Active Self-Supervised Learning: A Few Low-Cost Relationships Are All You Need

Vivien Cabannes* Meta AI
Leon Bottou Meta AI
Yann Lecun Meta AI
Randall Balestriero* Meta AI

Abstract

Self-Supervised Learning (SSL) has emerged as the solution of choice to learn transferable representations from unlabeled data. However, SSL requires to build samples that are known to be semantically akin, i.e. positive views. Requiring such knowledge is the main limitation of SSL and is often tackled by ad-hoc strategies e.g. applying known data-augmentations to the same input. In this work, we formalize and generalize this principle through Positive Active Learning (PAL) where an oracle queries semantic relationships between samples. PAL achieves three main objectives. First, it unveils a theoretically grounded learning framework beyond SSL, based on similarity graphs, that can be extended to tackle supervised and semi-supervised learning depending on the employed oracle. Second, it provides a consistent algorithm to embed a priori knowledge, e.g. some observed labels, into any SSL losses without any change in the training pipeline. Third, it provides a proper active learning framework yielding low-cost solutions to annotate datasets, arguably bringing the gap between theory and practice of active learning that is based on simple-to-answer-by-non-experts queries of semantic relationships between inputs.

1. Introduction

Learning representations of data that can be used to solve multiple tasks, out-of-the-box, and with minimal post-processing is one of the main goals of current AI research [39, 35, 24]. Such representations are generally found by processing given inputs through Deep neural Networks (DNs). The main question of interest around which contemporary research focuses on deals with the choice of the training setting that is employed to tune the DN's parameters. A few different strategies have emerged such as layerwise [6], reconstruction based [48], and more recently, based on Self-Supervised Learning (SSL) [14, 37]. In fact, due to the cost of labeling and the size of datasets constantly growing, recent methods have tried to drift away from traditional supervised learning [42]. From existing training solutions, joint-embedding SSL has emerged as one of the most promising ones [34]. It consists in learning representations that are invariant along some known transformations while preventing dimensional collapse of the representation. Such invariance is enforced by applying some known Data-Augmentation (DA), e.g. translations for images, to the same input and making sure that their corresponding representations are the same.

Despite tremendous progress, several limitations remain in the way of a widespread deployment of SSL. In particular, it is not clear how to incorporate *a priori* knowledge into SSL frameworks beyond the usual tweaking of the loss and DAs being employed, although some efforts are being made [16, 53, 52]. Indeed, it is not surprising that vision-language pre-training has replaced SSL as the state-of-the-art to learn image representation [26], as those models are better suited to incorporate information stemming from captions that often come alongside images collected on the Internet.

In this study, we propose to redefine existing SSL losses in terms of a similarity graph –where nodes represent data samples and edges reflect known inter-sample relationships. Our first contribution stemming from this formulation provides a *generic framework to think about learning in terms of similarity graph*: it yields a spectrum on which SSL and supervised learning can be seen as two extremes. Within this realm, those two extremes are connected through the similarity matrix, and in fact can be made equivalent by varying the similarity graph. Our second contribution naturally emerges from using such a similarity graph, unveiling an elegant framework to reduce the cost and expert requirement of active learning summarized by:

Tell me who your friends are,
and I will tell you who you are.

Active learning, which aims to reduce supervised learning cost by only asking an oracle for sample labels when needed [41, 20, 27, 31], can now be formulated in term of relative sample comparison, rather than absolute sample labeling.

*Equal contribution.

