ON LINEAR IDENTIFIABILITY OF LEARNED REPRESENTATIONS

Anonymous authors
Paper under double-blind review

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 modern 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). 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 are reproducible, in the sense that every time we relearn the representation function on the same data distribution, it is approximately the same, regardless of small deviations in the initialization of the parameters or the optimization procedure. 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. In this paper, we make this idea precise, and show when it can be expected to hold.

Advances in the theory of nonlinear Independent Components Analysis (ICA) 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)). In particular, we prove that a large class of discriminative and autoregressive models are identifiable in function space up to a linear transformation.

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),

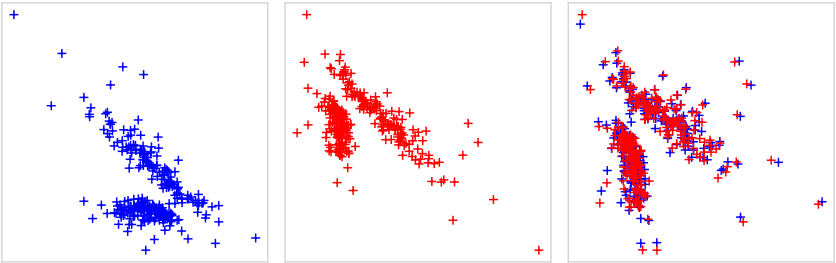

Figure 1: Left and Middle: Two learned neural network representation functions $\boldsymbol{f}_{\boldsymbol{\theta}_1}(\mathcal{B})$, $\boldsymbol{f}_{\boldsymbol{\theta}_2}(\mathcal{B})$ visualized on held-out data $\mathcal{B}$. The networks are learned word embeddings, trained on the Billion Word Dataset (Chelba et al., 2013) (see Appendix A.1 for model details and code release). Right: $\boldsymbol{A}\boldsymbol{f}_{\boldsymbol{\theta}_1}(\mathcal{B})$ and $\boldsymbol{f}_{\boldsymbol{\theta}_2}(\mathcal{B})$, 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 investigates the 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 the network architecture and training data too. This renders any analysis in full generality challenging. In this section, we will describe sufficient conditions on the data distribution and model that allow for identifiability up to a linear transformation. Using these conditions, we prove the identifiability results in Section 3. Interestingly, many discriminative models used for representation learning satisfy these conditions, as we show 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 a target variable whose value the model predicts. In case of object classification, this would be some 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 whose models obey equation (1) 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 and Ailon, 2015; Sohn, 2016), the set $\mathbf{S}$ contains one positive and k negative samples of the class $\mathbf{x}$ is associated with in the training data.

Canonical Discriminative Form Given a data distribution as above, the model family is defined by its parametric probability of target variable $\mathbf{y}$ given an observed data $\mathbf{x}$ and set $\mathbf{S}$ comprising the true label $\mathbf{y}$ and 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. See Section 4 and Appendix D for extensive examples and discussion.

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, in particular DNNs. 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. The proof in Section 3.2 elaborates additional assumptions on $\mathbf{f}_{\boldsymbol{\theta}}$ and $\mathbf{g}_{\boldsymbol{\theta}}$ needed for identifiability.

3 Model Identifiability

In this section, we derive the identifiability conditions of models in the family defined in Section 2. In short, sufficient conditions for identifiability by weakening the classical definition to admit identifiability up to a subset of the parameter space Θ.

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 $\mathbf{g}_{\boldsymbol{\theta}}$ parameterizes $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{S})$, e.g., probability of group membership, 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:*

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

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*.

We next present a simple derivation of the $\overset{L}{\sim}$-identifiability of members of the generalized discriminative family defined in Section 2. The proof reveals sufficient conditions that DNNs, parameterizing

$\mathbf{f}_{\boldsymbol{\theta}}$ and $\mathbf{g}_{\boldsymbol{\theta}}$, must satisfy for the model to be $\overset{\text{L}}{\sim}$-identifiable. For brevity, we will use shorthands $p' := p_{\boldsymbol{\theta}'}, p^* := p_{\boldsymbol{\theta}^*}$, $\mathbf{f}^* := \mathbf{f}_{\boldsymbol{\theta}^*}, \mathbf{f}' := \mathbf{f}_{\boldsymbol{\theta}'}, \mathbf{g}' := \mathbf{g}_{\boldsymbol{\theta}'}$, and drop superscripts when the statement applies to both.

