp. 1–4.

[284] A. Akbik, D. A. J. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *Proc. Int. Conf. Comput. Linguistics*, 2018, pp. 1638–1649.

[285] Y. Wu, M. Jiang, J. Xu, D. Zhi, and H. Xu, "Clinical named entity recognition using deep learning models," in *Proc. Annu. Symp. AMIA Symp.*, 2017, pp. 1812–1819.

[286] M. Ali, G. Tan, and A. Hussain, "Bidirectional recurrent neural network approach for Arabic named entity recognition," *Future Internet*, vol. 10, no. 12, p. 123, Dec. 2018. [Online]. Available: https://www.mdpi.com/1999-5903/10/12/123

[287] T. H. Nguyen and R. Grishman, "Relation extraction: Perspective from convolutional neural networks," in *Proc. VS@HLT-NAACL*, 2015, pp. 39–48.

[288] W. Zhou, K. Huang, T. Ma, and J. Huang, "Document-level relation extraction with adaptive thresholding and localized context pooling," 2020, *arXiv:2010.11304*.

[289] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2015, pp. 167–176.

[290] L. Li, Y. Liu, and M. Qin, "Extracting biomedical events with parallel multi-pooling convolutional neural networks," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 17, no. 2, pp. 599–607, Mar. 2020.

[291] X. Jin, X. Wang, X. Luo, S. Huang, and S. Gu, "Inter-sentence and implicit causality extraction from Chinese corpus," *Adv. Knowl. Discovery Data Mining*, vol. 12084, pp. 739–751, 2020.

[292] D. Ji, J. Gao, H. Fei, C. Teng, and Y. Ren, "A deep neural network model for speakers coreference resolution in legal texts," *Inf. Process. Manage.*, vol. 57, no. 6, Nov. 2020, Art. no. 102365.

[293] E. Allaway, S. Wang, and M. Ballesteros, "Sequential cross-document coreference resolution," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 4659–4671.

[294] A. Kuncoro, M. Ballesteros, L. Kong, C. Dyer, G. Neubig, and N. A. Smith, "What do recurrent neural network grammars learn about syntax?" in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1249–1258.

[295] J. R. Brennan, C. Dyer, A. Kuncoro, and J. T. Hale, "Localizing syntactic predictions using recurrent neural network grammars," *Neuropsychologia*, vol. 146, Sep. 2020, Art. no. 107479. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0028393220301500

[296] D. Yogatama, C. Dyer, W. Ling, and P. Blunsom, "Generative and discriminative text classification with recurrent neural networks," 2017, *arXiv:1703.01898*.

[297] R. Wang, Z. Li, J. Cao, T. Chen, and L. Wang, "Convolutional recurrent neural networks for text classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–6.

[298] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2227–2231.

[299] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[300] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward, "Semantic modelling with long-short-term memory for information retrieval," 2014, *arXiv:1412.6629*.

[301] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Proc. NIPS*, 2014, pp. 1–9.

[302] X. Qiu and X. Huang, "Convolutional neural tensor network architecture for community-based question answering," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 1–7.

[303] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, 2014, pp. 1–9.

[304] K. Cho, B. van Merrienboer, aglar Güehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1724–1734.

[305] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, *arXiv:1508.04025*.

[306] A. Bapna, M. Chen, O. Firat, Y. Cao, and Y. Wu, ''Training deeper neural machine translation models with transparent attention,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3028–3033.

[307] X. Du, J. Shao, and C. Cardie, ''Learning to ask: Neural question generation for reading comprehension,'' in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1342–1352.

[308] W.-T. Yih, M.-W. Chang, X. He, and J. Gao, ''Semantic parsing via staged query graph generation: Question answering with knowledge base,'' in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2015, pp. 1321–1331.

[309] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang, ''Abstractive text summarization using Sequence-to-sequence RNNs and beyond,'' in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.*, 2016, pp. 280–290.

[310] C. Khatri, G. Singh, and N. Parikh, ''Abstractive and extractive text summarization using document context vector and recurrent neural networks,'' 2018, *arXiv:1807.08000*.

[311] P. Etoori, M. Chinnakotla, and R. Mamidi, ''Automatic spelling correction for resource-scarce languages using deep learning,'' in *Proc. ACL, Student Res. Workshop*, 2018, pp. 146–152.

[312] S. Venugopalan, L. A. Hendricks, R. Mooney, and K. Saenko, ''Improving LSTM-based video description with linguistic knowledge mined from text,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1961–1966.

[313] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, ''Long-term recurrent convolutional networks for visual recognition and description,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2625–2634.

[314] X. Zhang, X. Liu, A. Ramachandran, C. Zhuge, S. Tang, P. Ouyang, Z. Cheng, K. Rupnow, and D. Chen, ''High-performance video content recognition with long-term recurrent convolutional network for FPGA,'' in *Proc. 27th Int. Conf. Field Program. Log. Appl. (FPL)*, Sep. 2017, pp. 1–4.

[315] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, ''Show, attend and tell: Neural image caption generation with visual attention,'' in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[316] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, ''Show and tell: A neural image caption generator,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3156–3164.

[317] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig, ''From captions to visual concepts and back,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1473–1482.

[318] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, ''Image captioning with semantic attention,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4651–4659.

[319] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, ''Bottom-up and top-down attention for image captioning and visual question answering,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6077–6086.

[320] S. S. Yadav and S. M. Jadhav, ''Deep convolutional neural network based medical image classification for disease diagnosis,'' *J. Big Data*, vol. 6, no. 1, pp. 1–18, Dec. 2019.

[321] L. Mou, P. Ghamisi, and X. X. Zhu, ''Deep recurrent neural networks for hyperspectral image classification,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[322] D. Ciregan, U. Meier, and J. Schmidhuber, ''Multi-column deep neural networks for image classification,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3642–3649.

[323] S. Tripathi, Z. C. Lipton, S. Belongie, and T. Nguyen, ''Context matters: Refining object detection in video with recurrent neural networks,'' 2016, *arXiv:1607.04648*.

[324] T. Hoeser and C. Kuenzer, ''Object detection and image segmentation with deep learning on earth observation data: A review—Part I: Evolution and recent trends,'' *Remote Sens.*, vol. 12, no. 10, p. 1667, May 2020.

[325] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, ''DRAW: A recurrent neural network for image generation,'' 2015, *arXiv:1502.04623*.

[326] G. Ning, Z. Zhang, C. Huang, X. Ren, H. Wang, C. Cai, and Z. He, ''Spatially supervised recurrent convolutional neural networks for visual object tracking,'' in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.

[327] H. Kieritz, W. Hübner, and M. Arens, ''Joint detection and online multi-object tracking,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1540–15408.

[328] M. Ren, R. Kiros, and R. S. Zemel, ''Exploring models and data for image question answering,'' in *Proc. NIPS*, 2015, pp. 2953–2961.

[329] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Parikh, and D. Batra, ''VQA: Visual question answering,'' *Int. J. Comput. Vis.*, vol. 123, no. 1, pp. 4–31, May 2017.

[330] D. Xu, W. Cheng, D. Luo, X. Liu, and X. Zhang, ''Spatio-temporal attentive RNN for node classification in temporal attributed graphs,'' in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3947–3953.

[331] J. B. Lee, R. Rossi, and X. Kong, ''Graph classification using structural attention,'' in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1666–1674.

[332] S. Geng, P. Gao, M. Chatterjee, C. Hori, J. L. Roux, Y. Zhang, H. Li, and A. Cherian, ''Dynamic graph representation learning for video dialog via multi-modal shuffled transformers,'' in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1415–1423.