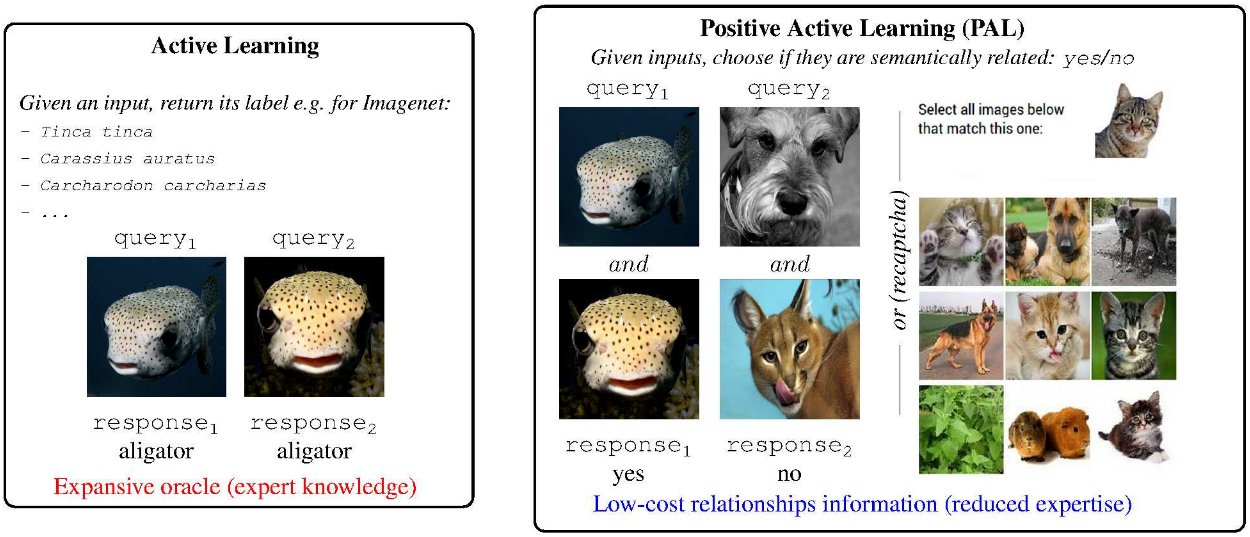

Figure 1. Active Self-Supervised Learning introduces PAL (**right box**), an alternative to active learning (**left box**) where the oracle is asked if a collection of inputs are semantically related or not. As opposed to active learning, expert knowledge is reduced as one need not to know all the possible classes but only how to distinguish inputs from different classes. PAL querying is flexible, as an illustrative example we exhibit an à la captcha version where a given input is presented along with a collection of other inputs, and the oracle can select among those inputs the positive ones.

This *much more efficient and low-cost approach is exactly the active learning strategy stemming from our framework*: rather than asking for labels, one rather asks if two (or more) inputs belong to the same classes or not, as depicted in Fig. 1. We coin such a strategy as *Positive Active Learning (PAL)*, and we will present some key analysis on the benefits of PAL over traditional active learning. We summarize our contributions below:

- We provide a *unified learning framework* based on the concept of similarity graph, which encompasses both self-supervised learning, supervised learning, as well as semi-supervised learning and many variants.
- We derive a *generic PAL algorithm* based on an oracle to query the underlying similarity graph Algorithm 1. The different learning frameworks (SSL, supervised, and so forth) are recovered by different oracles, who can be combined to benefit from each framework distinction.
- We show how PAL extends into an *active learning framework* based on similarity queries that provides low-cost efficient strategies to annotate a dataset (Fig. 1).

All statements of this study are proven in Appendix B, code to reproduce experiments is provided at `https://github.com/VivienCabannes/rates`.

2. Background on Self-Supervised Learning

This section provides a brief reminder of the main self-supervised learning (SSL) methods, their associated losses, and common notations for the remainder of the study.

A common strategy to learn a model in machine learning is to curate labeled examples $(\boldsymbol{x}_n, y_n)_n$, and to learn a model that given $\boldsymbol{x}_n \in \mathcal{X} \triangleq \mathbb{R}^D$ as input, outputs $y_n \in [C]$, hoping that this model will learn to recognize patterns and relations that generalizes to new, unseen input data. Yet, as the dataset grew larger, and annotating data has become a major bottleneck, machine learning has shifted its attention to learning methods that do not require knowledge of y_n. SSL has emerged as a powerful solution to circumvent the need for expensive and time-consuming labeling. It learns a embedding $f : \mathcal{X} \to \mathbb{R}^K$ for a small K by enforcing either reconstruction properties, or some invariance and symmetry onto a learned representation. SSL also relies on a set of observations $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N \in \mathbb{R}^{N\times D}$, yet instead of labels y_n, it requires known *pairwise positive relation* that indicates whether two samples are semantically similar or not. For simplicity, we shall focus on the joint-embedding framework, where those positive pairs are artificially generated on the fly by applying Data Augmentations (DA), e.g. adding white noise, masking, on the same input. Let denote $\mathcal{T}_1, \mathcal{T}_2 : \mathcal{X} \to \mathcal{X}$ the generators of two (random) DAs $\mathcal{T}_1(\boldsymbol{x})$ and $\mathcal{T}_2(\boldsymbol{x})$ from an input $\boldsymbol{x}$, $f_\theta : \mathbb{R}^D \to \mathbb{R}^K$ the parametric

