ON THE POWER OF FOUNDATION MODELS

Yang Yuan[1,2,3]

[1]IIIS, Tsinghua University
[2]Shanghai Artificial Intelligence Laboratory
[3]Shanghai Qi Zhi Institute

ABSTRACT

With infinitely many high-quality data points, infinite computational power, an infinitely large foundation model with a perfect training algorithm and guaranteed zero generalization error on the pretext task, can the model be used for everything? This question cannot be answered by the existing theory of representation, optimization or generalization, because the issues they mainly investigate are assumed to be nonexistent here. In this paper, we show that category theory provides powerful machinery to answer this question. We have proved three results. The first one limits the power of prompt-based learning, saying that the model can solve a downstream task with prompts if and only if the task is representable. The second one says fine tuning does not have this limit, as a foundation model with the minimum required power (up to symmetry) can theoretically solve downstream tasks for the category defined by pretext task, with fine tuning and enough resources. Our final result can be seen as a new type of generalization theorem, showing that the foundation model can generate unseen objects from the target category (e.g., images) using the structural information from the source category (e.g., texts). Along the way, we provide a categorical framework for supervised and self-supervised learning, which might be of independent interest.

1 Introduction

Foundation models have recently exhibited remarkable proficiency in addressing a myriad of complex downstream tasks that were very difficult or impossible for the previous methods (Ramesh et al., 2021; Rombach et al., 2022; Ramesh et al., 2022; Sohl-Dickstein et al., 2015; Brown et al., 2020; Radford et al., 2018, 2019; He et al., 2022). Being different in network structure, dataset and training algorithms, the foundation models are similar in terms of the training process: the model is first trained with a large unlabeled dataset on a pretext task, and then being applied to other downstream tasks with the parameters frozen. Freezing the parameter is necessary because training a foundation model is very expensive, and the downstream tasks usually have limited data samples, which makes retraining less attractive.

Given a set of downstream tasks, which of them are solvable by the foundation models, and which are not? This is a very fundamental question. The existing theory attacks this question through various perspectives like dataset quality and quantity, computational power, network structure, etc. However, as we collect trillions of data points, build gigantic GPU data centers, and optimize huge networks with billions of parameters, a simple question just pops up:

Where are we heading to?

Specifically, with infinitely many high-quality data points, infinite computational power, an infinitely large foundation model with a perfect training algorithm and guaranteed zero generalization error on the pretext task, can the model be used for every downstream task? As we will see, this question is not about model representation, optimization or generalization, but structural representation of the tasks. Category theory, as the theory of mathematical structures, is the ideal tool for answering this question.

For the downstream tasks, there are mainly two types of methods: prompt tuning and fine tuning. Prompt tuning does not train with the downstream tasks. Instead, it only sends a task specific prompt to the model, so that the model can "switch" its working mode for solving the task. Fine tuning trains a small network connecting to the foundation model with the labeled dataset for the downstream task.

In Theorem 1, we show that with prompt tuning, the model can solve the task if and only if the task is "representable" in the category defined by the pretext task. If the category does not have complicated structures, our theorem indicates that the power of prompt tuning is limited. For example, in Corollary 1, we show that by training with the pretext task of predicting the rotation of an image Gidaris et al. (2018), the foundation model is not able to solve complicated downstream tasks like segmentation or classification.

On the other hand, the results on fine tuning is more promising. Our Theorem 2 proves that for the foundation model with the minimum required power (up to symmetry) for the pretext task and enough resources including training data, it can potentially solve any downstream tasks for the category defined by pretext task. The role of pretext task is crucial in the sense that if the pretext task fails to extract adequate information from the unlabeled dataset, the power of fine tuning remains restricted.

Along the way, we have provided a categorical framework for machine learning. Interestingly, the framework injects the learning perspective to category theory as well. Therefore, we also proved a generalization theorem for structural learning (Theorem 3), which explains why self-supervised learning for text-image generation tasks can generate images like avocado chair, which do not exist in the dataset or real world. Theorem 3 can be easily generalized to the compositional theorem of multiple categories, see Theorem 4.

Unlike most machine learning theory papers, our paper does not have any assumptions on the data distribution or network structure. Instead, we take the bird's-eye view that is model oblivious, and only focuses on the structure defined by the pretext task. It is indeed possible that by designing a special network, one may get a more powerful model with better performance. However, we stick with our setting because:

- **Empirically, people do not customize network structures for different tasks.** Instead, they tend to use similar structures like ResNet He et al. (2016) or Transformer Vaswani et al. (2017). By the no free lunch theorem Shalev-Shwartz and Ben-David (2014), if the model does not contain task-specific prior information, it will not be able to completely solve all the tasks. In other words, the standard models are not universally competent, and it is likely that the limitation that we derived from the model oblivious setting, also applies to the practical settings.
- **Pretext task design is a central problem in self-supervised learning.** Currently, there are various datasets floating around, and a wide range of diverse tasks to solve, but the practitioners do not really know what kinds of pretext tasks to pick for solving a given task, or the limitations of each pretrained model. They get the intuition by trial and error with experiments, which are both expensive and noisy. Our framework will help them to think about this problem in a more mathematical and systematic way, and provide guidance for better pretext task design.

2 Related Work

2.1 Self-supervised learning

Self-supervised learning. Recently, researchers have proposed many self-supervised learning algorithms for foundation models, including contrastive methods (Chen et al., 2020; He et al., 2020; Grill et al., 2020; Chen and He, 2021; Noroozi and Favaro, 2016; Zbontar et al., 2021), masked image models (He et al., 2022; Dosovitskiy et al., 2020; Doersch et al., 2015; Pathak et al., 2016), masked language models (Devlin et al., 2018; Raffel et al., 2020), pure language models (Brown et al., 2020; Radford et al., 2018, 2019), and with other pretext tasks (Oord et al., 2018; Gidaris et al., 2018; Clark et al., 2020; Noroozi and Favaro, 2016; Pathak et al., 2017).

Multimodal learning. Self-supervised learning can also be applied to multimodal learning, including text + image (Ramesh et al., 2021; Rombach et al., 2022; Ramesh et al., 2022; Sohl-Dickstein et al., 2015), video + audio (Arandjelovic and Zisserman, 2018). For generating images for the multimodal tasks, diffusion model is the state-of-the-art approach (Rombach et al., 2022; Sohl-Dickstein et al., 2015; Dhariwal and Nichol, 2021; Ho et al., 2020).

Prompt tuning. There are various prompt tuning methods, including the discrete prompts (Brown et al., 2020; Jiang et al., 2020; Shin et al., 2020; Gao et al., 2020) and continuous prompts (Liu et al., 2021; Li and Liang, 2021).

