On Linear Identifiability of Learned Representations

Geoffrey Roeder[1] **Luke Metz**[2] **Diederik P. Kingma**[2]

Abstract

Identifiability is a desirable property of a statistical model: it implies that the true model parameters may be estimated to any desired precision, given sufficient computational resources and data. We study identifiability in the context of representation learning: discovering nonlinear data representations that are optimal with respect to some downstream task. When parameterized as deep neural networks, such representation functions lack identifiability in parameter space, because they are overparameterized by design. In this paper, building on recent advances in nonlinear Independent Components Analysis, we aim to rehabilitate identifiability by showing that a large family of discriminative models are in fact identifiable in *function* space, up to a linear indeterminacy. Many models for representation learning in a wide variety of domains have been identifiable in this sense, including text, images and audio, state-of-the-art at time of publication. We derive sufficient conditions for linear identifiability and provide empirical support for the result on both simulated and real-world data.

1. Introduction

An increasingly common methodology in machine learning is to improve performance on a primary down-stream task by first learning a high-dimensional representation of the data on a related, proxy task. In this paradigm, training a model reduces to fine-tuning the learned representations for optimal performance on a particular sub-task (Erhan et al., 2010). Deep neural networks (DNNs), as flexible function approximators, have been surprisingly successful in discovering effective high-dimensional representations for use in downstream tasks such as image classification (Sharif Razavian et al., 2014), text generation (Radford et al., 2018; Devlin et al., 2018), and sequential decision making (Oord et al., 2018).

When learning representations for downstream tasks, it would be useful if the representations were reproducible, in the sense that every time a network relearns the representation function on the same data distribution, they were approximately the same, regardless of small deviations in the initialization of the parameters or the optimization procedure. In some applications, such as learning real-world causal relationships from data, such reproducible learned representations are crucial for accurate and robust inference (Johansson et al., 2016; Louizos et al., 2017). A rigorous way to achieve reproducibility is to choose a model whose representation function is *identifiable* in function space. Informally speaking, identifiability in function space is achieved when, in the limit of infinite data, there exists a single, global optimum in function space. Interestingly, Figure 1 exhibits learned representation functions that appear to be the same up to a linear transformation, even on finite data and optimized without convergence guarantees (see Appendix A.1 for training details).

In this paper, we account for Figure 1 by making precise the relationship it exemplifies. We prove that a large class of discriminative and autoregressive models are identifiable in function space, up to a linear transformation. Our results extend recent advances in the theory of nonlinear Independent Components Analysis (ICA), which have recently provided strong identifiability results for generative models of data (Hyvärinen et al., 2018; Khemakhem et al., 2019; 2020; Sorrenson et al., 2020). Our key contribution is to bridge the gap between these results and *discriminative* models, commonly used for representation learning (e.g., (Hénaff et al., 2019; Brown et al., 2020)).

The rest of the paper is organized as follows. In Section 2, we describe a general discriminative model family, defined by its canonical mathematical form, which generalizes many supervised, self-supervised, and contrastive learning frameworks. In Section 3, we prove that learned representations in this family have an asymptotic property desirable for representation learning: equality up to a linear transformation. In Section 4, we show that this family includes a number of highly performant models, state-of-the-art at publication for their problem domains, including CPC (Oord et al., 2018),

[1]Princeton University [2]Google Brain. Correspondence to: Geoffrey Roeder <roeder@princeton.edu>, Diederik P. Kingma <durk@google.com>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

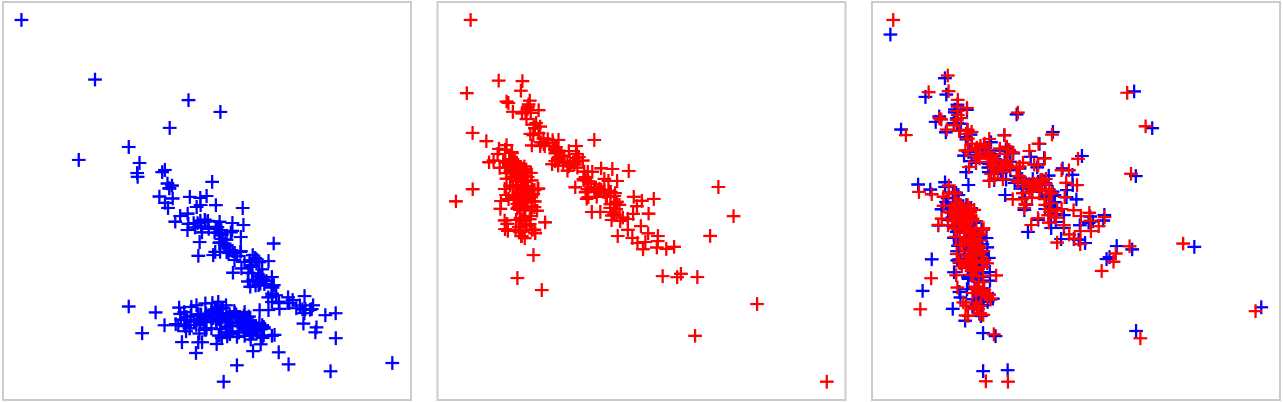

Figure 1. Left and Middle: Two learned DNN representation functions $\mathbf{f}_{\boldsymbol{\theta}_1}(\mathcal{B})$, $\mathbf{f}_{\boldsymbol{\theta}_2}(\mathcal{B})$ visualized on held-out data $\mathcal{B}$. The DNNs are word embedding models (Mnih & Teh, 2012) trained on the Billion Word Dataset (Chelba et al., 2013) (see Appendix A.1 for code release and training details). Right: $\boldsymbol{A}\mathbf{f}_{\boldsymbol{\theta}_1}(\mathcal{B})$ and $\mathbf{f}_{\boldsymbol{\theta}_2}(\mathcal{B})$, where $\boldsymbol{A}$ is a linear transformation learned after training. The overlap exhibits *linear identifiability* (see Section 3): different representation functions, learned on the same data distribution, live within linear transformations of each other in function space.

BERT (Devlin et al., 2018), and GPT-2 and GPT-3 (Radford et al., 2018; 2019; Brown et al., 2020). Section 5.2 investigates the actually realizable regime of *finite* data and *partial* optimization, showing that representations learned by members of the identifiable model family approach equality up to a linear transformation as a function of dataset size, neural network capacity, and optimization progress.

2. Model Family and Data Distribution

The learned embeddings of a DNN are a function not only of the parameters, but also the network architecture and size of dataset (viewed as a sample from the underlying data distribution). This renders any analysis in full generality challenging. To make such an analysis tractable, in this section, we begin by specifying a set of assumptions about the underlying data distribution and model family that must hold for the learned representations to be similar up to a linear transformation. These assumptions are, in fact, satisfied by a number of already published, highly performant models. We establish assumptions and definitions in this section, and exhibit models that satisfy them in depth in Section 4.

