mathematics

MDPI

Review

The Geometry of Feature Space in Deep Learning Models: A Holistic Perspective and Comprehensive Review

Minhyeok Lee

School of Electrical and Electronics Engineering, Chung-Ang University, Seoul 06974, Republic of Korea; mlee@cau.ac.kr

Abstract: As the field of deep learning experiences a meteoric rise, the urgency to decipher the complex geometric properties of feature spaces, which underlie the effectiveness of diverse learning algorithms and optimization techniques, has become paramount. In this scholarly review, a comprehensive, holistic outlook on the geometry of feature spaces in deep learning models is provided in order to thoroughly probe the interconnections between feature spaces and a multitude of influential factors such as activation functions, normalization methods, and model architectures. The exploration commences with an all-encompassing examination of deep learning models, followed by a rigorous dissection of feature space geometry, delving into manifold structures, curvature, wide neural networks and Gaussian processes, critical points and loss landscapes, singular value spectra, and adversarial robustness, among other notable topics. Moreover, transfer learning and disentangled representations in feature space are illuminated, accentuating the progress and challenges in these areas. In conclusion, the challenges and future research directions in the domain of feature space geometry are outlined, emphasizing the significance of comprehending overparameterized models, unsupervised and semi-supervised learning, interpretable feature space geometry, topological analysis, and multimodal and multi-task learning. Embracing a holistic perspective, this review aspires to serve as an exhaustive guide for researchers and practitioners alike, clarifying the intricacies of the geometry of feature spaces in deep learning models and mapping the trajectory for future advancements in this enigmatic and enthralling domain.

Keywords: feature space geometry; deep learning models; manifold structures; disentangled representations

MSC: 68T27

Citation: Lee, M. The Geometry of Feature Space in Deep Learning Models: A Holistic Perspective and Comprehensive Review. *Mathematics* **2023**, *11*, 2375. https://doi.org/10.3390/math11102375

Academic Editors: Zibin Zheng, Ruoxi Jia, Dan Li, Yuxun Zhou and Liang Xu

Received: 7 April 2023
Revised: 3 May 2023
Accepted: 18 May 2023
Published: 19 May 2023

1. Introduction

The pervasive presence of deep learning models in contemporary artificial intelligence research and applications has given rise to an urgent need for a thorough understanding of the underlying structures and properties of these models [1–5]. One of the most prominent and yet enigmatic aspects of deep learning models is the geometry of feature spaces [6], which constitutes the foundation upon which various learning algorithms and optimization techniques are established. In this scholarly work, a holistic perspective on the geometry of feature space in deep learning models hopes to be offered by meticulously examining the interconnections between feature spaces and a plethora of factors that impact their geometrical properties, such as activation functions, normalization methods, and model architectures [6–13].

The exploration begins with a review of the background in deep learning, encompassing a diverse array of models such as feedforward neural networks (FNNs), convolutional neural networks (CNNs), and transformer models, as well as an investigation of activation functions and normalization methods. Subsequently, an in-depth analysis of the geometry of feature space is undertaken, scrutinizing the relationships between feature space and

activation functions, and probing the interplay between feature space and normalization methods. Throughout the investigation, various aspects of isotropy, connectivity, and the combination of ReLU-inspired activations with normalization methods are investigated.

After a thorough examination of recent studies that have informed the understanding of feature space geometry, manifold structures, curvature, wide neural networks and Gaussian processes, critical points and loss landscapes, singular value spectra, and adversarial robustness, among other topics, are discussed. The state of the art in transfer learning and disentangled representations in feature space is also evaluated, emphasizing the advances and challenges in these areas.

As a deeper dive into the topic of deep learning is taken, the limitations in current review papers that concentrate on particular aspects or applications within the field [14–18] become apparent, while previous studies have made notable progress in shedding light on various aspects of deep learning models, a gap persists in providing a truly comprehensive and integrative understanding of the geometry of feature spaces. In particular, the extant literature often concentrates on discrete aspects, such as manifold structures, curvature, or adversarial robustness, without adequately situating these elements within the broader framework of deep learning models. Additionally, the rapidly evolving nature of deep learning research calls for a current synthesis that encompasses the most recent findings and methodological advancements.

To address these gaps, several innovative contributions to the field are offered in the current work. First, a cogent and unified framework that cohesively integrates the diverse aspects of feature space geometry in deep learning is introduced, promoting a comprehensive understanding of the subject. Second, by leveraging a collection of state-of-the-art research, a timely overview of the most recent breakthroughs and trends in the field is delivered, making it a valuable resource for both experienced professionals and newcomers alike. Lastly, the exploration of the challenges and future research directions in feature space geometry serves to stimulate and direct further investigation, thus paving the way for groundbreaking advancements in this intricate and compelling area.

A set of inclusion and exclusion criteria was established to ensure a focused and comprehensive mathematical examination of the geometry of feature space in deep learning. Studies were included if they primarily discussed the mathematical properties of feature spaces in deep learning models, with an emphasis on their geometrical aspects. In contrast, studies were excluded if they primarily focused on applications or implementation aspects without a significant contribution to the understanding of feature space geometry. This selection process allowed for concentration of efforts on providing a thorough and up-to-date review of the most relevant research in this domain.

The research questions examined in this paper are centered on providing a comprehensive mathematical understanding of the geometry of feature spaces in deep learning models. The relationships between feature spaces and activation functions, as well as the synergistic integration of ReLU-inspired activation functions and normalization techniques, are explored. Furthermore, how these interactions can lead to mutual benefits and potential improvements in model performance is investigated. The review also covers a wide range of topics in feature space geometry, including manifold structures, curvature, critical points, and adversarial robustness, as well as transfer learning and disentangled representations in feature space.

A series of distinct yet interconnected contributions aimed at providing a comprehensive understanding of the geometry of feature spaces in deep learning models is presented. Specifically, (1) a novel perspective on the relationships between feature space and activation functions in deep learning is offered, providing the intricate interplay between these key components; (2) the synergistic integration of ReLU-inspired activation functions and normalization techniques is explored, revealing the mutual benefits and potential improvements in model performance; (3) an extensive review of recent studies in feature space geometry in deep learning is provided, covering a wide range of topics such as manifold structures, curvature, critical points, and adversarial robustness; (4) the current

state of transfer learning and disentangled representations in feature space is discussed, highlighting both achievements and challenges in these areas; and (5) the challenges and future research directions in feature space geometry are identified and outlined, including overparameterized models, unsupervised and semi-supervised learning, interpretable geometry, topological analysis, and multimodal and multi-task learning. Collectively, these contributions serve to advance our understanding of feature space geometry in deep learning, providing a solid foundation for further exploration and innovation in this rapidly evolving domain.

The exploration of feature space geometry in deep learning is organized into several sections to provide a structured and coherent overview. Section 2 offers essential background information on deep learning architectures, activation functions, and normalization methods. Section 3 provides perspectives on the relationships between feature spaces and various factors that impact their geometrical properties. Section 4 delves into the recent studies that have shaped our understanding of feature space geometry, examining a wide range of topics and approaches. Section 5 discusses the challenges and future research directions in feature space geometry, outlining the areas where progress is needed to advance our knowledge. Finally, in Section 6, the conclusions are presented and the implications of the findings for the broader field of deep learning are reflected upon.

2. Background

2.1. Deep Learning Architectures

Deep learning architectures encompass a diverse array of artificial neural networks specifically engineered to unravel and interpret hierarchical structures within input data. These intricate models are composed of a multitude of interconnected layers, each diligently processing and transforming input data into increasingly abstract representations. In the following sections, the fundamental types of deep learning architectures and their mathematical underpinnings are explored, with a focus on feedforward neural networks, CNNs, and cutting-edge transformer models.

2.1.1. Feedforward Neural Networks (FNNs)

Feedforward neural networks (FNNs) represent the most elementary and foundational type of deep learning models [19]. They are comprised of input, hidden, and output layers that work in tandem to receive high-dimensional data, process it, and ultimately generate the final prediction. The layers are interconnected through synaptic weights, and the neurons within each layer employ activation functions to incorporate nonlinearity into the model. Consider a compact subset $\mathcal{X} \subset \mathbb{R}^D$ that characterizes the latent space of a sample space with high dimensionality, where D denotes the hypothetical dimension of the latent space. The input layer can be expressed as a vector $\mathbf{x} \in \mathcal{R}(\mathcal{X})$, with $\mathcal{R}$ symbolizing the mapping between the latent space and the sample space. The output of each layer can be formulated as follows:

$$\mathbf{h}^{(l)} = f^{(l)}(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}) \tag{1}$$

where $\mathbf{h}^{(l)}$ signifies the output of the l-th layer, $\mathbf{W}^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ denotes the weight matrix interconnecting layers $l-1$ and l, $\mathbf{b}^{(l)} \in \mathbb{R}^{n_l}$ represents the bias vector for layer l, n_l corresponds to the number of neurons in layer l, and $f^{(l)} : \mathbb{R}^{n_l} \to \mathbb{R}^{n_l}$ is the element-wise activation function. Widely adopted activation functions encompass the rectified linear unit (ReLU) $f(x) = \max(0, x)$, sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$, and hyperbolic tangent $\tanh(x)$.

For the input layer, $\mathbf{h}^{(0)} = \mathbf{x}$ holds, while the ultimate output layer L furnishes the prediction, denoted by $\mathbf{h}^{(L)}$. The FNN maps the high-dimensional data from the finite set $\mathcal{R}(\mathcal{X})$ to the output space via a sequence of transformations, encoding the data's inherent structure in the hidden layers' activations. This chain of transformations empowers the FNN to discern intricate patterns and associations within the high-dimensional data embedded in the compact set of the latent space.

The transformations applied to the input data as it advances through the network can be meticulously examined to comprehend how the hidden layers of FNNs accurately emulate the latent space. Given $\mathcal{X} \subset \mathbb{R}^D$ within the latent space and $\mathbf{x} \in \mathcal{R}(\mathcal{X})$, the transformations transpire through the weight matrices $\mathbf{W}^{(l)}$ and activation functions $f^{(l)}$, as explicated in Equation (1).

The transformations executed by the hidden layers can be perceived as a succession of mappings $\Phi^{(l)} : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$, where n_l corresponds to the quantity of neurons in layer l. The mapping function for layer l is articulated by:

$$\Phi^{(l)}(\mathbf{h}^{(l-1)}) = f^{(l)}(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}) \tag{2}$$

The synthesis of these mappings constitutes the comprehensive transformation employed by the FNN. For a network encompassing L layers, the ultimate mapping $\Phi : \mathcal{R}(\mathcal{X}) \to \mathbb{R}^{n_L}$ is delineated by:

$$\Phi(\mathbf{x}) = \Phi^{(L)} \circ \Phi^{(L-1)} \circ \cdots \circ \Phi^{(1)}(\mathbf{x}) \tag{3}$$

The approximation of $\mathcal{X}$ can be interpreted as the FNN's capacity to acquire a mapping $\Phi^{(l)}$ that renders the data analogous to $\mathcal{X}$, such that the distances between points in the input space are manifested in the hidden layer activations with the approximation of the latent space. This concept can be formalized via the notion of a Lipschitz continuous mapping, where for a specific constant $K > 0$:

$$|\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2)| \leq K|\mathbf{x}_1 - \mathbf{x}_2|, \quad \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X} \tag{4}$$

When an FNN is proficient in learning a Lipschitz continuous mapping Φ that fulfills the inequality stipulated in Equation (4), it can efficaciously approximate the latent space by conserving the data structure within the hidden layers' activations. This endows the FNN with the ability to discern intricate patterns and associations within the high-dimensional data, ultimately bolstering its generalization capabilities.

2.1.2. Convolutional Neural Networks (CNNs)

Convolutional neural networks (CNNs) represent a distinct class of deep learning models, predominantly tailored for processing grid-structured data, such as images. CNNs utilize convolutional layers that perform convolution operations on input data, enabling the model to discern local patterns and spatial hierarchies [20]. A convolution operation can be expressed as:

$$\mathbf{h}_{i,j}^{(l)} = f^{(l)}\left(\sum_{m,n} \mathbf{W}_{m,n}^{(l)} \mathbf{h}_{i+m,j+n}^{(l-1)} + \mathbf{b}^{(l)}\right) \tag{5}$$

where $\mathbf{h}_{i,j}^{(l)}$ signifies the output of the l-th convolutional layer at position (i, j), $\mathbf{W}_{m,n}^{(l)}$ denotes the weights of the convolutional kernel at position (m, n), and the sum encompasses the spatial extent of the kernel. CNNs frequently integrate pooling layers, which minimize the spatial dimensions of the feature maps, and fully connected layers, analogous to those present in FNNs.

To obtain an exhaustive understanding of how CNNs' hidden layers efficaciously reconstruct the compact subspace of the latent space containing spatial information, a thorough examination of the unique structure and operations of CNNs is required, concentrating particularly on the convolutional and pooling layers. These specialized layers are explicitly devised to capture local patterns and spatial hierarchies in grid-like data, such as images, more efficiently than FNNs. By leveraging these intrinsic advantages, CNNs can adeptly extract and represent the most pertinent and informative features of the input data, facilitating superior pattern recognition and classification accuracy.

Suppose that high-dimensional data is possessed, where the data exhibits robust spatial correlations by the properties of $\mathcal{X}$. The input layer of a CNN can be represented as a tensor $\mathbf{X} \in \mathcal{R}(\mathcal{X})$, and the output of each convolutional layer can be expressed as:

$$\mathbf{H}_{i,j}^{(l)} = f^{(l)}\left(\sum_{m,n} \mathbf{W}_{m,n}^{(l)} \mathbf{H}_{i+m,j+n}^{(l-1)} + \mathbf{b}^{(l)}\right) \tag{6}$$

where $\mathbf{H}_{i,j}^{(l)}$ represents the output of the l-th convolutional layer at position (i, j), $\mathbf{W}_{m,n}^{(l)}$ signifies the weights of the convolutional kernel at position (m, n), and the sum spans the spatial extent of the kernel.

In conjunction with convolutional layers, CNNs frequently integrate pooling layers that reduce the spatial dimensions of feature maps while retaining the most prominent spatial information. A pooling layer can be portrayed as:

$$\mathbf{P}_{i,j}^{(l)} = g^{(l)}\left(\mathbf{H}_{i:i+M,j:j+N}^{(l)}\right) \tag{7}$$

where $\mathbf{P}_{i,j}^{(l)}$ denotes the output of the l-th pooling layer at position (i, j), M and N represent the dimensions of the pooling window, and $g^{(l)}$ constitutes the pooling operation, such as max or average pooling.

The synergistic effect of convolutional and pooling layers in a CNN enables the more efficient recovery of the compact subspace of the latent space of spatial information by learning a series of mappings $\Psi^{(l)} : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$, where n_l signifies the number of features in layer l. The comprehensive transformation applied by the CNN can be expressed as:

$$\Psi(\mathbf{X}) = \Psi^{(L)} \circ \Psi^{(L-1)} \circ \cdots \circ \Psi^{(1)}(\mathbf{X}) \tag{8}$$

2.1.3. Transformer Models and Attention Mechanism

Transformer models have revolutionized natural language processing and are now widely employed for various sequential data tasks. These models rely on self-attention mechanisms to capture long-range dependencies in input data, rather than using recurrent or convolutional architectures. The fundamental building block of a transformer model is the scaled dot-product self-attention mechanism, which can be expressed as follows:

$$\textbf{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{9}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the query, key, and value matrices, respectively, and d_k is the dimension of the key vectors. These matrices are derived from the input embeddings through linear transformations using learnable weight matrices $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}^Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}^K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}^V \tag{10}$$

Transformer models employ multi-head attention to learn different types of dependencies in the input data. The output of each attention head is concatenated and linearly transformed to produce the final output:

$$\textbf{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}^O \tag{11}$$

where $\text{head}_i = \textbf{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i)$, and $\mathbf{W}^O$ is a learnable weight matrix. The transformer architecture consists of multiple layers of multi-head attention, followed by position-wise feedforward layers and layer normalization.

2.2. Activation Functions in Deep Learning

The choice of activation function in a deep learning model has a significant impact on the geometry of the feature space. The relationship between the choice of activation function and the geometry of the feature space is discussed in this section [21].

The activation function σ is typically chosen to introduce nonlinearity into the model. Common choices include the rectified linear unit (ReLU), sigmoid, and hyperbolic tangent (tanh) functions. The choice of activation function affects the shape of the decision boundary and the geometry of the feature space.

For example, the ReLU activation function is defined as:

$$\sigma(x) = \max(0, x) \tag{12}$$

The ReLU function introduces sparsity into the model, as it sets negative values to zero. This sparsity can lead to a fragmented and disconnected feature space, as some regions of the input space may be completely separated from others.

Conversely, the sigmoid activation function is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{13}$$

The sigmoid function introduces smoothness into the model, as it maps any input to a value between 0 and 1. This can lead to a more connected feature space, as points that are nearby in the input space are likely to have similar feature representations.

The choice of activation function can also affect the curvature of the feature space. For example, the hyperbolic tangent (tanh) function is defined as:

$$\sigma(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{14}$$

The tanh function maps any input to a value between -1 and 1, which can lead to a feature space with negative curvature. This negative curvature can be advantageous in some applications, as it allows the model to learn more complex decision boundaries.

The Gaussian Error Linear Unit (GELU) activation function [22] is a newer activation function that has shown improved performance over the commonly used ReLU function. The GELU function is defined as:

$$\sigma(x) = x\left(\frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right]\right) \tag{15}$$

where erf is the error function. GELU combines the smoothness of sigmoid and tanh functions with the sparsity of ReLU, leading to a feature space that can balance the trade-off between connectivity and fragmentation, thereby improving the model's ability to learn complex patterns and generalize to unseen data.