2.2 Theory for deep learning and self-supervised learning

Theory for deep learning has been an active research area recently. Optimization theory focuses on how and why first order methods like stochastic gradient descent finds the local/global optimum of the neural networks (Du et al., 2019b; Arora et al., 2019a; Du et al., 2018; Allen-Zhu et al., 2019b,a; Zou et al., 2020; Li and Yuan, 2017). Generalization theory focuses on how the performance of the model in the training set transfers to the population distribution (Allen-Zhu et al., 2019a; Arora et al., 2019a; Bartlett et al., 2017, 2020; Yin et al., 2019). Representation theory focuses on the representation power of the neural networks (Hornik et al., 1989; Cybenko, 1989; Raghu et al., 2017). There are also theoretical results on analyzing various aspects of reinforcement learning (Du et al., 2019a,c; Jin et al., 2018; Cai et al., 2020).

Theory for self-supervised learning. There are many interesting theory results for self-supervised learning learning (Wen and Li, 2021, 2022; Luo et al., 2022; HaoChen et al., 2021; Arora et al., 2019b; Tosh et al., 2021; Lee et al., 2021; Zimmermann et al., 2021). For example, HaoChen et al. (2021) show that SimCLR is essentially computing spectral graph clustering on the unlabeled dataset. Liu (2022) gives a better rectified flow algorithm with elegant theoretical guarantees for improving the diffusion models.

Application of category theory. Category theory has been applied to many research areas (Fong and Spivak, 2018), including physics (Marquis, 2008; Kuś and Skowron, 2019), design (Censi, 2015), and machine learning (Shiebler et al., 2021; Mahadevan, 2022a,b). The most relevant paper on category theory might be Bradley et al. (2022), where they provide an enriched category theory of natural language.

3 Preliminaries

Category theory is used in almost all areas of mathematics. Here we only introduce the necessary notions for understanding the results of our paper, and skip many important details (e.g., universe and diagram). Curious readers may check Mac Lane (2013); Riehl (2017); Adámek et al. (1990) for more comprehensive introductions.

A category $\mathcal{C}$ has a set of objects $\mathrm{Ob}(\mathcal{C})$, and a set of morphisms $\mathrm{Hom}_{\mathcal{C}}(X, Y)$ from X to Y for every $X, Y \in \mathrm{Ob}(\mathcal{C})$. Given $f \in \mathrm{Hom}_{\mathcal{C}}(X, Y), g \in \mathrm{Hom}_{\mathcal{C}}(Y, Z)$, we define their composition as $g \circ f \in \mathrm{Hom}_{\mathcal{C}}(X, Z)$. Notice that $\circ$ is associative, i.e., $(h \circ g) \circ f = h \circ (g \circ f)$. For every $X \in \mathrm{Ob}(\mathcal{C})$, there exists an unique identity morphism $\mathrm{id}_X \in \mathrm{Hom}_{\mathcal{C}}(X, X)$. A morphism $f : X \to Y$ is an isomorphism if there exists $g : X \leftarrow Y$ such that $f \circ g = \mathrm{id}_Y$ and $g \circ f = \mathrm{id}_X$. In this case, we say X and Y are isomorphic and write $X \simeq Y$.

Given a category $\mathcal{C}$, we define its opposite $\mathcal{C}^{\mathrm{op}}$ by setting $\mathrm{Ob}(\mathcal{C}^{\mathrm{op}}) = \mathrm{Ob}(\mathcal{C})$ and $\mathrm{Hom}_{\mathcal{C}^{\mathrm{op}}}(X, Y) = \mathrm{Hom}_{\mathcal{C}}(Y, X)$. Moreover, given $f \in \mathrm{Hom}_{\mathcal{C}^{\mathrm{op}}}(X, Y), g \in \mathrm{Hom}_{\mathcal{C}^{\mathrm{op}}}(Y, Z)$, the new composition is $g \overset{\mathrm{op}}{\circ} f = f \circ g \in \mathrm{Hom}_{\mathcal{C}^{\mathrm{op}}}(X, Z)$.

We define **Set** to be the category of sets, where the objects are sets, and $\mathrm{Hom}_{\mathbf{Set}}(X, Y)$ is the set of all functions with domain X and codomain Y. Notice that we ignore the subtleties about the universe for better presentation, so here just assume that **Set** does not contain strange objects like a set containing all sets.

Functor is like a function between two categories. Given two categories $\mathcal{C}, \mathcal{C}'$, a functor $F : \mathcal{C} \to \mathcal{C}'$ maps objects from $\mathcal{C}$ to $\mathcal{C}'$ with $F : \mathrm{Ob}(\mathcal{C}) \to \mathrm{Ob}(\mathcal{C}')$ and morphisms from $\mathcal{C}$ to $\mathcal{C}'$ with $F : \mathrm{Hom}_{\mathcal{C}}(X, Y) \to \mathrm{Hom}_{\mathcal{C}'}(F(X), F(Y))$ for all $X, Y \in \mathcal{C}$, so that F preserves identity and composition. Formally, we have $F(\mathrm{id}_X) = \mathrm{id}_{F(X)}$ for all $X \in \mathcal{C}$, and $F(g \circ f) = F(g) \circ F(f)$ for all $f : X \to Y, g : Y \to Z$. We call F an embedding functor, if it is injective on morphisms. Notice that if F is an embedding, it is also injective on objects. We call F a full functor, if for any $X, Y \in \mathcal{C}$, $\mathrm{Hom}_{\mathcal{C}}(X, Y) \to \mathrm{Hom}_{\mathcal{C}'}(FX, FY)$ is surjective.

The morphisms of functors, also called the natural transformation, is the way to transform the functors while preserving the structure. Given two categories $\mathcal{C}, \mathcal{C}'$, and two functors F_1, F_2 from $\mathcal{C}$ to $\mathcal{C}'$. A morphism of functors $\theta : F_1 \to F_2$ has a morphism $\theta_X : F_1(X) \to F_2(X)$ for all $X \in \mathcal{C}$ such that for all $f : X \to Y \in \mathrm{Hom}_{\mathcal{C}}(X, Y)$, we have $\theta_Y(F_1(f)(F_1(X))) = F_2(f)(\theta_X(F_1(X)))$. We write $F_1 \simeq F_2$ if there exists an isomorphism between F_1 and F_2. A functor $F : \mathcal{C} \to \mathcal{C}'$ is an isomorphism of categories if there exists $G : \mathcal{C}' \to \mathcal{C}$ such that $G \circ F(X) = X$ and $F \circ G(Y) = Y$ for all $X \in \mathcal{C}, Y \in \mathcal{C}'$, and similarly for the morphisms. In this case, we say $\mathcal{C}$ and $\mathcal{C}'$ are isomorphic and write $\mathcal{C} \simeq \mathcal{C}'$.

A category $\mathcal{A}$ is a subcategory of a category $\mathcal{B}$, if

1. $\mathrm{Ob}(\mathcal{A}) \subseteq \mathrm{Ob}(\mathcal{B})$,
2. for each $A, A' \in \mathrm{Ob}(\mathcal{A}), \mathrm{Hom}_{\mathcal{A}}(A, A') \subseteq \mathrm{Hom}_{\mathcal{B}}(A, A')$,
3. for each $\mathcal{A}$-object A, the $\mathcal{B}$-identity on A is the $\mathcal{A}$-identity on A;
4. the composition law in $\mathcal{A}$ is the restriction of the composition law in $\mathcal{B}$ to the morphisms of $\mathcal{A}$.