Data Distribution We assume the existence of a generalized dataset in the form of an empirical distribution $p_{\mathcal{D}}(\mathbf{x}, \mathbf{y}, \mathbf{S})$ over random variables $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{S}$ with the following properties:

- The random variable $\mathbf{x}$ is an input variable, typically high-dimensional, such as text or an image.
- The random variable $\mathbf{y}$ is the target variable whose value the model predicts. In case of object classification, $\mathbf{y}$ is a semantically meaningful class label. However, in our model family, $\mathbf{y}$ may also be a high-dimensional context variable, such a text, image, or sentence fragment.
- $\mathbf{S}$ is a set containing the possible values of $\mathbf{y}$ given $\mathbf{x}$, so $p_{\mathcal{D}}(\mathbf{y}|\mathbf{x}, \mathbf{S}) > 0 \iff \mathbf{y} \in \mathbf{S}$.

Note that the set of labels $\mathbf{S}$ is not fixed, but a random variable. This allows supervised, contrastive, and self-supervised learning frameworks to be analyzed together: the meaning of $\mathbf{S}$ encodes the task. For supervised classification, $\mathbf{S}$ is deterministic and contains class labels. For self-supervised pretraining, $\mathbf{S}$ contains randomly-sampled high-dimensional variables such as image embeddings. For deep metric learning (Hoffer & Ailon, 2015; Sohn, 2016), the set $\mathbf{S}$ contains one positive and k negative samples of the class to which $\mathbf{x}$ belongs.

Canonical Discriminative Form Given a data distribution as above, a generalized discriminative model family may be defined by its parameterization of the probability of a target variable $\mathbf{y}$ conditioned on an observed variable $\mathbf{x}$ and a set $\mathbf{S}$ that contains not only the true target label $\mathbf{y}$, but also a collection of distractors $\mathbf{y}'$:

$$p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{S}) = \frac{\exp(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})^\top \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}))}{\sum_{\mathbf{y}' \in \mathbf{S}} \exp(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})^\top \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}'))}, \tag{1}$$

The codomain of the functions $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$ and $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y})$ is $\mathbb{R}^M$, and the domains vary according to modelling task. For notational convenience both are parameterized by $\boldsymbol{\theta} \in \Theta$, but $\mathbf{f}$ and $\mathbf{g}$ may use disjoint parts of $\boldsymbol{\theta}$, meaning that they do not necessarily share parameters.

With $\mathcal{F}$ and $\mathcal{G}$ we denote the function spaces of $\mathbf{f}_{\boldsymbol{\theta}}$ and $\mathbf{g}_{\boldsymbol{\theta}}$ respectively. Our primary domain of interest is when $\mathbf{f}_{\boldsymbol{\theta}}$ and $\mathbf{g}_{\boldsymbol{\theta}}$ are highly flexible function approximators, such as

DNNs. This brings certain analytical challenges. In neural networks, different choices of parameters $\boldsymbol{\theta}$ can result in the same functions $\mathbf{f}_{\boldsymbol{\theta}}$ and $\mathbf{g}_{\boldsymbol{\theta}}$, hence the map $\Theta \to \mathcal{F} \times \mathcal{G}$ is many-to-one. In the context of representation learning, the function $\mathbf{f}_{\boldsymbol{\theta}}$ is typically viewed as a nonlinear feature extractor, e.g., the learned representation of the input data. While other choices meet the membership conditions for the family defined by the canonical form of Equation (1), in the remainder, we will focus on DNNs. We next present a definition of identifiability suitable for DNNs, and prove that members of the above family satisfy it, under additional mild assumptions.

3. Model Identifiability

In this section, we derive identifiability conditions for models in the family defined in Section 2.

3.1. Identifiability in Parameter Space

Identifiability analysis answers the question of whether it is theoretically possible to learn the parameters of a statistical model exactly. Specifically, given some estimator $\boldsymbol{\theta}'$ for model parameters $\boldsymbol{\theta}^*$, identifiability is the property that, for any $\{\boldsymbol{\theta}', \boldsymbol{\theta}^*\} \subset \Theta$,

$$p_{\boldsymbol{\theta}'} = p_{\boldsymbol{\theta}^*} \quad \Longrightarrow \quad \boldsymbol{\theta}' = \boldsymbol{\theta}^*. \tag{2}$$

Models that do not have this property are said to be non-identifiable. This happens when different values $\{\boldsymbol{\theta}', \boldsymbol{\theta}^*\} \subset \Theta$ can give rise to the same model distribution $p_{\boldsymbol{\theta}'}(\mathbf{y}|\mathbf{x}, \mathbf{S}) = p_{\boldsymbol{\theta}^*}(\mathbf{y}|\mathbf{x}, \mathbf{S})$. In such a case, observing an empirical distribution $p_{\boldsymbol{\theta}^*}(\mathbf{y}|\mathbf{x}, \mathbf{S})$, and fitting a model $p_{\boldsymbol{\theta}'}(\mathbf{y}|\mathbf{x}, \mathbf{S})$ to it perfectly does not guarantee that $\boldsymbol{\theta}' = \boldsymbol{\theta}^*$.

Neural networks exhibit various symmetries in parameter space such that there is almost always a many-to-one correspondence between a choice of $\boldsymbol{\theta}$ and resulting probability function $p_{\boldsymbol{\theta}}$. A simple example in neural networks is that one can swap the (incoming and outgoing) connections of two neurons in a hidden layer. This changes the value of the parameters, but does not change the network's function. Thus, when representation functions $\mathbf{f}_{\boldsymbol{\theta}}$ or $\mathbf{g}_{\boldsymbol{\theta}}$ are parameterized as DNNs, Equation (2) is not satisfiable.

3.2. Identifiability in Function Space

For reliable and efficient representation learning, we want learned representations $\mathbf{f}_{\boldsymbol{\theta}}$ from two identifiable models to be *sufficiently* similar for interchangeable use in downstream tasks. The most general property we wish to preserve among learned representations is their ability to discriminate among statistical patterns corresponding to categorical groupings. In the model family defined in Section 2, the data and context functions $\mathbf{f}_{\boldsymbol{\theta}}$ and $\mathbf{g}_{\boldsymbol{\theta}}$ parameterize $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{S})$, the probability of label assignment, through a normalized inner product. This induces a hyperplane boundary, for discrimination, in a joint space of learned representations for data $\mathbf{x}$ and context $\mathbf{y}$. Therefore, in the following, we will derive identifiability conditions *up to a linear transformation*, using a notion of similarity in parameter space inspired by Hyvärinen et al. (2018).