The Exponential Linear Unit (ELU) activation function [23] is an alternative to the ReLU function. The ELU function is defined as:

$$\sigma(x) = \begin{cases} x, & x \geq 0 \\ \alpha(e^{x} - 1), & x < 0 \end{cases} \tag{16}$$

where α is a hyperparameter that controls the slope of the negative part of the function. Like the GELU function, the ELU function aims to be more robust to the vanishing gradient problem compared to the ReLU function. One interesting property of the ELU function is that it has a smooth gradient, unlike the ReLU function, which has a discontinuous gradient at zero. This smoothness can make the ELU function easier to optimize using gradient-based methods.

The Scaled Exponential Linear Unit with Squish (SWISH) activation function [24] is a more recent alternative to the ReLU function, showing promise in achieving state-of-the-art performance on several benchmarks. The SWISH function is defined as:

$$\sigma(x) = x\sigma(\beta x) \tag{17}$$

where σ is the sigmoid function and β is a trainable parameter. The SWISH function has a similar shape to the ReLU function but with a smooth gradient. The SWISH function possesses a gating mechanism that dynamically adjusts the slope of the function based on the input, making it more adaptable to different types of input data.

The activation functions are summarized in Table 1. The choice of activation function can also affect the curvature of the feature space. For example, the tanh function can lead to a feature space with negative curvature, which can be advantageous in some applications, as it allows the model to learn more complex decision boundaries. The GELU and ELU activation functions address the limitations of the ReLU function, such as the vanishing gradient problem and the discontinuous gradient at zero. The GELU function presents a smooth approximation to the ReLU function and can be viewed as a form of soft attention mechanism. The ELU function, with its smooth gradient, can be easier to optimize using gradient-based methods compared to the ReLU function.

Table 1. Representative Activation Functions in Deep Learning.

Activation Function	Mathematical Formula	Descriptions
ReLU	$\sigma(x) = \max(0, x)$	Introduces sparsity into the model
Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$	Introduces smoothness into the model
Hyperbolic tangent (tanh)	$\sigma(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	Can lead to a feature space with negative curvature
Gaussian Error Linear Unit (GELU)	$\sigma(x) = x\left(\frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right]\right)$	Combines smoothness with sparsity
Exponential Linear Unit (ELU)	$\sigma(x) = \begin{cases} x, & x \geq 0 \\ \alpha(e^{x} - 1), & x < 0 \end{cases}$	Aims to be more robust to the vanishing gradient problem
Scaled Exponential Linear Unit with Squish (SWISH)	$\sigma(x) = x\sigma(\beta x)$	Possesses a gating mechanism that dynamically adjusts the slope of the function

2.3. Normalization Methods in Deep Learning

Normalization is a prevalent technique in deep learning to enhance the performance of models. Normalization methods aim to rescale the activations in the feature space so that they have zero mean and unit variance. The relationship between the choice of normalization method and the geometry of the feature space is discussed in this section.

Let $f(x)$ be a deep neural network with m layers and activation function $f^{(l)}$. The output of the i-th layer is denoted by $\mathbf{h}^{(i)}(x) \in \mathbb{R}^{d_i}$, where d_i is the dimensionality of the feature space at layer i.

Batch normalization (BN) [25] is a widely used normalization method in deep learning. It aims to rescale the activations in the feature space so that they have zero mean and unit variance with respect to the mini-batch of input data. BN is defined as:

$$\text{BN}(\mathbf{h}^{(i)}(x)) = \gamma_i \frac{\mathbf{h}^{(i)}(x) - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} + \beta_i \tag{18}$$

where μ_i and σ_i^2 are the mean and variance of the activations in the mini-batch, ϵ is a small constant to prevent division by zero, and γ_i and β_i are learnable parameters controlling the scaling and shifting of the activations.

Layer normalization (LN) [26] is another common normalization method. It aims to rescale the activations in the feature space so that they have zero mean and unit variance with respect to the entire layer of input data. LN is defined as:

$$\mathrm{LN}(\mathbf{h}^{(i)}(x)) = \gamma_i \frac{\mathbf{h}^{(i)}(x) - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta_i \tag{19}$$

where μ and σ^2 are the mean and variance of the activations in the entire layer, and γ_i and β_i are learnable parameters controlling the scaling and shifting of the activations.

Group normalization (GN) [27] is another normalization technique that has been shown to improve the performance of deep learning models. GN is similar to BN, but instead of normalizing the activations with respect to the mini-batch, it normalizes the activations with respect to groups of channels. In GN, the output of the i-th layer is denoted by $\mathbf{h}^{(i)}(x) \in \mathbb{R}^{d_i \times c_i \times h_i \times w_i}$, where d_i, c_i, h_i, and w_i are the depth, number of channels, height, and width of the feature map at layer i, respectively. The output of the final layer, $\mathbf{h}^{(m)}(x)$, is the feature representation of the input x.

GN divides the channels into G groups and normalizes the activations within each group separately. The mean and variance used to normalize the activations are computed only over the channels within each group. The GN operation is defined as:

$$\mathrm{GN}(\mathbf{h}^{(i)}(x)) = \gamma_g \frac{\mathbf{h}^{(i)}(x) - \mu_g}{\sqrt{\sigma_g^2 + \epsilon}} + \beta_g \tag{20}$$

where μ_g and σ_g^2 are the mean and variance of the activations within group g, and ϵ, γ_g, and β_g are learnable parameters that control the scaling and shifting of the activations.

The GN is a remarkable technique that has become increasingly popular in deep learning due to its many advantages. One significant benefit of GN is that it can drastically reduce the reliance on the mini-batch size during training. Unlike other normalization techniques, GN normalizes the activations within each group separately, which can make it more suitable for small batch sizes or when the batch size fluctuates during training.

Additionally, GN can lead to a more isotropic feature space than other normalization techniques, such as BN. This is because GN normalizes the activations with respect to groups of channels, which can effectively minimize the impact of the channel dimension on the normalization process. Consequently, GN is particularly beneficial for models with deep or wide architectures, where the channel dimension may be large.

Normalization methods play a significant role in deep learning to enhance the performance of models by rescaling activations in the feature space. This section covered three representative normalization methods in deep learning, including BN, LN, and GN, and provided their mathematical formulas and descriptions. BN rescales activations in the feature space with respect to mini-batch mean and variance, while LN rescales activations with respect to layer mean and variance. In contrast, GN rescales activations with respect to groups of channels, which can make it more suitable for small batch sizes or deep and wide architectures. Table 2 provides a summary of the key features of these normalization methods. Choosing the most appropriate normalization method depends on the nature of the data and the specific deep learning task at hand.

Table 2. Representative Normalization Methods in Deep Learning.

Normalization Methods	Mathematical Formula	Descriptions
Batch Normalization (BN)	$\gamma_i \frac{\mathbf{h}^{(i)}(x)-\mu_i}{\sqrt{\sigma_i^2+\epsilon}} + \beta_i$	Rescales activations in feature space with respect to mini-batch mean and variance
Layer Normalization (LN)	$\gamma_i \frac{\mathbf{h}^{(i)}(x)-\mu}{\sqrt{\sigma^2+\epsilon}} + \beta_i$	Rescales activations in feature space with respect to layer mean and variance
Group Normalization (GN)	$\gamma_g \frac{\mathbf{h}^{(i)}(x)-\mu_g}{\sqrt{\sigma_g^2+\epsilon}} + \beta_g$	Rescales activations in feature space with respect to groups of channels

3. Feature Space Geometry in Deep Learning

3.1. Relationships between Feature Space and Activation Functions in Deep Learning

The selection of an activation function can have a profound impact on the geometry of the feature space, dictating vital aspects such as the shape of the decision boundary, the sparsity or smoothness of the feature space, and the curvature of the feature space. To elucidate the mathematical underpinnings of these relationships, a detailed analysis shall be provided on the effects of various activation functions on the feature space geometry. In particular, the effect of the activation function on the decision boundary can be succinctly captured by the gradient of the output with respect to the input:

$$\nabla_x \mathbf{h}^{(l)} = \nabla_x f^{(l)}(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}). \tag{21}$$

For the ReLU activation function, the gradient is piecewise constant:

$$\nabla_x \sigma(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}. \tag{22}$$

This piecewise constant gradient leads to a fragmented and disconnected feature space, as some regions of the input space may be completely separated from others.

For the sigmoid activation function, the gradient is smooth and can be expressed as:

$$\nabla_x \sigma(x) = \sigma(x)(1 - \sigma(x)). \tag{23}$$

This smooth gradient leads to a more connected feature space, as points that are nearby in the input space are likely to have similar feature representations.

For the tanh activation function, the gradient is also smooth and can be expressed as:

$$\nabla_x \sigma(x) = 1 - \sigma(x)^2. \tag{24}$$

The tanh function can lead to a feature space with negative curvature, as its gradient is nonmonotonic and can change sign depending on the input value. This negative curvature allows the model to learn more complex decision boundaries.

The curvature of the feature space can be further analyzed by computing the Hessian matrix, which is the matrix of second-order partial derivatives of the output with respect to the input:

$$\mathbf{H}(x) = \nabla_x^2 \mathbf{h}^{(i)}(x) = \nabla_x^2 f^{(i)}(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)}). \tag{25}$$

The eigenvalues of the Hessian matrix determine the local curvature of the feature space. For example, if all eigenvalues are positive, the feature space has a positive curvature, while if some eigenvalues are negative, the feature space has a negative curvature.

For the GELU activation function, the gradient is smooth and can be expressed as:

$$\nabla_x \sigma(x) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right] + \frac{x}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right). \tag{26}$$

The GELU function presents a smooth approximation to the ReLU function, combining the smoothness of sigmoid and tanh functions with the sparsity of ReLU. This leads to a feature space that can balance the trade-off between connectivity and fragmentation, thereby improving the model's ability to learn complex patterns and generalize to unseen data.

For the ELU activation function, the gradient is smooth and can be expressed as:

$$\nabla_x \sigma(x) = \begin{cases} 1, & x \geq 0 \\ \alpha e^x, & x < 0 \end{cases}. \tag{27}$$

With its smooth gradient, the ELU function can be easier to optimize using gradient-based methods compared to the ReLU function. The curvature of the feature space induced by the ELU function depends on the value of the hyperparameter α and the input value x.

For the SWISH activation function, the gradient is smooth and can be expressed as:

$$\nabla_x \sigma(x) = \beta\sigma(\beta x) + (1 - \beta\sigma(\beta x))x\sigma(\beta x). \tag{28}$$

The SWISH function possesses a gating mechanism that dynamically adjusts the slope of the function based on the input, making it more adaptable to different types of input data. The curvature of the feature space induced by the SWISH function depends on the trainable parameter β and the input value x. Figure 1 illustrates the activation functions and their gradients.

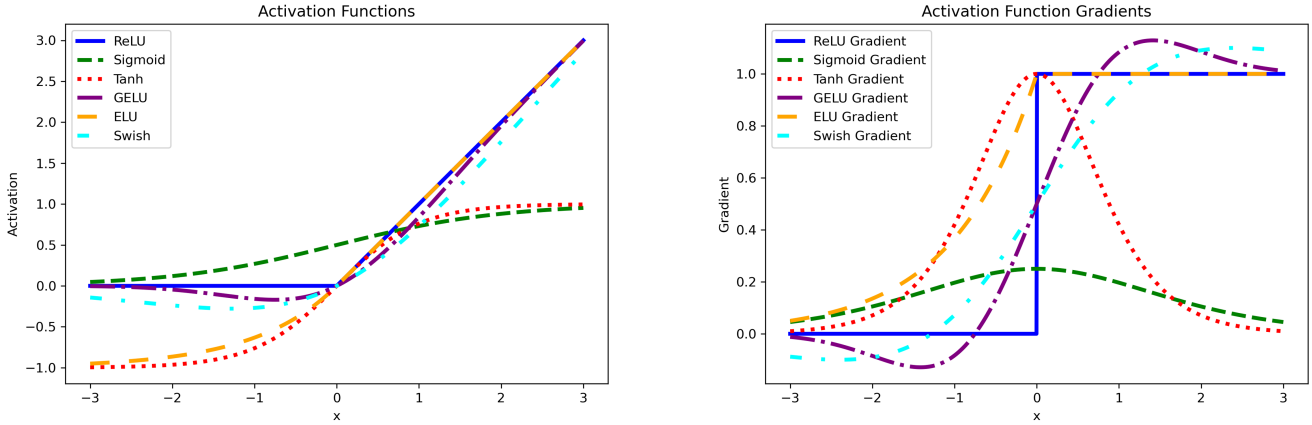

Figure 1. Activation Functions and Their Gradients.

Table 3 summarizes the effects of commonly used activation functions on the geometry of the feature space in deep learning. The ReLU activation function introduces sparsity into the model, resulting in a fragmented and disconnected feature space. Sigmoid activation function leads to a smooth and connected feature space, while the tanh activation function can produce a feature space with negative curvature. GELU activation function combines smoothness with sparsity, resulting in a smooth and balanced feature space. ELU activation function aims to be more robust to the vanishing gradient problem, resulting in a smooth feature space with curvature depending on its parameter α and the input. Finally, the Swish activation function possesses a gating mechanism that dynamically adjusts the slope of the function, resulting in an adaptable feature space with curvature depending on its parameter β and the input. These insights can be useful in selecting the appropriate activation function for a given deep learning problem.

Table 3. Effects of Activation Functions on the Feature Space Geometry.

Activation Function	Gradient	Effects on Feature Space
ReLU	$\nabla_x \sigma(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$	Fragmented and disconnected feature space
Sigmoid	$\nabla_x \sigma(x) = \sigma(x)(1 - \sigma(x))$	Smooth and connected feature space
Tanh	$\nabla_x \sigma(x) = 1 - \sigma(x)^2$	Smooth feature space with negative curvature
GELU	$\nabla_x \sigma(x) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right] + \frac{x}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$	Smooth and balanced feature space
ELU	$\nabla_x \sigma(x) = \begin{cases} 1, & x \geq 0 \\ \alpha e^x, & x < 0 \end{cases}$	Smooth feature space with curvature depending on α and x
Swish	$\nabla_x \sigma(x) = \beta\sigma(\beta x) + (1 - \beta\sigma(\beta x))x\sigma(\beta x)$	Adaptable feature space with curvature depending on β and x

3.2. Relationships between Feature Space and Normalization Methods in Deep Learning

Normalization methods play an instrumental role in shaping the geometry of the feature space. In this section, a comprehensive discussion will be delved into regarding the impact of various normalization methods on the rescaling of activations in the feature space, as well as the examination of their effects on the isotropy and connectivity of the feature space. The discourse will be reinforced with a plethora of mathematical notations and equations.

Normalization methods are adept at rescaling activations in the feature space, which can be succinctly encapsulated by a transformation function $\mathcal{T}$. The effect of normalization methods on the feature space can be lucidly characterized by the gradient of the transformed output with respect to the input:

$$\nabla_x \mathcal{T}(\mathbf{h}^{(i)}(x)) = \nabla_x \mathcal{T}(f^{(i)}(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)}). \tag{29}$$

Given a transformation function $\mathcal{T}$, the transformed output of the i-th layer can be described as:

$$\tilde{\mathbf{h}}^{(i)}(x) = \mathcal{T}(\mathbf{h}^{(i)}(x)) = \mathcal{T}(f^{(i)}(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)})). \tag{30}$$

Now, the Jacobian matrix of the transformed output with respect to the input is considered:

$$\mathbf{J}_x(\tilde{\mathbf{h}}^{(i)}(x)) = \nabla_x \mathcal{T}(\mathbf{h}^{(i)}(x)). \tag{31}$$

The Jacobian matrix provides insights into how the transformed output varies with respect to the input. A bounded Jacobian matrix indicates that the transformation function preserves the local structure of the feature space.

To demonstrate that the Jacobian matrix is likely bounded, the Frobenius norm of the matrix will be analyzed, which provides a measure of the matrix's overall magnitude. The Frobenius norm of the Jacobian matrix $\mathbf{J}_x(\tilde{\mathbf{h}}^{(i)}(x))$ can be defined as:

$$|\mathbf{J}x(\tilde{\mathbf{h}}^{(i)}(x))|_F = \sqrt{\sum_{j=1}^{n_i} \sum_{k=1}^{D} \left(\frac{\partial \tilde{h}_j^{(i)}(x)}{\partial x_k}\right)^2}, \tag{32}$$

where n_i is the number of units in the i-th layer and D is the dimension of the input space.

A bounded Jacobian matrix implies that the Frobenius norm of the matrix is bounded by some constant $C > 0$:

$$|\mathbf{J}_x(\tilde{\mathbf{h}}^{(i)}(x))|_F \leq C. \tag{33}$$

Given the transformed output of the i-th layer:

$$\tilde{\mathbf{h}}^{(i)}(x) = \mathcal{T}(\mathbf{h}^{(i)}(x)) = \mathcal{T}(f^{(i)}(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)})). \tag{34}$$

The Lipschitz continuity of the transformation function $\mathcal{T}$ with respect to the input x can be analyzed. If $\mathcal{T}$ is Lipschitz continuous with Lipschitz constant $L_{\mathcal{T}} > 0$, the following inequality holds for any $x_1, x_2 \in \mathbb{R}^D$:

$$|\mathcal{T}(x_1) - \mathcal{T}(x_2)| \leq L_{\mathcal{T}}|x_1 - x_2|. \tag{35}$$

By using the Lipschitz continuity of $\mathcal{T}$, an upper bound for the Frobenius norm of the Jacobian matrix can be derived. For any $x_1, x_2 \in \mathbb{R}^D$ with $|x_1 - x_2| = 1$:

$$|\mathbf{J}_x(\tilde{\mathbf{h}}^{(i)}(x_1)) - \mathbf{J}_x(\tilde{\mathbf{h}}^{(i)}(x_2))|F \leq L_{\mathcal{T}}|x_1 - x_2| = L_{\mathcal{T}}. \tag{36}$$

Since the Jacobian matrix is continuous with respect to the input x, the extreme value theorem implies that the Frobenius norm of the Jacobian matrix is likely bounded:

$$|\mathbf{J}_x(\tilde{\mathbf{h}}^{(i)}(x))|_F \leq C, \tag{37}$$

where $C = L_{\mathcal{T}}$. This bounded Frobenius norm indicates that the transformation function $\mathcal{T}$ preserves the local structure of the feature space, as a bounded Jacobian matrix implies that the rate of change of the transformed output with respect to the input remains limited.