Moreover, $\mathcal{A}$ is a full subcategory of $\mathcal{B}$, if it is a subcategory of $\mathcal{B}$ and for each for each $A, A' \in \mathrm{Ob}(\mathcal{A}), \mathrm{Hom}_{\mathcal{A}}(A, A') = \mathrm{Hom}_{\mathcal{B}}(A, A')$. Every subcategory $\mathcal{A}$ of category $\mathcal{B}$ naturally defines an inclusion functor $E : \mathcal{A} \hookrightarrow \mathcal{B}$, which is an embedding. For such embeddings, we have the following lemma.

Lemma 1 (Adámek et al. (1990))**.** *A functor $F : \mathcal{C} \to \mathcal{B}$ is a full embedding if and only if there exists a full subcategory $\mathcal{A}$ of $\mathcal{B}$ with inclusion functor $E : \mathcal{A} \hookrightarrow \mathcal{B}$ and an isomorphism $G : \mathcal{C} \to \mathcal{A}$ with $F = E \circ G$.*

3.1 Reproducing Kernel Hilbert Space (RKHS)

Given two objects $X, Y \in \mathcal{C}$, consider a feature map $f : \mathcal{C} \to \mathcal{H}$, where the feature space $\mathcal{H}$ is usually much larger than $\mathcal{C}$. We may define a kernel k that measures the similarity of X and Y as $k(x, y) \triangleq \langle f(x), f(y)\rangle_{\mathcal{H}}$, i.e., the inner product between the two object after mapping them to the feature space. For any vector $T \in \mathcal{H}$, it also corresponds to a function $T(\cdot) : \mathcal{C} \to \mathbb{R}$, defined as $T(x) = \langle T, f(x)\rangle_{\mathcal{H}}$. Specifically, $f(y)$ as a vector in $\mathcal{H}$ also represents the function $k(\cdot, y) : \mathcal{C} \to \mathbb{R}$, because for any $x \in \mathcal{C}$, we have $k(x, y) = \langle f(x), f(y)\rangle_{\mathcal{H}}$. Formally, we have:

Definition 1 (Reproducing kernel Hilbert space)**.** *Let $\mathcal{H}$ be a Hilbert space of $\mathbb{R}$-valued functions defined on a non-empty set $\mathcal{C}$. A function $k : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ is called a reproducing kernel of $\mathcal{H}$, and $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS), if k satisfies*

- $\forall x \in \mathcal{C}, k(\cdot, x) \in \mathcal{H}$,
- $\forall x \in \mathcal{C}, \forall f \in \mathcal{H}, \langle f, k(\cdot, x)\rangle_{\mathcal{H}} = f(x)$.

4 Categorical Framework of Supervised Learning

In supervised learning, we have a population distribution $D = (D_X, D_Y)$, representing the ground truth distribution of the input and output data points. The training set $(X_{\text{train}}, Y_{\text{train}})$ and test set $(X_{\text{test}}, Y_{\text{test}})$ are uniformly sampled from (D_X, D_Y). We hope to learn a function $f : X \to Y$ so that $f(x)$ accurately predicts the label $x \in X$. We also define a loss function $L(f, x, y)$ to measure the distance between the prediction $f(x)$ and the correct label, which is hopefully close to 0. The loss function on a dataset (X, Y) is $L(f, X, Y) \triangleq \mathbb{E}_{(x,y)\sim(X,Y)} L(f, x, y)$. The task of supervised learning, is to minimize the population loss $L_{\text{population}} \triangleq \mathbb{E}_{(X,Y)\sim(D_X,D_Y)} L(f, X, Y)$, with access of the training set $(X_{\text{train}}, Y_{\text{train}})$.

Using the language of category theory, we have two categories $\mathcal{X}$ and $\mathcal{Y}$, with objects x, y as input and output data points, respectively. To avoid confusion, below we switch notation from x, y to X, Y to represent the objects, as later we will not use D_X, D_Y any more. The population distribution can be seen as a functor F from $\mathcal{X}$ to $\mathcal{Y}$, representing the correct label Y given the input X. Due to the inherent noise in the real world, there may not always exist the unique correct label for each input X. In other words, the Bayesian optimal solution does not give zero population loss. We will discussion this issue in Section 7.3, and for now let us simply assume that the unique correct labels always exist.

With this formulation, training/test set can be seen as samples over objects in $\mathcal{X}$ with correct labels in $\mathcal{Y}$. Therefore, supervised learning investigates the following question: can we learn a functor F with samples of $X \in \mathcal{X}$ and $F(X) \in \mathcal{Y}$?

Consider the special case that both categories are discrete, meaning that the only morphisms existed are identity morphisms like $\mathrm{id}_X \in \mathrm{Hom}_{\mathcal{X}}(X, X)$ and $\mathrm{id}_Y \in \mathrm{Hom}_{\mathcal{Y}}(Y, Y)$. In other words, both categories are sets without any morphisms between different objects. In this case, learning F is impossible, because it maps a set to another set without any prior knowledge. The no free lunch theorem tells us that unless we have sample size larger than half of the set size, the functor we learned will have constant generalization error with constant probability.

Generalization theory deals with this problem by assuming that F is in a predefined hypothesis class $\mathcal{H}$, or close to a function in $\mathcal{H}$. In category theory, we do not add restrictions to the functors, but to the structure of the categories instead. For example, if we know $\mathcal{X}$ has a linear structure, and F preserves the structure, it is possible to learn F with a few samples.

This formulation seems useless, as it does not even characterize the loss function, which is crucial in supervised learning. This is because **category theory takes the bird's-eye view.** Our categorical framework will not replace the classical framework, or generate better supervised learning algorithms. Instead, it treats the existing supervised learning algorithms a subroutine or a building block, which can be incorporated into a bigger picture. Therefore, it does not care about the loss functions or optimization process, which are treated as the implementation details. Instead, it focuses on the structure of the categories and functors, and tries to understand whether certain functors are learnable or not. It also investigates whether the ability of mastering at one or more tasks can be generalized to other tasks.

5 Categorical Framework of Self-supervised Learning

5.1 Pretext tasks and morphisms

In self-supervised learning, we have a population distribution D of data points without labels, and try to extract useful information from D by setting up pretext tasks. As summarized in Section 2, there are different kinds of pretext tasks, including contrastive methods, masked image/language models, pure language models, etc. The model trained with the pretext task can be seen as a feature extractor $f \in D \to \mathcal{H}$, mapping the input x to its feature representation $f(x)$.