[333] K. Xu, L. Wu, Z. Wang, Y. Feng, M. Witbrock, and V. Sheinin, ''Graph2Seq: Graph to sequence learning with attention-based neural networks,'' 2018, *arXiv:1804.00823*.

[334] R. S. Nair and P. Supriya, ''Robotic path planning using recurrent neural networks,'' in *Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2020, pp. 1–5.

[335] Y. Li, S. Li, and B. Hannaford, ''A novel recurrent neural network for improving redundant manipulator motion planning completeness,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2956–2961.

[336] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu, ''Attention mechanisms in computer vision: A survey,'' *Comput. Visual Media*, vol. 8, no. 3, pp. 331–368, Mar. 2022. [Online]. Available: http://dx.doi.org/10.1007/s41095-022-0271-y

[337] M. Liu, P. Shi, and L. Li, ''A survey of attention mechanism in deep learning,'' 2019, *arXiv:1902.07892*.

[338] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, ''Efficient transformers: A survey,'' *ACM Comput. Surv.*, Apr. 2022. [Online]. Available: https://doi.org/10.1145/3530811

[339] T. Lin, Y. Wang, X. Liu, and X. Qiu, ''A survey of transformers,'' *AI Open*, vol. 3, pp. 111–132, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666651022000146

[340] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, ''Transformer-XL: Attentive language models beyond a fixed-length context,'' in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2978–2988. [Online]. Available: https://aclanthology.org/P19-1285

[341] Q. Fournier, G. M. Caron, and D. Aloise, ''A practical survey on faster and lighter transformers,'' 2021, *arXiv:2103.14636*.

[342] P. Shaw, J. Uszkoreit, and A. Vaswani, ''Self-attention with relative position representations,'' 2018, *arXiv:1803.02155*.

[343] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, ''Image transformer,'' in *Proc. 35th Int. Conf. Mach. Learn.* (Proceedings of Machine Learning Research), vol. 80, J. Dy and A. Krause, Eds. PMLR, Jul. 2018, pp. 4055–4064. [Online]. Available: https://proceedings.mlr.press/v80/parmar18a.html

[344] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, ''Transformers are RNNs: Fast autoregressive transformers with linear attention,'' in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2020, pp. 5156–5165.

[345] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, ''Linformer: Self-attention with linear complexity,'' 2020, *arXiv:2006.04768*.

[346] Y. Tay, D. Bahri, D. Metzler, D.-C. Juan, Z. Zhao, and C. Zheng, ''Synthesizer: Rethinking self-attention for transformer models,'' 2021. [Online]. Available: https://openreview.net/forum?id=H-SPvQtMwm

[347] J. Dass, S. Wu, H. Shi, C. Li, Z. Ye, Z. Wang, and Y. Lin, ''ViTALiTy: Unifying low-rank and sparse approximation for vision transformer acceleration with a linear Taylor attention,'' 2022, *arXiv:2211.05109*.

[348] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed, ''Big bird: Transformers for longer sequences,'' in *Proc. 34th Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran Associates, 2020.

[349] J. Ainslie, S. Ontanon, C. Alberti, V. Cvicek, Z. Fisher, P. Pham, A. Ravula, S. Sanghai, Q. Wang, and L. Yang, "ETC: Encoding long and structured inputs in transformers," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, Nov. 2020, pp. 268–284. [Online]. Available: https://aclanthology.org/2020.emnlp-main.19

[350] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[351] J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans, "Axial attention in multidimensional transformers," 2020. [Online]. Available: https://openreview.net/forum?id=H1e5GJBtDr

[352] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Advances in Neural Information Processing Systems*, vol. 28, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/7137debd45ae4d0ab9aa953017286b20-Paper.pdf

[353] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. New Orleans, LA, USA: Association for Computational Linguistics, Jun. 2018, pp. 2227–2237. [Online]. Available: https://aclanthology.org/N18-1202

[354] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran Associates, 2017, pp. 6297–6308.

[355] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:49313245

[356] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf

[357] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2020. [Online]. Available: https://openreview.net/forum?id=SyxS0T4tvS

[358] F. Yu, J. Tang, W. Yin, Y. Sun, H. Tian, H. Wu, and H. Wang, "ERNIE-ViL: Knowledge enhanced vision-language representations through scene graph," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 3208–3216.

[359] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," in *Proc. Int. Conf. Learn. Represent.*, 2020. [Online]. Available: https://openreview.net/forum?id=r1xMH1BtvB

[360] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2021.

[361] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[362] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 730–734.

[363] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," 2014, *arXiv:1404.5997*.

[364] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016.

[365] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018.

[366] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, "A survey on evaluation of large language models," 2023.

[367] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models," 2023.

[368] *GPT-4 Technical Report*, OpenAI, vol. abs/2303.08774, 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[369] R. Anil et al., "PaLM 2 technical report," vol. 4, 2023, *arXiv:2305.10403*.

[370] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.

[371] A. Payandeh, D. Pluth, J. Hosier, X. Xiao, and V. K. Gurbani, "How susceptible are LLMs to Logical Fallacies?" 2023, *arXiv:2308.09853*.

[372] I. O. Gallegos, R. A. Rossi, J. Barrow, M. Mehrab Tanjim, S. Kim, F. Dernoncourt, T. Yu, R. Zhang, and N. K. Ahmed, "Bias and fairness in large language models: A survey," 2023, *arXiv:2309.00770*.

[373] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "A comprehensive overview of large language models," 2023, *arXiv:2307.06435*.

[374] J. Kaddour, J. Harris, M. Mozes, H. Bradley, R. Raileanu, and R. McHardy, "Challenges and applications of large language models," 2023, *arXiv:2307.10169*.

[375] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (IndRNN): Building a longer and deeper RNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5457–5466.

[376] A. R. Voelker, I. Kajić, and C. Eliasmith, "Legendre memory units: Continuous-time representation in recurrent neural networks," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates, 2019.

[377] A. R. Voelker and C. Eliasmith, "Improving spiking dynamical networks: Accurate delays, higher-order synapses, and time cells," *Neural Comput.*, vol. 30, no. 3, pp. 569–609, Mar. 2018, doi: 10.1162/neco_a_01046.

[378] S. Wang, M. Khabsa, and H. Ma, "To pretrain or not to pretrain: Examining the benefits of pretrainng on resource rich tasks," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, Jul. 2020, pp. 2209–2213. [Online]. Available: https://aclanthology.org/2020.acl-main.200

[379] V. S. F. Garnot, L. Landrieu, S. Giordano, and N. Chehata, "Satellite image time series classification with pixel-set encoders and temporal self-attention,"in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2020, pp. 12322–12331. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.01234

[380] K. Lu, A. Grover, P. Abbeel, and I. Mordatch, "Pretrained transformers as universal computation engines," 2021, *arXiv:2103.05247*.

[381] Z. Li, Y. Wu, B. Peng, X. Chen, Z. Sun, Y. Liu, and D. Paul, "SeTransformer: A transformer-based code semantic parser for code comment generation," *IEEE Trans. Rel.*, vol. 72, no. 1, pp. 258–273, Mar. 2023.

[382] T. Kano, S. Sakti, and S. Nakamura, "Transformer-based direct speech-to-speech translation with transcoder," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Jan. 2021, pp. 958–965.

[383] P. Li, P. Zhong, K. Mao, D. Wang, X. Yang, Y. Liu, J. Yin, and S. See, "ACT: An attentive convolutional transformer for efficient text classification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 15, 2021, pp. 13261–13269.