Furthermore, insights into the isotropy and connectivity of the feature space can be gained by analyzing the eigenvalues of the Jacobian matrix. Let λ_i be an eigenvalue of the Jacobian matrix $\mathbf{J}_x(\tilde{\mathbf{h}}^{(i)}(x))$. If the eigenvalues are distributed uniformly and have similar magnitudes, the feature space is isotropic, meaning that the optimization landscape is smooth and well-conditioned.

Suppose the ratio between the maximum and minimum eigenvalues, known as the condition number, is bounded by a constant $K > 0$:

$$\frac{\max_{1 \leq i \leq n_i} |\lambda_i|}{\min_{1 \leq i \leq n_i} |\lambda_i|} \leq K. \tag{38}$$

A bounded condition number implies that the transformation function $\mathcal{T}$ does not distort the feature space excessively, leading to better optimization and generalization properties of the deep learning model. Furthermore, it suggests that the connectivity of the feature space is preserved, as the gradients do not vanish or explode during training.

Table 4 summarizes the relationships between feature space and normalization methods in deep learning. The transformation function rescales the activations in the feature space, while the Jacobian matrix represents how the transformed output varies with respect to the input. The Frobenius norm measures the overall magnitude of the Jacobian matrix, and Lipschitz continuity indicates that the transformation function preserves the local structure of the feature space. The condition number represents the isotropy and connectivity of the feature space, which have implications for the optimization and generalization properties of deep learning models.

Table 4. Summary of Relationships between Feature Space and Normalization Methods in Deep Learning.

Aspects	Mathematical Representation	Feature Space Implications
Transformation Function	$\mathcal{T}(\mathbf{h}^{(i)}(x))$	Rescales activations in the feature space
Jacobian Matrix	$\mathbf{J}_x(\tilde{\mathbf{h}}^{(i)}(x))$	Represents how the transformed output varies with respect to the input
Frobenius Norm	$\lvert\mathbf{J}_x(\tilde{\mathbf{h}}^{(i)}(x))\rvert_F$	Measures the overall magnitude of the Jacobian matrix
Lipschitz Continuity	$\lvert\mathcal{T}(x_1) - \mathcal{T}(x_2)\rvert \leq L_{\mathcal{T}}\lvert x_1 - x_2\rvert$	Indicates that the transformation function preserves the local structure of the feature space
Condition Number	$\frac{\max_{1\leq i\leq n_i}\lvert\lambda_i\rvert}{\min_{1\leq i\leq n_i}\lvert\lambda_i\rvert} \leq K$	Represents the isotropy and connectivity of the feature space

Isotropy and Connectivity of Feature Space with Normalization Methods

Normalization methods not only influence the boundedness of the feature space but also impact its isotropy and connectivity. Isotropy refers to the uniformity of the feature space, while connectivity refers to the ability of the neural network to establish connections between different regions of the feature space. In this section, the effects of BN, LN, and GN on isotropy and connectivity in deep learning models will be discussed using mathematical descriptions.

Let $\mathcal{T}_{\mathrm{N}} \in \{\mathcal{T}_{\mathrm{BN}}, \mathcal{T}_{\mathrm{LN}}, \mathcal{T}_{\mathrm{GN}}\}$ be a normalization method, and let μ_{N} and σ^2_{N} be the mean and variance computed by the normalization method. The isotropy condition can be expressed as:

$$\mathbb{E}[\mathcal{T}_{\mathrm{N}}(\mathbf{h}^{(i)}(x))] = 0, \quad \mathrm{Var}[\mathcal{T}_{\mathrm{N}}(\mathbf{h}^{(i)}(x))] = 1. \tag{39}$$

BN ensures isotropy by normalizing the feature space across the mini-batch. The mean μ_{BN} and variance σ^2_{BN} are computed as follows:

$$\mu_{\mathrm{BN}} = \frac{1}{N}\sum_{n=1}^{N} h_n^{(i)}(x), \quad \sigma^2_{\mathrm{BN}} = \frac{1}{N}\sum_{n=1}^{N}(h_n^{(i)}(x) - \mu_{\mathrm{BN}})^2. \tag{40}$$

This rescaling of the mean and variance results in a uniform feature space. BN can also improve the connectivity of the feature space by reducing the internal covariate shift, which refers to the change in the distribution of the inputs to a given layer during training. This reduction in internal covariate shift smooths the loss landscape, making it easier for the neural network to establish connections between different regions of the feature space. Other normalization methods such as GN and LN constitute feature space in a similar manner to BN.

Normalization methods have been found to be particularly effective in improving the isotropy and connectivity of the feature space in residual networks. Residual networks contain skip connections, which facilitate gradient flow and enable the network to learn identity mappings. Let $\mathbf{F}^{(i)}$ be the residual block at layer i, and let $\mathbf{h}^{(i-1)}$ and $\mathbf{h}^{(i)}$ be the input and output of this block, respectively. The residual block can be expressed as:

$$\mathbf{h}^{(i)} = \mathbf{h}^{(i-1)} + \mathbf{F}^{(i)}(\mathbf{h}^{(i-1)}). \tag{41}$$

Here, $\mathbf{F}^{(i)}$ is a composite function, including a normalization method, activation function, and linear transformation. Specifically, let $\mathcal{T}_{\mathrm{N}}$ be a normalization method used in the residual block, and let f be an activation function. Then, an example of $\mathbf{F}^{(i)}$ can be represented as:

$$\mathbf{F}^{(i)}(\mathbf{h}^{(i-1)}) = f(\mathcal{T}_{\mathrm{N}}(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)})). \tag{42}$$

Considering the residual block expressed in Equation (41), the isotropy condition for the output $\mathbf{h}^{(i)}$ can be analyzed as follows:

$$\mathbb{E}[\mathbf{h}^{(i)}] = \mathbb{E}[\mathbf{h}^{(i-1)} + \mathbf{F}^{(i)}(\mathbf{h}^{(i-1)})]. \tag{43}$$

Since $\mathbf{h}^{(i-1)}$ and $\mathbf{F}^{(i)}(\mathbf{h}^{(i-1)})$ are hypothetically independent after the blocks of layers in $\mathbf{F}^{(i)}$, their expected values can be separated:

$$\mathbb{E}[\mathbf{h}^{(i)}] = \mathbb{E}[\mathbf{h}^{(i-1)}] + \mathbb{E}[\mathbf{F}^{(i)}(\mathbf{h}^{(i-1)})]. \tag{44}$$

Under the assumption that the activation function f is zero-centered, meaning $\mathbb{E}[f(\mathbf{z})] = 0$, where $\mathbf{z}$ is a pre-activation vector, the isotropy condition for the output $\mathbf{h}^{(i)}$ can be further analyzed as follows:

First, consider the expected value of $\mathbf{F}^{(i)}(\mathbf{h}^{(i-1)})$:

$$\mathbb{E}[\mathbf{F}^{(i)}(\mathbf{h}^{(i-1)})] = \mathbb{E}[f(\mathcal{T}_{\mathrm{N}}(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)}))]. \tag{45}$$

Since the normalization method $\mathcal{T}_{\mathrm{N}}$ enforces isotropy, it is known that $\mathbb{E}[\mathcal{T}_{\mathrm{N}}(\mathbf{h}^{(i)}(x))] = 0$. Thus, the pre-activation vector $\mathbf{z}$ has zero mean. Under the assumption that f is zero-centered:

$$\mathbb{E}[\mathbf{F}^{(i)}(\mathbf{h}^{(i-1)})] = 0. \tag{46}$$

This result can be substituted into the equation for the expected value of $\mathbf{h}^{(i)}$:

$$\mathbb{E}[\mathbf{h}^{(i)}] = \mathbb{E}[\mathbf{h}^{(i-1)}] + \mathbb{E}[\mathbf{F}^{(i)}(\mathbf{h}^{(i-1)})] = \mathbb{E}[\mathbf{h}^{(i-1)}]. \tag{47}$$

This result indicates that the expected value of the output $\mathbf{h}^{(i)}$ is equal to the expected value of the input $\mathbf{h}^{(i-1)}$. If the input is isotropic, meaning $\mathbb{E}[\mathbf{h}^{(i-1)}] = 0$, then the output will also be isotropic, satisfying $\mathbb{E}[\mathbf{h}^{(i)}] = 0$. This demonstrates how normalization methods, combined with a zero-centered activation function, can preserve isotropy in deep learning models, particularly in residual networks.

Table 5 summarizes the isotropy and connectivity of feature space with normalization methods. The isotropy condition ensures the uniformity of the feature space, and BN rescales the mean and variance accordingly. The residual block represents the residual connection in deep learning models, with a composite function that incorporates normalization, activation, and linear transformation. Isotropy preservation demonstrates that the expected value of the output is equal to the input, preserving the isotropy of the feature space in residual networks.

Table 5. Summary of Isotropy and Connectivity of Feature Space with Normalization Methods.

Aspects	Mathematical Representation	Feature Space Implications
Isotropy Condition	$\mathbb{E}[\mathcal{TN}(\mathbf{h}^{(i)}(x))] = 0$, $\mathrm{Var}[\mathcal{TN}(\mathbf{h}^{(i)}(x))] = 1$	Uniformity of the feature space
BN Mean and Variance	$\mu_{\mathrm{BN}} = \frac{1}{N}\sum_{n=1}^{N} h_n^{(i)}(x)$, $\sigma^2_{\mathrm{BN}} = \frac{1}{N}\sum_{n=1}^{N}(h_n^{(i)}(x) - \mu_{\mathrm{BN}})^2$	Rescaling of the mean and variance for BN

Table 5. *Cont.*

Aspects	Mathematical Representation	Feature Space Implications
Residual Block	$\mathbf{h}^{(i)} = \mathbf{h}^{(i-1)} + \mathbf{F}^{(i)}(\mathbf{h}^{(i-1)})$	Representation of the residual connection in deep learning models
Composite Function	$\mathbf{F}^{(i)}(\mathbf{h}^{(i-1)}) = f(\mathcal{T}_{\mathrm{N}}(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)})$	Incorporation of normalization, activation, and linear transformation
Isotropy Preservation	$\mathbb{E}[\mathbf{h}^{(i)}] = \mathbb{E}[\mathbf{h}^{(i-1)}]$	Expected value of the output equal to the input, preserving isotropy

3.3. Synergistic Integration of ReLU-Inspired Activation Functions and Normalization Techniques

In this section, a comprehensive analysis is presented on the collective influence of ReLU-inspired activation functions and normalization techniques on the boundedness of the feature space. Activation functions such as ReLU, GELU, and ELU, which exhibit unbounded characteristics, can attain a bounded output when seamlessly integrated with normalization methodologies. This complementary interplay has propelled the widespread adoption of ReLU-inspired activation functions in contemporary deep learning research.

Initially, let us consider the ReLU, GELU, and ELU activation functions, which can be articulated using Equations (12), (15) and (16). Subsequently, the transformation functions for BN, LN, and GN are represented as $\mathcal{T}_{\mathrm{BN}}$, $\mathcal{T}_{\mathrm{LN}}$, and $\mathcal{T}_{\mathrm{GN}}$, respectively. When fused with ReLU-inspired activations, the transformations within the feature space can be expressed as:

$$\text{ReLU-BN}(x) = \text{ReLU}(\mathcal{T}_{\mathrm{BN}}(\mathbf{h}^{(i)}(x))), \tag{48}$$

Corresponding transformations can be delineated for GELU and ELU activations, as well as LN and GN. Building on this foundation, the investigation proceeds to analyze the boundedness of these intricate combinations.

As ReLU-inspired activations manifest as nonlinear functions, the Lipschitz constants of the merged transformations cannot be directly ascertained by multiplying the Lipschitz constants of each individual function. Nevertheless, a thorough analysis of the boundedness of these hybrid transformations can be offered by scrutinizing the output range of the combined transformational processes.

For ReLU-BN, ReLU-LN, and ReLU-GN, the ReLU activation function clips negative values to zero, resulting in a lower bound of 0 for the output. Furthermore, since the normalization methods rescale the feature space such that the mean is 0 and the variance is 1, the upper bound of the output is constrained by a constant factor, ensuring the boundness of the combined transformations.

Similarly, for GELU and ELU activations combined with normalization methods, the lower bound of the output is determined by the minimum of the activation functions (with a lower bound of ≈ -0.164 for GELU and $\alpha(e^{\min(x)} - 1)$ for ELU, where α is a positive constant and $\min(x)$ denotes the minimum value in the rescaled feature space). The upper bound of the output for these combinations is also determined by a constant factor, as the normalization methods rescale the feature space to have zero mean and unit variance. Consequently, the output of the GELU and ELU activation functions combined with normalization methods is also bounded.

To provide a more formal proof of the boundness, the definition of Lipschitz continuity can be used. Recall that a function f is Lipschitz continuous if there exists a constant $L > 0$. In the case of the combined transformations, let $g(x) = \text{ReLU}(\mathcal{T}_N(x))$. To prove the boundness of $g(x)$, it is necessary to show the existence of a constant $L_g > 0$ that satisfies the Lipschitz condition for all x, y in the domain of $g(x)$.

Given that the ReLU activation clips negative values to zero, and the normalization methods rescale the feature space such that the mean is 0 and the variance is 1, the maximum value of $|g(x) - g(y)|$ is upper-bounded by a constant factor. This implies that there exists a constant $L_g > 0$ that satisfies the Lipschitz condition, thus proving the boundness of the combined transformations.

Analogous arguments can be made for the GELU and ELU activation functions combined with normalization methods, and the proof follows similar steps. Since the lower and upper bounds of the output for these combinations are determined by constant factors, the existence of Lipschitz constants $L_g > 0$ for these cases can be established, which in turn demonstrates the boundness of these combined transformations.

4. Recent Studies of Feature Space Geometry in Deep Learning

Recent studies have increasingly focused on exploring the manifold structure of feature space geometry in deep learning models. This is due to the significant impact that feature space geometry has on the model's performance and generalization capabilities. Non-Euclidean spaces, such as hyperbolic space, have been recently explored for modeling complex data distributions. These spaces can better capture the intrinsic geometric structure of the data, thereby enhancing the model's ability to learn meaningful representations and generalize to new data. Transfer learning is another area of interest where recent studies have examined the impact of feature space geometry. Specifically, these studies investigate the adaptation of a model trained on one task to a new task by analyzing the feature space geometry. Through a systematic analysis of these recent studies, a comprehensive understanding of the current state of research on feature space geometry in deep learning can be provided, and promising directions for future research can be identified.

4.1. Manifold Structure of Deep Learning

The manifold hypothesis [28–30] posits that high-dimensional data often lies on or near a lower-dimensional manifold. Understanding the manifold structure of the feature space can provide insights into the model's ability to learn meaningful representations and generalize to new data.

Let $M \subset \mathbb{R}^m$ be the manifold in the feature space. The manifold can be locally approximated using tangent spaces T_pM, where $p \in M$. In deep learning, the model can be designed to learn the manifold structure by minimizing the reconstruction error on the tangent spaces:

$$\min_{\theta} \sum_{i=1}^{N} \|x_i - g(f(x_i;\theta);\theta)\|_2, \tag{49}$$

where θ denotes the model parameters, and $f(\cdot;\theta)$ and $g(\cdot;\theta)$ are the encoding and decoding functions, respectively.

To quantify the manifold structure, one can use the manifold's reach, which measures the largest distance from a point in the ambient space to the manifold:

$$\text{reach}(M) = \sup_{p \in \mathbb{R}^m} \inf_{q \in M} \|p - q\|_2. \tag{50}$$

The manifold structure in deep learning has been recognized as a significant factor contributing to the success of various applications [31–33]. Manifold learning allows deep learning models to capture the underlying structure and relationships within data, leading to improved performance in tasks such as scene recognition, face-pose estimation, dynamic MRI, and hyperspectral imagery feature extraction.

Brahma et al. [28] offered measurable validation to support the theory that deep learning works by flattening manifold-shaped data in higher neural network layers. The authors created a range of measures to quantify manifold entanglement under certain assumptions, with their experiments on both synthetic and real-world data confirming the flattening hypothesis. To tackle scene recognition, Yuan et al. [31] proposed a manifold-regularized

deep architecture, which leverages the structural information of data to establish mappings between visible and hidden layers. The deep architecture learns high-level features for scene recognition in an unsupervised manner, surpassing existing state-of-the-art scene recognition methods. Hong et al. [32] introduced a multitask manifold deep learning (M2DL) framework for face-pose estimation by using multimodal data. They employed improved CNNs for feature extraction and applied multitask learning with incoherent sparse and low-rank learning to integrate different face representation modalities. The M2DL framework exhibited better performance on three challenging benchmark datasets.