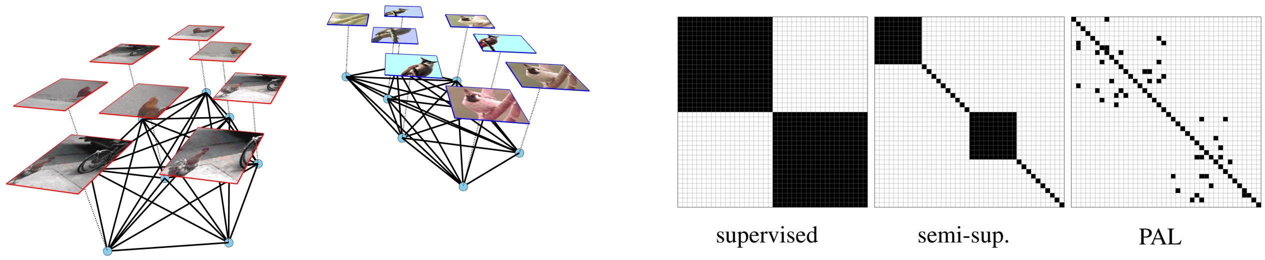

Figure 2. **Left:** Depiction of the "knowledge graph" arising from binary classification with supervised learning. Notice the two connected components, each corresponding to a single class. Each sample is associated with a node of the graph (blue circle) and the known positive relation between samples is represented by an edge. This knowledge is summarized into the $\boldsymbol{G}$ matrix depicted on the right. **Right:** Examples of the $N \times N$ symmetric graph-adjacency matrices $\boldsymbol{G}$ for the case of binary classification with supervised (same graph as on the left), semi-supervised and active learning. Each nonzero entry $(\boldsymbol{G})_{i,j}$ represents the known positive relation between sample i and j.

model to be learned, and

$$\boldsymbol{Z}^{(1)} \triangleq \begin{bmatrix} f_\theta(\mathcal{T}_1(\boldsymbol{x}_1)) \\ \vdots \\ f_\theta(\mathcal{T}_1(\boldsymbol{x}_N)) \end{bmatrix}, \boldsymbol{Z}^{(2)} \triangleq \begin{bmatrix} (f_\theta(\mathcal{T}_2(\boldsymbol{x}_1)) \\ \vdots \\ (f_\theta(\mathcal{T}_2(\boldsymbol{x}_N)) \end{bmatrix}, \quad (1)$$

where $(\boldsymbol{z}_n^{(1)}, \boldsymbol{z}_n^{(2)})$, the n^{th} row of $\boldsymbol{Z}^{(1)}$ and $\boldsymbol{Z}^{(2)}$ respectively, form the n^{th} positive pair associated to sample $\boldsymbol{x}_n$. Using (1), different SSL losses will employ different measures of invariance and dimensional collapse. Typically, the losses are minimized with gradient descent and backpropagation to learn θ.

VICReg. With the above notations, the VICReg loss [3] reads, with hyper-parameter $\alpha, \beta > 0$,

$$\mathcal{L}_{\text{VIC}} = \alpha \sum_{k=1}^{K} \text{ReLU}\left(1 - \sqrt{\boldsymbol{C}_{k,k}}\right) + \beta \sum_{k \neq l} \boldsymbol{C}_{k,l}^2 + \frac{1}{N} \|\boldsymbol{Z}^{(1)} - \boldsymbol{Z}^{(2)}\|_2^2, \qquad \boldsymbol{C} \triangleq \text{Cov}(\begin{bmatrix} \boldsymbol{Z}^{(1)} \\ \boldsymbol{Z}^{(2)} \end{bmatrix}). \quad (2)$$