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 $\mathbf{y}_A, \mathbf{y}_B \in \mathbf{S}$, 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 $\mathbf{g}^*(\mathbf{y}_A^{(i)}) - \mathbf{g}^*(\mathbf{y}_B^{(i)}),\ i \in \{1, \dots, M\}$. See Section 3.3 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,

$$p' = p^* \quad \Longrightarrow \quad \boldsymbol{\theta}' \overset{\text{L}}{\sim} \boldsymbol{\theta}^*. \tag{4}$$

To establish the result, we proceed by directly constructing an invertible linear transformations that satisfies Definition 1. Consider $\mathbf{y}_A, \mathbf{y}_B \in \mathbf{S}$. The likelihood ratios for these points

$$\frac{p'(\mathbf{y}_A|\mathbf{x}, \mathbf{S})}{p'(\mathbf{y}_B|\mathbf{x}, \mathbf{S})} = \frac{p^*(\mathbf{y}_A|\mathbf{x}, \mathbf{S})}{p^*(\mathbf{y}_B|\mathbf{x}, \mathbf{S})} \tag{5}$$

are equal. Substituting our model definition from equation (1), we find:

$$\frac{\exp(\mathbf{f}'(\mathbf{x})^\top \mathbf{g}'(\mathbf{y}_A))}{\exp(\mathbf{f}'(\mathbf{x})^\top \mathbf{g}'(\mathbf{y}_B))} = \frac{\exp(\mathbf{f}^*(\mathbf{x})^\top \mathbf{g}^*(\mathbf{y}_A))}{\exp(\mathbf{f}^*(\mathbf{x})^\top \mathbf{g}^*(\mathbf{y}_B))}, \tag{6}$$

where the normalizing constants cancelled out on the left- and right-hand sides. Taking the logarithm, this simplifies to:

$$(\mathbf{g}'(\mathbf{y}_A) - \mathbf{g}'(\mathbf{y}_B))^\top \mathbf{f}'(\mathbf{x}) = (\mathbf{g}^*(\mathbf{y}_A) - \mathbf{g}^*(\mathbf{y}_B))^\top \mathbf{f}^*(\mathbf{x}). \tag{7}$$

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

We next collect M distinct tuples $(\mathbf{y}_A^{(i)}, \mathbf{y}_B^{(i)})$ so that by repeating Equation (7) M times and by the diversity condition noted above, the resulting difference vectors are linearly independent. We collect these vectors together as the columns of $(M \times M)$-dimensional matrices $\mathbf{L}'$ and $\mathbf{L}^*$, forming the following system of M linear equations:

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

Since $\mathbf{L}'$ and $\mathbf{L}^*$ are invertible, we rearrange:

$$\mathbf{f}'(\mathbf{x}) = (\mathbf{L}^* \mathbf{L}'^{-1})^\top \mathbf{f}^*(\mathbf{x}). \tag{8}$$

Hence, $\mathbf{f}'(\mathbf{x}) = \mathbf{A}\mathbf{f}^*(\mathbf{x})$ where $\mathbf{A} = (\mathbf{L}^* \mathbf{L}'^{-1})$, completing the first half of the proof. See Appendix C for the remainder.

3.3 Discussion of Diversity Condition

Theorem 1 requires invertible $(M \times M)$ matrices $\mathbf{L}'$ and $\mathbf{L}^*$. This requirement is similar to the conditions in earlier work on nonlinear ICA such as (Hyvärinen et al., 2018), as discussed in Section 6. Informally, this means that there needs to be a sufficient number of possible values $\mathbf{y} \in \mathbf{S}$. In the case of a supervised classification with K classes, $\mathbf{S}$ is fixed and of size K. Then, we need $K \geq M + 1$ in order to generate M difference vectors $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}^{(1)}) - \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}^{(j)})$, $j = 2, \dots, M+1$. In case of self-supervised or deep metric learning, where $\mathbf{S}$ and $\mathbf{y}$ may be stochastically generated from $\mathbf{x}$, there will typically be a diversity of values of $\mathbf{y}$.