[384] Z. Wang, Y. Ma, Z. Liu, and J. Tang. (2020). *R-Transformer: RECURRENT Neural Network Enhanced Transformer*. [Online]. Available: https://openreview.net/forum?id=HJx4PAEYDH

[385] D. Fellner, T. I. Strasser, and W. Kastner, "Applying deep learning-based concepts for the detection of device misconfigurations in power systems," *Sustain. Energy, Grids Netw.*, vol. 32, Dec. 2022, Art. no. 100851. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352467722001266

[386] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.

[387] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Minneapolis, MN USA: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[388] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, and V. Stoyanov, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.

[389] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[390] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[391] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," 2019, *arXiv:1906.00446*.

[392] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," 2021, *arXiv:2102.12092*.

[393] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020, *arXiv:2005.12872*.

[394] L. L. Li, B. Yang, M. Liang, W. Zeng, M. Ren, S. Segal, and R. Urtasun, "End-to-end contextual perception and prediction with interaction transformer," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5784–5791.

[395] M. H. Nazeri and M. Bohlouli, "Exploring reflective limitation of behavior cloning in autonomous vehicles," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2021, pp. 1252–1257.

[396] L. Chen, Y. Wang, Z. Miao, Y. Mo, M. Feng, Z. Zhou, and H. Wang, "Transformer-based imitative reinforcement learning for multi-robot path planning," *IEEE Trans. Ind. Informat.*, vol. 19, no. 10, pp. 10233–10243, Oct. 2023.

[397] J. Wang, Q. Yang, S. Shen, R. Xiong, K. Zhang, and J. Chen, "Transformer-based multi-agent motion planning," 2021, *arXiv:2103.04358*.

[398] U. Gupta, L. N. Giribabu, A. Gupta, and A. Agrawal, "The beauty and the beast: An unbiased look at the performance of transformers in mobile robotics," 2020, *arXiv:2005.05776*.

[399] Y. Zhang, C. Jiang, L. Bai, and M. Liu, "Transformers in robotics: A survey," 2021, *arXiv:2105.10223*.

[400] Y. Chang, Z. Yang, Y. Liu, C. Yang, H. Yang, and J. Wang, "Robot control with transformer," 2021, *arXiv:2101.11552*.

[401] H. Kim, Y. Ohmura, and Y. Kuniyoshi, "Transformer-based deep imitation learning for dual-arm robot manipulation," 2021, *arXiv:2108.00385*.

[402] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (IndRNN): Building a longer and deeper RNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5457–5466.

[403] D. Lee and S. N. Yoon, "Application of artificial intelligence-based technologies in the healthcare industry: Opportunities and challenges," *Int. J. Environ. Res. Public Health*, vol. 18, no. 1, p. 271, Jan. 2021. https://www.mdpi.com/1660-4601/18/1/271

[404] T. Reitmaier, E. Wallington, D. K. Raju, O. Klejch, J. Pearson, M. Jones, P. Bell, and S. Robinson, "Opportunities and challenges of automatic speech recognition systems for low-resource language speakers," in *Proc. CHI Conf. Hum. Factors Comput. Syst.* New York, NY, USA: Association for Computing Machinery, 2022, pp. 1–17, doi: 10.1145/3491102.3517639.

[405] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Nov. 2010.

[406] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Dec. 2009.

[407] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," 2019, *arXiv:1911.02685*.

[408] M.-W. Chang, L.-A. Ratinov, D. Roth, and V. Srikumar, "Importance of semantic representation: Dataless classification," in *Proc. Aaai*, vol. 2, 2008, pp. 830–835.

[409] Z. Zhu, K. Lin, A. K. Jain, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," 2020, *arXiv:2009.07888*.

[410] P. Zhao, S. C. H. Hoi, J. Wang, and B. Li, "Online transfer learning," *Artif. Intell.*, vol. 216, pp. 76–102, Nov. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370214000800

[411] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. 24th Int. Conf. Mach. Learn.* New York, NY, USA: Association for Computing Machinery, Jun. 2007, pp. 193–200, doi: 10.1145/1273496.1273521.

[412] C. Wan, R. Pan, and J. Li, "Bi-weighting domain adaptation for cross-language text classification," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1535–1541.

[413] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in NLP," in *Proc. ACL*, 2007, pp. 264–271.

[414] X. Liao, Y. Xue, and L. Carin, "Logistic regression with an auxiliary data source," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, 2005, pp. 505–512.

[415] P. Wu and T. G. Dietterich, "Improving SVM accuracy by training on auxiliary data sources," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 110.

[416] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, p. 9, May 2016, doi: 10.1186/s40537-016-0043-6.

[417] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," Aug. 2018, *arXiv:1808.01974*.

[418] J. Stüber, M. Kopicki, and C. Zito, "Feature-based transfer learning for robotic push manipulation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 5643–5650.

[419] H. Ren, W. Liu, M. Shan, and X. Wang, "A new wind turbine health condition monitoring method based on VMD-MPE and feature-based transfer learning," *Measurement*, vol. 148, Dec. 2019, Art. no. 106906. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224119307638

[420] X. Zhong, S. Guo, H. Shan, L. Gao, D. Xue, and N. Zhao, "Feature-based transfer learning based on distribution similarity," *IEEE Access*, vol. 6, pp. 35551–35557, 2018.

[421] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[422] S. Gao and N. Bezzo, "A conformal mapping-based framework for robot-to-robot and sim-to-real transfer learning," 2021, *arXiv:2109.09214*.

[423] Y. Lockner and C. Hopmann, "Induced network-based transfer learning in injection molding for process modelling and optimization with artificial neural networks," *Int. J. Adv. Manuf. Technol.*, vol. 112, nos. 11–12, pp. 3501–3513, Feb. 2021.

[424] Q. Yang, W. Shi, J. Chen, and W. Lin, "Deep convolution neural network-based transfer learning method for civil infrastructure crack detection," *Autom. Construct.*, vol. 116, Aug. 2020, Art. no. 103199. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580519316000

[425] P. Cao, S. Zhang, and J. Tang, "Preprocessing-free gear fault diagnosis using small datasets with deep convolutional neural network-based transfer learning," *IEEE Access*, vol. 6, pp. 26241–26253, 2018.

[426] S. Sakhavi and C. Guan, "Convolutional neural network-based transfer learning and knowledge distillation using multi-subject data in motor imagery BCI," in *Proc. 8th Int. IEEE/EMBS Conf. Neural Eng. (NER)*, May 2017, pp. 588–591.

[427] F. Xue, Q. Wang, and G. Guo, "TransFER: Learning relation-aware facial expression representations with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3581–3590.

[428] R. Kumaraswamy, P. Odom, K. Kersting, D. Leake, and S. Natarajan, "Transfer learning via relational type matching," in *Proc. IEEE Int. Conf. Data Mining*, Nov. 2015, pp. 811–816.

[429] X. Qin, J. Yang, P. Li, W. Sun, and W. Liu, "A novel relational-based transductive transfer learning method for PolSAR images via time-series clustering," *Remote Sens.*, vol. 11, no. 11, p. 1358, Jun. 2019. [Online]. Available: https://www.mdpi.com/2072-4292/11/11/1358

[430] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," 2017, *arXiv:1711.02536*.

[431] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," 2018, *arXiv:1809.02176*.

[432] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," 2017, *arXiv:1702.05464*.

[433] Z. Zhang, X. Li, L. Wen, L. Gao, and Y. Gao, "Fault diagnosis using unsupervised transfer learning based on adversarial network," in *Proc. IEEE 15th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2019, pp. 305–310.

[434] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[435] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," 2019, *arXiv:1905.07129*.

[436] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.

[437] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[438] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11966–11976.

[439] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "Wav2vec: Unsupervised pre-training for speech recognition," 2019, *arXiv:1904.05862*.

[440] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," 2014, *arXiv:1412.5567*.