SimCLR. The SimCLR loss [14] with temperature hyper-parameter $\tau > 0$ reads

$$\mathcal{L}_{\text{Sim}} = -\sum_{i=1}^{N} \frac{\boldsymbol{C}_{i,i}}{\tau} + \log\left(\sum_{i \neq j}^{N} \exp\left(\frac{\boldsymbol{C}_{i,j}}{\tau}\right)\right), \qquad \boldsymbol{C}_{i,j} \triangleq \text{CoSim}(\boldsymbol{Z}^{(1)}, \boldsymbol{Z}^{(2)})_{ij} \triangleq \frac{\left\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(2)} \right\rangle}{\|\boldsymbol{z}_i^{(1)}\| \, \|\boldsymbol{z}_j^{(2)}\|}, \quad (3)$$

BarlowTwins. BarlowTwins [51] is built on the cross-correlation matrix $\boldsymbol{C}_{ij} = \text{CoSim}(\boldsymbol{Z}^{(1)\top}, \boldsymbol{Z}^{(2)\top})$, with the hyper-parameter λ it reads

$$\mathcal{L}_{\text{BT}} = \sum_{k=1}^{K} (1 - \boldsymbol{C}_{ii})^2 + \lambda \sum_{i \neq j} \boldsymbol{C}_{ij}^2. \quad (4)$$

Spectral Contrastive Loss. Finally, the spectral contrastive loss [28] is theory-friendly proxy for SSL reading

$$\mathcal{L}_{\text{VIC}^2} = -2\left\langle \boldsymbol{Z}^{(1)}, \boldsymbol{Z}^{(2)} \right\rangle + \frac{1}{N} \sum_{i \neq j} \left\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(2)} \right\rangle^2. \quad (5)$$

In particular, as proven in Appendix B.1.1, (5) recovers VICReg (2) when the ReLU-hinge loss is replaced by the mean-square error, hence the denomination VIC^2.

The Commonality between SSL Losses. All the above Eqs. (2) to (5) losses combine two terms: (i) a matching term between positive pairs, and (ii) a term to avoid collapse towards predicting a constant solution for all inputs. (i) can take different forms such as the squared norm between $\boldsymbol{Z}^{(1)}$ and $\boldsymbol{Z}^{(2)}$ (2), the opposite of their scalar product (5), or of their cosine (3), or the square norm between the centered-cosine and one (4). (ii) can also take different forms such as the infoNCE softmax (5), or an energy that enforces richness of the learn feature, such as the variance-covariance regularization in (2) forcing the different coordinates of f_θ to be orthogonal [12].

While at face-value those losses seem distinct, they actually all simply consist and combine some variants of (i) and (ii), and even more importantly, they all rely on the exact same information of positive inter-sample relation for (i). This is exactly what the next Section 3 will dive into, as a means to unify SSL losses, along with supervised learning methods.

3. The Ubiquity of Similarity Graphs

The goal of this section is to unify SSL and supervised learning through the introduction of a special object: a *similarity graph.*

3.1. The Graphs for (Self-)Supervised Learning

Throughout this study, a similarity graph denotes a graph for which nodes represent data samples, and edges reflect similarity relationships. Formally, such a graph is expressed through its symmetric adjacency matrix $\boldsymbol{G} \in \mathbb{R}^{N \times N}$, the semantic relation between inputs i and j being encoded in the real entry $\boldsymbol{G}_{i,j}$. The remainder of this section will demonstrate how (i) SSL losses are implicitly based on a similarity graph (ii) how those losses tackle the supervised learning problem when provided a richer graph $\boldsymbol{G}$.