Definition 1. *Let $\overset{L}{\sim}$ be a pairwise relation on Θ defined as:*

$$\boldsymbol{\theta}' \overset{L}{\sim} \boldsymbol{\theta}^* \iff \begin{array}{c} \mathbf{f}_{\boldsymbol{\theta}'}(\mathbf{x}) = \boldsymbol{A}\mathbf{f}_{\boldsymbol{\theta}^*}(\mathbf{x}) \\ \mathbf{g}_{\boldsymbol{\theta}'}(\mathbf{y}) = \boldsymbol{B}\mathbf{g}_{\boldsymbol{\theta}^*}(\mathbf{y}) \end{array} \tag{3}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are invertible $M \times M$ matrices.

See Appendix B for proof that $\overset{L}{\sim}$ is an equivalence relation. In the remainder, we refer to identifiability up to the equivalence relation $\overset{L}{\sim}$ as *$\overset{L}{\sim}$-identifiable* or *linearly identifiable*.

3.3. Derivation of Identifiability Conditions

We next present a simple proof of the $\overset{L}{\sim}$-identifiability of members of the generalized discriminative family defined in Section 2. This result reveals sufficient conditions under which a discriminative probabilistic model $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{S})$ has a useful property: the learned representations of the input $\mathbf{x}$ and target random variables $\mathbf{y}$ for any two pairs of parameters $(\boldsymbol{\theta}', \boldsymbol{\theta}^*)$ are related as $\boldsymbol{\theta}' \overset{L}{\sim} \boldsymbol{\theta}^*$, that is, $\mathbf{f}_{\boldsymbol{\theta}'}(\mathbf{x}) = \boldsymbol{A}\mathbf{f}_{\boldsymbol{\theta}^*}(\mathbf{x})$ and $\mathbf{g}_{\boldsymbol{\theta}'}(\mathbf{y}) = \boldsymbol{B}\mathbf{g}_{\boldsymbol{\theta}^*}(\mathbf{y})$.

First, we review the notation for the proof, which is introduced in detail in Section 2. We then highlight an important requirement on the diversity of the data distribution, which must be satisfied for the proof statement to hold. We prove the result immediately after.

Notation. The target random variables $\mathbf{y}$, associated with input random variables $\mathbf{x}$, may be class labels (as in supervised classification), or they could be stochastically generated from datapoints $\mathbf{x}$ as, e.g., perturbed image patches (as in self-supervised learning). We account for this additional stochasticity as a set-valued random variable $\mathbf{S}$, containing all possible values of $\mathbf{y}$ conditioned on some $\mathbf{x}$. For brevity, we will use shorthands that drop the parameters $\boldsymbol{\theta}$: $p' := p_{\boldsymbol{\theta}'}, p^* := p_{\boldsymbol{\theta}^*}, \mathbf{f}^* := \mathbf{f}_{\boldsymbol{\theta}^*}, \mathbf{f}' := \mathbf{f}_{\boldsymbol{\theta}'}, \mathbf{g}' := \mathbf{g}_{\boldsymbol{\theta}'}$.

Diversity condition. We assume that for any $(\boldsymbol{\theta}', \boldsymbol{\theta}^*)$ for which it holds that $p' = p^*$, and for any given $\mathbf{x}$, by repeated sampling $\mathbf{S} \sim p_{\mathcal{D}}(\mathbf{S}|\mathbf{x})$ and picking two points $\mathbf{y}_A, \mathbf{y}_B \in \mathbf{S}$, we can construct a set of M distinct tuples $\{(\mathbf{y}_A^{(i)}, \mathbf{y}_B^{(i)})\}_{i=1}^M$ such that the matrices $\mathbf{L}'$ and $\mathbf{L}^*$ are invertible, where $\mathbf{L}'$ consists of columns $(\mathbf{g}'(\mathbf{y}_A^{(i)}) - \mathbf{g}'(\mathbf{y}_B^{(i)}))$, and $\mathbf{L}^*$ consists of columns $\mathbf{g}^*(\mathbf{y}_A^{(i)}) - \mathbf{g}^*(\mathbf{y}_B^{(i)})$, $i \in \{1, \ldots, M\}$. See Section 3.4 for detailed discussion.

Theorem 1. Under the diversity condition, models in the family defined by Equation (1) are linearly identifiable. That is, for any $\boldsymbol{\theta}', \boldsymbol{\theta}^* \in \Theta$, and $\mathbf{f}^*, \mathbf{f}', \mathbf{g}^*, \mathbf{g}', p^*, p'$ defined as in Section 2,

$$p' = p^* \implies \boldsymbol{\theta}' \overset{\mathrm{L}}{\sim} \boldsymbol{\theta}^*.$$

Proof. We proceed by directly constructing an invertible linear transformation that satisfies Definition 1. Consider $\mathbf{y}_A, \mathbf{y}_B \in \mathbf{S}$. The likelihood ratios for these points

$$\frac{p'(\mathbf{y}_A|\mathbf{x}, \mathbf{S})}{p'(\mathbf{y}_B|\mathbf{x}, \mathbf{S})} = \frac{p^*(\mathbf{y}_A|\mathbf{x}, \mathbf{S})}{p^*(\mathbf{y}_B|\mathbf{x}, \mathbf{S})} \tag{4}$$

are equal. Substituting the model definition from Equation (1), we find:

$$\frac{\exp(\mathbf{f}'(\mathbf{x})^\top \mathbf{g}'(\mathbf{y}_A))}{\exp(\mathbf{f}'(\mathbf{x})^\top \mathbf{g}'(\mathbf{y}_B))} = \frac{\exp(\mathbf{f}^*(\mathbf{x})^\top \mathbf{g}^*(\mathbf{y}_A))}{\exp(\mathbf{f}^*(\mathbf{x})^\top \mathbf{g}^*(\mathbf{y}_B))}, \tag{5}$$

where the normalizing constants have cancelled out on both the left- and right-hand sides. Evaluating the logarithm of both sides and simplifying yields

$$\begin{aligned} &(\mathbf{g}'(\mathbf{y}_A) - \mathbf{g}'(\mathbf{y}_B))^\top \mathbf{f}'(\mathbf{x}) \\ =&(\mathbf{g}^*(\mathbf{y}_A) - \mathbf{g}^*(\mathbf{y}_B))^\top \mathbf{f}^*(\mathbf{x}). \end{aligned} \tag{6}$$

Note that this equation is true for any triple $(\mathbf{x}, \mathbf{y}_A, \mathbf{y}_B)$ for which $p_{\mathcal{D}}(\mathbf{x}, \mathbf{y}_A, \mathbf{y}_B) > 0$.