[441] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," 2021, *arXiv:2106.07447*.

[442] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," 2017, *arXiv:1702.08502*.

[443] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: A survey," 2020, *arXiv:2009.13303*.

[444] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, "Transfusion: Understanding transfer learning for medical imaging," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 3347–3357.

[445] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. V. Le, "Rethinking pre-training and self-training," 2020, *arXiv:2006.06882*.

[446] Y. Zhang and Q. Yang, "A survey on multi-task learning," 2017, *arXiv:1707.08114*.

[447] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, "Towards a unified view of parameter-efficient transfer learning," 2021, *arXiv:2110.04366*.

[448] N. Nejatishahidin, P. Fayyazsanavi, and J. Kosecka, "Object pose estimation using mid-level visual representations," 2022, *arXiv:2203.01449*.

[449] L. Alzubaidi, M. A. Fadhel, O. Al-Shamma, J. Zhang, J. Santamaría, Y. Duan, and S. R. Oleiwi, "Towards a better understanding of transfer learning for medical imaging: A case study," *Appl. Sci.*, vol. 10, no. 13, p. 4523, Jun. 2020.

[450] S. Chen, K. Ma, and Y. Zheng, "Med3D: Transfer learning for 3D medical image analysis," 2019, *arXiv:1904.00625*.

[451] M. Maqsood, F. Nazir, U. Khan, F. Aadil, H. Jamal, I. Mehmood, and O.-Y. Song, "Transfer learning assisted classification and detection of Alzheimer's disease stages using 3D MRI scans," *Sensors*, vol. 19, no. 11, p. 2645, Jun. 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/11/2645

[452] J. Wacker, M. Ladeira, and J. E. V. Nascimento, "Transfer learning for brain tumor segmentation," 2019, *arXiv:1912.12452*.

[453] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[454] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[455] L. Alzubaidi, M. Al-Amidie, A. Al-Asadi, A. J. Humaidi, O. Al-Shamma, M. A. Fadhel, J. Zhang, J. Santamaría, and Y. Duan, "Novel transfer learning approach for medical imaging with limited labeled data," *Cancers*, vol. 13, no. 7, p. 1590, Mar. 2021. [Online]. Available: https://www.mdpi.com/2072-6694/13/7/1590

[456] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," 2019, *arXiv:1906.08237*.

[457] T. B. Brown et al., "Language models are few-shot learners," 2020, *arXiv:2005.14165*.

[458] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.

[459] K. Subramanyam Kalyan, A. Rajasekharan, and S. Sangeetha, "AMMUS: A survey of transformer-based pretrained models in natural language processing," 2021, *arXiv:2108.05542*.

[460] W. Zhang, R. He, H. Peng, L. Bing, and W. Lam, "Cross-lingual aspect-based sentiment analysis with aspect term code-switching," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 9220–9230. [Online]. Available: https://aclanthology.org/2021.emnlp-main.727

[461] H. Nayel, E. Amer, A. Allam, and H. Abdallah, "Machine learning-based model for sentiment and sarcasm detection," in *Proc. 6th Arabic Natural Lang. Process. Workshop*. Kyiv, Ukraine: Association for Computational Linguistics, Apr. 2021, pp. 386–389. [Online]. Available: https://aclanthology.org/2021.wanlp-1.51

[462] J. Zhou, J. Tian, R. Wang, Y. Wu, W. Xiao, and L. He, "SentiX: A sentiment-aware pre-trained model for cross-domain sentiment analysis," in *Proc. 28th Int. Conf. Comput. Linguistics*. Barcelona, Spain: International Committee on Computational Linguistics, Dec. 2020, pp. 568–579. [Online]. Available: https://aclanthology.org/2020.coling-main.49

[463] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "SuperGLUE: A stickier benchmark for general-purpose language understanding systems," 2019, *arXiv:1905.00537*.

[464] M. Guo, Y. Yang, D. Cer, Q. Shen, and N. Constant, "MultiReQA: A cross-domain evaluation for retrieval question answering models," in *Proc. 2nd Workshop Domain Adaptation for NLP*. Kyiv, Ukraine: Association for Computational Linguistics, Apr. 2021, pp. 94–104. [Online]. Available: https://aclanthology.org/2021.adaptnlp-1.10

[465] W. Yu, L. Wu, Y. Deng, R. Mahindru, Q. Zeng, S. Guven, and M. Jiang, "A technical question answering system with transfer learning," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*. Stroudsburg, PA, USA: Association for Computational Linguistics, Oct. 2020, pp. 92–99. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.13

[466] M. V. Nguyen, T. N. Nguyen, B. Min, and T. H. Nguyen, "Crosslingual transfer learning for relation and event extraction via word category and class alignments," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 5414–5426. [Online]. Available: https://aclanthology.org/2021.emnlp-main.440

[467] Q. Do and J. Gaspers, "Cross-lingual transfer learning with data selection for large-scale spoken language understanding," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 1455–1460. [Online]. Available: https://aclanthology.org/D19-1153

[468] N. Das, S. Bodapati, M. Sunkara, S. Srinivasan, and D. H. Chau, "Best of both worlds: Robust accented speech recognition with adversarial transfer learning," 2021, *arXiv:2103.05834*.

[469] J. Luo, J. Wang, N. Cheng, E. Xiao, J. Xiao, G. Kucsko, P. O'Neill, J. Balam, S. Deng, A. Flores, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, and J. Li, "Cross-language transfer learning and domain adaptation for End-to-End automatic speech recognition," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2021, pp. 1–6.

[470] B. Sertolli, Z. Ren, B. W. Schuller, and N. Cummins, "Representation transfer learning from deep end-to-end speech recognition networks for the classification of health states from speech," *Comput. Speech Lang.*, vol. 68, Jul. 2021, Art. no. 101204.

[471] J. Cho, M. K. Baskar, R. Li, M. Wiesner, S. H. Mallidi, N. Yalta, M. Karafiát, S. Watanabe, and T. Hori, "Multilingual sequence-to-sequence speech recognition: Architecture, transfer learning, and language modeling," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2018, pp. 521–527.

[472] M. Weber, M. Auch, C. Doblander, P. Mandl, and H.-A. Jacobsen, "Transfer learning with time series data: A systematic mapping study," *IEEE Access*, vol. 9, pp. 165409–165432, 2021.

[473] Y. Rotem, N. Shimoni, L. Rokach, and B. Shapira, "Transfer learning for time series classification using synthetic data generation," in *Proc. Int. Symp. Cyber Secur., Cryptol., Mach. Learn.* Cham, Switzerland: Springer, 2022, pp. 232–246.

[474] C. Xu, J. Wang, J. Zhang, and X. Li, "Anomaly detection of power consumption in yarn spinning using transfer learning," *Comput. Ind. Eng.*, vol. 152, Feb. 2021, Art. no. 107015.

[475] B. Maschler and M. Weyrich, ''Deep transfer learning for industrial automation: A review and discussion of new techniques for data-driven machine learning,'' *IEEE Ind. Electron. Mag.*, vol. 15, no. 2, pp. 65–75, Jun. 2021.

[476] R. Ye and Q. Dai, ''Implementing transfer learning across different datasets for time series forecasting,'' *Pattern Recognit.*, vol. 109, Jan. 2021, Art. no. 107617. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320320304209

[477] Q. Gu and Q. Dai, ''A novel active multi-source transfer learning algorithm for time series forecasting,'' *Int. J. Speech Technol.*, vol. 51, no. 3, pp. 1326–1350, Mar. 2021, doi: 10.1007/s10489-020-01871-5.