Supervised learning. In addition to the N input samples $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, supervised learning has access to paired labels $\boldsymbol{y} \triangleq [y_1, \ldots, y_N]$. For clarity, we focus here on categorical labels i.e. y_n belongs to $\{1, \ldots, C\}$ for C the number of classes.[1] The one-hot encoding of $\boldsymbol{y}$ will be denoted by the matrix $\boldsymbol{Y} \in \mathbb{R}^{N \times C}$. In terms of the similarity graph $\boldsymbol{G}$, the label-based relation becomes naturally encoded as $\boldsymbol{G}_{i,j} = \mathbf{1}_{\{y_i \neq y_j\}}$, or equivalently

$$\boldsymbol{G}(\boldsymbol{Y}) = \boldsymbol{Y}\boldsymbol{Y}^{\top} \tag{6}$$

A key observation that we must emphasize is that the graph $\boldsymbol{G}$ almost explicitly encodes for the labels $\boldsymbol{Y}$, which will be made explicit with Theorem 2.

Multiple Epochs with Data Augmentation. When DA is employed, and training is carried for E epochs, the original N input samples are transformed into $N \times E$ "augmented" samples. For more generality, since DA will also be used in SSL, let's denote by $A \in \mathbb{N}^*$ the number of augmentations –where here $A = E$. We now have the augmented dataset

$$\boldsymbol{X}^{(A)} \triangleq [\underbrace{\mathcal{T}(\boldsymbol{x}_1), \ldots, \mathcal{T}(\boldsymbol{x}_1)}_{\text{repeated } A \text{ times}}, \ldots, \mathcal{T}(\boldsymbol{x}_N), \ldots, \mathcal{T}(\boldsymbol{x}_N)]^{\top},$$

where each $\mathcal{T}$ has its own randomness. When available, i.e. for supervised learning, the corresponding "augmented" labels $\boldsymbol{Y}^{(A)}$ are given by repeating A times each row of $\boldsymbol{Y}$, formally written with the Kronecker product $\boldsymbol{Y}^{(\text{sup})} \triangleq \boldsymbol{Y} \otimes \mathbf{1}_A$, and from that, we can now define the supervised dataset and associated graph extending (6) to the case of multiple epochs and DA training

$$\boldsymbol{X}^{(\text{sup})} \triangleq \boldsymbol{X}^{(E)}, \qquad \boldsymbol{G}^{(\text{sup})} \triangleq \boldsymbol{Y}^{(\text{sup})\top}\boldsymbol{Y}^{(\text{sup})}. \tag{7}$$

The resulting graph (7) is depicted on the left part of Fig. 2.

Self-Supervised Learning. SSL does not rely on labels, but on positive pairs/tuples/views generated at each epoch. Let us denote by V the number of positive views generated, commonly $V = 2$ for positive pairs as modeled in (1). With E the total number of epochs, SSL produces $V \times E$ samples semantically related to each original sample $\boldsymbol{x}_n$ through the course of training *i.e.* in SSL $A = V \times E$ while in supervised learning $A = E$. The total number of samples is thus $N \times E \times V$, defining the dataset and associated graph

$$\boldsymbol{X}^{(\text{ssl})} \triangleq \boldsymbol{X}^{(V \times E)}, \ \boldsymbol{G}^{(\text{ssl})}_{i,j} = \mathbf{1}_{\{\lfloor i/VE \rfloor = \lfloor j/VE \rfloor\}}, \tag{8}$$

where the associated similarity graph $\boldsymbol{G}^{(\text{ssl})}$ –now of size $NEV \times NEV$– captures if two samples were generated as DA of the same original input.

[1] While we focus here on classification for simplicity, our approach is easily extendable for generic problems involving a loss ℓ by defining the graph as $\boldsymbol{G}_{ij} = -\ell(y_i, y_j)$. In the classification, ℓ could be the zero-one loss $\ell(y_i, y_j) = \mathbf{1}_{\{y_i \neq y_j\}}$, and $\boldsymbol{G}_{ij} \simeq 1 - \ell(y_i, y_j)$. See Appendix B.2.1 for details.

3.2. Self-Supervised Learning on a Graph

This section reformulates the different SSL losses through the sole usage of the similarity graph $\boldsymbol{G}^{(\text{ssl})}$. To lighten notations, and without loss of generality, we *redefine* $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ to denote the full dataset, i.e. $N \leftarrow NEV$ with $\boldsymbol{X} = \boldsymbol{X}^{(\text{sup})}$ for supervised learning with $V \times E$ epochs, or with $\boldsymbol{X} = \boldsymbol{X}^{(\text{ssl})}$ in SSL with E epochs with V views for the SSL case. The model embedding is shortened to $\boldsymbol{Z} \triangleq f_\theta(\boldsymbol{X}) \in \mathbb{R}^{N \times K}$ as per Eq. (1).