We next collect M distinct tuples $(\mathbf{y}_A^{(i)}, \mathbf{y}_B^{(i)})$. Using identity (6), we now construct a system of M linear equations that relates $\mathbf{f}'$ and $\mathbf{f}^*$.

Let $\mathbf{L}'$ be the $(M \times M)$-dimensional matrix whose i-th column is the difference vector $(\mathbf{g}'(\mathbf{y}_A^{(i)}) - \mathbf{g}'(\mathbf{y}_B^{(i)}))$. Similarly, let $\mathbf{L}^*$ be the $(M \times M)$-dimensional matrix whose i-th column is $(\mathbf{g}^*(\mathbf{y}_A^{(i)}) - \mathbf{g}^*(\mathbf{y}_B^{(i)}))$. Then, the system of M linear equations is

$$\mathbf{L}'^\top \mathbf{f}'(\mathbf{x}) = \mathbf{L}^{*\top} \mathbf{f}^*(\mathbf{x}).$$

By the diversity condition, $\mathbf{L}'$ is invertible. We left-multiply by $\mathbf{L}'^{-\top}$, yielding

$$\mathbf{f}'(\mathbf{x}) = (\mathbf{L}^* \mathbf{L}'^{-1})^\top \, \mathbf{f}^*(\mathbf{x}). \tag{7}$$

Hence, $\mathbf{f}'(\mathbf{x}) = \mathbf{A}\mathbf{f}^*(\mathbf{x})$ for $\mathbf{A} = (\mathbf{L}^* \mathbf{L}'^{-1})$. Because $\mathbf{L}^*$ is also invertible, so is $\mathbf{A}$. This completes the proof that $p' = p^* \implies \mathbf{f}_{\boldsymbol{\theta}'}(\mathbf{x}) = \boldsymbol{A}\mathbf{f}_{\boldsymbol{\theta}^*}(\mathbf{x})$ for invertible $\mathbf{A}$. See Appendix C for the remainder, which proves the corresponding result for $\mathbf{g}$, and completes the proof of Theorem 1.

3.4. When Does the Diversity Condition Hold?

The diversity condition guarantees the existence of the matrix $\mathbf{A}$ in Equation (7) by ensuring that the matrices $\mathbf{L}'$ and $\mathbf{L}^*$ are non-singular. Informally, this requires that the set of possible values of $\mathbf{y}$ given $\mathbf{x}$ must be big enough–the size of the set $\mathbf{S}$ is greater than some number–and the function $\mathbf{g}_{\boldsymbol{\theta}}$ has enough unique points in its range to ensure that there exist M difference vectors that span its range.[1]

For example, consider a supervised learning model with K classes. The random variable $\mathbf{S}$ is clamped to the possible labels for an image $\mathbf{x}$, and is of size K. In order for the diversity condition to hold, the number of classes $K \geq M + 1$ so that there can exist M difference vectors $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}^{(1)}) - \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}^{(j)})$, $j = 2, \ldots, M + 1$. In case of self-supervised or deep metric learning, where $\mathbf{S}$ and $\mathbf{y}$ may be randomly generated from $\mathbf{x}$, this requirement is easy to satisfy. The same is true for language models with large vocabularies. However, for supervised classification with a small number of classes, this requirement on the size of $\mathbf{S}$ may be restrictive, as we discuss further in Section 4. We stress here that our goal is to study *representation* learning, rather than supervised classification, so that the fact our result applies to supervised learning at all is an interesting curiosity.

Along with requiring number of classes $|\mathbf{S}| = K \geq M + 1$, we implicitly assumed that the context representation function $\mathbf{g}_{\boldsymbol{\theta}}$ has the following property: there exist M difference vectors in the range of $\mathbf{g}_{\boldsymbol{\theta}}$ (of the form in Equation (6) that span it. This is a mild assumption in the context of DNNs: for random initialization and iterative weight updates, this property follows from the stochasticity of the distribution used to initialize the network. Briefly, a set of $M + 1$ unique points $\mathbf{y}^{(j)}$ such that the M vectors $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}^{(1)}) - \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}^{(j)}), j = 2, \ldots, M + 1$ are not linearly independent has measure zero. For other choices of $\mathbf{g}_{\boldsymbol{\theta}}$, care must be taken to ensure this condition is satisfied.

What can be said when $\mathbf{L}'$ and $\mathbf{L}^*$ are ill-conditioned, that is, the ratio between maximum and minimum singular value $\frac{\sigma_{\max}(\mathbf{L})}{\sigma_{\min}(\mathbf{L})}$ (dropping superscripts when a statement apply to both) is large? In the context of a data representation matrix such as $\mathbf{L}$, this implies that there exists at least one column $\boldsymbol{\ell}_j$ of $\mathbf{L}$ and constants λ_k for $k \neq j$ such that $\|\boldsymbol{\ell}_j - \sum_{k \neq j} \lambda_k \boldsymbol{\ell}_k\|_2 < \varepsilon$ for small ε. In other words, some column is nearly a linear combination of the others. This implies, in turn, that there exists some tuple $(\mathbf{y}^{(k)}, \mathbf{y}^{(i)})$ such that the resulting difference vector $\boldsymbol{\ell}_j = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}_A^{(k)}) - \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}_B^{(i)})$ can nearly (in the sense above) be written as a linear combination of the other columns. Such near singularity is in this case is caused by the choice of samples $\mathbf{y}$ that yield the difference vectors. The issue could be handled by resampling different data points until the condition number of the matrices is satisfactory. This amounts to strengthening

[1] We note here that a second, weaker diversity condition is also required on the data distribution and model with respect to $\mathbf{x}$ and $\mathbf{f}$. This is discussed in Appendix C.

the diversity condition. We leave more detailed analysis to future work, as the result will depend on the choice of $\mathbf{f}$ and $\mathbf{g}$.

4. Examples of Linearly Identifiable Models

The form of Equation (1) is already used as a general approach for a variety of machine learning problems. We present a non-exhaustive sample of such publications, chosen to exhibit the range of applications. Many of these approaches were state-of-the-art at the time of their release: Contrastive Predictive Coding (Hénaff et al., 2019), BERT (Devlin et al., 2018), GPT-2 and GPT-3 (Radford et al., 2018; 2019; Brown et al., 2020), XLNET (Yang et al., 2019), and the triplet loss for deep metric learning (Sohn, 2016). In this section, we discuss how to interpret the functional components of these frameworks with respect to the generalized data distribution of Section 2 and canonical parameterization of Equation (1). See Appendix D for reductions to the canonical form of Equation (1).

Supervised Classification. Although the scope of this paper is identifiable *representation* learning, under certain conditions, standard supervised classifiers can learn identifiable representations as well. In this case, the number of classes must be strictly greater than the feature dimension, as noted in Section 3.4. We simulate such a model in Section 5.1 to show evidence of its linear identifiability. We stress that representation learning as *pretraining* for classification is a way to ensure that the conditions on label diversity are met, rather than relying on the supervised classifier itself to generate identifiable representations. This paradigm is discussed in the next subsection.