Using the language of category theory, we have a category $\mathcal{C}$, where each object in $\mathcal{C}$ is a data point $X \in D$. For $X, Y \in \mathcal{C}$, $\mathrm{Hom}_{\mathcal{C}}(X, Y)$ is defined by the pretext task. More specifically, **all the existing pretext tasks can be seen as building morphisms between two objects in $\mathcal{C}$**:

- Contrastive methods. Contrastive methods like SimCLR (Chen et al., 2020) modify $X \in \mathcal{C}$ to get two semantically similar copies $X', X'' \in \mathcal{C}$ as the positive pair, and pick another different $Y \in \mathcal{C}$ as the negative sample. The pretext task says $f(X'), f(X'')$ should be close and both of them should be far away from $f(Y)$. It has been proved that contrastive learning is doing spectral clustering on the similarity graph defined by the pretext task (Tan et al., 2023; HaoChen et al., 2021). Specifically, the similarity graph is defined as for every two objects $X, Y \in \mathcal{C}$, they are connected with an edge $\mathrm{Hom}_{\mathcal{C}}(X, Y)$ of weight proportional to $\Pr((X, Y)$ gets sampled as a positive pair). Therefore, in this category, each morphism between two objects is just an edge connecting these objects, and can be represented with a real number.
- Masked image/language models. For this kind of pretext tasks, we mask a portion of the full object and ask the model to predict the masked part. We may treat the revealed part as $X \in \mathcal{C}$, masks as $Y \in \mathcal{C}$, and define the morphism $\mathrm{Hom}_{\mathcal{C}}(X, Y)$ as the full object. For other pairs of objects, their morphisms are defined as empty sets. Therefore, the pretext task defines a category of sub-objects, and the pretrained model can find the correct complement for the given sub-object.
- Pure language models. Each object $X \in \mathcal{C}$ is a sentence of length at most[1] N, and $\mathrm{Hom}_{\mathcal{C}}(X, Y)$ is the probability that Y is a sentence successive to X that differs by the last token. For example, if X is "I am ", Y can be "I am happy". See the detailed discussion in Section 7.2.

Now, we should formally define what we meant by an ideal foundation model.

5.2 Ideal foundation model

Definition 2 (Ideal foundation model)**.** *Given a category $\mathcal{C}$ defined by a pretext task, a foundation model $f : \mathcal{C} \to \mathcal{H}$ is ideal, if there exists a data-oblivious function $k_f : \mathcal{H} \times \mathcal{H} \to$ **Set** so that for any $X, Y \in \mathcal{C}$, $k_f(f(X), f(Y)) = \mathrm{Hom}_{\mathcal{C}}(X, Y)$.*

Here, data-oblivious means k_f is predefined without seeing the data, but the subscript f means k_f as a predefined function, can access f and f^{-1} as oracles. In other words, k_f is a "simple" function, but can borrow the power of foundation model f for expressing more complicated morphisms. Take the compression algorithms for example. The compression algorithms can be data dependent, i.e., optimized for certain data distribution. However, as a data-oblivious function k_f, it cannot access the data distribution, but can call the compression algorithm to decompress the data, because the action of calling an algorithm is data-oblivious. We define k_f for different pretext tasks below.

- For contrastive methods, $\mathrm{Hom}_{\mathcal{C}}(X, Y) \in \mathbb{R}$, so $\mathcal{H}$ can be seen as the RKHS with k_f as a kernel function. For example, in SimCLR (Chen et al., 2020) and MoCo (He et al., 2020), k_f is the Gaussian kernel for the two object embeddings.
- For masked models, X is the revealed part, Y is the masks. In MAE (He et al., 2022), $f(X)$ and $f(Y)$ are used for encoding both revealed part and masks. Afterwards, k_f concatenates $f(X)$ and $f(Y)$, then run the decoder $f^{-1}(\cdot)$ to recover the full object.
- For pure language models, we want to compute $\Pr(Y|X)$ where Y and X differ by the last token. k_f runs a linear function[2] and a softmax function to compute the probability distribution of the next token. Based on this distribution, k_f runs $f^{-1}(f(Y))$ to extract Y, which determines $\Pr(Y|X)$.

In Definition 2, the ideal foundation model is not unique. Theoretically, it is possible that the model f has the general intelligence, therefore can do everything even without learning the data in $\mathcal{C}$. In this case, we can make no meaningful arguments about the power of f. Therefore, we should look at the opposite side, and ask:

Is f guaranteed to have some property, if it is ideal with $\mathcal{C}$ defined by a pretext task?

This is the main motivation of our paper. Before moving forward, we would like to point out that Definition 2 does not rule out the possibility that f simply memorizes all possible morphisms. The intuition is, the pretext task is so

[1] Length restriction is not important as the category can contain infinitely many objects.

[2] This linear function can be predefined with random initialization, without affect the empirical performance of the model.

difficult, that even memorizing all the morphisms in $\mathcal{C}$ is already powerful enough. Moreover, a simple memorization function will not have an efficient implementation empirically (although category theory does not care about the implementation).

Below we first introduce an important notion called Yoneda embedding.

Definition 3 (Yoneda embedding $h_{\mathcal{C}}$)**.** *Given any $X \in \mathcal{C}$, $h_{\mathcal{C}}(X) \triangleq \mathrm{Hom}_{\mathcal{C}}(\cdot, X)$, which takes input $Y \in \mathcal{C}$ and outputs $\mathrm{Hom}_{\mathcal{C}}(Y, X)$.*

Define $\mathcal{C}^{\wedge}$ as the category of functors from $\mathcal{C}^{\mathrm{op}}$ to **Set**, we have $\mathrm{Hom}_{\mathcal{C}}(\cdot, X) \in \mathcal{C}^{\wedge}$ and $h_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}^{\wedge}$. Notice that we can define $\mathrm{Hom}_{\mathcal{C}}(X, \cdot)$ similarly, which defines another kind of Yoneda embedding.

How do we represent $\mathcal{C}^{\wedge}$ empirically? To answer this question, we should first learn arguably the most important theorem in category theory, as follows.

Lemma 2 (Yoneda lemma)**.** *For $A \in \mathcal{C}^{\wedge}$, and $X \in \mathcal{C}$, $\mathrm{Hom}_{\mathcal{C}^{\wedge}}(h_{\mathcal{C}}(X), A) \simeq A(X)$.*

In category theory, when two objects (or functors) are isomorphic, we treat them as equal. The technical details for making them equal to each other is omitted, as we observe that this is not a problem for modern networks empirically. For example, if there exists an isomorphism between two objects X and Y, so that $\phi(X) = Y$, empirically a deep neural network can easily fit this ϕ. More generally, techniques like rectified flow (Liu, 2022) can fit very complicated isomorphisms between random noise and images, using neural network with ODE.

Similar to the RKHS, we assume that $\mathcal{C}^{\wedge}$ is a space $\mathcal{H}$ equipped with a function $k_f : \mathcal{H} \times \mathcal{H} \to$ **Set**, such that $\mathrm{Hom}_{\mathcal{C}^{\wedge}}(h_{\mathcal{C}}(X), h_{\mathcal{C}}(Y)) \triangleq k_f(h_{\mathcal{C}}(X), h_{\mathcal{C}}(Y))$, where $f = h_{\mathcal{C}}$. Note that such $\mathcal{C}^{\wedge}$ always exists when $h_{\mathcal{C}}$ is given. For example, k_f can first compute $h_{\mathcal{C}}^{-1}(h_{\mathcal{C}}(X)) \simeq X$, and then send it to $h_{\mathcal{C}}(Y)$ to get $\mathrm{Hom}_{\mathcal{C}}(X, Y)$, which is isomorphic to $\mathrm{Hom}_{\mathcal{C}^{\wedge}}(h_{\mathcal{C}}(X), h_{\mathcal{C}}(Y))$ by Lemma 2.