Note that by the diversity requirement, we implicitly assume $\mathbf{g}_{\boldsymbol{\theta}}$ to have the following property: the M difference vectors span the range of $\mathbf{g}_{\boldsymbol{\theta}}$. This is a mild assumption in the context of DNNs: for random initialization and iterative weight updates, this property follows from the sampling distribution used to choose the initial weights. Briefly, a set of $M + 1$ unique points $\mathbf{y}^{(j)}$ such that the M vectors $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}^{(1)}) - \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}^{(j)})$, $j = 2, \dots, M+1$ 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.

4 EXAMPLES OF LINEARLY IDENTIFIABLE MODELS

The conditions for Theorem 1 hold in a variety of models for different problem contexts, often state-of-the-art at time of publication: Contrastive Predictive Coding (Hénaff et al., 2019), BERT (Devlin et al., 2018), GPT-2 and GPT-3 (Radford et al., 2018, 2019, **?**), XLNET (Yang et al., 2019), and the triplet loss for deep metric learning (Sohn, 2016). Functions $\mathbf{f}_{\boldsymbol{\theta}}$ and $\mathbf{g}_{\boldsymbol{\theta}}$ have been implemented using a variety of deep learning architectures. We discuss these here. For reductions to the canonical form of Equation 1, please refer to Appendix D.

Deep Supervised Classification. Classification models that deploy DNNs as feature extractors satisfy the sufficient conditions given in Section 3 when: the network from input to the layer prior to the logits is the representation function $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$ and weights in the final projection layer are given by a context map $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y})$ that depends only on the labels. The class labels $\mathbf{y}$ are integers 0 to $K-1$, and $\mathbf{w}_i = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}=i)$ is the i-th column of a weight matrix $\mathbf{W}$, and the vector representation of the i-th class. Other than a simulation study to empirically validate the above claim (Section 5.1), we do not explore supervised classification further. Self-supervised and multi-task pretraining have been empirically shown to be more data efficient for learning effective representations.

Self-Supervised Pretraining for Image Classification. Self-supervised learning is supervised learning with algorithmically-generated labels. Typically, self-supervision is used to pretrain network weights using a classification task with synthetically-generated data and labels, in order to improve performance a downstream, related task. Contrastive Predictive Coding (CPC) (Oord et al., 2018) is a paradigmatic self-supervised method, often applied to image domains, and is a member of the model family of Section 2. CPC as applied to images involves: (1) pre-processing the image into 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. The collection of all patches from the sequence, from a given minibatch of images, the set $\mathbf{S}$.

Multi-task Pretraining for Natural Language Generation. Autoregressive language models, such as (Mikolov et al., 2010; Dai and 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 and Schmidhuber, 1997) or an autoregessive 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, typically implemented as a simple look-up table.

BERT (Devlin et al., 2018) learns 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 pre-training 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–that is, the domain of deep learning practice. Results in this section present evidence for: (1) close convergence in the small dimensional, large data regime; and (2) monotonic increase in linear similarity of learned representations as a function of dataset size and model capacity in the high dimensional regime. Note that due to the

generality of $\mathbf{f}_{\boldsymbol{\theta}}$ and $\mathbf{g}_{\boldsymbol{\theta}}$ and Theorem 1, the number of possible experiments is huge. Here, we focus on a core set of linearly identifiable models, interpolating from a low dimensional simulation study of deep supervised classification to GPT-2 (Radford et al., 2019), an approximately $1.5 * 10^9$-parameter generative model of natural language. Please see Appendix A for details needed to reproduce.