Representations learned during supervised classification can be linearly identifiable under the following model specification. The input random variables $\mathbf{x}$ represent some data domain to be classified, such as images or word embeddings. The target variables $\mathbf{y}$ represent label assignments for $\mathbf{x}$, typically semantically meaningful. These are often encoded these as the standard basis vectors $\mathbf{e}_{\mathbf{y}}$, a "one-hot encoding." The set $\mathbf{S}$ contains all K possible values of $\mathbf{y}$. In this case, notice that $\mathbf{S}$ is not stochastic: the empirical distribution $p_{\mathcal{D}}(\mathbf{S}|\mathbf{x})$ is modelled as a Dirac measure with all probability mass on the set $\mathbf{S} = \{0, \ldots, K-1\}$ (using integers, here, to represent distinct labels) . The representation function $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$ of a classifier is often implemented as DNN that maps from the input layer to the layer just prior to the model logits. The context map $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y})$ is given by the weights in the final, linear projection layer, which outputs unnormalized logits. Concretely, $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}) = \mathbf{W}\mathbf{e}_{\mathbf{y}}$, where $\mathbf{W} \in \mathbb{R}^{M\times M}$ is a learnable weight matrix. In order satisfy the diversity condition, the dimension M of the number of classes K must be strictly greater than the dimension of the learned representation M, that is, $|\mathbf{S}| \geq M+1$. Finally, the output of the final, linear projection layer is normalized through a Softmax function, yielding the parameterization of Equation (1).

Self-Supervised Pretraining for Image Classification. Self-supervised learning is a framework that first pretrains a DNN before deploying it on some other, related task. The pretraining task often takes the form of Equation (1) and meets the sufficient conditions to be linearly identifiable. A paradigmatic example is Contrastive Predictive Coding (CPC) (Oord et al., 2018). CPC is a general pretraining framework, but we focus for the sake of clarity on its use in image models here. CPC as applied to images involves: (1) preprocessing an image into augmented patches, (2) assigning labels according to which image the patch came from, and then (3) predicting the representations of the patches whether below, to the right, to the left, or above a certain level (Oord et al., 2018).

The context function of CPC, $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y})$, encodes a particular position in the sequence of patches, and the representation function, $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$, is an autoregressive function of the previous k patches, according to some predefined patch ordering. Given some $\mathbf{x}$, the collection of all patches from the sequence, from a given minibatch of images, is the set $\mathbf{S} \sim p_{\mathcal{D}}(\mathbf{S}|\mathbf{x})$, where the randomness enters via the patch preprocessing algorithm. Since the preprocessing phase is part of the algorithm design, it is straightforward to make it sufficiently diverse (enough transformations of enough patches) so as to meet the requirements for the model to be linearly identifiable.

Multi-task Pretraining for Natural Language Generation. Autoregressive language models, such as (Mikolov et al., 2010; Dai & Le, 2015) and more recently GPT-2 and GPT-3 (Radford et al., 2018; 2019; Brown et al., 2020), are typically also instances of the model family of Equation (1). Data points $\mathbf{x}$ are the past tokens, $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$ is a nonlinear representation of the past estimated by either an LSTM (Hochreiter & Schmidhuber, 1997) or an autoregressive Transformer model (Vaswani et al., 2017), $\mathbf{y}$ is the next token, and $\mathbf{w}_i = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y} = i)$ is a learned representation of the next token, often implemented as a simple look-up table, as in supervised classification.

BERT (Devlin et al., 2018) is also a member of the linearly identifiable family. This model pretrains word embeddings through a denoising autoencoder-like (Vincent et al., 2008) architecture. For a given sequence of tokenized text, some fixed percentage of the symbols are extracted and set aside, and their original values set to a special null symbol, "corrupting" the original sequence. The pretraining task in BERT is to learn a continuous representation of the extracted symbols conditioned on the remainder of the text. A transformer (Vaswani et al., 2017) function approximator is used to map from the corrupted sequence into a continuous space. The transformer network is the $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$ function of Equation (1). The context map $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y})$ is a lookup map into the learned basis vector for each token.

5. Experiments

The derivation in Section 3 shows that, for models in the general discriminative family defined in Section 2, the functions $\mathbf{f}_{\boldsymbol{\theta}}$ and $\mathbf{g}_{\boldsymbol{\theta}}$ are identifiable up to a linear transformation given unbounded data and assuming model convergence. The question remains as to how close a model trained on finite data and without convergence guarantees will approach this limit. One subtle issue is that poor architecture choices (such as too few hidden units, or inadequate inductive priors) or insufficient data samples when training can interfere with model estimation and thereby linear identifiability of the learned representations, due to underfitting. In this section, we study this issue over a range of models, from low-dimensional language embedding and supervised classification (Figures 1 and 2 respectively) to GPT-2 (Radford et al., 2019), an approximately $1.5 * 10^9$-parameter generative model of natural language (Figure 4). See Appendix A and the code release for details needed to reproduce.

Through these experiments, we show that (1) in the small dimensional, large data regime, linearly identifiable models yield learned representations that lie approximately within a linear transformation of each other (Figures 1 and 2) as predicted by Theorem 1; and (2) in the high dimensional, large data regime, linearly identifiable models yield learned representations that exhibit a strong trend towards linear identifiability. The learned representations approach a linear transformation of each other monotonically, as a function of dataset sample size, neural network capacity (number of hidden units), and optimization progress. In the case of GPT-2, which has benefited from substantial tuning by engineers to improve model estimation, we find strong evidence of linear identifiability.