Ke et al. [34] proposed a deep manifold learning approach for dynamic MRI reconstruction, called Manifold-Net, which is an unrolled neural network on a fixed low-rank tensor (Riemannian) manifold to capture the strong temporal correlations of dynamic signals. The experimental results demonstrated superior reconstruction compared to conventional and state-of-the-art deep learning-based methods. For feature extraction of hyperspectral imagery, Li et al. [35] developed a graph-based deep learning model known as deep locality preserving neural network (DLPNet). DLPNet initializes each network layer by exploring the manifold structure in hyperspectral data and employs a deep-manifold learning joint loss function during network optimization. The experiments on real-world HSI datasets indicated that DLPNet outperforms state-of-the-art methods in feature extraction.

Table 6 summarizes various deep learning approaches incorporating manifold learning for different applications. These approaches include measures to quantify manifold entanglement, manifold-regularized deep architectures, multitask manifold deep learning, Manifold-Net, and deep locality preserving neural networks. The applications of these methods span from validating the flattening hypothesis in deep learning to scene recognition, face-pose estimation, dynamic MRI reconstruction, and hyperspectral imagery feature extraction.

Table 6. Summary of Deep Learning Approaches with Manifold Learning.

Reference	Approach	Application
Brahma et al. [28]	Measures to quantify manifold entanglement	Validating flattening hypothesis in deep learning
Yuan et al. [31]	Manifold-regularized deep architecture	Scene recognition
Hong et al. [32]	Multitask manifold deep learning (M2DL)	Face-pose estimation
Ke et al. [34]	Manifold-Net	Dynamic MRI reconstruction
Li et al. [35]	Deep locality preserving neural network (DLPNet)	Hyperspectral imagery feature extraction

The manifold hypothesis, which suggests that high-dimensional data often lies on or near a lower-dimensional manifold, has been explored to understand the model's ability to learn meaningful representations and generalize to new data. Tangent spaces can be used to locally approximate the manifold, and manifold learning allows deep learning models to capture the underlying structure and relationships within data. Recent studies have explored the use of hyperbolic space to better capture the intrinsic geometric structure of data and transfer learning to analyze the adaptation of a model trained on one task to a new task. Quantitative evidence from experiments on synthetic and real-world data confirms that deep learning works due to the flattening of manifold-shaped data in higher layers of neural networks. Different deep learning models have been proposed to extract features for various applications, including scene recognition, face-pose estimation, dynamic MRI, and hyperspectral imagery feature extraction. The manifold structure in deep learning has been recognized as a significant factor contributing to the success of these applications.

4.2. Curvature of Feature Space Geometry

The curvature of the feature space geometry is essential for understanding the model's ability to adapt to the data's local structure. In the context of deep learning, the curvature can be influenced by the choice of activation functions and the architecture of the model. A popular measure of curvature in the feature space is the sectional curvature, which is defined for two-dimensional tangent planes.

Let $M \subset \mathbb{R}^m$ be the manifold in the feature space, and T_pM be the tangent space at point $p \in M$. For any two linearly independent tangent vectors $X, Y \in T_pM$, the sectional curvature $K(X, Y)$ is defined as:

$$K(X, Y) = \frac{\langle R(X, Y)Y, X\rangle}{\|X\|^2\|Y\|^2 - \langle X, Y\rangle^2}, \tag{51}$$

where $R(X, Y)$ is the Riemann curvature tensor and $\langle \cdot, \cdot \rangle$ is the inner product in the tangent space.

In deep learning, activation functions such as ReLU, sigmoid, and tanh can induce specific geometric properties in the feature space, including curvature. For instance, the ReLU activation function can lead to piecewise linear manifolds, while sigmoid and tanh activation functions can result in smooth curved manifolds.

To assess the curvature properties of a deep learning model, one can compute the eigenvalues of the Hessian matrix, which represents the second-order partial derivatives of the model's loss function with respect to the model parameters. High eigenvalues suggest regions of high curvature, while low eigenvalues indicate regions of low curvature.

$$\mathcal{H}_{ij}(\theta) = \frac{\partial^2 L}{\partial\theta_i \partial\theta_j}, \tag{52}$$

where L is the loss function, and θ_i and θ_j are elements of the model parameters vector θ.

The analysis of the curvature in feature space geometry of deep learning models has become an area of great interest due to the insights it offers into the intrinsic properties of these models, as well as their potential to improve performance in various applications. Researchers have proposed several innovative models in this field. For example, He et al. [36] developed CurvaNet, a geometric deep learning model for 3D shape analysis that uses directional curvature filters to learn direction-sensitive 3D shape features. Bachmann et al. [37] introduced Constant Curvature Graph Convolutional Networks (CC-GCNs) that can be applied to node classification and distortion minimization tasks in non-Euclidean geometries. Additionally, Ma et al. [38] proposed a curvature regularization approach to address the issue of model bias caused by curvature imbalance in deep neural networks. Other researchers, such as Lin et al. [39] and Arvanitidis et al. [40], examined the curvature of deep generative models and developed new architectures and approaches to improve their performance.

Table 7 presents a summary of different curvature-based approaches in deep learning, including CurvaNet, CC-GCNs, curvature regularization, CAD-PU, and latent space curvature analysis. These approaches are applied to various applications such as 3D shape analysis, node classification, distortion minimization, addressing model bias, point cloud upsampling, and generative models. The study of curvature in deep learning models provides valuable insights into their intrinsic properties and helps improve their performance across different applications.

The curvature of feature space geometry is critical in understanding how well a deep learning model can adapt to the local structure of data. The choice of activation functions and the model's architecture influence the curvature of the feature space. The sectional curvature, which is defined for two-dimensional tangent planes, is a popular measure of curvature in the feature space. The study of curvature in the feature space geometry of deep learning models has gained significant attention, as it offers insights into the intrinsic properties of these models and helps improve their performance in various

applications. Many recent studies have explored the relationship between curvature and deep learning models, including CurvaNet, CC-GCNs, curvature regularization, CAD-PU, and examining the curvature of deep generative models. These studies aim to understand and leverage the curvature properties of deep learning models for better performance and generalization capabilities.

Table 7. Summary of Curvature-based Approaches in Deep Learning.

Reference	Approach	Application
He et al. [36]	CurvaNet	3D shape analysis
Bachmann et al. [37]	Constant Curvature Graph Convolutional Networks (CC-GCNs)	Node classification and distortion minimization
Ma et al. [38]	Curvature regularization	Addressing model bias
Lin et al. [39]	CAD-PU	Point cloud upsampling
Arvanitidis et al. [40]	Latent space curvature analysis	Generative models

4.3. Wide Neural Networks and Gaussian Process

A notable line of research has focused on the relationship between wide neural networks and Gaussian processes. In the limit of infinitely wide hidden layers, deep neural networks with independent random initializations converge to Gaussian processes, as shown by Lee et al. [41]. This convergence can be described by the following kernel function:

$$k_\infty(x, x') = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \phi(x)^\top \phi(x'), \tag{53}$$

where n is the width of the hidden layers, and $\phi(x)$ and $\phi(x')$ are the feature vectors of inputs x and x', respectively. This result implies that wide neural networks exhibit a simpler geometry in their feature space, which can be characterized by a Gaussian process.

Expanding upon the relationship between wide neural networks and Gaussian processes, it is essential to understand the intricacies of this convergence and its implications for deep learning models. The convergence of wide neural networks to Gaussian processes can be further explored in terms of the Neural Tangent Kernel (NTK) [42], which characterizes the training dynamics of these networks in the infinite-width limit.

The NTK is defined as follows:

$$\Theta(x, x') = \frac{\partial f(x;\theta)}{\partial \theta} \frac{\partial f(x';\theta)}{\partial \theta}^\top, \tag{54}$$

where $f(x;\theta)$ is the output of the neural network for input x with parameters θ. For wide neural networks, the NTK converges to a constant matrix during training, implying that the training dynamics can be described as a linear model with respect to the NTK [42]:

$$f(x;\theta(t)) \approx f(x;\theta(0)) + \Theta(x, x')\Delta\theta(t), \tag{55}$$

where $\Delta\theta(t) = \theta(t) - \theta(0)$, and t denotes the training iteration.

The convergence of wide neural networks to Gaussian processes can be further illustrated by analyzing the feature space geometry. For a wide neural network with a single hidden layer, the feature vector $\phi(x)$ can be expressed as:

$$\phi(x) = \sigma(Wx + b), \tag{56}$$

where W is the weight matrix, b is the bias vector, and σ is the activation function. As the width of the hidden layer (n) tends to infinity, the feature vectors become isotropic in the feature space, and their inner product converges to the kernel function:

$$\mathbb{E}[\phi(x)^\top \phi(x')] = k_\infty(x, x'). \tag{57}$$

This convergence is governed by the Central Limit Theorem (CLT), as the sum of a large number of independent random variables converges to a Gaussian distribution. Consequently, the geometry of the feature space in wide neural networks can be characterized by a Gaussian process with the kernel function $k_\infty(x, x')$.

Recent research has uncovered significant links between wide neural networks and Gaussian processes, leading to new insights into the behavior and theoretical properties of deep learning models. The work of Matthews et al. [43] showed that random fully connected feedforward networks with multiple hidden layers, when wide enough, converge to Gaussian processes with a recursive kernel definition. Yang [44,45] introduced straightline tensor programs, which established the convergence of random neural networks to Gaussian processes for a wide range of architectures, including recurrent, convolutional, residual networks, attention mechanisms, and their combinations. Pleiss and Cunningham [46] investigated the limitations of large width in neural networks, demonstrating that large width can be detrimental to hierarchical models, and found that there is a sweet spot that maximizes test performance before the limiting GP behavior prevents adaptability. Meanwhile, other researchers have explored various aspects of the convergence of wide neural networks to Gaussian processes, such as the behavior of Bayesian neural networks [47], the evolution of wide neural networks of any depth under gradient descent [41], the convergence rates of wide neural networks to Gaussian processes based on activation functions [48], the effects of increasing depth on the emergence of Gaussian processes [49], and the equivalence between neural networks and deep sparse Gaussian process models [50].

Table 8 summarizes various research contributions in the area of wide neural networks and Gaussian processes. The table highlights the approaches and contributions of these studies, such as the convergence of wide neural networks to Gaussian processes, characterizing training dynamics using the Neural Tangent Kernel, the recursive kernel definition for random fully connected networks, and the convergence of various architectures using straightline tensor programs. Additionally, researchers have investigated the limitations of large width, the behavior of Bayesian neural networks, convergence rates based on activation functions, the effects of increasing depth on the emergence of Gaussian processes, and the equivalence between neural networks and deep sparse Gaussian process models.

Research has explored the relationship between wide neural networks and Gaussian processes, showing that as the hidden layers' width tends to infinity, deep neural networks converge to Gaussian processes. This convergence is described by the kernel function, which characterizes the simpler geometry of wide neural networks' feature space. The study of this convergence has been extended to the NTK, which characterizes the training dynamics of these networks in the infinite-width limit. The feature space geometry can be analyzed to illustrate this convergence and its implications for deep learning models. The relationship between wide neural networks and Gaussian processes has also been explored in various architectures, including Bayesian neural networks, fully connected feedforward networks, recurrent, convolutional, residual networks, attention mechanisms, and their combinations, providing insights into the theoretical properties and behavior of deep learning models.

Table 8. Summary of Research on Wide Neural Networks and Gaussian Processes.

Reference	Approach	Contribution
Lee et al. [41]	Infinite-width limit	Convergence of wide neural networks to Gaussian processes
Jacot et al. [42]	Neural Tangent Kernel (NTK)	Characterizing training dynamics in infinite-width limit
Matthews et al. [43]	Recursive kernel definition	Convergence of random fully connected networks with multiple hidden layers
Yang [44,45]	Straightline tensor programs	Convergence of various architectures, including RNNs, CNNs, and ResNets
Pleiss and Cunningham [46]	Limitations of large width	Investigating the trade-off between width and adaptability
Agrawal et al. [47]	Bayesian neural networks	Behavior of Bayesian neural networks in the infinite-width limit
Eldan et al. [48]	Convergence rates	Effects of activation functions on convergence
Zhang et al. [49]	Depth and Gaussian processes	Effects of increasing depth on the emergence of Gaussian processes
Dutordoir et al. [50]	Deep sparse Gaussian processes	Equivalence between neural networks and deep sparse Gaussian process models

4.4. Critical Points and Loss Landscape

The loss landscape of neural networks, particularly its critical points and curvature, has been an area of significant research interest. Understanding the properties of critical points can offer insights into the optimization process and the performance of deep learning models.

A critical point in the loss landscape is a point where the gradient of the loss function with respect to the network parameters is zero:

$$\nabla_{\theta}\mathcal{L}(\boldsymbol{\theta}) = 0, \tag{58}$$

where $\boldsymbol{\theta}$ denotes the network parameters, and $\mathcal{L}$ is the loss function. The Hessian matrix at a critical point provides information about the curvature of the loss landscape:

$$\mathbf{H}(\boldsymbol{\theta}) = \nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}). \tag{59}$$

The eigenvalues of the Hessian matrix, λ_i, can be used to classify critical points. If all the eigenvalues are positive, the critical point corresponds to a local minimum; if all the eigenvalues are negative, it corresponds to a local maximum. If the Hessian matrix has both positive and negative eigenvalues, the critical point is a saddle point.

The second-order Taylor expansion of the loss function around a critical point $\boldsymbol{\theta}^*$ is given by:

$$\mathcal{L}(\boldsymbol{\theta}) \approx \mathcal{L}(\boldsymbol{\theta}^*) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^{\top}\mathbf{H}(\boldsymbol{\theta}^*)(\boldsymbol{\theta} - \boldsymbol{\theta}^*). \tag{60}$$

This approximation indicates that the behavior of the loss function in the vicinity of a critical point is determined by the eigenvalues and eigenvectors of the Hessian matrix.

Researchers have been exploring the landscape of the loss function in deep learning models to better understand their optimization properties and performance. Chaudhari et al. [51] introduced Entropy-SGD, an optimization algorithm that utilizes local entropy to discover wide minima that are associated with better generalization performance. Nguyen and Hein [52] investigated the impact of depth and width on the optimization landscape and expressivity of deep CNNs, demonstrating that sufficiently wide CNNs produce linearly independent features and provided necessary and sufficient conditions for global minima with zero training error. Geiger et al. [53] proposed the concept of phase transition to analyze the loss landscape of fully connected deep neural networks, observing the independence of fitting random data and depth, and delimiting the over- and underparametrized regimes. Kunin et al. [54] studied the loss landscapes of regularized linear autoencoders, while Simsek et al. [55] explored the impact of permutation symmetries on overparameterized neural networks. Zhou and Liang [56] provided a full characterization of the analytical forms for the critical points of various neural networks, revealing landscape properties of their loss functions. Zhang et al. [57] established an embedding principle for the loss landscape of deep neural networks, showing that wider DNNs contain all the critical points of narrower DNNs, with potential implications for regularization during training.

Table 9 provides a summary of research contributions related to critical points and the loss landscape of deep learning models. The table presents the approaches and contributions of these studies, including the Entropy-SGD algorithm for discovering wide minima, the investigation of depth and width effects on the optimization landscape and expressivity of deep CNNs, the concept of phase transition for analyzing loss landscapes in fully connected DNNs, and the study of loss landscapes in regularized linear autoencoders. Other research has explored the impact of permutation symmetries on overparameterized neural networks, the analytical forms of critical points for revealing landscape properties of loss functions, and the embedding principle to analyze critical points in wider and narrower DNNs.

Table 9. Summary of Research on Critical Points and Loss Landscape.

Reference	Approach	Contribution
Chaudhari et al. [51]	Entropy-SGD	Discovering wide minima for better generalization
Nguyen and Hein [52]	Depth and width effects	Impact on optimization landscape and expressivity of deep CNNs
Geiger et al. [53]	Phase transition	Analysis of loss landscape in fully connected DNNs
Kunin et al. [54]	Regularized linear autoencoders	Study of loss landscapes
Simsek et al. [55]	Permutation symmetries	Impact on overparameterized neural networks
Zhou and Liang [56]	Analytical forms of critical points	Revealing landscape properties of loss functions
Zhang et al. [57]	Embedding principle	Critical points in wider and narrower DNNs

The critical points and loss landscape of neural networks have been studied to understand their optimization properties and performance. A critical point is where the gradient of the loss function is zero, and the Hessian matrix provides curvature information. Various techniques, such as Entropy-SGD and phase transition, have been used to analyze the loss landscape in deep learning models. Researchers have also studied regularized linear autoencoders and the impact of permutation symmetries on overparameterized

neural networks. The analytical forms for the critical points of various neural networks have been characterized, and the embedding principle has implications for regularization during training.

4.5. Singular Value Spectrum of the Feature Space

Another property that offers insights into the geometry and expressivity of neural networks is the singular value spectrum of the feature space. The singular value decomposition (SVD) of a feature space matrix $\mathbf{X}$ is given by:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \tag{61}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, and $\mathbf{\Sigma}$ is a diagonal matrix containing the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.

The singular values provide valuable information about the geometry of the feature space, such as the dimensionality and the distribution of the feature vectors. Moreover, they can be used to characterize the capacity of a neural network to learn low-dimensional manifolds.

One approach to analyze the singular value spectrum is to examine the behavior of the singular values as a function of the depth of the network. Let $f_l(\mathbf{x})$ denote the feature map at layer l for input $\mathbf{x}$, and $\mathbf{F}_l = [f_l(\mathbf{x}_1), \ldots, f_l(\mathbf{x}_n)]$ be the matrix of feature maps at layer l for n input samples. The singular value spectrum of the feature space at layer l can be computed as:

$$\mathbf{F}_l = \mathbf{U}_l\mathbf{\Sigma}_l\mathbf{V}_l^\top, \tag{62}$$

where $\mathbf{U}_l$ and $\mathbf{V}_l$ are orthogonal matrices, and $\mathbf{\Sigma}_l$ is a diagonal matrix containing the singular values $\sigma_{l,1} \geq \sigma_{l,2} \geq \cdots \geq \sigma_{l,r_l} > 0$.