Measuring linear similarity between learned representations. Our goal is to 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. We select a subset $\mathcal{B} \subset \mathcal{D}$ of the training data and compute $\boldsymbol{f}_{\boldsymbol{\theta}_1}(\mathcal{B})$ and $\boldsymbol{f}_{\boldsymbol{\theta}_2}(\mathcal{B})$, for two models with parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ respectively. Assume without loss of generality that $\boldsymbol{f}_{\boldsymbol{\theta}_1}(\mathcal{B})$ and $\boldsymbol{f}_{\boldsymbol{\theta}_2}(\mathcal{B})$ are centered. Then, CCA finds the optimal linear transformations $\boldsymbol{C}$ and $\boldsymbol{D}$ such that the pairwise correlations $\rho_i = \text{Corr}[\boldsymbol{C}_i^\top \boldsymbol{f}_{\boldsymbol{\theta}_1}(\mathcal{B}), \boldsymbol{D}_i^\top \boldsymbol{f}_{\boldsymbol{\theta}_2}(\mathcal{B})]$ are maximized. If one vector is a linear transform of another, CCA applied to both will learn this transformation and the mean of $\boldsymbol{\rho}$ will be 1; if they are perfectly uncorrelated then the mean of $\boldsymbol{\rho}$ is 0. We use the mean of $\boldsymbol{\rho}$ as a proxy for the existence of a linear transformation between $\boldsymbol{f}_{\boldsymbol{\theta}_1}(\mathcal{B})$ and $\boldsymbol{f}_{\boldsymbol{\theta}_2}(\mathcal{B})$. 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 images $\boldsymbol{f}_{\boldsymbol{\theta}_2}(\mathcal{B})$ and $\boldsymbol{f}_{\boldsymbol{\theta}_1}(\mathcal{B})$, 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

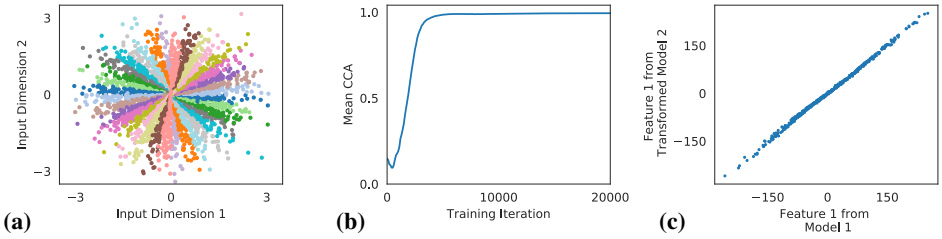

Figure 2: **Deep Supervised Classification**. **(a)** Data distribution for a linearly identifiable K-way classification problem. **(b)** Mean (centered) CCA between the learned representations over the course of training. After approx. 4000 iterations, CCA finds a linear transformation that rotate the 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.

We conducted a simulation study of linearly identifiable K-way classification, where all assumptions and sufficient conditions are met by construction. First, we designed a data distribution with the properties required by Section 2. The data distribution $p_{\mathcal{D}}(\mathbf{x}, \mathbf{y}, \mathbf{S})$ was generated as follows: points $\mathbf{x}$ are sampled from a 2-D Gaussian with $\sigma = 3$ and assigned pre-labels among K classes radially, for $K = 18$ (σ chosen arbitrarily and K to ensure $K \geq M + 1$). The context map is $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}) = \mathbf{W}\mathbf{e}_{\mathbf{y}}$ where $\mathbf{e}_{\mathbf{y}}$ is a one-hot encoding of the class identity.

For this experiment, we want to minimize noise in the model parameters due to model misspecification. Model misspecification would add noise optimizing the parameters of the function approximators $\mathbf{f}_{\boldsymbol{\theta}'}$ and $\mathbf{g}_{\boldsymbol{\theta}'}$ to $\mathbf{f}_{\boldsymbol{\theta}^\star}$ and $\mathbf{g}_{\boldsymbol{\theta}^\star}$. We eliminated model misspecification by fitting two neural networks $f_{\boldsymbol{\theta}^\star}$ and $g_{\boldsymbol{\theta}^\star}$ to predict the labels according to Equation (1). In particular, both are 4-hidden-layer MLPs with two 64 unit layers and one 2-D bottle neck layer. We then used these representation functions to predict ground truth labels for each $\mathbf{x}$. Finally, we chose $\mathbf{f}_{\boldsymbol{\theta}'}$ and $\mathbf{g}_{\boldsymbol{\theta}'}$ to be the same architecture as $\mathbf{f}_{\boldsymbol{\theta}^\star}$ and $\mathbf{g}_{\boldsymbol{\theta}^\star}$, ensuring that the parameters $\boldsymbol{\theta}'$ and $\boldsymbol{\theta}^\star$ live in the same space Θ. This enforces that the function approximators $\mathbf{f}_{\boldsymbol{\theta}'}$ and $\mathbf{g}_{\boldsymbol{\theta}'}$ are able to represent the true data-generating process.