Note on methodology: measuring linear similarity between learned representations. How can we measure whether pairs of learned representations live within a linear transformation of each other in function space? We adapt Canonical Correlation Analysis (CCA) (Hotelling, 1936) for this purpose, which finds the optimal linear transformations to maximize correlation among two random vectors. On a randomly selected held-out subset $\mathcal{B} \subset \mathcal{D}$ of the training data we compute $\mathbf{f}_{\boldsymbol{\theta}_1}(\mathcal{B})$ and $\mathbf{f}_{\boldsymbol{\theta}_2}(\mathcal{B})$ for two models with parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ respectively. Assume without loss of generality that $\mathbf{f}_{\boldsymbol{\theta}_1}(\mathcal{B})$ and $\mathbf{f}_{\boldsymbol{\theta}_2}(\mathcal{B})$ are centered. CCA finds the optimal linear transformations $\boldsymbol{C}$ and $\boldsymbol{D}$ such that the pairwise correlations ρ_i between the i^{th} columns of $\boldsymbol{C}^\top \mathbf{f}_{\boldsymbol{\theta}_1}(\mathcal{B})$ and $\boldsymbol{D}^\top \mathbf{f}_{\boldsymbol{\theta}_2}(\mathcal{B})$ are maximized. We collect correlations together in $\boldsymbol{\rho}$. If after linear transformation the two matrices are aligned, the mean of $\boldsymbol{\rho}$ will be 1; if they are instead uncorrelated, then the mean of $\boldsymbol{\rho}$ will be 0. We use the mean of $\boldsymbol{\rho}$ as a proxy for the existence of a linear transformation between $\mathbf{f}_{\boldsymbol{\theta}_1}(\mathcal{B})$ and $\mathbf{f}_{\boldsymbol{\theta}_2}(\mathcal{B})$. For DNNs, it is a well known phenomenon that most of the variability in a learned representation tends to concentrate in a low-dimensional subspace, leaving many noisy, random dimensions (Morcos et al., 2018). Such random noise can result in spurious high correlations in CCA. A solution to this problem is to apply Principal Components Analysis (PCA) (Pearson, 1901) to each of the two matrices $\mathbf{f}_{\boldsymbol{\theta}_2}(\mathcal{B})$ and $\mathbf{f}_{\boldsymbol{\theta}_1}(\mathcal{B})$, projecting onto their top-k principal components, before applying CCA. This technique is known as SVCCA (Raghu et al., 2017).

5.1. Simulation Study: Classification by DNNs

We report first on a simulation study of linearly identifiable K-way classification, where all assumptions and sufficient conditions of Theorem 1 are guaranteed to be met. We generated a synthetic data distribution with the properties required by Section 2, and chose DNNs that had sufficient capacity to learn a specified nonlinear relationship between inputs $\mathbf{x}$ and targets $\mathbf{y}$. In short, the data distribution $p_{\mathcal{D}}(\mathbf{x}, \mathbf{y}, \mathbf{S})$ consists of inputs $\mathbf{x}$ sampled from a 2-D Gaussian with $\sigma = 3$. The targets $\mathbf{y}$ were assigned among $K = 18$ classes according to their radial position (angle swept out by a ray fixed at the origin). The number of classes K was chosen to ensure $K \geq \dim[\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})] + 1$, the diversity condition. See Appendix D.1 for more details.

To evaluate linear similarity, we trained two randomly initialized models of $p_{\mathcal{D}}(\mathbf{y}|\mathbf{x}, \mathbf{S})$. Plots show $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$, the data representation function, on random $\mathbf{x}$. Figure 2b shows that the mean CCA increases to its maximum value over training, demonstrating that the feature spaces converge to the same solution up to a linear transformation modulo model estimation noise. Similarly, Figure 2c shows that the learned representations exhibit a strongly linear relationship.

5.2. Self-Supervised Learning for Image Classification

We next investigate high-dimensional, self-supervised representation learning on CIFAR-10 (Krizhevsky et al., 2009) using CPC (Oord et al., 2018; Hénaff et al., 2019). For a given input image, this model predicts the identity of a bottom image patch representation given a top patch representation (Figure 3a.) Here, $\mathbf{S}$ comprises the true patch with a set of distractor patches from across the current minibatch. For each model we define both $\mathbf{f}_{\boldsymbol{\theta}'}$ and $\mathbf{g}_{\boldsymbol{\theta}'}$ as a 3-layer MLP with 256 units per layer (except where noted otherwise) and fix output dimensionality of 64.

In Figure 3b, CCA coefficients are plotted over the course of training. As training progresses, alignment between the learned representations increases. In Figure 3c, we artifi-

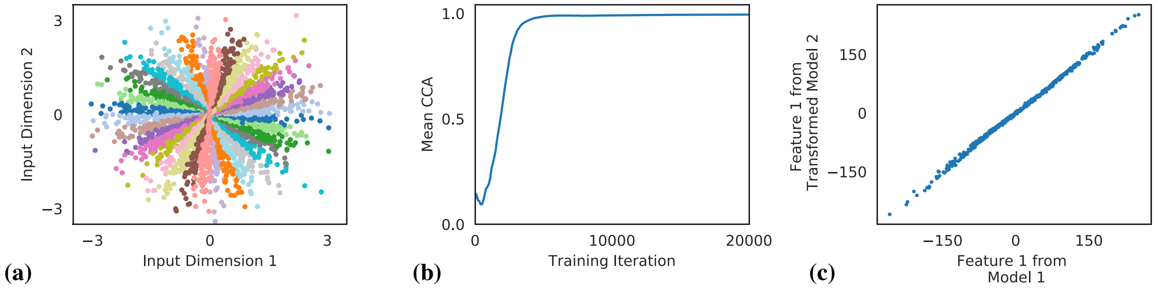

Figure 2. **Supervised Classification**. **(a)** Data distribution for a linearly identifiable K-way classification problem. **(b)** Mean (centered) CCA between the learned representations over the course of training. After approx. 4000 iterations, CCA finds a linear transformation that rotate the learned representations into alignment, up to optimization error. **(c)** Learned representations after transformation via optimal linear transformation. The first dimension of the first model's feature space is plotted against the first dimension of second. The learned representations have a nearly linear relationship, modulo estimation noise.

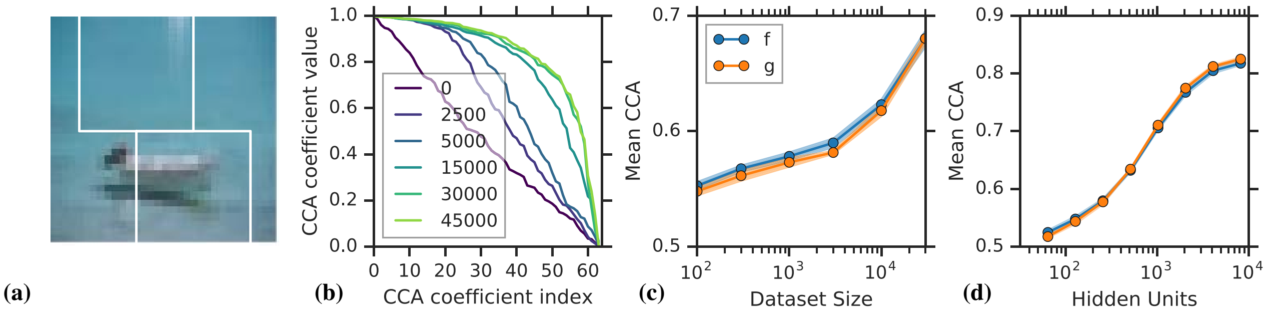

Figure 3. **Self-Supervised Representation Learning.** Error bars are computed over 5 pairs of models. **(a)** Input data. Two patches are taken (one from top half, and one from the bottom half) of an image at random. Using a contrastive loss, we predict the identity of the bottom patch encoding from the top. **(b)** Linear similarity of learned representations at checkpoints (see legend). As models converge, linear similarity increases. **(c)** Linear similarity as we increase the amount of data for $\mathbf{f}_{\boldsymbol{\theta}}$ and $\mathbf{g}_{\boldsymbol{\theta}}$. Error bars are computed over 5 pairs of models. **(d)** As we increase model size, linear similarity after convergence increases for both $\mathbf{f}_{\boldsymbol{\theta}}$ and $\mathbf{g}_{\boldsymbol{\theta}}$.