Intuitively, $h_{\mathcal{C}}$ takes X and outputs all relationships of X, so we immediately have the following lemma.

Lemma 3. *$h_{\mathcal{C}}$ is ideal.*

The next lemma says, $h_{\mathcal{C}}$ is not only ideal, but with "the minimum power", up to symmetry[3].

Lemma 4. *Given any other ideal foundation model f, all the morphisms in $h_{\mathcal{C}}$ are also contained in f.*

Proof. Since f gets all the pretext tasks correctly, and k_f is data-oblivious, we know the morphism $\mathrm{Hom}_{\mathcal{C}}(X, Y)$ is stored in f for every $X, Y \in \mathcal{C}$. In other words, for every $h \in \mathrm{Hom}_{\mathcal{C}}(X, Y)$, either $h \in f(X)$ or $h \in f(Y)$. Notice that $h_{\mathcal{C}}$ contains exactly all the morphisms of $\mathrm{Hom}_{\mathcal{C}}(\cdot, \cdot)$, we proved our lemma. □

The above two lemmas tell us that we should work with $h_{\mathcal{C}}$, as it is the weakest ideal model.

5.3 Downstream tasks

After the model is trained, we apply it to the downstream tasks with two different approaches: prompt tuning and fine tuning. In this section, we investigate the power of both approaches, with different outcomes. We first define what we mean by solving a downstream task.

Definition 4 (Task)**.** *A task T is a functor in $\mathcal{C}^{\wedge}$.*

Definition 5 (Task solving)**.** *We say the model solves a task T, if for any input $X \in \mathcal{C}$, the model outputs a solution that is isomorphic to $T(X)$.*

Given a task defined as a functor $T \in \mathcal{C}^{\wedge}$, prompt tuning means we freeze the parameters of the model, and only use a task specific prompt P (usually in text or image), followed with the actual input of the task X, to get the output $T(X)$. Therefore, the prompt P and input X are the only two inputs to the model. By Lemma 2, if we directly send $h_{\mathcal{C}}(X)$ and T as a functor in $\mathcal{C}^{\wedge}$ to k_f, we have

$$T(X) \simeq \mathrm{Hom}_{\mathcal{C}^{\wedge}}(h_{\mathcal{C}}(X), T) \triangleq k_f(h_{\mathcal{C}}(X), T) \tag{1}$$

That is, k_f can accurately compute $T(X)$ with these two representations. However, notice that the prompt has to be sent through $h_{\mathcal{C}}$ before sending to k_f, which means we will send $h_{\mathcal{C}}(P)$ instead of T for the input of k_f. This brings up another important definition in category theory.

[3] We say $h_{\mathcal{C}}$ has the minimum power up to symmetry, because there are other foundation models that are equally powerful.

Definition 6 (Representable functor)**.** *A functor $T \in \mathcal{C}^\wedge$ is representable if there is an isomorphism $h_\mathcal{C}(X) \simeq T$ for some $X \in \mathcal{C}$. Such object X is called a representative of T.*

Based on this definition, we have the following characterization of prompt tuning.

Theorem 1 (Power on prompt tuning)**.** *$h_\mathcal{C}$ can solve T with prompt tuning, if and only if task $T \in \mathcal{C}^\wedge$ is representable. When T is representable, the optimal prompt is the representative of T.*

Proof. When the task T is representable, we can use its representative X as the prompt for T. By (1), we know $h_\mathcal{C}$ can solve T exactly.

On the other hand, if the task T is not representable, there exists no object $X \in \mathcal{C}$ such that $h_\mathcal{C}(X) \simeq T$. Therefore, assume we use task specific prompt P, and get the representation $h_\mathcal{C}(P)$. Since $h_\mathcal{C}(P) \not\simeq T$, we can always find an object $Y \in \mathcal{C}$ such that $h_\mathcal{C}(P)(Y) \not\simeq T(Y)$. In that case, our model using k_f and $h_\mathcal{C}$ cannot solve T on Y. □

Remark. There are some interesting results on continuous prompts (Liu et al., 2021; Li and Liang, 2021), which directly tunes the prompts in the feature space of the neural network, therefore the resulting prompt is not a representation of any real words/sentences. By doing this, it is possible to obtain more power than the representable tasks, but the enhancement depends on the expressive power of the feature space. Below we provide a simple application of Theorem 1.

Corollary 1. *For the pretext task of predicting image rotations (Gidaris et al., 2018), prompt tuning cannot solve complicated downstream tasks like segmentation or classification.*

Proof. The pretext task of predicting image rotations will rotate a given image $X \in \mathcal{C}$, by four different degrees: 0°, 90°, 180°, and 270°. Therefore, the four objects form a simple group of 4 elements. For any object $X \in \mathcal{C}$, we know $\mathrm{Hom}_\mathcal{C}(\cdot, X)$ contains exactly 4 morphisms. Clearly, tasks likes segmentation or classification are not representable by objects in $\mathcal{C}$. □

Remark. The statement of Corollary 1 is a bit counter-intuitive, because as presented in Gidaris et al. (2018), the model trained with the rotation pretext task indeed works for downstream tasks like classification and segmentation. However, our definition of solving a task in Definition 2 means the model should generates correct output for **every input**, so being partially accurate is not considered a success. This also matches with the goal of our paper: with infinite resource, can rotation pretext task used for complicated downstream tasks? Corollary 1 gives the negative answer.

Note that in Theorem 1 we consider the model with the minimum power, while empirically, the network structure may contain prior knowledge like the convolutional layers for some specific downstream tasks. However, these prior knowledge is inherently noisy and hard to characterize, therefore cannot be relied for completely solving the downstream tasks.

The power of prompt tuning is limited, and can be characterized by representable functors. What about fine tuning? In the case of fine tuning, we may use the Kan extension. However, the details of Kan extension is fairly abstract, so here we only provide one related theorem:

Lemma 5 (Yoneda Extension of Functors)**.** *Let $F : \mathcal{C} \to$ **Set** be any functor, then there exists an extension functor $h_\mathcal{C}^\dagger F : \mathcal{C}^\wedge \to$ **Set** such that $h_\mathcal{C}^\dagger F \circ h_\mathcal{C} \simeq F$.*

The actual Yoneda extension of functors theorem is more general (see e.g., Proposition 2.7.1 in Masaki Kashiwara (2006)), it can be applied to any category $\mathcal{A}$, but here we only use **Set** to avoid unnecessary technical details of inductive limits. Using Lemma 5, we immediately get the following theorem.

Theorem 2 (Power on fine tuning)**.** *Ideally, given enough resources, $h_\mathcal{C}$ can solve any task $F : \mathcal{C} \to$ **Set**.*

Proof. Applying Lemma 5, we know that with enough training data for the downstream task, computational power, and a fine tuning model h that learns $h_\mathcal{C}^\dagger F$ perfectly, we can concatenate h with f, to solve the task F. □

The downstream tasks considered in Theorem 2 are based on the structure in $\mathcal{C}$, not the data content in the dataset. As a result, the category defined by Gidaris et al. (2018) still has very simple group-structure, but with Theorem 2 it is possible to solve more diverse tasks. For example, we can map all the objects to the same output, which cannot be achieved with prompt tuning. Theorem 2 conveys the importance of pretext task, as more informative pretext tasks will create more structured category $\mathcal{C}$, which further improves the power of fine tuning on $\mathcal{C}^\wedge$.