[478] B. Kulis, K. Saenko, and T. Darrell, ''What you saw is not what you get: Domain adaptation using asymmetric kernel transforms,'' in *Proc. CVPR*, Jun. 2011, pp. 1785–1792.

[479] X. Shi, Q. Liu, W. Fan, P. S. Yu, and R. Zhu, ''Transfer learning on heterogenous feature spaces via spectral transformation,'' in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 1049–1054.

[480] C. Wang and S. Mahadevan, ''Heterogeneous domain adaptation using manifold alignment,'' in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, p. 1541.

[481] R. Chattopadhyay, Q. Sun, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, ''Multisource domain adaptation and its application to early detection of fatigue,'' *ACM Trans. Knowl. Discovery From Data*, vol. 6, no. 4, pp. 1–26, Dec. 2012.

[482] J. Gao, W. Fan, J. Jiang, and J. Han, ''Knowledge transfer via multiple model local structure mapping,'' in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2008, pp. 283–291.

[483] J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han, ''Graph-based consensus maximization among multiple supervised and unsupervised models,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 22, 2009, pp. 585–593.

[484] P. Luo, F. Zhuang, H. Xiong, Y. Xiong, and Q. He, ''Transfer learning from multiple source domains via consensus regularization,'' in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, Oct. 2008, pp. 103–112.

[485] O. Moreno, B. Shapira, L. Rokach, and G. Shani, ''TALMUD: Transfer learning for multiple domains,'' in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2012, pp. 425–434.

[486] B. Cao, N. N. Liu, and Q. Yang, ''Transfer learning for collective link prediction in multiple heterogenous domains,'' in *Proc. ICML*, 2010, pp. 159–166.

[487] M. Jiang, P. Cui, F. Wang, Q. Yang, W. Zhu, and S. Yang, ''Social recommendation across multiple relational domains,'' in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2012, pp. 1422–1431.

[488] B. Li, Q. Yang, and X. Xue, ''Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction,'' in *Proc. 21st Int. Joint Conf. Artif. Intell.* Pasadena, CA, USA: Morgan Kaufmann Publishers, 2009, pp. 2052–2057.

[489] B. Li, Q. Yang, and X. Xue, ''Transfer learning for collaborative filtering via a rating-matrix generative model,'' in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Jun. 2009, pp. 617–624.

[490] W. Pan, E. Xiang, N. Liu, and Q. Yang, ''Transfer learning in collaborative filtering for sparsity reduction,'' in *Proc. AAAI Conf. Artif. Intell.*, 2010, vol. 24, no. 1, pp. 230–235.

[491] W. Pan, N. N. Liu, E. W. Xiang, and Q. Yang, ''Transfer learning to predict missing ratings via heterogeneous user feedbacks,'' in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, p. 2318.

[492] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, ''Self-taught learning: Transfer learning from unlabeled data,'' in *Proc. 24th Int. Conf. Mach. Learn.*, Jun. 2007, pp. 759–766.

[493] Y. Zhang, B. Cao, and D.-Y. Yeung, ''Multi-domain collaborative filtering,'' 2012, *arXiv:1203.3535*.

[494] L. Zhao, S. J. Pan, E. W. Xiang, E. Zhong, Z. Lu, and Q. Yang, ''Active transfer learning for cross-system recommendation,'' in *Proc. 27th AAAI Conf. Artif. Intell.*, 2013, pp. 1205–1211.

[495] H. A. Ogoe, S. Visweswaran, X. Lu, and V. Gopalakrishnan, ''Knowledge transfer via classification rules using functional mapping for integrative modeling of gene expression data,'' *BMC Bioinf.*, vol. 16, no. 1, pp. 1–15, Dec. 2015.

[496] C. Widmer and G. Rätsch, ''Multitask learning in computational biology,'' in *Proc. ICML Workshop Unsupervised Transf. Learn.*, 2012, pp. 207–216.

[497] M. Kan, J. Wu, S. Shan, and X. Chen, ''Domain adaptation for face recognition: Targetize source domain bridged by common subspace,'' *Int. J. Comput. Vis.*, vol. 109, nos. 1–2, pp. 94–109, Aug. 2014.

[498] A. Farhadi, D. Forsyth, and R. White, ''Transfer learning in sign language,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[499] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, ''Visual domain adaptation: A survey of recent advances,'' *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 53–69, May 2015.

[500] L. Shao, F. Zhu, and X. Li, ''Transfer learning for visual categorization: A survey,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 1019–1034, May 2015.

[501] D. A. Adama, A. Lotfi, and R. Ranson, ''A survey of vision-based transfer learning in human activity recognition,'' *Electronics*, vol. 10, no. 19, p. 2412, Oct. 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/19/2412

[502] N. Hernandez-Cruz, C. Nugent, S. Zhang, and I. McChesney, ''The use of transfer learning for activity recognition in instances of heterogeneous sensing,'' *Appl. Sci.*, vol. 11, no. 16, p. 7660, Aug. 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/16/7660

[503] Y. Abdulazeem, H. M. Balaha, W. M. Bahgat, and M. Badawy, ''Human action recognition based on transfer learning approach,'' *IEEE Access*, vol. 9, pp. 82058–82069, 2021.

[504] E. Cole, X. Yang, K. Wilber, O. M. Aodha, and S. Belongie, ''When does contrastive visual representation learning work?'' 2021, *arXiv:2105.05837*.

[505] T. Standley, A. R. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, ''Which tasks should be learned together in multi-task learning?'' 2019, *arXiv:1905.07553*.

[506] Z. Wang, Z. Dai, B. Póczos, and J. Carbonell, ''Characterizing and avoiding negative transfer,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11285–11294.

[507] R. Sousa, L. M. Silva, L. A. Alexandre, J. Santos, and J. M. De Sá, ''Transfer learning: current status, trends and challenges,'' in *Proc. 20th Portuguese Conf. Pattern Recognit., RecPad*, 2014, pp. 57–58.

[508] X. Glorot, A. Bordes, and Y. Bengio, ''Domain adaptation for large-scale sentiment classification: A deep learning approach,'' in *Proc. ICML*, 2011, pp. 513–520.

[509] S. Moon and J. Carbonell, ''Completely heterogeneous transfer learning with attention–what and what not to transfer,'' in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1–2.

[510] E. Strubell, A. Ganesh, and A. McCallum, ''Energy and policy considerations for deep learning in NLP,'' 2019, *arXiv:1906.02243*.

[511] L. F. W. Anthony, B. Kanding, and R. Selvan, ''Carbontracker: Tracking and predicting the carbon footprint of training deep learning models,'' 2020, *arXiv:2007.03051*.

[512] D. Kim, W. Lim, M. Hong, and H. Kim, ''The structure of deep neural network for interpretable transfer learning,'' in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2019, pp. 1–4.

[513] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, ''Learning transferable visual models from natural language supervision,'' 2021, *arXiv:2103.00020*.

[514] J. Lu, D. Batra, D. Parikh, and S. Lee, ''ViLBERT: Pretraining task-agnostic visiolinguistic representations for Vision-and-Language tasks,'' 2019, *arXiv:1908.02265*.

[515] H. Shan, Y. Zhang, Q. Yang, U. Kruger, M. K. Kalra, L. Sun, W. Cong, and G. Wang, ''3-D convolutional encoder–decoder network for low-dose CT via transfer learning from a 2-D trained network,'' *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1522–1534, Jun. 2018.

[516] A. Ebrahimi-Ghahnavieh, S. Luo, and R. Chiong, ''Transfer learning for Alzheimer's disease detection on MRI images,'' in *Proc. IEEE Int. Conf. Ind., Artif. Intell., Commun. Technol. (IAICT)*, Jul. 2019, pp. 133–138.