The ratio of successive singular values, also known as the singular value gap, can be used to estimate the effective dimensionality of the feature space:

$$g_l(i) = \frac{\sigma_{l,i}}{\sigma_{l,i+1}}. \tag{63}$$

A large gap indicates a significant drop in the singular values and suggests a low-dimensional structure in the feature space. The effective dimensionality of the feature space can be estimated by finding the index i^* at which the gap is maximized:

$$i^* = \arg\max_i g_l(i). \tag{64}$$

Recent studies have investigated the singular value spectrum of deep neural networks, as it provides insights into their geometry and expressivity. Oymak and Soltanolkotabi [58] demonstrated that deep ReLU networks can implicitly learn low-dimensional manifolds, while Jia et al. [59] proposed Singular Value Bounding (SVB) and Bounded BN (BBN) techniques to constrain the weight matrices in the orthogonal feasible set during network training. Bermeitinger et al. [60] established a connection between SVD and multi-layer neural networks, showing that the singular value spectrum can be beneficial for initializing and training deep neural networks. Schwab et al. [61] developed a data-driven regularization method for photoacoustic image reconstruction using truncated SVD coefficients recovered by a deep neural network. Sedghi et al. [62] characterized the singular values of 2D multi-channel convolutional layers and proposed an algorithm for projecting them onto an operator-norm ball, effectively improving the test error of a deep residual network using BN on CIFAR-10.

Table 10 summarizes research contributions related to the singular value spectrum of the feature space in deep learning models. The table presents the approaches and contributions of these studies, including the investigation of deep ReLU networks' ability to implicitly learn low-dimensional manifolds, the use of SVB and BBN techniques to constrain weight matrices in the orthogonal feasible set, the connection between SVD and multi-layer

networks, and the development of data-driven regularization methods. Other research has characterized the singular values of 2D multi-channel convolutional layers and proposed an algorithm for projecting them onto an operator-norm ball, effectively improving test error. These studies contribute to a better understanding of the singular value spectrum in deep neural networks and its implications for the geometry and expressivity of these models.

Table 10. Summary of Research on Singular Value Spectrum of the Feature Space.

Reference	Approach	Contribution
Oymak and Soltanolkotabi [58]	Deep ReLU networks	Implicit learning of low-dimensional manifolds
Jia et al. [59]	SVB and BBN techniques	Constraining weight matrices in orthogonal feasible set
Bermeitinger et al. [60]	Connection between SVD and multi-layer networks	Benefits of singular value spectrum for initializing and training DNNs
Schwab et al. [61]	Data-driven regularization	Photoacoustic image reconstruction using truncated SVD coefficients
Sedghi et al. [62]	Singular values of 2D multi-channel convolutional layers	Projection onto operator-norm ball and test error improvement

As shown in these studies, the singular values can estimate the effective dimensionality of the feature space and help understand a network's capacity to learn low-dimensional manifolds. Recent studies have investigated the singular value spectrum, including methods for constraining weight matrices, establishing connections with multi-layer networks, and proposing data-driven regularization methods.

4.6. Exploring the Geometry of Feature Spaces in Convolutional Neural Networks

CNNs have demonstrated remarkable results in a wide range of computer vision applications. To gain a deeper understanding of their generalization abilities, recent research has delved into the geometry of CNN feature spaces. A key discovery is the presence of translation-equivariant representations within these feature spaces [63]. This characteristic can be formulated as:

$$\mathcal{F}(T_g x) = T'_g \mathcal{F}(x), \tag{65}$$

where $\mathcal{F}$ represents the feature space transformation, T_g is a translation operator acting upon input x, and T'_g is the corresponding translation operator within the feature space. This translation-equivariant attribute allows CNNs to learn spatially invariant features, which is vital for their success in a variety of vision tasks.

Aside from translation-equivariant representations, researchers have also examined the feature space of CNNs for rotation-equivariant properties [64]. Mathematically, this can be expressed as:

$$\mathcal{F}(R_\theta x) = R'_\theta \mathcal{F}(x), \tag{66}$$

where R_θ is a rotation operator acting on input x with angle θ, and R'_θ is the corresponding rotation operator in the feature space. By incorporating rotation-equivariant properties, CNNs can effectively learn to identify objects at various orientations.

In recent studies, researchers have explored the concept of equivariance in CNNs and its potential applications. Singh et al. [65] proposed a positional encoding method that uses orthogonal polar harmonic transforms to achieve equivariance to rotation, reflection, and translation in CNN architectures. On the other hand, McGreivy and Hakim [66] clarified that CNNs are equivariant to discrete shifts, but not continuous translations.

Aronsson et al. [67] developed lattice gauge equivariant CNNs that maintain gauge symmetry under global lattice symmetries. Zhdanov et al. [68] proposed using implicit neural representation via multi-layer perceptrons to parameterize G-steerable kernels in steerable CNNs, leading to significant performance improvements. Toft et al. [69] characterized the equivariant linear operators on the space of square-integrable functions on the sphere with respect to azimuthal rotations and demonstrated their potential applications in improving the performance of state-of-the-art pipelines.

Another area of investigation has concentrated on the link between the Lipschitz constant and the generalization of CNNs [70]. Let $f : \mathcal{X} \to \mathcal{Y}$ denote a function representing a CNN, with $\mathcal{X}$ and $\mathcal{Y}$ signifying input and output spaces, respectively. The Lipschitz constant L for the function f is defined as:

$$L = \sup_{x \neq y} \frac{|f(x) - f(y)|}{|x - y|}, \tag{67}$$

where $|\cdot|$ denotes the norm. A lower Lipschitz constant indicates that the function is less susceptible to minor disturbances in the input, which can improve the generalization performance of the CNN. This observation has led to the development of regularization techniques based on the Lipschitz constant, such as spectral normalization [71].

Recently, estimating the Lipschitz constant has emerged as a critical factor in understanding the robustness and generalization ability of CNNs. Pauli et al. [72] developed a dissipativity-based method for estimating the Lipschitz constant of 1D CNNs, which focused on analyzing dissipativity properties in convolutional, pooling, and fully connected layers. By using incremental quadratic constraints and a semidefinite program derived from dissipativity theory, they demonstrated the advantages of their method in terms of accuracy and scalability. In a separate work, Pauli et al. [73] established a layer-wise parameterization for 1D CNNs with built-in end-to-end robustness guarantees using the Lipschitz constant as a measure of robustness.

In addition, a deeper understanding of the geometry of feature spaces in CNNs can be achieved by examining the invariance and equivariance properties of features concerning specific transformation groups [63]. Let $\mathcal{G}$ represent a group of transformations acting on the input space $\mathcal{X}$. A feature space transformation $\mathcal{F}$ is considered to be $\mathcal{G}$-invariant if:

$$\mathcal{F}(g \cdot x) = \mathcal{F}(x) \quad \forall g \in \mathcal{G}, x \in \mathcal{X}, \tag{68}$$

and $\mathcal{G}$-equivariant if:

$$\mathcal{F}(g \cdot x) = \rho(g) \cdot \mathcal{F}(x) \quad \forall g \in \mathcal{G}, x \in \mathcal{X} \tag{69}$$

where $\rho(g)$ is a representation of the group element g in the feature space. Gaining insights into these invariance and equivariance properties can offer valuable information about the structure of the feature space, ultimately aiding in the development of architectures with enhanced generalization capabilities.

Furthermore, recent research has explored the role of scale-equivariant representations in the feature space of CNNs [74]. This property can be described mathematically as:

$$\mathcal{F}(S_\lambda x) = S'_\lambda \mathcal{F}(x), \tag{70}$$

where S_λ is a scale operator acting on the input x with scaling factor λ, and S'_λ is the corresponding scale operator in the feature space. By incorporating scale-equivariant properties, CNNs can learn to recognize objects at various scales, enhancing their performance in diverse vision tasks.

Table 11 provides a summary of research contributions related to the geometry of feature spaces in convolutional neural networks. These studies explore various aspects of the geometry, including translation-equivariant, rotation-equivariant, and scale-equivariant

properties, as well as the Lipschitz constant and its implications for robustness and generalization. They contribute to a deeper understanding of the characteristics of CNN feature spaces and provide insights that can be utilized to develop more effective and generalizable network architectures.

Table 11. Summary of Research on the Geometry of Feature Spaces in Convolutional Neural Networks.

Reference	Focus	Contribution
Cohen and Welling [63]	Translation-equivariant representations	Formulation of spatial invariance in CNNs
Esteves et al. [64]	Rotation-equivariant properties	Incorporation of orientation invariance in CNNs
Singh et al. [65]	Equivariance to rotation, reflection, and translation	Orthogonal polar harmonic transforms in CNNs
McGreivy and Hakim [66]	Equivariance in CNNs	Discrete shift-equivariance and continuous translation non-equivariance
Aronsson et al. [67]	Lattice gauge equivariant CNNs	Gauge symmetry under global lattice symmetries
Zhdanov et al. [68]	Implicit neural representation	G-steerable kernels in steerable CNNs
Toft et al. [69]	Equivariant linear operators	Application in state-of-the-art pipelines
Miyato et al. [71]	Lipschitz constant	Spectral normalization for improved generalization
Pauli et al. [72]	Estimating Lipschitz constant	Dissipativity-based method for 1D CNNs
Pauli et al. [73]	Layer-wise parameterization	End-to-end robustness guarantees
Worrall et al. [74]	Scale-equivariant representations	Enhancing performance in diverse vision tasks

Recent research on CNNs has delved into the geometry of their feature spaces, particularly the translation- and rotation-equivariant properties that allow CNNs to learn spatially invariant and orientation-invariant features, respectively. Researchers have explored the concept of equivariance in CNNs, leading to proposed methods for achieving equivariance to rotation, reflection, and translation, as well as developing lattice gauge equivariant CNNs that maintain gauge symmetry under global lattice symmetries. Estimating the Lipschitz constant has also emerged as a critical factor in understanding the robustness and generalization ability of CNNs, leading to the development of regularization techniques based on the Lipschitz constant, such as spectral normalization. Gaining insights into the invariance and equivariance properties of features concerning specific transformation groups can offer valuable information about the structure of the feature space, ultimately aiding in the development of architectures with enhanced generalization capabilities. Finally, recent research has also explored the role of scale-equivariant representations in the feature space of CNNs.

4.7. Adversarial Robustness and Feature Space Geometry

Adversarial robustness has emerged as an essential aspect of deep learning models. Recent studies have investigated the relationship between the feature space geometry and adversarial robustness. One such study by Fawzi et al. [75] revealed that adversarial

examples lie near the decision boundaries of the classifier in the feature space, and their existence is related to the curvature of the decision boundary.

The robustness of a classifier can be characterized by the margin around the decision boundary:

$$\rho(\boldsymbol{\theta}) = \min_{(x,y)\in\mathcal{D}} y \cdot f_{\boldsymbol{\theta}}(x), \tag{71}$$

where $\mathcal{D}$ denotes the data distribution, $f_{\boldsymbol{\theta}}(x)$ is the classifier output, and y is the true label. A larger margin $\rho(\boldsymbol{\theta})$ corresponds to increased robustness against adversarial examples.

Another recent study by Tsipras et al. [76] examined the trade-off between standard accuracy and adversarial robustness. They analyzed the linear approximation of the classifier loss function, $L(\boldsymbol{\theta})$, around a data point (x, y), and derived the adversarial perturbation, δ, as follows:

$$\delta = -\epsilon \cdot \text{sign}(\nabla_x L(\boldsymbol{\theta}, x, y)), \tag{72}$$

where ϵ is a small constant determining the maximum allowed perturbation. This adversarial perturbation causes a decrease in the margin around the decision boundary, implying a trade-off between standard accuracy and adversarial robustness.

In a different study, Hein et al. [77] introduced a geometric perspective on adversarial robustness by defining the concept of robustness certificates. They proposed a measure called the cross-Lipschitz regularization (CLR) that quantifies the robustness of a classifier, $f_{\boldsymbol{\theta}}(x)$, as follows:

$$\text{CLR}(f_{\boldsymbol{\theta}}) = \sup_{x,x'\in\mathcal{X}} \frac{|f_{\boldsymbol{\theta}}(x) - f_{\boldsymbol{\theta}}(x')|_2}{|x - x'|_2}, \tag{73}$$

where $\mathcal{X}$ is the input space. A smaller CLR value indicates a more robust classifier. This measure captures the sensitivity of the classifier output to input perturbations and is related to the Lipschitz constant of the classifier.

Recent studies have focused on enhancing the adversarial robustness of deep learning models by investigating the geometry of the feature space. Various methods have been proposed, including the defense layer by Goel et al. [78] that aims to prevent the generation of adversarial noise and Dual Manifold Adversarial Training (DMAT) introduced by Lin et al. [79], which exploits the underlying manifold information of data to achieve comparable robustness to standard adversarial training against L_p attacks. Additionally, Chen and Liu [80] provided a comprehensive overview of adversarial robustness research methods for deep learning models, while Gavrikov and Keuper [81] investigated the properties of convolution filters in adversarially trained models. Moreover, Ghaffari Laleh et al. [82] studied the susceptibility of CNNs and vision transformers (ViTs) to white- and black-box adversarial attacks in clinically relevant weakly-supervised classification tasks, demonstrating ViTs' higher robustness to such attacks attributed to their more robust latent representation of clinically relevant categories compared to CNNs.

Table 12 provides a summary of research contributions related to adversarial robustness and feature space geometry. These studies investigate various aspects of adversarial robustness, including the relationship between adversarial examples and decision boundary curvature, the trade-off between standard accuracy and robustness, and the development of robustness certificates. They contribute to a better understanding of the geometry of the feature space in the context of adversarial robustness and provide insights for developing more robust and reliable deep learning models.

Adversarial attacks are a major concern for the reliability and safety of deep learning models. Recent research has explored the relationship between adversarial robustness and the geometry of the feature space. Adversarial examples tend to lie near the decision boundaries of the classifier, and their existence is related to the curvature of the decision boundary. The margin around the decision boundary is a measure of the robustness of a classifier, with a larger margin indicating increased robustness. There is a trade-off between standard accuracy and adversarial robustness, and the sensitivity of the classifier output to input perturbations can be quantified using the cross-Lipschitz robustness measure.

Table 12. Summary of Research on Adversarial Robustness and Feature Space Geometry.

Reference	Focus	Contribution
Fawzi et al. [75]	Adversarial examples	Relationship with decision boundary curvature
Tsipras et al. [76]	Trade-off between accuracy and robustness	Linear approximation of classifier loss function
Hein et al. [77]	Robustness certificates	Cross-Lipschitz regularization (CLR)
Goel et al. [78]	Defense layer for CNNs	Preventing generation of adversarial noise
Lin et al. [79]	Dual Manifold Adversarial Training (DMAT)	Exploiting underlying manifold information
Chen and Liu [80]	Overview of adversarial robustness methods	Comprehensive review of research
Gavrikov and Keuper [81]	Convolution filters in adversarially-trained models	Investigation of filter properties
Ghaffari Laleh et al. [82]	CNNs and ViTs under adversarial attacks	Comparison of robustness in weakly-supervised classification tasks

4.8. Feature Space Geometry and Transfer Learning

Transfer learning is a widely-used technique in deep learning, in which pre-trained models are fine-tuned on a new task. The feature space geometry plays a crucial role in the success of transfer learning. A study by Yosinski et al. [83] found that the feature spaces of lower layers in neural networks tend to be more general and transferable than those of higher layers.

The transferability of a feature space can be quantified by measuring the similarity between the feature spaces of the source and target tasks:

$$\tau(\mathcal{F}_s, \mathcal{F}_t) = \frac{\langle \mathcal{F}_s, \mathcal{F}_t \rangle}{|\mathcal{F}_s||\mathcal{F}_t|}, \tag{74}$$

where $\mathcal{F}_s$ and $\mathcal{F}_t$ denote the feature spaces of the source and target tasks, respectively, and $\tau(\cdot, \cdot)$ measures the cosine similarity between the feature spaces. A high similarity score indicates that the feature space is more transferable between the tasks.

Several recent studies have focused on exploring the transferability of features between different domains of time series data, real-time crash risk models, and image classification tasks. Otović et al. [84] investigated the transferability of features in time series data and found that transfer learning is likely to improve or not negatively affect the model's predictive performance or its training convergence rate. Man et al. [85] proposed a method combining Wasserstein Generative Adversarial Network and transfer learning to address the spatio-temporal transferability issue of real-time crash risk models. Their findings show that transfer learning can improve model transferability under extremely imbalanced settings, resulting in models that are transferable temporally, spatially, and spatio-temporally. Pándy et al. [86] proposed a novel method, Gaussian Bhattacharyya Coefficient, for quantifying transferability between a source model and a target dataset. Their results showed that GBC outperforms state-of-the-art transferability metrics on most evaluation criteria in semantic segmentation settings and performs well on dataset transferability and architecture selection problems for image classification.

The feature space geometry has continued to be a significant area of research in transfer learning, with several studies exploring different aspects of this relationship. Xu et al. [87] investigated the connection between the similarity of feature spaces and the performance of

transfer learning. They proposed a task similarity measure based on the normalized mutual information (NMI) between the feature space distributions of the source and target tasks:

$$\mathrm{NMI}(\mathcal{F}_s, \mathcal{F}_t) = \frac{2 \cdot I(\mathcal{F}_s; \mathcal{F}_t)}{H(\mathcal{F}_s) + H(\mathcal{F}_t)}, \tag{75}$$

where $I(\cdot;\cdot)$ denotes the mutual information and $H(\cdot)$ denotes the entropy. This measure quantifies the degree of dependence between the source and target feature spaces, and higher values of NMI indicate a stronger relationship, leading to better transfer learning performance.