Even with very informative $\mathcal{C}$, Theorem 2 only provides a raw upper bound with strong assumptions on the resources needed. By contrast, empirically the fine tuning step is usually done with a small network. Therefore, instead of treating it as a strong backup of what people are doing right now, it is more like exhibiting the future possibility. In other words, it implies that by transforming the objects to the corresponding feature space $\mathcal{C}^\wedge$, we have captured all the information encoded in $\mathcal{C}$.

For readers who are familiar with machine learning theory, Theorem 2 may look similar to the theory of over-parameterization at the first glance. However, they are analyzing different steps of self-supervised learning. Over-parameterization analyzes the pretraining step, saying that under certain assumptions, the optimization and generalization error will be very small for the pretext task, as long as the model is big enough and the learning rate is small enough. By contrast, Theorem 2 analyzes the fine tuning step after pretraining. Even if we have successfully pretrained a network on Imagenet with contrastive learning and zero generalization error, it remains unclear whether the model can be used for image segmentation as the downstream task, unless someone verifies it empirically. But Theorem 2 says, as long as the model is ideal, the feature layer contains all the information of $\mathcal{C}$ for any downstream tasks.

6 Multimodal Learning

In the previous section, we have seen how the foundation models are related to learning a category defined by a pretext task. What happens if we have multiple different categories? We can first use the pretext tasks to learn different foundation models separately, then connect these models together in the embedding space. This is exactly how multimodal models like CLIP Radford et al. (2021), Dall-E 2 Ramesh et al. (2022), Multilingual CLIP Carlsson et al. (2022) and AltCLIP Chen et al. (2022) work.

In this section, we analyze the functors between different categories. Similar to the previous section, we consider the case that the functors are perfectly learned, and investigate the implications under this assumption. This setting hides unnecessary details like how loss is defined or how network structure is designed. However, it provides interesting insights of how to connect different categories together, and what to expect after the functor connection.

As the starting point, we would like to emphasize that we assume the categories we consider are "natural", which means they are not only self consistent, but also consistent with each other. For example, when we use texts like "white, red, blue" or "big, small, tiny" to describe a chair in the language category, there are corresponding images in the image category. Such connection can be described by functors.

6.1 Generalization Theorem

We first assume that our functor from the embedding space learns the object mapping between the two categories perfectly, then show the structural information will be preserved with this functor. Notice that below we still use the notations of $h_\mathcal{B}, h_\mathcal{C}$ and $\mathcal{B}^\wedge, \mathcal{C}^\wedge$ to denote the foundation models and the corresponding embedding spaces, but our results hold for any other ideal foundation models and their embedding spaces as well.

Definition 7 (feature-aligned functor)**.** *Given two categories $\mathcal{B}, \mathcal{C}$, a full embedding $F : \mathcal{C} \to \mathcal{B}$, denote the corresponding foundation model as $h_\mathcal{B}, h_\mathcal{C}$. A function $\hat{F} : \mathcal{C}^\wedge \to \mathcal{B}^\wedge$ is feature-aligned with F if for any $X \in \mathrm{Ob}(\mathcal{C})$, $\hat{F}(h_\mathcal{C}(X)) \simeq h_\mathcal{B}(F(X))$.*

Theorem 3 (Generalization theorem for structural learning)**.** *Consider two categories $\mathcal{B}, \mathcal{C}$ and a full embedding $F : \mathcal{C} \to \mathcal{B}$. In the learning scenario, an ideal foundation model $h_\mathcal{C}$ for $\mathcal{C}$ together with a feature-aligned functor $\hat{F} : \mathcal{C}^\wedge \to \mathcal{B}^\wedge$, preserves the structure of $\mathcal{C}$ in a full subcategory $\mathcal{A}$ of $\mathcal{B}$: for any $X, Y \in \mathcal{C}$, $\mathrm{Hom}_\mathcal{C}(X, Y) \simeq \mathrm{Hom}_{\mathcal{A}^\wedge}(\hat{F}(h_\mathcal{C}(X)), \hat{F}(h_\mathcal{C}(Y)))$. Moreover, when $h_\mathcal{B}$ is available and invertible, we have $F(X) \simeq h_\mathcal{B}^{-1}(\hat{F}(h_\mathcal{C}(X)))$ for any $X \in \mathcal{C}$.*

Proof. By Lemma 1, there exists a full subcategory $\mathcal{A}$ of $\mathcal{B}$ with inclusion functor E, which is isomorphic to $\mathcal{C}$ with functor G, and $F = E \circ G$. Therefore, for any $X, Y \in \mathcal{C}$,

$$\begin{aligned}
&\mathrm{Hom}_\mathcal{C}(X, Y) \simeq \mathrm{Hom}_\mathcal{A}(GX, GY) = \mathrm{Hom}_\mathcal{B}(FX, FY)\\
=&\mathrm{Hom}_\mathcal{A}(FX, FY) \simeq \mathrm{Hom}_{\mathcal{A}^\wedge}(h_\mathcal{A}(FX), h_\mathcal{A}(FY))\\
=&\mathrm{Hom}_{\mathcal{A}^\wedge}(h_\mathcal{B}(FX), h_\mathcal{B}(FY))\\
\simeq&\mathrm{Hom}_{\mathcal{A}^\wedge}(\hat{F}(h_\mathcal{C}(X)), \hat{F}(h_\mathcal{C}(Y)))
\end{aligned}$$

The first equality holds as G is isomorphic. The second equality holds because F is full. The third equality holds as E maps the objects in $\mathcal{A}$ to the same objects in $\mathcal{B}$. The fourth equality uses the Yoneda lemma. The fifth equality holds because A is a full subcategory. The last equality holds since $\hat{F}$ is feature-aligned. If $h_{\mathcal{B}}$ is invertible, we have $F(X) = h_{\mathcal{B}}^{-1}(h_{\mathcal{B}}(F(X))) \simeq h_{\mathcal{B}}^{-1}(\hat{F}(h_{\mathcal{C}}(X)))$. □

Theorem 3 is much more powerful that it appears. We call it the generalization theorem, because it provides another kind of generalization, different from the existing generalization theory on stability or Rademacher complexity. It tells us that, the structural information of one category can be recovered in the feature space by the structural information of another category with a feature-aligned functor.

6.2 Application of Theorem 3

In this subsection, we apply Theorem 3 to analyze two models: CLIP and Dall-E 2.

CLIP. The dataset of CLIP contains millions of (image, text) pairs. During pretraining, for each batch of N pairs of data points, CLIP uses an image encoder and a text encoder to get N pairs of embeddings, and learns a function for matching the N pairs out of $N \times N$ possible connections. Since the network used in CLIP is invertible, Theorem 3 immediately gives us the following corollary.