[517] M. Long, Y. Cao, J. Wang, and M. Jordan, ''Learning transferable features with deep adaptation networks,'' in *Proc. 32nd Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 37, F. Bach and D. Blei, Eds. Lille, France, Jul. 2015, pp. 97–105. [Online]. Available: https://proceedings.mlr.press/v37/long15.html

[518] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, ''Learning transferable architectures for scalable image recognition,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8697–8710.

[519] S. Lee, D. Kim, N. Kim, and S.-G. Jeong, ''Drop to adapt: Learning discriminative features for unsupervised domain adaptation,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 91–100.

[520] X. Wang, Y. Jin, M. Long, J. Wang, and M. I. Jordan, "Transferable normalization: Towards improving transferability of deep neural networks," in *Advances in Neural Information Processing Systems*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/fd2c5e4680d9a01dba3aada5ece22270-Paper.pdf

[521] X. Wang, L. Li, W. Ye, M. Long, and J. Wang, "Transferable attention for domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 5345–5352.

[522] J. Bai, A. Agarwala, M. Agrawala, and R. Ramamoorthi, "Automatic cinemagraph portraits," in *Proc. Eurographics Symp. Rendering (EGSR)*. Goslar, Germany: Eurographics Association, 2013, pp. 17–25, doi: 10.1111/cgf.12147.

[523] Y. Zhou, Y. Song, and T. L. Berg, "Image2GIF: Generating cinemagraphs using recurrent deep Q-Networks," 2018, *arXiv:1801.09042*.

[524] F. Zhang, Y. Li, S. You, and Y. Fu, "Learning temporal consistency for low light video enhancement from single images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4965–4974.

[525] S. Son, S. Lee, S. Nah, R. Timofte, and K. M. Lee, "NTIRE 2021 challenge on video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 166–181.

[526] C. Gao, A. Saraf, J. Kopf, and J.-B. Huang, "Dynamic view synthesis from dynamic monocular video," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5692–5701.

[527] Y. Xu, J. Zhang, and D. Tao, "Out-of-boundary view synthesis towards full-frame video stabilization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4822–4831.

[528] Y.-L. Liu, W.-S. Lai, M.-H. Yang, Y.-Y. Chuang, and J.-B. Huang, "Hybrid neural fusion for full-frame video stabilization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2279–2288.

[529] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh, "Neural volumes: Learning dynamic renderable volumes from images," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 65:1–65:14, Jul. 2019.

[530] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[531] R. Tucker and N. Snavely, "Single-view view synthesis with multiplane images," 2020, *arXiv:2004.11364*.

[532] J. Flynn, M. Broxton, P. Debevec, M. DuVall, G. Fyffe, R. Overbeck, N. Snavely, and R. Tucker, "DeepView: View synthesis with learned gradient descent," 2019, *arXiv:1906.07316*.

[533] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," 2019, *arXiv:1905.00889*.

[534] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," 2019, *arXiv:1901.05103*.

[535] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," 2018, *arXiv:1812.03828*.

[536] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3D-Structure-Aware neural scene representations," 2019, *arXiv:1906.01618*.

[537] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision," 2019, *arXiv:1912.07372*.

[538] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," 2020, *arXiv:2003.08934*.

[539] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," 2020, *arXiv:2011.12948*.

[540] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-NeRF: Neural radiance fields for dynamic scenes," 2020, *arXiv:2011.13961*.

[541] D. Rebain, W. Jiang, S. Yazdani, K. Li, K. Moo Yi, and A. Tagliasacchi, "DeRF: Decomposed radiance fields," 2020, *arXiv:2011.12490*.

[542] B. Jiang, X. Ren, M. Dou, X. Xue, Y. Fu, and Y. Zhang, "LoRD: Local 4D implicit representation for high-fidelity dynamic human modeling," 2022, *arXiv:2208.08622*.

[543] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised NeRF: Fewer views and faster training for free," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12872–12881.

[544] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," 2021, *arXiv:2111.11215*.

[545] A. W. Bergman, P. Kellnhofer, and G. Wetzstein, "Fast training of neural lumigraph representations using meta learning," 2021, *arXiv:2106.14942*.

[546] L. Wu, J. Yong Lee, A. Bhattad, Y. Wang, and D. Forsyth, "DIVeR: Real-time and accurate neural radiance fields with deterministic integration for volume rendering," 2021, *arXiv:2111.10427*.

[547] L. Liu, J. Gu, K. Z. Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 33, 2020, pp. 15651–15663.

[548] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "NeRF++: Analyzing and improving neural radiance fields," 2020, *arXiv:2010.07492*.

[549] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentin, "FastNeRF: High-fidelity neural rendering at 200FPS," 2021, *arXiv:2103.10380*.

[550] L. Wang, J. Zhang, X. Liu, F. Zhao, Y. Zhang, Y. Zhang, M. Wu, J. Yu, and L. Xu, "Fourier PlenOctrees for dynamic radiance field rendering in real-time," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13514–13524.

[551] T. Hu, S. Liu, Y. Chen, T. Shen, and J. Jia, "EfficientNeRF: Efficient neural radiance fields," 2022, *arXiv:2206.00878*.

[552] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, "GRAF: Generative radiance fields for 3D-aware image synthesis," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 33, 2020, pp. 20154–20166.

[553] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "PixelNeRF: Neural radiance fields from one or few images," 2020, *arXiv:2012.02190*.

[554] W. Jang and L. Agapito, "CodeNeRF: Disentangled neural radiance fields for object categories," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12929–12938.

[555] J. Chibane, A. Bansal, V. Lazova, and G. Pons-Moll, "Stereo radiance fields (SRF): Learning view synthesis for sparse views of novel scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7911–7920.

[556] H. Baatz, J. Granskog, M. Papas, F. Rousselle, and J. Novák, "NeRF-tex: Neural reflectance field textures," *Comput. Graph. Forum*, vol. 41, no. 6, pp. 287–301, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14449

[557] X. Chen, Q. Zhang, X. Li, Y. Chen, Y. Feng, X. Wang, and J. Wang, "Hallucinated neural radiance fields in the wild," 2021, *arXiv:2111.15246*.

[558] S. Liu, X. Zhang, Z. Zhang, R. Zhang, J.-Y. Zhu, and B. Russell, "Editing conditional radiance fields," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5753–5763.

[559] B. Yang, Y. Zhang, Y. Xu, Y. Li, H. Zhou, H. Bao, G. Zhang, and Z. Cui, "Learning object-compositional neural radiance field for editable scene rendering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13759–13768.

[560] M. Niemeyer and A. Geiger, "GIRAFFE: Representing scenes as compositional generative neural feature fields," 2020, *arXiv:2011.12100*.

[561] K. Kania, K. Moo Yi, M. Kowalski, T. Trzciński, and A. Tagliasacchi, "CoNeRF: Controllable neural radiance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18602–18611.

[562] J. Sun, X. Wang, Y. Zhang, X. Li, Q. Zhang, Y. Liu, and J. Wang, "FENeRF: Face editing in neural radiance fields," 2021, *arXiv:2111.15490*.

[563] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, "CLIP-NeRF: Text-and-image driven manipulation of neural radiance fields," 2021, *arXiv:2112.05139*.

[564] G. Ras, N. Xie, M. Van Gerven, and D. Doran, "Explainable deep learning: A field guide for the uninitiated," *J. Artif. Intell. Res.*, vol. 73, pp. 329–397, Jan. 2022.

[565] B. Kim, R. Khanna, and O. O. Koyejo, "Examples are not enough, learn to criticize! Criticism for interpretability," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Barcelona, Spain. Red Hook, NY, USA: Curran Associates Inc., 2016, pp. 2288–2296.