Xie et al. [88] studied the importance of feature alignment in transfer learning. They introduced a feature alignment loss, which aims to minimize the distance between the source and target feature space distributions:

$$\mathcal{L}_{\mathrm{align}}(\mathcal{F}_s, \mathcal{F}_t) = \frac{1}{n} \sum i = 1^n \|\mathcal{F}_s(x_i) - \mathcal{F}_t(x_i)\|_2^2, \tag{76}$$

where n is the number of samples and x_i is the i-th sample. By optimizing this loss, the feature spaces become more aligned, resulting in improved transfer learning performance.

Raghu et al. [89] proposed a method called Transfusion to enhance transfer learning by selectively transferring features. They introduced a transfer matrix $\mathcal{T}$, which maps the source feature space to the target feature space:

$$\mathcal{F}_t(x) = \mathcal{T}(\mathcal{F}_s(x)), \tag{77}$$

where the transfer matrix $\mathcal{T}$ is learned during the fine-tuning process. This approach allows for selective transfer of features, which can improve the transferability and adaptability of pre-trained models.

Zamir et al. [90] explored the task structure in transfer learning and introduced the concept of taskonomy, which models the relationships between tasks using a directed graph. They proposed an optimization problem to find the optimal transfer learning strategy:

$$\min_{\mathcal{G}} \sum_{t \in \mathcal{T}} \mathrm{Cost}(\mathcal{F}_t) \quad \text{subject to} \quad \mathrm{Constraints}(\mathcal{G}), \tag{78}$$

where $\mathcal{G}$ is the task graph, $\mathcal{T}$ is the set of tasks, and the cost function represents the performance of transfer learning. By solving this optimization problem, they found the optimal task relationships that yield the best transfer learning performance.

Table 13 provides a summary of research contributions related to feature space geometry and transfer learning. These studies investigate various aspects of transfer learning, such as the transferability of lower layers in neural networks, the transferability of features between different domains, and the development of novel methods to quantify transferability. Additionally, they explore techniques to improve transfer learning performance, such as feature alignment, selective transfer, and optimizing task relationships.

Transfer learning is a widely-used technique in deep learning that involves fine-tuning pre-trained models on a new task. The feature space geometry plays a crucial role in the success of transfer learning. Several recent studies have explored the transferability of features between different domains of time series data, real-time crash risk models, and image classification tasks. The findings indicate that transfer learning can improve model transferability and result in models that are transferable temporally, spatially, and spatio-temporally. Additionally, recent research has focused on different aspects of the feature space geometry and transfer learning, including the connection between the similarity of feature spaces and the performance of transfer learning, the importance of feature alignment in transfer learning, and selective transfer of features to improve adaptability. Moreover, task structure has been explored to model the relationships between tasks using a directed graph and find the optimal task relationships that yield the best transfer learning performance.

Table 13. Summary of Research on Feature Space Geometry and Transfer Learning.

Reference	Focus	Contribution
Yosinski et al. [83]	Transferability of lower layers	Lower layers more general and transferable
Otović et al. [84]	Time series data transferability	Improved predictive performance and convergence rate
Man et al. [85]	Spatio-temporal transferability in crash risk models	Improved model transferability under imbalanced settings
Pándy et al. [86]	Gaussian Bhattacharyya Coefficient	Quantifying transferability in semantic segmentation
Xu et al. [87]	Normalized mutual information (NMI)	Quantifying task similarity for better transfer learning
Xie et al. [88]	Feature alignment loss	Improved transfer learning performance through alignment
Raghu et al. [89]	Transfusion for selective transfer	Enhanced transferability and adaptability of models
Zamir et al. [90]	Taskonomy	Optimal task relationships for best transfer learning performance

4.9. Disentangled Representations in Feature Space

Disentangled representations are a desirable property in deep learning models, as they facilitate the separation of underlying explanatory factors in the data. The geometry of the feature space can provide insights into the degree of disentanglement achieved by a model. Mathematically, a disentangled representation can be characterized by the independence between the latent factors:

$$p(\boldsymbol{z}) = \prod_{i=1}^{n} p(z_i), \tag{79}$$

where $\boldsymbol{z}$ is a latent representation vector and z_i represents individual latent factors. The geometry of the feature space in disentangled representation learning has been investigated in various studies, such as the β-VAE [91] and InfoGAN [92].

The pursuit of disentangled representations has led to numerous recent studies proposing various mathematical frameworks and models to achieve a higher degree of disentanglement.

Locatello et al. [93] introduced a fairness-aware framework for disentangled representations. They proposed a regularization term to encourage fairness by minimizing the mutual information between the sensitive attribute s and the disentangled latent factors $\boldsymbol{z}$:

$$\mathcal{L}_{\text{fair}}(\boldsymbol{z}, s) = -\sum i = 1^{n} I(z_i; s), \tag{80}$$

where $I(\cdot;\cdot)$ denotes the mutual information. By minimizing this loss, the learned representations become more disentangled with respect to the sensitive attribute, leading to fairer representations.

Achille et al. [94] studied the relationship between the information bottleneck principle and disentangled representations. They derived an upper bound on the mutual information between the input data $\boldsymbol{x}$ and the disentangled latent factors $\boldsymbol{z}$:

$$I(\boldsymbol{x}; \boldsymbol{z}) \leq \sum_{i=1}^{n} H(z_i) - \beta \mathcal{L}_{\text{IB}}(\boldsymbol{z}), \tag{81}$$

where $H(\cdot)$ is the entropy, β is a trade-off parameter, and $\mathcal{L}_{\text{IB}}(\cdot)$ is the information bottleneck loss. This result suggests that optimizing the information bottleneck objective can lead to the emergence of disentangled representations.

Kim and Mnih [95] introduced a method called Disentangling by Factorizing (DF) to learn disentangled representations in a purely unsupervised manner. They proposed a factorized variational autoencoder (FVAE) that models the joint distribution of data $\boldsymbol{x}$ and latent factors $\boldsymbol{z}$ as:

$$p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z}) \prod_{i=1}^{n} p(z_i), \tag{82}$$

where $p(\boldsymbol{x}|\boldsymbol{z})$ is the likelihood and $p(z_i)$ are the prior distributions over latent factors. They also introduced a structured inference network $q(\boldsymbol{z}|\boldsymbol{x})$ to approximate the true posterior $p(\boldsymbol{z}|\boldsymbol{x})$. The model is trained by optimizing the evidence lower bound (ELBO):

$$\mathcal{L}_{\text{ELBO}}(\boldsymbol{x}, \boldsymbol{z}) = \mathbb{E}q(\boldsymbol{z}|\boldsymbol{x})[\log p(\boldsymbol{x}|\boldsymbol{z})] - \sum_{i=1}^{n} \text{KL}(q(z_i|\boldsymbol{x})|p(z_i)), \tag{83}$$

where $\text{KL}(\cdot|\cdot)$ denotes the Kullback–Leibler divergence. By optimizing this objective, the model learns disentangled representations that separate the underlying factors of variation in the data.

Table 14 provides a summary of research contributions related to disentangled representations in feature space. These studies investigate various aspects of disentangled representations, such as balancing reconstruction and disentanglement, using mutual information for unsupervised disentanglement, and incorporating fairness-aware frameworks. They contribute to a better understanding of the geometry of the feature space in the context of disentangled representations and provide insights for developing models that can separate the underlying explanatory factors in the data.

Table 14. Summary of Research on Disentangled Representations in Feature Space.

Reference	Focus	Contribution
Higgins et al. [91]	β-VAE	Balancing reconstruction and disentanglement
Chen et al. [92]	InfoGAN	Mutual information for unsupervised disentanglement
Locatello et al. [93]	Fairness-aware framework	Regularization for fair disentangled representations
Achille et al. [94]	Information bottleneck principle	Connection to disentangled representations
Kim and Mnih [95]	Disentangling by Factorizing (DF)	Unsupervised disentanglement via factorized VAE

The pursuit of disentangled representations in deep learning has led to various studies exploring the geometry of the feature space and proposing models to achieve a higher degree of disentanglement. Disentangled representations facilitate the separation of underlying explanatory factors in the data and are characterized by the independence between the latent factors. Recent studies have proposed various mathematical frameworks and models to achieve a higher degree of disentanglement.

5. Challenges and Future Directions

In this section, the challenges and future directions in the field of feature space geometry in deep learning will be discussed. The study of feature space geometry is a rapidly growing area of research, and numerous open questions remain, offering exciting oppor-

tunities for the development of novel mathematical techniques and insights. Several key challenges and possible future research directions are outlined below.

5.1. Understanding the Geometry of Overparameterized Models

Overparameterized models [96–98] have shown remarkable success in deep learning, often achieving better generalization despite the increased capacity. A better understanding of the geometry of feature spaces in overparameterized models could shed light on the mechanisms behind their improved generalization. Mathematically, the characterization of the feature space geometry in such models can be challenging, as the dimensionality of the space grows significantly:

$$\dim(\mathcal{F}) = \mathcal{O}(N), \tag{84}$$

where $\dim(\mathcal{F})$ denotes the dimensionality of the feature space and N represents the number of parameters in the model. Future research could focus on the development of efficient mathematical techniques to analyze high-dimensional feature spaces in overparameterized models.

Understanding the geometry of overparameterized models poses a challenge due to implicit regularization, wherein optimization tends to favor simpler solutions that generalize well, despite the model having a large capacity. This implicit regularization can be viewed as a constraint on the parameter space:

$$\mathcal{R}(\boldsymbol{\theta}) \leq C, \tag{85}$$

where $\mathcal{R}(\boldsymbol{\theta})$ is a regularizer, $\boldsymbol{\theta}$ denotes the model parameters, and C is a constant. The nature of this implicit regularizer is not yet fully understood, and further research is needed to reveal its mathematical properties.

Additionally, the nonconvexity of the optimization landscape, with local minima and saddle points, creates another challenge for understanding geometry in overparameterized models. Mathematical techniques analyzing the Hessian matrix of the loss function can aid in gaining insights into the convergence properties and generalization capabilities of such models:

$$\boldsymbol{H}(\boldsymbol{\theta}) = \nabla^2 \mathcal{L}(\boldsymbol{\theta}), \tag{86}$$

where $\boldsymbol{H}(\boldsymbol{\theta})$ is the Hessian matrix and $\mathcal{L}(\boldsymbol{\theta})$ denotes the loss function. Analyzing the eigenvalues and eigenvectors of the Hessian can help understand the local curvature of the optimization landscape and reveal the properties of the feature space geometry.

Future research can focus on investigating the geometry of feature spaces in the context of NTKs [42], which provide a linearized approximation of the model's dynamics during training, making them a powerful tool for analyzing overparameterized models:

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}') = \nabla_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{x})^T \nabla_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{x}'), \tag{87}$$

where $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}')$ is the NTK, $\boldsymbol{x}$ and $\boldsymbol{x}'$ are input data points, and $\mathcal{F}(\boldsymbol{x})$ denotes the feature space representation.

Table 15 provides a summary of research challenges related to understanding the geometry of overparameterized models. These challenges include analyzing high-dimensional feature spaces, investigating the nature and properties of implicit regularization, understanding the nonconvex optimization landscape, and utilizing Neural Tangent Kernels (NTKs) to study the geometry of feature spaces. Addressing these challenges could lead to a better understanding of the mechanisms behind the improved generalization capabilities of overparameterized models.

Table 15. Summary of Research Challenges in Understanding the Geometry of Overparameterized Models.

Challenge	Description
High-dimensional feature spaces	Developing efficient mathematical techniques for high-dimensional analysis
Implicit regularization	Investigating the nature and properties of the implicit regularizer
Non-convex optimization landscape	Analyzing the Hessian matrix to understand local curvature and feature space properties
Neural Tangent Kernels (NTKs)	Investigating the geometry of feature spaces using linearized approximations of model dynamics

5.2. Feature Space Geometry in Unsupervised and Semi-Supervised Learning

While the majority of research on feature space geometry has focused on supervised learning, the study of feature space geometry in unsupervised and semi-supervised learning [99,100] settings remains relatively unexplored. Understanding the geometry of feature spaces in these settings could be crucial for developing more effective representation learning techniques. Future work could involve the development of mathematical tools and theories that capture the geometry of feature spaces in unsupervised and semi-supervised settings, such as:

$$\mathcal{G}(\mathcal{F}_{\text{unsup}}) = \mathcal{T}(\mathcal{F}_{\text{sup}}), \tag{88}$$

where $\mathcal{G}$ denotes a geometric transformation, and $\mathcal{F}_{\text{unsup}}$ and $\mathcal{F}_{\text{sup}}$ represent the feature spaces in unsupervised and supervised settings, respectively.

Measuring the similarity between learned feature spaces and the underlying data structure is a difficult task in unsupervised and semi-supervised learning. Mutual information can be employed to quantify this similarity:

$$\mathcal{I}(\boldsymbol{X}; \boldsymbol{Z}) = \sum_{\boldsymbol{x} \in \boldsymbol{X}} \sum_{\boldsymbol{z} \in \boldsymbol{Z}} p(\boldsymbol{x}, \boldsymbol{z}) \log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{x}) p(\boldsymbol{z})}, \tag{89}$$

where $\boldsymbol{X}$ and $\boldsymbol{Z}$ represent the input data and the learned representations, respectively, and $p(\cdot)$ denotes the corresponding probability distributions. A high mutual information between $\boldsymbol{X}$ and $\boldsymbol{Z}$ indicates that the learned representations capture the underlying structure of the data. Future research could develop methods for optimizing mutual information in unsupervised and semi-supervised settings.

Inductive biases [101,102] play a significant role in shaping the geometry of feature spaces in unsupervised and semi-supervised learning. By restricting the hypothesis space, inductive biases can have a profound impact on the geometry of feature spaces:

$$\mathcal{H}_{\text{biased}} = \mathcal{H}_{\text{unbiased}} \cap \mathcal{B}, \tag{90}$$

where $\mathcal{H}_{\text{biased}}$ and $\mathcal{H}_{\text{unbiased}}$ denote the biased and unbiased hypothesis spaces, respectively, and $\mathcal{B}$ represents the set of constraints introduced by the inductive bias. Future work could investigate the impact of different inductive biases on the geometry of feature spaces in unsupervised and semi-supervised learning and develop techniques for designing models with appropriate biases.

A promising research direction in this field is to develop unified mathematical frameworks for the study of feature space geometry in supervised, unsupervised, and semi-supervised learning. An area of focus in this regard could be to investigate the interplay between the loss functions used in these different settings:

$$\mathcal{L}_{\text{semi}}(\boldsymbol{\theta}) = \alpha \mathcal{L}_{\text{sup}}(\boldsymbol{\theta}) + (1 - \alpha) \mathcal{L}_{\text{unsup}}(\boldsymbol{\theta}), \tag{91}$$

where $\mathcal{L}_{\text{semi}}(\boldsymbol{\theta})$, $\mathcal{L}_{\text{sup}}(\boldsymbol{\theta})$, and $\mathcal{L}_{\text{unsup}}(\boldsymbol{\theta})$ denote the loss functions for semi-supervised, supervised, and unsupervised learning, respectively, and $\alpha \in [0, 1]$ is a weighting factor. Understanding the interplay between these loss functions and their impact on the geometry of feature spaces could lead to novel algorithms for representation learning.

An important area of research to consider is the impact of objective functions on the geometry of feature spaces and how this affects optimization dynamics. An approach to investigating this could be analyzing the gradients of loss functions with respect to the feature space, which can provide insights into how the choice of objective function influences the geometry of the feature space:

$$\nabla_z \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z}) = \frac{\partial \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z})}{\partial \boldsymbol{z}}, \tag{92}$$

where ∇_z denotes the gradient with respect to the learned representation $\boldsymbol{z}$, and $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z})$ represents the loss function associated with the model parameters $\boldsymbol{\theta}$. Analyzing the properties of these gradients and their interplay with the geometry of feature spaces may provide insights into the optimization challenges faced by unsupervised and semi-supervised learning algorithms.

Developing theoretical frameworks for understanding the robustness of feature spaces in unsupervised and semi-supervised learning is another important direction for future research. The geometry of the feature space plays a significant role in the model's ability to handle noisy data or resist adversarial attacks. A measure of robustness could be defined by analyzing the sensitivity of the feature space to small perturbations in the input data:

$$\mathcal{R}(\boldsymbol{z}) = \max_{\boldsymbol{z}'} \frac{\|\boldsymbol{z} - \boldsymbol{z}'\|}{\|\boldsymbol{x} - \boldsymbol{x}'\|}, \tag{93}$$

where $\boldsymbol{x}$ and $\boldsymbol{x}'$ represent the original and perturbed input data, respectively, and $\boldsymbol{z}$ and $\boldsymbol{z}'$ are their corresponding learned representations. A low sensitivity of the feature space to perturbations in the input data implies a more robust model. Investigating the relationship between the geometry of feature spaces and robustness can help develop more resilient unsupervised and semi-supervised learning techniques.

Exploring the interplay between feature space geometry and architectural components is another promising direction for future research. Components such as activation functions, pooling layers, and normalization techniques can significantly affect the geometry of the feature space. A mathematical characterization of this relationship can be achieved by modeling the effect of these components on the feature space:

$$\mathcal{F}_{\text{mod}} = \mathcal{M}(\mathcal{F}_{\text{orig}}, \boldsymbol{\phi}), \tag{94}$$

where $\mathcal{F}_{\text{mod}}$ and $\mathcal{F}_{\text{orig}}$ denote the modified and original feature spaces, respectively, $\mathcal{M}$ represents a transformation function capturing the effect of the architectural component, and $\boldsymbol{\phi}$ denotes the parameters associated with the component. Understanding the impact of architectural choices on the geometry of feature spaces could lead to more effective model designs for unsupervised and semi-supervised learning tasks.