Corollary 2 (Creativity of CLIP)**.** *Let $\mathcal{C}$ be a category that describes images, which is a subcategory of language category $\mathcal{C}'$. Let $\mathcal{B}$ be the image category. Assume CLIP learns a feature-aligned functor $\hat{F} : \mathcal{C}^\wedge \to \mathcal{B}^\wedge$ for a full embedding F, then CLIP is able to create new images that can be described in $\mathcal{C}$, but do not existed in the training set.*

Remark. Corollary 2 explains why CLIP can draw images like "avocado chair" that did not exist previously. The full embedding assumption is crucial, otherwise the embedding computed in $\mathcal{A}^\wedge$ cannot be used for generating images in $\mathcal{B}$.

Dall-E 2. Dall-E 2 is the combination of Clip and the diffusion model (Rombach et al., 2022; Sohl-Dickstein et al., 2015; Dhariwal and Nichol, 2021; Ho et al., 2020). If Clip can be seen as the feature-aligned functor between two isomorphic subcategories, why do we need an extra diffusion model? This is because these two categories are not purely isomorphic. When we type "a photo of dog", there exists millions of different matching images. Therefore, Dall-E2 modifies the definition of image category, so that each object is a probability distribution of images. From this perspective, the diffusion model is a generator of images, while Clip learns the functor from the category of texts to the category of probability distributions of images.

6.3 Compositional Theorem

By applying Theorem 3 multiples times through a list of categories, we immediately get the following theorem.

Theorem 4 (Compositional Theorem)**.** *Consider a list of categories $\{\mathcal{B}_i\}_{i=1}^n$, and $n-1$ full embeddings $\{F_i\}_{i=1}^{n-1}$, where $F_i : \mathcal{A}_i \to \mathcal{B}_{i+1}$, where $\mathcal{A}_1 = \mathcal{B}_1$, and for $i > 1$, $\mathcal{A}_i$ is the full subcategory of B_i induces by F_{i-1}. Denote the foundation model for $\mathcal{B}_i$ as $h_{\mathcal{B}_i}$, and feature-aligned functors as $\{\hat{F}_i\}_{i=1}^{n-1}$. Denote $F \triangleq F_{n-1}F_{n-2}\cdots F_1$ and $\hat{F} \triangleq \hat{F}_{n-1}\hat{F}_{n-2}\cdots\hat{F}_1$. By composing all the $\hat{F}_i$s together, we preserve the structure of $\mathcal{B}_1$ in $\mathcal{A}_n$: for any $X, Y \in \mathcal{C}$, $\mathrm{Hom}_{\mathcal{B}_1}(X,Y) = \mathrm{Hom}_{\mathcal{A}_n^\wedge}(\hat{F}(h_{\mathcal{B}_1}(X)), \hat{F}(h_{\mathcal{B}_1}(Y)))$. Moreover, when $h_{\mathcal{B}_n}$ is invertible, we have $F(X) = h_{\mathcal{B}_n}^{-1}(\hat{F}(h_{\mathcal{C}}(X)))$ for any $X \in \mathcal{C}$.*

If we want to map objects from $\mathcal{B}_1$ and $\mathcal{B}_n$, but the training data between the two categories is limited, Theorem 4 provides an alternative way. It suffices to find a path between the two categories, and learn the functors for each edge of the path. This is especially useful when some of the categories are pretrained (like GPT or CLIP). For example, Carlsson et al. (2022) and Chen et al. (2022) use multilingual text pairs to train the functors between various language categories and the English category, which are further connected to the image category using CLIP. By doing that, they easily extend CLIP to the multilingual settings.

7 Discussion

7.1 Applying to small categories

In our abstract and introduction, we raised the question about an infinitely large model with infinite resources, which makes our theorem look unrealistic. However, these assumptions are purely rhetoric, and our real assumption is simply the model being ideal, i.e., the model perfectly solves the pretext task. This assumption is much weaker than the infinite one, because it means our theorems can be directly applied to small datasets, which correspond to smaller categories. For instance, consider a dataset containing only 100k sentences, and a model is trained to learn the data using a language model where each sentence is connected to its neighboring sentences with certain probabilities. While the sentences or words in the corresponding category C may not be as informative as the language category that humans possess, it is well-defined, and the ideal model is also well-defined. Training an ideal model for a small dataset of this scale is feasible. According to our Theorem 1, such model can only perfectly solve tasks that are representable in $\mathcal{C}$.

7.2 Modeling probabilities

In this subsection, we provide the full definition of the category of language models. In our new category $\mathcal{C}$, every object $X \in \mathcal{C}$ is a probability measure μ on the space of sentences $\mathbb{S}$ of length at most N. when μ concentrates at a single point $s \in \mathbb{S}$, it means we have a deterministic sentence s. Otherwise, μ represents a probability distribution over all possible sentences.

Given any sentence $s_1 = (w_1, \cdots, w_N)$, we will define a probability distribution ν on the next token w_{N+1}, so that we can define an object Y with the same probability measure on the sentences of concatenating $(w_2, \cdots, w_N)$ to w_{N+1}. By defining X as the object contains single sentence s_1, we have a morphism from X to Y. We call Y the canonical successor of X, and its probability distribution (on length N sentences) the canonical successive measure of X.

Given any object $Z \in \mathcal{C}$, assume Z has nonzero probabilities on various objects, like $X_1, \cdots, X_n$, with probability measure μ. We define canonical successive distribution of X_i as ν_i. In this case, we define its canonical successive distribution μ' as:

$$\mu'(s) = \int_{X_i} \int_s d\nu_i(s) d\mu(X_i)$$

Intuitively, if s_1 is the first sentence of a paragraph, we may use an object with measure concentrated at s_1 to represent it. However, due to the nature of natural language (not the dataset noise), the next word of s_1 is not deterministic. So we define s_2 that drops the first word and adds the next word of s_1, which should be represented as a probability distribution (canonical successive distribution). What about the next word of s_2? Without the realization of s_1, we have to enumerate all the possibilities of s_1, and use the canonical successive distribution of each of them to define s_2. By doing so, we have defined a category with probabilities, where the pretext task defines its morphisms.

7.3 Bayesian optimal classifier

The notion of Bayesian optimal classifier is widely used in the generalization theory for supervised learning. It simply means that due to the inherent noise in the labels, even the best model will not be able to get zero loss in the population distribution. For example, given a blurry image of a dog, it might be difficult to tell its exact breed.

However, in our categorical framework, we take the physicist's view of the world, and assume that **all the objects and relationships are self consistent**, without any noises. For example, Alaskan Husky and Siberian Husky indeed look similar from their photos, but they are fundamentally different breeds, in terms of their origins, size, appearance, temperament and so on. Our framework models these conceptual differences, and treat the photos of the dogs (which have noise) as the outcome of the last step of diffusion as discussed in Section 6. If the reader is familiar with the functional programming, the categorical framework can be seen as the "pure functional" part, which is predictable and precise. By contrast, the process that deals with the actual input and output, can be seen as the "side effects", where the notion of Bayesian optimal classifier comes in.