[566] J. Ault, J. P. Hanna, and G. Sharon, "Learning an interpretable traffic signal control policy," 2019, *arXiv:1912.11023*.

[567] A. S. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," 2017, *arXiv:1711.09404*.

[568] R. Binns, "Fairness in machine learning: Lessons from political philosophy," 2017, *arXiv:1712.03586*.

[569] T. Küstner, A. Liebgott, L. Mauch, P. Martirosian, F. Bamberg, K. Nikolaou, B. Yang, F. Schick, and S. Gatidis, "Automated reference-free detection of motion artifacts in magnetic resonance images," *Magn. Reson. Mater. Phys., Biol. Med.*, vol. 31, no. 2, pp. 243–256, Apr. 2018, doi: 10.1007/s10334-017-0650-z.

[570] F. Khoshnevisan, J. Ivy, M. Capan, R. Arnold, J. Huddleston, and M. Chi, "Recent temporal pattern mining for septic shock early prediction," in *Proc. IEEE Int. Conf. Healthcare Informat. (ICHI)*, Jun. 2018, pp. 229–240.

[571] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," 2017, *arXiv:1608.05745*.

[572] A. H. Alibak, M. Khodarahmi, P. Fayyazsanavi, S. M. Alizadeh, A. J. Hadi, and E. Aminzadehsarikhanbeglou, "Simulation the adsorption capacity of polyvinyl alcohol/carboxymethyl cellulose based hydrogels towards methylene blue in aqueous solutions using cascade correlation neural network (CCNN) technique," *J. Cleaner Prod.*, vol. 337, Feb. 2022, Art. no. 130509.

[573] J. Pourkia, S. Rahimi, and K. T. Baghaei, "Hospital data interpretation: A self-organizing map approach," in *Fuzzy Techniques: Theory and Applications*, R. B. Kearfott, I. Batyrshin, M. Reformat, M. Ceberio, and V. Kreinovich, Eds. Cham, Switzerland: Springer, 2019, pp. 493–504.

[574] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008. [Online]. Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html

[575] F. Doshi-Velez, B. Wallace, and R. Adams, "Graph-sparse LDA: A topic model with structured sparsity," 2014, *arXiv:1410.4510*.

[576] R. Mayer and H.-A. Jacobsen, "Scalable deep learning on distributed infrastructures: Challenges, techniques, and tools," *ACM Comput. Surv.*, vol. 53, no. 1, pp. 1–37, Feb. 2020, doi: 10.1145/3363554.

[577] C. Boden, A. Spina, T. Rabl, and V. Markl, "Benchmarking data flow systems for scalable machine learning," in *Proc. 4th ACM SIGMOD Workshop Algorithms Syst. MapReduce Beyond (BeyondMR)*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 1–10, doi: 10.1145/3070607.3070612.

[578] Y. Chen and H. Li, "SMALE: Enhancing scalability of machine learning algorithms on extreme scale computing platforms," Duke Univ., Durham, NC, USA, Tech. Rep., Feb. 2022. [Online]. Available: https://www.osti.gov/biblio/1846568

[579] G. De Francisci Morales, "SAMOA: A platform for mining big data streams," in *Proc. 22nd Int. Conf. World Wide Web (WWW)*. New York, NY, USA: Association for Computing Machinery, 2013, pp. 777–778, doi: 10.1145/2487788.2488042.

[580] B. Goankar, K. Cook, and L. Macyszyn, "Ethical issues arising due to bias in training A.I. Algorithms in healthcare and data sharing as a potential solution," *AI Ethics J.*, vol. 1, no. 2, pp. 1–9, Sep. 2020.

[581] G. George, M. R. Haas, and A. Pentland, "From the editors: Big data and management," *Acad. Manage. J.*, vol. 57, no. 2, pp. 321–326, 2014. [Online]. Available: http://www.jstor.org/stable/43589260

[582] B. K. Beaulieu-Jones, Z. S. Wu, C. Williams, and C. S. Greene, "Privacy-preserving generative deep neural networks support clinical data sharing," *Circulat., Cardiovascular Qual. Outcomes*, vol. 12, no. 7, Jul. 2019, Art. no. e005122. [Online]. Available: http://dx.doi.org/10.1161/CIRCOUTCOMES.118.005122

[583] M. Al-Rubaie and J. M. Chang, "Privacy-preserving machine learning: Threats and solutions," *IEEE Secur. Privacy*, vol. 17, no. 2, pp. 49–58, Mar. 2019.

[584] S. Gupta and A. Gupta, "Dealing with noise problem in machine learning data-sets: A systematic review," *Proc. Comput. Sci.*, vol. 161, pp. 466–474, Jan. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919318575

[585] C. F. Caiafa, Z. Sun, T. Tanaka, P. Marti-Puig, and J. Solé-Casals, "Machine learning methods with noisy, incomplete or small datasets," *Appl. Sci.*, vol. 11, no. 9, p. 4132, Apr. 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/9/4132

[586] H. Chen, J. Chen, and J. Ding, "Data evaluation and enhancement for quality improvement of machine learning," *IEEE Trans. Rel.*, vol. 70, no. 2, pp. 831–847, Jun. 2021.

[587] H. Kaur, H. S. Pannu, and A. K. Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–36, Aug. 2019, doi: 10.1145/3343440.

[588] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, pp. 1–54, Dec. 2019.

[589] B. Settles, "Active learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1648, 2009.

[590] A. Coates, B. Huval, T. Wang, D. J. Wu, A. Y. Ng, and B. Catanzaro, "Deep learning with cots HPC systems," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, vol. 28, 2013, pp. III-1337–III-1345.

[591] W. Wang, M. Zhang, G. Chen, H. V. Jagadish, B. C. Ooi, and K.-L. Tan, "Database meets deep learning: Challenges and opportunities," *ACM SIGMOD Rec.*, vol. 45, no. 2, pp. 17–22, Sep. 2016, doi: 10.1145/3003665.3003669.

[592] W. Xiao, R. Bhardwaj, R. Ramjee, M. Sivathanu, N. Kwatra, Z. Han, P. Patel, X. Peng, H. Zhao, Q. Zhang, F. Yang, and L. Zhou, "Gandiva: Introspective cluster scheduling for deep learning," in *Proc. 13th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*. Carlsbad, CA, USA: USENIX Association, Oct. 2018, pp. 595–610. [Online]. Available: https://www.usenix.org/conference/osdi18/presentation/xiao

[593] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman, "Project Adam: Building an efficient and scalable deep learning training system," in *Proc. 11th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*. Broomfield, CO, USA: USENIX Association, Oct. 2014, pp. 571–582. [Online]. Available: https://www.usenix.org/conference/osdi14/technical-sessions/presentation/chilimbi

[594] H. Cui, H. Zhang, G. R. Ganger, P. B. Gibbons, and E. P. Xing, "GeePS: Scalable deep learning on distributed GPUs with a GPU-specialized parameter server," in *Proc. 11th Eur. Conf. Comput. Syst.*, Apr. 2016, pp. 1–16.

[595] R. Vuduc, A. Chandramowlishwaran, J. Choi, M. Guney, and A. Shringarpure, "On the limits of GPU acceleration," in *Proc. 2nd USENIX Conf. Hot Topics Parallelism (HotPar)*. Berkeley, CA, USA: USENIX Association, 2010, p. 13.

[596] L. You, H. Jiang, J. Hu, C. H. Chang, L. Chen, X. Cui, and M. Zhao, "GPU-accelerated faster mean shift with Euclidean distance metrics," in *Proc. IEEE 46th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, Jun. 2022, pp. 211–216.