cially limited the size of the dataset, and plot mean correlation after training and convergence. This shows that increasing availability of data correlates with closer alignment. In Figure 3d, we fix dataset size and artificially limit the model capacity (number hidden units) to investigate the effect of model size on the learned representations, varying the number of hidden units from 64 to 8192. This show that increasing model capacity correlates with increase in alignment of learned representations.

5.3. GPT-2

Finally, we report on a study of GPT-2 (Radford et al., 2019), a massive-scale language model. The identifiable representation is the set of features just before the last linear layer of the model. We use pretrained models from HuggingFace (Wolf et al., 2019). HuggingFace provides four different versions of the GPT-2: `gpt2`, `gpt2-medium`, `gpt2-large` and `gpt2-xl`, which differ mainly in the hyper-parameters that determine the width and depth of the neural network layers. For approximately 2000 input sentences, per timestep, for each model, we extracted representations at the last layer (which is identifiable) in addition to the representations per timestep given by three earlier layers in the model. Then, we performed SVCCA on each possible pair of models, on each of the four representations. SVCCA was performed with 16, 64, 256 and 768 principal components, computed by applying SVD separately for each representations of each model. We chose 768 as the largest number of principal components, since that is the representation size for the smallest model in the repository (`gpt2`). We then averaged the CCA correlation coefficients across the pairs of models. Figure 4 shows the results. The results align well with our theory, namely that the representations at the last layer are more linearly related than the representations at other layers of the model.

5.4. Interpretation and Summary

Theorem 1 establishes linear identifiability as an asymptotic property of a model that holds in the limit of infinite data and exact estimation. The experiments of this section have shown that for linear identifiable models, when the dimen-

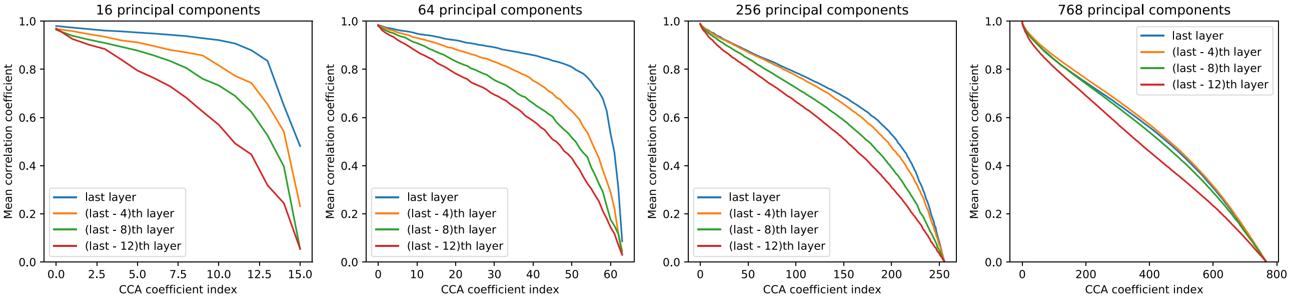

Figure 4. **Text Embeddings by GPT-2**. GPT-2 results. Representations of the last hidden layer (which is identifiable), in addition to three earlier layers (not necessarily identifiable) for four GPT-2 models. For each representation layer, SVCCA is computed over to all pairs of models, over which correlation coefficients were averaged. SVCCA was applied with 16, 64, 256 and 768 principal components. The learned representations in the *last*, identifiable layer more correlated than representations learned in preceding layers.

sionality is small relative to dataset size (Figures 1 and 2), the learned embeddings are closely linearly related, up to noise. Problems of model estimation and sufficient dataset size are more pronounced in high dimensions. Nevertheless, in GPT-2, representations among different trained models do in fact approach a mean correlation coefficient of 1.0 after training (Figure 4, blue line), providing strong evidence of linear identifiability.

6. Related Works

Prior to Hyvärinen & Morioka (2016), identifiability analysis was uncommon in deep learning. We build on advances in the theory of nonlinear ICA (Hyvärinen & Morioka, 2016; Hyvärinen et al., 2018; Khemakhem et al., 2019). In this section, we carefully distinguish our results from prior and concurrent works. Our diversity assumption is similar to diversity assumptions in these earlier works, while differing on certain conditions. The main difference is that their results apply to related but distinct families of models compared to the general discriminative family outlined in this paper. Arguably most related is Theorem 3 of Hyvärinen et al. (2018) and its proof, which shows that a class of contrastive discriminative models will estimate, up to an affine transformation, the true latent variables of a nonlinear ICA model. The main difference with our result is that they additionally assume: (1) that the mapping between observed variables and latent representations is invertible; and (2) that the discriminative model is binary logistic regression exhibiting universal approximation (Hornik et al., 1989), estimated with a contrastive objective. In addition, (Hyvärinen et al., 2018) does not present conditions for affine identifiability for their version of the context representation function $\mathbf{g}$. It should be noted that Theorem 1 in (Hyvärinen et al., 2018) provides a potential avenue for further generalization of our Theorem 1 to discriminative models with non-linear interaction between $\mathbf{f}$ and $\mathbf{g}$.

Concurrent work (Khemakhem et al., 2020) has expanded the theory of identifiable nonlinear ICA to a class of conditional energy-based models (EBMs) with universal density approximation capability, therefore imposing milder assumptions than previous nonlinear ICA results. Their version of affine identifiability is similar to our result of linear identifiability in Section 3.2. The main differences are that Khemakhem et al. (2020) focus in both theory and experiment on EBMs. This allows for alternative versions of the diversity condition, assuming that the Jacobians of their versions of $\mathbf{f}$ or $\mathbf{g}$ are full rank. This is only possible if $\mathbf{x}$ or $\mathbf{y}$ are assumed continuous-valued; note that we do not make such an assumption. Khemakhem et al. (2020) also presents an architecture for which the conditions provably hold, in addition to sufficient conditions for identifiability up to element-wise scaling, which we did not explore in this work. While we build on these earlier results, we are, to the best of our knowledge, the first to apply identifiability analysis to state-of-the-art discriminative and autoregressive generative models.

ecent work on the asymptotics of fully-connected, infinitely wide neural networks (Lee et al., 2017) has shown that they converge to a Gaussian Process with a particular approximable kernel, extending earlier work on single-layer networks (Neal, 1995). Jacot et al. (2018) prove that the evolution of a neural network during training can also be described by a kernel, termed the Neural Tangent Kernel (NTK). Both are fine-grained analysis that place restrictions on the forms of the neural networks under analysis in order to produce strong analytic results. Like NTK, we take a function-space perspective, but our analysis considers learned representation functions and their optimal solution sets.