7.4 Relationship to RKHS

Our categorical framework can be seen as a natural generalization to the RKHS framework, where in RKHS a kernel function outputs a value in $\mathbb{R}$, and our framework generalizes it to **Set**.

For example, applying the Yoneda Lemma on RKHS, we get the property: $\forall x \in \mathcal{C}, \forall f \in \mathcal{H}, \langle k(\cdot, x), f\rangle_{\mathcal{H}} = f(x)$, where f can be seen as a functor in $\mathcal{C}^\wedge$, $k(\cdot, x)$ is a representable functor, and $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ computes $\mathrm{Hom}_{\mathcal{C}}$. Therefore in RKHS, the morphisms between two objects are represented by a single real number computed by $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. This case works perfectly for algorithms like SimCLR where the relationship between two objects is a real number representing the similarity. However, in practice, the relationship between two objects can be much more than a real number, especially for NLP tasks. In that case, RKHS is insufficient, and the general Yoneda lemma is necessary.

Acknowledgements

This paper was greatly inspired by the fruitful interactions with the Qianfang (functor) team at the Beijing Academy of Artificial Intelligence. Special thanks are extended to Yue Cao and anonymous reviewers for their helpful discussions and suggestions. This paper is supported by the Ministry of Science and Technology of the People's Republic of China, the 2030 Innovation Megaprojects "Program on New Generation Artificial Intelligence" (Grant No. 2021AAA0150000).

References

Adámek, J., Herrlich, H., and Strecker, G. (1990). *Abstract and concrete categories*. Wiley-Interscience.

Allen-Zhu, Z., Li, Y., and Liang, Y. (2019a). Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32.

Allen-Zhu, Z., Li, Y., and Song, Z. (2019b). A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR.

Arandjelovic, R. and Zisserman, A. (2018). Objects that sound. In *Proceedings of the European conference on computer vision (ECCV)*, pages 435–451.

Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. (2019a). Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR.

Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., and Saunshi, N. (2019b). A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. (2017). Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30.

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. (2020). Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070.

Bradley, T.-D., Terilla, J., and Vlassopoulos, Y. (2022). An enriched category theory of language: from syntax to semantics. *La Matematica*, pages 1–30.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Cai, Q., Yang, Z., Jin, C., and Wang, Z. (2020). Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pages 1283–1294. PMLR.

Carlsson, F., Eisen, P., Rekathati, F., and Sahlgren, M. (2022). Cross-lingual and multilingual clip. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6848–6854.

Censi, A. (2015). A mathematical theory of co-design. *arXiv preprint arXiv:1512.08055*.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Chen, X. and He, K. (2021). Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758.

Chen, Z., Liu, G., Zhang, B.-W., Ye, F., Yang, Q., and Wu, L. (2022). Altclip: Altering the language encoder in clip for extended language capabilities. *arXiv preprint arXiv:2211.06679*.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794.

Doersch, C., Gupta, A., and Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Du, S., Krishnamurthy, A., Jiang, N., Agarwal, A., Dudik, M., and Langford, J. (2019a). Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pages 1665–1674. PMLR.

Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. (2019b). Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR.

Du, S. S., Kakade, S. M., Wang, R., and Yang, L. F. (2019c). Is a good representation sufficient for sample efficient reinforcement learning? *arXiv preprint arXiv:1910.03016*.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. (2018). Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*.

Fong, B. and Spivak, D. I. (2018). Seven sketches in compositionality: An invitation to applied category theory. *arXiv preprint arXiv:1803.05316*.

Gao, T., Fisch, A., and Chen, D. (2020). Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284.

HaoChen, J. Z., Wei, C., Gaidon, A., and Ma, T. (2021). Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34:5000–5011.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Hou, Z., Sun, F., Chen, Y.-K., Xie, Y., and Kung, S.-Y. (2022). Milan: Masked image pretraining on language assisted representation.

Jiang, Z., Xu, F. F., Araki, J., and Neubig, G. (2020). How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. (2018). Is q-learning provably efficient? *Advances in neural information processing systems*, 31.

Kuś, M. and Skowron, B. (2019). *Category Theory in Physics, Mathematics, and Philosophy*. Springer.

Lee, J. D., Lei, Q., Saunshi, N., and Zhuo, J. (2021). Predicting what you already know helps: Provable self-supervised learning. *Advances in Neural Information Processing Systems*, 34:309–323.

Li, X. L. and Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Li, Y. and Yuan, Y. (2017). Convergence analysis of two-layer neural networks with relu activation. *Advances in neural information processing systems*, 30.

Liu, Q. (2022). Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*.

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., and Tang, J. (2021). Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Luo, Z., Weng, C., Wu, S., Zhou, M., and Ge, R. (2022). One objective for all models–self-supervised learning for topic models. *arXiv preprint arXiv:2203.03539*.

Mac Lane, S. (2013). *Categories for the working mathematician*, volume 5. Springer Science & Business Media.

Mahadevan, S. (2022a). Categoroids: Universal conditional independence. *arXiv preprint arXiv:2208.11077*.

Mahadevan, S. (2022b). Unifying causal inference and reinforcement learning using higher-order category theory. *arXiv preprint arXiv:2209.06262*.

Marquis, J.-P. (2008). *From a geometrical point of view: A study of the history and philosophy of category theory*, volume 14. Springer Science & Business Media.

Masaki Kashiwara, P. S. (2006). *Categories and Sheaves*. Springer.

Noroozi, M. and Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Pathak, D., Girshick, R., Dollár, P., Darrell, T., and Hariharan, B. (2017). Learning features by watching objects move. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2701–2710.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training. *OpenAI blog*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Sohl-Dickstein, J. (2017). On the expressive power of deep neural networks. In *international conference on machine learning*, pages 2847–2854. PMLR.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR.

Riehl, E. (2017). *Category theory in context*. Courier Dover Publications.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shiebler, D., Gavranović, B., and Wilson, P. (2021). Category theory in machine learning. *arXiv preprint arXiv:2106.07032*.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., and Singh, S. (2020). Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR.

Tan, Z., Zhang, Y., Yang, J., and Yuan, Y. (2023). Contrastive learning is spectral clustering on similarity graph.

Tosh, C., Krishnamurthy, A., and Hsu, D. (2021). Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pages 1179–1206. PMLR.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wen, Z. and Li, Y. (2021). Toward understanding the feature learning process of self-supervised contrastive learning. In *International Conference on Machine Learning*, pages 11112–11122. PMLR.

Wen, Z. and Li, Y. (2022). The mechanism of prediction head in non-contrastive self-supervised learning. *arXiv preprint arXiv:2205.06226*.

Yin, D., Kannan, R., and Bartlett, P. (2019). Rademacher complexity for adversarially robust generalization. In *International conference on machine learning*, pages 7085–7094. PMLR.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12310–12320. PMLR.

Zimmermann, R. S., Sharma, Y., Schneider, S., Bethge, M., and Brendel, W. (2021). Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, pages 12979–12990. PMLR.

Zou, D., Cao, Y., Zhou, D., and Gu, Q. (2020). Gradient descent optimizes over-parameterized deep relu networks. *Machine learning*, 109(3):467–492.