[597] H. Cheng, M. Zhang, and J. Q. Shi, "A survey on deep neural network pruning-taxonomy, comparison, analysis, and recommendations," 2023, *arXiv:2308.06767*.

[598] Y. He and L. Xiao, "Structured pruning for deep convolutional neural networks: A survey," 2023, *arXiv:2303.00566*.

[599] N. Lee, T. Ajanthan, and P. Torr, "SNIP: Single-shot network pruning based on connection sensitivity," 2019, *arXiv:1810.02340*.

[600] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*.

[601] T. Zhang and Z. Zhu, "Interpreting adversarially trained convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7502–7511.

[602] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," 2017, *arXiv:1708.08296*.

[603] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Attention branch network: Learning of attention mechanism for visual explanation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10697–10706.

[604] K. Dwivedi, G. Roig, A. Kembhavi, and R. Mottaghi, "What do navigation agents learn about their environment?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10266–10275.

[605] T. Thrush, R. Jiang, M. Bartolo, A. Singh, A. Williams, D. Kiela, and C. Ross, "Winoground: Probing vision and language models for visio-linguistic compositionality," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5228–5238.

[606] K. Sinha, R. Jia, D. Hupkes, J. Pineau, A. Williams, and D. Kiela, "Masked language modeling and the distributional hypothesis: Order word matters pre-training for little," 2021, *arXiv:2104.06644*.

[607] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

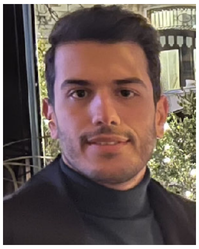
[608] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[609] H. Xu, Y. Ma, H.-C. Liu, D. Deb, H. Liu, J.-L. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *Int. J. Autom. Comput.*, vol. 17, no. 2, pp. 151–178, Apr. 2020.

[610] G. R. Machado, E. Silva, and R. R. Goldschmidt, "AAdversarial machine learning in image classification: A survey toward the defender's perspective," *ACM Comput. Surv.*, vol. 55, no. 1, pp. 1–38, 2021.
[611] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu, "Benchmarking adversarial robustness on image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 318–328.
[612] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," 2017, *arXiv:1711.00117*.
[613] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial attacks on deep-learning models in natural language processing: A survey," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 3, pp. 1–41, May 2020.
[614] E. Wallace, M. Gardner, and S. Singh, "Interpreting predictions of NLP models," in *Proc. Conf. Empirical Methods Natural Lang. Process., Tutorial Abstr.*, 2020, pp. 20–23.
[615] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5231–5240.
[616] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? Adversarial examples against automatic speech recognition," 2018, *arXiv:1801.00554*.
[617] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 1–7.
[618] F. Karim, S. Majumdar, and H. Darabi, "Adversarial attacks on time series," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3309–3320, Oct. 2021.
[619] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Adversarial attacks on deep neural networks for time series classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.
[620] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Characterizing and evaluating adversarial examples for offline handwritten signature verification," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 8, pp. 2153–2166, Aug. 2019.
[621] A. K. Jain, D. Deb, and J. J. Engelsma, "Biometrics: Trust, but verify," 2021, *arXiv:2105.06625*.
[622] J. Fei, Z. Xia, P. Yu, and F. Xiao, "Adversarial attacks on fingerprint liveness detection," *EURASIP J. Image Video Process.*, vol. 2020, no. 1, pp. 1–11, Dec. 2020.
[623] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 4–20, Jan. 2004.
[624] H. Zhao, J. Chi, Y. Tian, and G. J. Gordon, "Trade-offs and guarantees of adversarial representation learning for information obfuscation," in *Advances in Neural Information Processing Systems*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates, 2020, pp. 9485–9496. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/6b8b8e3bd6ad94b985c1b1f1b7a94cb2-Paper.pdf
[625] D. Kang, Y. Sun, T. Brown, D. Hendrycks, and J. Steinhardt, "Transfer of adversarial robustness between perturbation types," 2019, *arXiv:1905.01034*.

AMIRREZA PAYANDEH received the Master of Science (M.S.) degree in computer science from the University of North Carolina at Charlotte. He is currently pursuing the Ph.D. degree in computer science with George Mason University. He gained a solid understanding of the field with the University of North Carolina at Charlotte. His work significantly revolves around robotics, natural language processing, and computer vision applications as an Artificial Intelligence Scientist.

KOUROSH T. BAGHAEI (Member, IEEE) received the M.S. degree in computer science from Mississippi State University. He is currently pursuing the Ph.D. degree with George Mason University, Fairfax, VA, USA. His research interests include natural language processing and vision and language navigation.



POOYA FAYYAZSANAVI received the Master of Science (M.S.) degree in computer science from George Mason University, where he is currently pursuing the Ph.D. degree in computer science. He acquired a strong foundation in the field with George Mason University. He is also an accomplished Computer Vision Scientist with a proven track record in the research industry. His research interests include computer vision, robotics, and data mining applications, he possesses expertise in human pose estimation, self-supervised learning, object pose estimation, and navigation. He is also actively engaged in the development of self-supervised-based models in deep learning, pushing the boundaries of the field, and exploring the potential of unlabeled data. His extensive knowledge of computer vision tasks, coupled with a commitment to rigorous research methodologies, has led to significant contributions to academic publications and presentations at conferences.

SOMAYEH BAKHTIARI RAMEZANI (Member, IEEE) received the B.S. degree in computer engineering and the M.S. degree in information technology engineering from the Iran University of Science and Technology, in 2004 and 2008, respectively. She is currently pursuing the Ph.D. degree in computer science with Mississippi State University. She is also a Graduate Research Assistant with the Predictive Analytics and Technology Integration (PATENT) Laboratory in collaboration with the Institute for Systems Engineering Research. Prior to joining Mississippi State University, in 2019, she was with several companies in the energy and healthcare sectors as a HPC Programmer and a Data Scientist. She is also a 2021 SIGHPC Computational and Data Science Fellow. Her research interests include probabilistic modeling and optimization of dynamic systems, the application of ML, quantum computation, and time-series segmentation in the healthcare sector. She is a member of ACM, the President of the ACM-W Student Chapter with Mississippi State University, and the Chair of the IEEE-WIE AG Mississippi Section.

ZHIQIAN CHEN is currently an Assistant Professor with the Department of Computer Science and Engineering, Mississippi State University. Before joining Mississippi State University, in 2020, he was a Research Assistant with Virginia Tech. At present, he is engaged in the study of machine learning as it pertains to graph and network problems. His research interests include network flow and its various applications, including but not limited to brain networks, power networks, biological networks, traffic networks, and social networks.

SHAHRAM RAHIMI (Member, IEEE) is currently a Professor and the Head of the Department of Computer Science and Engineering, Mississippi State University. Prior to that, he led the Department of Computer Science, Southern Illinois University, for five years. He is also a recognized Leader in the area of artificial and computational intelligence, with over 220 peer-reviewed publications and a few patents or pending patents in this area. He is a member of the IEEE New Standards Committee in Computational Intelligence. He provides advice to a staff and administration at the federal government on predictive analytics for foreign policy. He was a recipient of the 2016 Illinois Rising Star Award from ISBA, selected among 100s of highly qualified candidates. His intelligent algorithm for patient flow optimization and hospital staffing is currently used in over 1000 emergency departments across the nation. He was named one of the top ten AI technology for healthcare, in 2018, by *HealthTech Magazines*. He has secured over 20M of federal and industry funding as a PI or a co-PI in the last 20 years. He has also organized 15 conferences and workshops in the areas of computational intelligence and multi-agent systems over the past two decades. He has served as the Editor-in-Chief for two leading computational intelligence journals and is on the editorial board for several other journals.