7. Conclusion

We have shown that representations learned by a large family of discriminative models are identifiable up to a linear transformation, providing a novel perspective on representation learning using DNNs. Since identifiability is a property of a model class, and identification is realized in the asymptotic limit of data and compute, we perform experiments in the more realistic setting with finite datasets and finite compute. Our empirical results show that as the representational capacity of the model and dataset size increases, learned representations indeed tend towards solutions that are equal up to only a linear transformation.

8. Acknowledgements

We thank Ryan Adams for in-depth conversations on methodology, Kevin Murphy for helpful feedback on an early version, Aapo Hyvärinen, Ilyes Khemakhem, and Jascha Sohl-Dickstein for helpful conversations and feedback, George Tucker and Alexi Alemi for feedback on an early version of the theorem, and members of the Google Brain Team and Princeton Laboratory for Intelligent Probabilistic Systems for valuable remarks and comments. Geoffrey Roeder is supported in part by the National Science and Engineering Research Council of Canada (PGSD3-518716-2018).

References

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., and Wanderman-milne, S. Jax: Composable transformations of Python+NumPy programs, 2018. URL Http://Github.Com/Google/Jax.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., and Others. Language Models are Few-Shot Learners. *Arxiv Preprint Arxiv:2005.14165*, 2020.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, t. One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. *Arxiv Preprint Arxiv:1312.3005*, 2013.

Dai, A. M. and Le, Q. V. Semi-Supervised Sequence Learning. In *Advances in Neural information Processing Systems*, pp. 3079–3087, 2015.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Arxiv Preprint Arxiv:1810.04805*, 2018.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. Why Does Unsupervised Pre-training Help Deep Learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.

Hénaff, O. J., Razavi, A., Doersch, C., Eslami, S., and Oord, A. V. D. Data-Efficient Image Recognition with Contrastive Predictive Coding. *Arxiv Preprint Arxiv:1905.09272*, 2019.

Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

Hoffer, E. and Ailon, N. Deep Metric Learning Using Triplet Network. In *International Workshop On Similarity-Based Pattern Recognition*, pp. 84–92. Springer, 2015.

Hornik, K., Stinchcombe, M., and White, H. Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, 2(5):359–366, 1989.

Hotelling, H. Relations Between Two Sets of Variates. *Biometrika*, 28(3/4):321–377, 1936.

Hyvärinen, A. and Morioka, H. Unsupervised Feature Extraction by Time-Contrastive Learning and Nonlinear ICA. In *Advances in Neural information Processing Systems*, pp. 3765–3773, 2016.

Hyvärinen, A., Sasaki, H., and Turner, R. E. Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning. *Arxiv Preprint Arxiv:1805.08651*, 2018.

Jacot, A., Gabriel, F., and Hongler, C. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *Advances in Neural information Processing Systems*, pp. 8571–8580, 2018.

Johansson, F., Shalit, U., and Sontag, D. Learning representations for counterfactual inference. In *International conference on machine learning*, pp. 3020–3029, 2016.

Khemakhem, I., Kingma, D. P., and Hyvärinen, A. Variational Autoencoders and Nonlinear ICA: A Unifying Framework. *Arxiv Preprint Arxiv:1907.04809*, 2019.

Khemakhem, I., Monti, R. P., Kingma, D. P., and Hyvärinen, A. ICE-BeeM: Identifiable Conditional Energy-based Deep Models. *Arxiv Preprint Arxiv:2002.11537*, 2020.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. *Arxiv Preprint Arxiv:1412.6980*, 2014.

Krizhevsky, A., Hinton, G., and Others. Learning Multiple Layers of Features from Tiny Images. 2009.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-dickstein, J. Deep Neural Networks as Gaussian Processes. *Arxiv Preprint Arxiv:1711.00165*, 2017.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. Generating Wikipedia by Summarizing Long Sequences. *Arxiv Preprint Arxiv:1801.10198*, 2018.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, pp. 6446–6456, 2017.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., and Khudanpur, S. Recurrent Neural Network Based Language Model. In *Eleventh Annual Conference of The international Speech Communication Association*, 2010.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural information Processing Systems*, pp. 3111–3119, 2013.

Mnih, A. and Hinton, G. E. A Scalable Hierarchical Distributed Language Model. In *Advances in Neural information Processing Systems*, pp. 1081–1088, 2009.

Mnih, A. and Teh, Y. W. A Fast and Simple Algorithm for Training Neural Probabilistic Language Models. *Arxiv Preprint Arxiv:1206.6426*, 2012.

Morcos, A. S., Raghu, M., and Bengio, S. Insights on Representational Similarity in Neural Networks with Canonical Correlation, 2018.

Neal, R. M. *Bayesian Learning for Neural Networks*. PhD thesis, University of toronto, 1995.

Oord, A. V. D., Li, Y., and Vinyals, O. Representation Learning with Contrastive Predictive Coding. *Arxiv Preprint Arxiv:1807.03748*, 2018.

Pearson, K. LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving Language Understanding by Generative Pre-training. 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language Models are Unsupervised Multitask Learners. *Openai Blog*, 1(8), 2019.

Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and interpretability. In *Advances in Neural information Processing Systems*, pp. 6076–6085, 2017.

Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *Proceedings of The Ieee Conference On Computer Vision and Pattern Recognition Workshops*, pp. 806–813, 2014.

Sohn, K. Improved Deep Metric Learning with Multi-class N-Pair Loss Objective. In *Advances in Neural information Processing Systems*, pp. 1857–1865, 2016.

Sorrenson, P., Rother, C., and Köthe, U. Disentanglement by Nonlinear ICA with General Incompressible-flow Networks (Gin). *Arxiv:2001.04872 [Cs, Stat]*, January 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is All You Need. In *Advances in Neural information Processing Systems*, pp. 5998–6008, 2017.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proceedings of The 25th international Conference On Machine Learning*, pp. 1096–1103, 2008.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. Huggingface's Transformers: State-of-the-art Natural Language Processing. *Arxiv*, Abs/1910.03771, 2019.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. XLNET: Generalized Autoregressive Pretraining for Language Understanding. *Arxiv Preprint Arxiv:1906.08237*, 2019.