Table 16 summarizes the research challenges in feature space geometry for unsupervised and semi-supervised learning. These challenges include developing geometric transformations for mathematical tools, optimizing mutual information for representation quality, investigating the impact of inductive biases, creating unified mathematical frameworks, analyzing the influence of objective functions, studying the robustness of feature spaces, and exploring the impact of architectural components. Addressing these challenges could lead to a better understanding of unsupervised and semi-supervised learning and help develop more effective representation learning techniques.

Table 16. Summary of Research Challenges in Feature Space Geometry for Unsupervised and Semi-Supervised Learning.

Challenge	Description
Geometric transformations	Developing mathematical tools to capture geometry in unsupervised and semi-supervised settings
Mutual information	Optimizing mutual information to measure similarity between learned representations and data structure
Inductive biases	Investigating the impact of inductive biases on feature space geometry
Unified mathematical frameworks	Exploring the interplay between loss functions and feature space geometry across learning settings
Objective function impact	Analyzing gradients of loss functions to understand their influence on feature space geometry
Robustness	Developing theoretical frameworks to study robustness of feature spaces
Architectural components	Exploring the impact of architectural choices on feature space geometry

5.3. Interpretable Feature Space Geometry

Interpretability is an important aspect of deep learning models [103,104], especially in safety-critical applications. A better understanding of the feature space geometry could facilitate the development of more interpretable models. Future research could focus on the identification of mathematical properties and structures within feature spaces that correspond to human-understandable representations:

$$\mathcal{I}(\mathcal{F}) = \mathcal{H}(\mathcal{R}), \tag{95}$$

where $\mathcal{I}$ denotes an interpretability function, $\mathcal{F}$ is the feature space, $\mathcal{H}$ is a transformation into human-understandable representations, and $\mathcal{R}$ represents the human-understandable representation space.

A major challenge in developing interpretable feature spaces lies in quantifying the degree of interpretability. One possible approach is to define a measure of interpretability based on the similarity between the learned feature space $\mathcal{F}$ and the human-understandable representation space $\mathcal{R}$:

$$\mathcal{S}(\mathcal{F}, \mathcal{R}) = \frac{\langle \mathcal{F}, \mathcal{R} \rangle}{\|\mathcal{F}\| \|\mathcal{R}\|}, \tag{96}$$

where $\mathcal{S}(\cdot, \cdot)$ denotes the cosine similarity between the two spaces. A high similarity score indicates that the learned feature space closely aligns with human-understandable representations.

In order to enhance the interpretability of deep learning models, further investigation is needed to examine how different architectural choices and training strategies impact the geometry of feature spaces. The selection of activation functions, regularization techniques, and inductive biases can all contribute to the interpretability of learned feature spaces and should be explored:

$$\mathcal{F}_{\text{interp}} = \mathcal{A}(\mathcal{F}_{\text{orig}}, \boldsymbol{\psi}), \tag{97}$$

where $\mathcal{F}_{\text{interp}}$ and $\mathcal{F}_{\text{orig}}$ denote the interpretable and original feature spaces, respectively, $\mathcal{A}$ represents a transformation function capturing the effect of architectural choices and training strategies, and $\boldsymbol{\psi}$ denotes the associated parameters.

Another avenue for future research is the development of mathematical frameworks that combine disentangled representations with interpretable feature spaces. By analyzing the relationship between disentangled latent factors and human-understandable representations, models can be designed to both disentangle explanatory factors and be interpretable:

$$\mathcal{D}(\mathcal{F}_{\text{interp}}) = \mathcal{E}(\mathcal{R}), \tag{98}$$

where $\mathcal{D}$ denotes a disentanglement function, and $\mathcal{E}$ is a transformation into a disentangled representation space. Understanding the interplay between interpretability and disentanglement could pave the way for more interpretable and effective deep learning models.

A promising direction for research is exploring the role of feature space geometry in Explainable AI (XAI) techniques [105,106]. By developing methods that can project high-dimensional feature spaces onto lower-dimensional, human-understandable spaces, deep learning models can provide more intuitive explanations for their predictions:

$$\mathcal{P}(\mathcal{F}) = \mathcal{Q}(\mathcal{R}), \tag{99}$$

where $\mathcal{P}$ denotes a projection function, and $\mathcal{Q}$ is a transformation into a lower-dimensional, human-understandable space. Developing such projection techniques could greatly enhance the explainability and trustworthiness of deep learning models, especially in safety-critical applications.

Table 17 summarizes the research challenges in interpretable feature space geometry. The challenges include quantifying interpretability in feature spaces, investigating the impact of architectural choices and training strategies, understanding the interplay between interpretability and disentanglement, and exploring the role of feature space geometry in Explainable AI techniques. Addressing these challenges could contribute to the development of more interpretable and trustworthy deep learning models.

Table 17. Summary of Research Challenges in Interpretable Feature Space Geometry.

Challenge	Description
Quantifying interpretability	Developing measures to quantify the degree of interpretability in feature spaces
Impact of architectural choices	Investigating the effect of architectural choices and training strategies on feature space interpretability
Interplay between interpretability and disentanglement	Understanding the relationship between interpretable and disentangled feature spaces
Feature space geometry in XAI	Exploring the role of feature space geometry in Explainable AI techniques

5.4. Topological Analysis of Feature Space Geometry

Topological Data Analysis (TDA) techniques [107–109], such as persistent homology [110], have emerged as powerful tools for analyzing complex and high-dimensional data structures. Applying these techniques to the study of feature space geometry in deep learning models could provide valuable insights into the topological properties of these spaces and their relation to the performance of the models.

Persistent homology is a TDA technique that characterizes the topology of a space through its persistent homology groups:

$$\mathcal{PH}_k(\mathcal{F}) = H_k(\mathcal{F}\alpha) \to H_k(\mathcal{F}_\beta) \mid \alpha \leq \beta, \tag{100}$$

where $\mathcal{PH}_k(\mathcal{F})$ denotes the k-th persistent homology group of the feature space $\mathcal{F}$, $H_k(\cdot)$ represents the k-th homology group, and $\mathcal{F}\alpha$ and $\mathcal{F}_\beta$ are subspaces of $\mathcal{F}$ parameterized by the filtration values α and β, respectively.

A significant challenge in using persistent homology to analyze feature space geometry is the complexity and high dimensionality of deep learning models. To overcome this, researchers could explore the development of efficient algorithms and data structures, including dimensionality reduction techniques, that enable the computation of persistent homology in high-dimensional feature spaces:

$$\mathcal{F}_{\text{reduced}} = \mathcal{R}(\mathcal{F}), \tag{101}$$

where $\mathcal{F}_{\text{reduced}}$ denotes the reduced-dimension feature space and $\mathcal{R}$ is a dimensionality reduction function that preserves the topological structure of the original space. Researchers could define a topological complexity measure that quantifies the impact of these properties on the model's performance:

$$\mathcal{C}(\mathcal{F}) = \sum_{k=0}^{n} \omega_k \mathcal{B}_k(\mathcal{PH}_k(\mathcal{F})), \tag{102}$$

where $\mathcal{C}(\mathcal{F})$ denotes the topological complexity of the feature space $\mathcal{F}$, ω_k represents the weight assigned to the k-th persistent homology group, and $\mathcal{B}_k$ is a function that computes the topological complexity contribution from the k-th persistent homology group.

Developing novel methods for visualizing and understanding the high-dimensional topological structures of feature spaces is another challenge. A potential approach could involve creating visualizations based on the persistence diagram, which is a summary of the topological information contained in the persistent homology groups:

$$\mathcal{P}(\mathcal{PH}_k(\mathcal{F})) = (b_i, d_i) \mid b_i \leq d_i,\ i = 1, \ldots, m, \tag{103}$$

where $\mathcal{P}(\mathcal{PH}_k(\mathcal{F}))$ denotes the persistence diagram of the k-th persistent homology group of the feature space $\mathcal{F}$, b_i and d_i represent the birth and death times of the i-th topological feature, and m is the number of topological features in the k-th persistent homology group.

Another promising direction for research is to examine the correlation between the topological characteristics of feature spaces and deep learning model performance. This could help uncover relationships that can be exploited to design better models:

$$\rho(\mathcal{PH}_k(\mathcal{F}), \mathcal{P}(\mathcal{M})) = \text{corr}(f(\mathcal{PH}_k(\mathcal{F})), g(\mathcal{P}(\mathcal{M}))), \tag{104}$$

where $\rho(\cdot, \cdot)$ measures the correlation between the topological properties of the feature space $\mathcal{F}$ and the performance $\mathcal{P}(\mathcal{M})$ of the deep learning model $\mathcal{M}$, and $f(\cdot)$ and $g(\cdot)$ are suitable transformation functions.

Developing novel methods for visualizing and understanding the high-dimensional topological structures of feature spaces is crucial for gaining insights into their geometry. Potential approaches could involve the use of Mapper algorithms [111] and visualization techniques such as t-distributed Stochastic Neighbor Embedding (t-SNE) [112] to generate informative visualizations of the topological structure:

$$\mathcal{V}(\mathcal{F}) = \text{t-SNE}(\mathcal{M}(\mathcal{F})), \tag{105}$$

where $\mathcal{V}(\mathcal{F})$ represents the visualization of the feature space $\mathcal{F}$, and $\mathcal{M}(\mathcal{F})$ denotes the Mapper output. By combining these visualization techniques with the analysis of topological properties, researchers could gain a deeper understanding of the underlying structure and complexity of feature spaces in deep learning models.

To enhance the interpretability and generalization of deep learning models, future research could investigate the integration of topological constraints in their design. This could involve developing regularization techniques based on topological properties to improve model performance:

$$\mathcal{L}_{\text{topo}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{orig}}(\boldsymbol{\theta}) + \lambda \mathcal{T}(\mathcal{PH}_k(\mathcal{F})), \tag{106}$$

where $\mathcal{L}_{\text{topo}}(\boldsymbol{\theta})$ denotes the topologically regularized loss function, $\mathcal{L}_{\text{orig}}(\boldsymbol{\theta})$ represents the original loss function, λ is a regularization parameter, and $\mathcal{T}(\mathcal{PH}_k(\mathcal{F}))$ is a function that measures the topological complexity of the feature space.

Table 18 summarizes the research challenges in topological analysis of feature space geometry. The challenges include developing efficient algorithms for high-dimensional spaces, creating methods for visualizing topological structures, examining the correlation between topological characteristics and model performance, and integrating topological constraints in deep learning model design. Addressing these challenges could lead to a better understanding of feature space geometry and contribute to the development of more interpretable and generalizable deep learning models.

Table 18. Summary of Research Challenges in Topological Analysis of Feature Space Geometry.

Challenge	Description
Efficient algorithms for high-dimensional spaces	Developing efficient algorithms and data structures for computing persistent homology in high-dimensional feature spaces
Visualization of topological structures	Creating novel methods to visualize and understand high-dimensional topological structures in feature spaces
Correlation with model performance	Examining the relationship between topological characteristics of feature spaces and deep learning model performance
Topological constraints in model design	Investigating the integration of topological constraints in deep learning model design and regularization techniques

5.5. Feature Space Geometry in Multimodal and Multi-Task Learning

Multimodal and multi-task learning are essential approaches in deep learning that aim to leverage multiple input modalities (e.g., text, images, audio) and learn shared representations across different tasks [113–115]. Studying the feature space geometry of models in these settings can provide insights into the interplay between task-specific and shared representations and the role of feature space geometry in transfer learning across different modalities and tasks.

A significant challenge in multimodal learning is to study the feature space geometry of models trained on multiple input modalities. To address this challenge, researchers could define a joint feature space $\mathcal{F}_{\text{joint}}$ that incorporates information from all modalities:

$$\mathcal{F}_{\text{joint}} = \bigoplus_{i=1}^{M} \mathcal{F}_i, \tag{107}$$

where $\bigoplus$ denotes the direct sum operation, M is the number of input modalities, and $\mathcal{F}i$ represents the feature space of the i-th input modality. Studying the properties of $\mathcal{F}_{\text{joint}}$ could provide insights into the interactions between different modalities and their influence on the model's performance.

In multi-task learning settings, the impact of task-specific and shared representations on the geometry of feature spaces is of particular interest. To analyze this, researchers could define a partition of the feature space into task-specific and shared components:

$$\mathcal{F} = \mathcal{F}_{\text{shared}} \oplus \bigoplus_{j=1}^{T} \mathcal{F}_{\text{task},j}, \tag{108}$$

where T is the number of tasks, $\mathcal{F}_{\text{shared}}$ denotes the shared feature space across all tasks, and $\mathcal{F}_{\text{task},j}$ represents the task-specific feature space for the j-th task. By examining the

geometric properties of these components and their interactions, researchers could gain a better understanding of the interplay between shared and task-specific representations.

Exploring the role of feature space geometry in transfer learning across different modalities and tasks is another challenge. To study this, researchers could define a transfer function $\mathcal{T}$ that maps the feature space of the source domain $\mathcal{F}_{\text{src}}$ to the target domain $\mathcal{F}_{\text{tgt}}$:

$$\mathcal{F}_{\text{tgt}} = \mathcal{T}(\mathcal{F}_{\text{src}}), \tag{109}$$

Investigating the properties of $\mathcal{T}$ and its impact on the geometry of $\mathcal{F}_{\text{tgt}}$ could reveal the factors that contribute to successful transfer learning across different modalities and tasks.

6. Conclusions

The multifaceted nature of the geometry of feature space in deep learning models has been elucidated in this comprehensive review. By delving into the intricate relationships between feature spaces and various aspects of deep learning models, such as activation functions, normalization methods, and model architectures, a panoramic view of the current state of the field has been provided in this review. Furthermore, recent studies have been examined, which have contributed significantly to the understanding of manifold structures, curvature, critical points, adversarial robustness, and transfer learning, among other topics.

Throughout this paper, the importance of comprehending the geometric properties of feature spaces has been underscored, as such understanding can lead to the development of novel and efficient deep learning architectures, optimization techniques, and regularization methods. The discussion of challenges and future directions in this paper has shed light on several pertinent research areas, including overparameterized models, unsupervised and semi-supervised learning, interpretable feature space geometry, topological analysis, and multimodal and multi-task learning.

In conclusion, the study of the geometry of feature space in deep learning models is a vast and multidimensional domain that interweaves diverse theoretical and practical perspectives. A profound comprehension of the geometric underpinnings of deep learning models has the potential to pave the way for groundbreaking advancements in artificial intelligence and machine learning. By integrating topological, set-theoretic, and real number analysis methods, researchers can delve deeper into the complexities of feature space geometry and unlock hitherto unexplored facets of deep learning. As the exploration of the frontiers of deep learning continues, it is believed that a holistic understanding of the geometry of feature space will serve as a cornerstone for shaping the future of this dynamic field.

Funding: This work was supported by a research grant funded by Generative Artificial Intelligence System Inc. (GAIS).

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Chai, J.; Zeng, H.; Li, A.; Ngai, E.W. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Mach. Learn. Appl.* **2021**, *6*, 100134. [CrossRef]
2. Zhai, X.; Kolesnikov, A.; Houlsby, N.; Beyer, L. Scaling vision transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–22 June 2022; pp. 12104–12113.
3. Wortsman, M.; Ilharco, G.; Kim, J.W.; Li, M.; Kornblith, S.; Roelofs, R.; Lopes, R.G.; Hajishirzi, H.; Farhadi, A.; Namkoong, H.; et al. Robust fine-tuning of zero-shot models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–22 June 2022; pp. 7959–7971.
4. Wu, S.; Roberts, K.; Datta, S.; Du, J.; Ji, Z.; Si, Y.; Soni, S.; Wang, Q.; Wei, Q.; Xiang, Y.; et al. Deep learning in clinical natural language processing: A methodical review. *J. Am. Med. Inform. Assoc.* **2020**, *27*, 457–470. [CrossRef] [PubMed]