To evaluate linear similarity, we trained two randomly initialized models of $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{S})$ with the same architecture as the data generating process. We visualize the image of $\mathbf{f}_{\boldsymbol{\theta}}$, the data representation function. 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 up to estimation noise. Similarly, Figure 2c shows that the learned representations, after mapping through the optimal linear transformation from CCA, have a strongly linear relationship.

5.2 Self-Supervised Learning for Image Classification

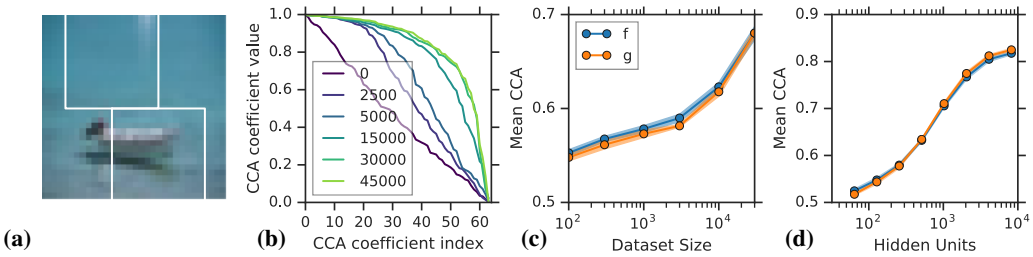

Figure 3: **Self-Supervised Representation Learning. (a)** Input data. Two patches are taken (one from top half, and one from the bottom half) of an image at random. Using a contrastive loss, we predict the identity of the bottom patch encoding from the top patch encoding. **(b)** Linear similarity of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

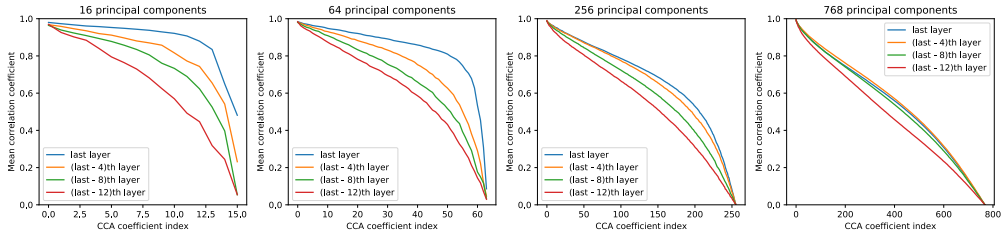

Figure 4: **Text Embeddings by GPT-2**. GPT-2 results. We computed representations of the last hidden layer (which is identifiable), in addition to three earlier layers (not necesarilly identifiable) for four GPT-2 models (Radford et al., 2019). For each representation layer, we then applied SVCCA to all possible pairs of models and averaged their correlation coefficients. SVCCA was applied with 16, 64, 256 and 768 principal components. The results show that, in line with expectations, the learned representations in the *last* layer are much more correlated between the four models than the representations learned in preceding layers.

We next investigate Theorem 1 when we can no longer expect efficient model estimation or nice distributional properties: 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 a the true patch with a set of distractor patches from across the minibatch. For each model we define both $\mathbf{f}_{\theta'}$ and $\mathbf{g}_{\theta'}$ as a 3-layer MLP with 256 units per layer (except where noted otherwise) and fix output dimensionality of 64.

In Figure 3b, we plot CCA coefficients over the course of training. As training progresses, the linear similarity between the learned representations increases. In Figure 3c, we artificially limit the size of the dataset, and plot mean correlation after retraining and convergence. This shows that increasing availability of data correlates with closer linear similarity. 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 size correlates strongly with increase in linear similarity of learned representations. Despite lack of estimation guarantees, these experiments validate Theorem 1.

5.3 GPT-2

Next, we performed experiments with GPT-2 (Radford et al., 2019), a large-scale language model. The identifiable representation is, as usual, the set of features just before the last linear layer of the model. We use pre-trained models from HuggingFace (Wolf et al., 2019). This repository 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.

6 Related Works

Prior to Hyvärinen and Morioka (2016), identifiability analysis was uncommon in deep learning. We build on advances in the theory of nonlinear ICA (Hyvärinen and 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.

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 one increases the representational capacity of the model and dataset size, learned representations indeed tend towards solutions that are equal up to only a linear transformation.

REFERENCES

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

K. Pearson. 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.

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

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

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

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

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

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

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

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

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

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