5. Nassif, A.B.; Shahin, I.; Attili, I.; Azzeh, M.; Shaalan, K. Speech recognition using deep neural networks: A systematic review. *IEEE Access* **2019**, *7*, 19143–19165. [CrossRef]
6. Kansizoglou, I.; Bampis, L.; Gasteratos, A. Deep feature space: A geometrical perspective. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 6823–6838. [CrossRef] [PubMed]
7. Bronstein, M.M.; Bruna, J.; Cohen, T.; Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv* **2021**, arXiv:2104.13478.
8. Pineda, J.; Midtvedt, B.; Bachimanchi, H.; Noé, S.; Midtvedt, D.; Volpe, G.; Manzo, C. Geometric deep learning reveals the spatiotemporal features of microscopic motion. *Nat. Mach. Intell.* **2023**, *5*, 71–82. [CrossRef]
9. Ghosh, R.; Motani, M. Local Intrinsic Dimensional Entropy. *arXiv* **2023**, arXiv:2304.02223.
10. Magai, G.; Ayzenberg, A. Topology and geometry of data manifold in deep learning. *arXiv* **2022**, arXiv:2204.08624.
11. Nguyen, N.D.; Huang, J.; Wang, D. A deep manifold-regularized learning model for improving phenotype prediction from multi-modal data. *Nat. Comput. Sci.* **2022**, *2*, 38–46. [CrossRef]
12. Li, X.; Jiao, Z.; Zhang, H.; Zhang, R. Deep Manifold Learning with Graph Mining. *arXiv* **2022**, arXiv:2207.08377.
13. Xu, Z.; Wen, S.; Wang, J.; Liu, G.; Wang, L.; Yang, Z.; Ding, L.; Zhang, Y.; Zhang, D.; Xu, J.; et al. AMCAD: Adaptive Mixed-Curvature Representation based Advertisement Retrieval System. In Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 9–12 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 3439–3452.
14. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 1–74. [CrossRef]
15. Shrestha, A.; Mahmood, A. Review of deep learning algorithms and architectures. *IEEE Access* **2019**, *7*, 53040–53065. [CrossRef]
16. Sarker, I.H. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Comput. Sci.* **2021**, *2*, 420. [CrossRef]
17. Suganyadevi, S.; Seethalakshmi, V.; Balasamy, K. A review on deep learning in medical image analysis. *Int. J. Multimed. Inf. Retr.* **2022**, *11*, 19–38. [CrossRef]
18. Lakshmanna, K.; Kaluri, R.; Gundluru, N.; Alzamil, Z.S.; Rajput, D.S.; Khan, A.A.; Haq, M.A.; Alhussen, A. A review on deep learning techniques for IoT data. *Electronics* **2022**, *11*, 1604. [CrossRef]
19. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
20. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6999–7019. [CrossRef]
21. Dubey, S.R.; Singh, S.K.; Chaudhuri, B.B. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing* **2022**, *503*, 92–108. [CrossRef]
22. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
23. Clevert, D.A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv* **2015**, arXiv:1511.07289.
24. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.
25. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 7–9 July 2015; pp. 448–456.
26. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
27. Wu, Y.; He, K. Group normalization. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
28. Brahma, P.P.; Wu, D.; She, Y. Why deep learning works: A manifold disentanglement perspective. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *27*, 1997–2008. [CrossRef] [PubMed]
29. Bengio, Y.; Courville, A.C.; Vincent, P. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR, abs/1206.5538* **2012**, *1*, 2012.
30. Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; Bronstein, M.M. Geometric deep learning on graphs and manifolds using mixture model cnns. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5115–5124.
31. Yuan, Y.; Mou, L.; Lu, X. Scene recognition by manifold regularized deep learning architecture. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 2222–2233. [CrossRef] [PubMed]
32. Hong, C.; Yu, J.; Zhang, J.; Jin, X.; Lee, K.H. Multimodal face-pose estimation with multitask manifold deep learning. *IEEE Trans. Ind. Inform.* **2018**, *15*, 3952–3961. [CrossRef]
33. Li, Z.; Huang, H.; Zhang, Z.; Shi, G. Manifold-Based Multi-Deep Belief Network for Feature Extraction of Hyperspectral Image. *Remote Sens.* **2022**, *14*, 1484. [CrossRef]
34. Ke, Z.; Cui, Z.X.; Huang, W.; Cheng, J.; Jia, S.; Ying, L.; Zhu, Y.; Liang, D. Deep manifold learning for dynamic MR imaging. *IEEE Trans. Comput. Imaging* **2021**, *7*, 1314–1327. [CrossRef]
35. Li, Z.; Huang, H.; Duan, Y.; Shi, G. DLPNet: A deep manifold network for feature extraction of hyperspectral imagery. *Neural Netw.* **2020**, *129*, 7–18. [CrossRef]

36. He, W.; Jiang, Z.; Zhang, C.; Sainju, A.M. CurvaNet: Geometric deep learning based on directional curvature for 3D shape analysis. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020; pp. 2214–2224.
37. Bachmann, G.; Bécigneul, G.; Ganea, O. Constant curvature graph convolutional networks. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 13–18 July 2020; pp. 486–496.
38. Ma, Y.; Jiao, L.; Liu, F.; Yang, S.; Liu, X.; Li, L. Curvature-Balanced Feature Manifold Learning for Long-Tailed Classification. *arXiv* **2023**, arXiv:2303.12307.
39. Lin, J.; Shi, X.; Gao, Y.; Chen, K.; Jia, K. Cad-pu: A curvature-adaptive deep learning solution for point set upsampling. *arXiv* **2020**, arXiv:2009.04660.
40. Arvanitidis, G.; Hansen, L.K.; Hauberg, S. Latent space oddity: On the curvature of deep generative models. *arXiv* **2017**, arXiv:1710.11379.
41. Lee, J.; Bahri, Y.; Novak, R.; Schoenholz, S.S.; Pennington, J.; Sohl-Dickstein, J. Deep neural networks as gaussian processes. *arXiv* **2017**, arXiv:1711.00165.
42. Jacot, A.; Gabriel, F.; Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 8580–8589.
43. Matthews, A.G.d.G.; Rowland, M.; Hron, J.; Turner, R.E.; Ghahramani, Z. Gaussian process behaviour in wide deep neural networks. *arXiv* **2018**, arXiv:1804.11271.
44. Yang, G. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv* **2019**, arXiv:1902.04760.
45. Yang, G. Wide feedforward or recurrent neural networks of any architecture are gaussian processes. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 9551–9960.
46. Pleiss, G.; Cunningham, J.P. The limitations of large width in neural networks: A deep Gaussian process perspective. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 3349–3363.
47. Agrawal, D.; Papamarkou, T.; Hinkle, J. Wide neural networks with bottlenecks are deep Gaussian processes. *J. Mach. Learn. Res.* **2020**, *21*, 7056–7121.
48. Eldan, R.; Mikulincer, D.; Schramm, T. Non-asymptotic approximations of neural networks by Gaussian processes. In Proceedings of the Conference on Learning Theory, PMLR, Boulder, CO, USA, 15–19 August 2021; pp. 1754–1775.
49. Zhang, S.Q.; Wang, F.; Fan, F.L. Neural network gaussian processes by increasing depth. *IEEE Trans. Neural Netw. Learn. Syst.* 2022, *early access*. [CrossRef]
50. Dutordoir, V.; Hensman, J.; van der Wilk, M.; Ek, C.H.; Ghahramani, Z.; Durrande, N. Deep neural networks as point estimates for deep Gaussian processes. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 9443–9455.
51. Chaudhari, P.; Choromanska, A.; Soatto, S.; LeCun, Y.; Baldassi, C.; Borgs, C.; Chayes, J.; Sagun, L.; Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. *J. Stat. Mech. Theory Exp.* **2019**, *2019*, 124018. [CrossRef]
52. Nguyen, Q.; Hein, M. Optimization landscape and expressivity of deep CNNs. In Proceedings of the International Conference on Machine Learning, PMLR, Vienna, Austria, 25–31 July 2018; pp. 3730–3739.
53. Geiger, M.; Spigler, S.; d'Ascoli, S.; Sagun, L.; Baity-Jesi, M.; Biroli, G.; Wyart, M. Jamming transition as a paradigm to understand the loss landscape of deep neural networks. *Phys. Rev. E* **2019**, *100*, 012115. [CrossRef] [PubMed]
54. Kunin, D.; Bloom, J.; Goeva, A.; Seed, C. Loss landscapes of regularized linear autoencoders. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 3560–3569.
55. Simsek, B.; Ged, F.; Jacot, A.; Spadaro, F.; Hongler, C.; Gerstner, W.; Brea, J. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 18–24 July 2021; pp. 9722–9732.
56. Zhou, Y.; Liang, Y. Critical points of neural networks: Analytical forms and landscape properties. *arXiv* **2017**, arXiv:1710.11205.
57. Zhang, Y.; Zhang, Z.; Luo, T.; Xu, Z.J. Embedding principle of loss landscape of deep neural networks. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 14848–14859.
58. Oymak, S.; Soltanolkotabi, M. Toward moderate overparameterization: Global convergence guarantees for training shallow neural networks. *IEEE J. Sel. Areas Inf. Theory* **2020**, *1*, 84–105. [CrossRef]
59. Jia, K.; Tao, D.; Gao, S.; Xu, X. Improving Training of Deep Neural Networks via Singular Value Bounding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
60. Bermeitinger, B.; Hrycej, T.; Handschuh, S. Singular Value Decomposition and Neural Networks. In Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2019: Deep Learning, Munich, Germany, 17–19 September 2019; Tetko, I.V., Kůrková, V., Karpov, P., Theis, F., Eds.; Springer: Cham, Switzerland, 2019; pp. 153–164.
61. Schwab, J.; Antholzer, S.; Nuster, R.; Paltauf, G.; Haltmeier, M. Deep Learning of truncated singular values for limited view photoacoustic tomography. In Proceedings of the Photons Plus Ultrasound: Imaging and Sensing 2019, San Francisco, CA, USA, 3–6 February 2019; Oraevsky, A.A., Wang, L.V., Eds.; International Society for Optics and Photonics, SPIE: Bellingham, WA, USA, 2019; Volume 10878, p. 1087836.
62. Sedghi, H.; Gupta, V.; Long, P.M. The Singular Values of Convolutional Layers. *arXiv* **2018**, arXiv:1805.10408.

63. Cohen, T.; Welling, M. Group Equivariant Convolutional Networks. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; Proceedings of Machine Learning Research; Balcan, M.F., Weinberger, K.Q., Eds.; PMLR: New York, NY, USA, 2016; Volume 48, pp. 2990–2999.
64. Esteves, C.; Allen-Blanchette, C.; Makadia, A.; Daniilidis, K. Learning SO(3) Equivariant Representations with Spherical CNNs. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
65. Singh, J.; Singh, C.; Rana, A. Orthogonal Transforms for Learning Invariant Representations in Equivariant Neural Networks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 1523–1530.
66. McGreivy, N.; Hakim, A. Convolutional Layers Are Not Translation Equivariant. *arXiv* **2022**, arXiv:2206.04979.
67. Aronsson, J.; Müller, D.I.; Schuh, D. Geometrical aspects of lattice gauge equivariant convolutional neural networks. *arXiv* **2023**, arXiv:2303.11448.
68. Zhdanov, M.; Hoffmann, N.; Cesa, G. Implicit Neural Convolutional Kernels for Steerable CNNs. *arXiv* **2022**, arXiv:2212.06096.
69. Toft, C.; Bökman, G.; Kahl, F. Azimuthal Rotational Equivariance in Spherical Convolutional Neural Networks. In Proceedings of the 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 3808–3814.
70. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
71. Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv* **2018**, arXiv:1802.05957.
72. Pauli, P.; Gramlich, D.; Allgöwer, F. Lipschitz constant estimation for 1D convolutional neural networks. *arXiv* **2022**, arXiv:2211.15253.
73. Pauli, P.; Wang, R.; Manchester, I.R.; Allgöwer, F. Lipschitz-bounded 1D convolutional neural networks using the Cayley transform and the controllability Gramian. *arXiv* **2023**, arXiv:2303.11835.
74. Worrall, D.E.; Garbin, S.J.; Turmukhambetov, D.; Brostow, G.J. Harmonic networks: Deep translation and rotation equivariance. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5028–5037.
75. Fawzi, A.; Fawzi, O.; Frossard, P. Analysis of classifiers' robustness to adversarial perturbations. *Mach. Learn.* **2018**, *107*, 481–508. [CrossRef]
76. Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; Madry, A. Robustness may be at odds with accuracy. *arXiv* **2018**, arXiv:1805.12152.
77. Hein, M.; Andriushchenko, M. Formal guarantees on the robustness of a classifier against adversarial manipulation. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 2263–2273.
78. Goel, A.; Agarwal, A.; Vatsa, M.; Singh, R.; Ratha, N.K. DNDNet: Reconfiguring CNN for adversarial robustness. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 22–23.
79. Lin, W.A.; Lau, C.P.; Levine, A.; Chellappa, R.; Feizi, S. Dual manifold adversarial robustness: Defense against lp and non-lp adversarial attacks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 3487–3498.
80. Chen, P.Y.; Liu, S. Holistic adversarial robustness of deep learning models. *arXiv* **2022**, arXiv:2202.07201.
81. Gavrikov, P.; Keuper, J. Adversarial robustness through the lens of convolutional filters. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 139–147.
82. Ghaffari Laleh, N.; Truhn, D.; Veldhuizen, G.P.; Han, T.; van Treeck, M.; Buelow, R.D.; Langer, R.; Dislich, B.; Boor, P.; Schulz, V.; et al. Adversarial attacks and adversarial robustness in computational pathology. *Nat. Commun.* **2022**, *13*, 5711. [CrossRef]
83. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 3320–3328.
84. Otović, E.; Njirjak, M.; Jozinović, D.; Mauša, G.; Michelini, A.; Stajduhar, I. Intra-domain and cross-domain transfer learning for time series data—How transferable are the features? *Knowl.-Based Syst.* **2022**, *239*, 107976. [CrossRef]
85. Man, C.K.; Quddus, M.; Theofilatos, A. Transfer learning for spatio-temporal transferability of real-time crash prediction models. *Accid. Anal. Prev.* **2022**, *165*, 106511. [CrossRef] [PubMed]
86. Pándy, M.; Agostinelli, A.; Uijlings, J.; Ferrari, V.; Mensink, T. Transferability estimation using bhattacharyya class separability. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9172–9182.
87. Xu, H.; Wang, M.; Wang, B. A Difference Standardization Method for Mutual Transfer Learning. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2022; pp. 24683–24697.
88. Xie, Z.; Wen, Z.; Wang, Y.; Wu, Q.; Tan, M. Towards effective deep transfer via attentive feature alignment. *Neural Netw.* **2021**, *138*, 98–109. [CrossRef] [PubMed]
89. Raghu, M.; Zhang, C.; Kleinberg, J.; Bengio, S. Transfusion: Understanding transfer learning for medical imaging. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 3347–3357.
90. Zamir, A.R.; Sax, A.; Shen, W.; Guibas, L.J.; Malik, J.; Savarese, S. Taskonomy: Disentangling task transfer learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3712–3722.

91. Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
92. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 2180–2188.
93. Locatello, F.; Abbati, G.; Rainforth, T.; Bauer, S.; Schölkopf, B.; Bachem, O. On the fairness of disentangled representations. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 14611–14624.
94. Achille, A.; Soatto, S. Emergence of invariance and disentanglement in deep representations. *J. Mach. Learn. Res.* **2018**, *19*, 1947–1980.
95. Kim, H.; Mnih, A. Disentangling by factorising. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 2649–2658.
96. Liu, C.; Zhu, L.; Belkin, M. Toward a theory of optimization for over-parameterized systems of non-linear equations: The lessons of deep learning. *arXiv* **2020**, arXiv:2003.00307.
97. Liu, C.; Zhu, L.; Belkin, M. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Appl. Comput. Harmon. Anal.* **2022**, *59*, 85–116. [CrossRef]
98. Carlini, N.; Tramer, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.B.; Song, D.; Erlingsson, U.; et al. Extracting Training Data from Large Language Models. In Proceedings of the USENIX Security Symposium, Virtual Event, 11–13 August 2021; Volume 6.
99. Reddy, Y.; Viswanath, P.; Reddy, B.E. Semi-supervised learning: A brief review. *Int. J. Eng. Technol.* **2018**, *7*, 81. [CrossRef]
100. Grira, N.; Crucianu, M.; Boujemaa, N. Unsupervised and semi-supervised clustering: A brief survey. *Rev. Mach. Learn. Tech. Process. Multimed. Content* **2004**, *1*, 9–16.
101. Battaglia, P.W.; Hamrick, J.B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. Relational inductive biases, deep learning, and graph networks. *arXiv* **2018**, arXiv:1806.01261.
102. Goyal, A.; Bengio, Y. Inductive biases for deep learning of higher-level cognition. *Proc. R. Soc. A* **2022**, *478*, 20210068. [CrossRef]
103. Zhang, Q.s.; Zhu, S.C. Visual interpretability for deep learning: A survey. *Front. Inf. Technol. Electron. Eng.* **2018**, *19*, 27–39. [CrossRef]
104. Li, X.; Xiong, H.; Li, X.; Wu, X.; Zhang, X.; Liu, J.; Bian, J.; Dou, D. Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond. *Knowl. Inf. Syst.* **2022**, *64*, 3197–3234. [CrossRef]
105. Samek, W.; Montavon, G.; Vedaldi, A.; Hansen, L.K.; Müller, K.R. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*; Springer Nature: Cham, Switzerland, 2019; Volume 11700.
106. Xu, F.; Uszkoreit, H.; Du, Y.; Fan, W.; Zhao, D.; Zhu, J. Explainable AI: A brief survey on history, research areas, approaches and challenges. In Proceedings of the Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, 9–14 October 2019; Proceedings, Part II 8; Springer: Berlin/Heidelberg, Germany, 2019; pp. 563–574.
107. Wasserman, L. Topological data analysis. *Annu. Rev. Stat. Its Appl.* **2018**, *5*, 501–532. [CrossRef]
108. Bubenik, P. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.* **2015**, *16*, 77–102.
109. Chazal, F.; Michel, B. An introduction to topological data analysis: Fundamental and practical aspects for data scientists. *Front. Artif. Intell.* **2021**, *4*, 667963. [CrossRef]
110. Edelsbrunner, H.; Harer, J. Persistent homology—A survey. *Contemp. Math.* **2008**, *453*, 257–282.
111. Singh, G.; Mémoli, F.; Carlsson, G.E. Topological methods for the analysis of high dimensional data sets and 3d object recognition. *PBG@ Eurograph.* **2007**, *2*, 091–100.
112. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
113. Choi, S.R.; Lee, M. Estimating the Prognosis of Low-Grade Glioma with Gene Attention Using Multi-Omics and Multi-Modal Schemes. *Biology* **2022**, *11*, 1462. [CrossRef] [PubMed]
114. Chen, S.; Jin, Q.; Zhao, J.; Wang, S. Multimodal multi-task learning for dimensional and continuous emotion recognition. In Proceedings of the 7th Annual Workshop on Audio/Visual Emotion Challenge, Mountain View, CA, USA, 23–27 October 2017; pp. 19–26.
115. Sawhney, R.; Mathur, P.; Mangal, A.; Khanna, P.; Shah, R.R.; Zimmermann, R. Multimodal multi-task financial risk forecasting. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 456–465.