Theorem 1. *VICReg (2), SimCLR (3), and BarlowTwins (4) losses can be expressed in term of the graph $\boldsymbol{G}$ (8)*

$$\mathcal{L}_{\text{VIC}^2}(\boldsymbol{Z}; \boldsymbol{G}) = \| \boldsymbol{Z}\boldsymbol{Z}^T - \boldsymbol{G}\|_F^2,$$

$$\mathcal{L}_{\text{Sim}}(\boldsymbol{Z}; \boldsymbol{G}) = -\sum_{i,j \in [N]} \boldsymbol{G}_{i,j} \log \left(\frac{\exp(\tilde{\boldsymbol{z}}_i^{\top} \tilde{\boldsymbol{z}}_j)}{\sum_{k \in [N]} \exp(\tilde{\boldsymbol{z}}_i^{\top} \tilde{\boldsymbol{z}}_k)} \right),$$

$$\mathcal{L}_{\text{BT}}(\boldsymbol{Z}; \boldsymbol{G}) = \left\| \tilde{\boldsymbol{Z}}^{\top} \boldsymbol{G} \tilde{\boldsymbol{Z}} - I \right\|^2,$$

where $\boldsymbol{D} = \operatorname{diag}(\boldsymbol{G}\mathbf{1})$ is the degree matrix of $\boldsymbol{G}$; with $\tilde{\boldsymbol{z}} \triangleq \boldsymbol{z}/\|\boldsymbol{z}\|$ and $\tilde{\boldsymbol{Z}}$ the column normalized $\boldsymbol{Z}$ so that each column has unit norm.

In essence, SSL is about making sure that sample's representations match for samples that were deemed similar through the design of data augmentation. As such, it is not surprising that one can rewrite SSL losses through the sole usage of the similarity graph. From Theorem 1, the attentive observer would notice how VICReg is akin to Laplacian Eigenmaps or multidimensional scaling, SimCLR is akin to Cross-entropy and BarlowTwins is akin to Canonical Correlation Analysis; observations already discovered in the literature [2] and reinforced above.

Beyond recovering such representation learning losses, our goal is to go one step further and to tie SSL and supervised learning through the lens of $\boldsymbol{G}$, which follows in the next section.

3.3. Self-Supervised is a G Away from Supervised

What happens if one takes the different SSL losses, but replaces the usual SSL graph $\boldsymbol{G}^{(\text{ssl})}$ with the supervised one $\boldsymbol{G}^{(\text{sup})}$?

It turns out that the learned representations emerging from such losses are identical –up to some negligible symmetries that can be corrected for when learning a linear probe– to the one hot-encoding of $\boldsymbol{Y}$. To make our formal statement (Theorem 2) clearer, we introduce the set of optimal representations that minimize a given loss:

$$\mathcal{S}_{\text{method}}(\boldsymbol{G}) \triangleq \underset{\boldsymbol{Z}\in\mathbb{R}^{N\times K}}{\arg\min}\, \mathcal{L}_{\text{method}}(\boldsymbol{Z};\boldsymbol{G}),$$

where "method" refers to the different losses.

Theorem 2 (Interpolation optimum). *When $K \geq C$, and $\boldsymbol{Z} = f_\theta(\boldsymbol{X})$ is unconstrained (e.g. interpolation regime with a rich functions class), the SSL losses as per Theorem 1 with the supervised graph (7) solve the supervised learning problem with:*

$$\mathcal{S}_{\text{VIC}^2}(\boldsymbol{G}^{(\text{sup})}) = \left\{\boldsymbol{Y}\boldsymbol{R} \,\middle|\, \boldsymbol{R}\in\mathbb{R}^{C\times K}; \boldsymbol{R}\boldsymbol{R}^\top = \boldsymbol{I}_C\right\},$$
$$\mathcal{S}_{\text{Sim}}(\boldsymbol{G}^{(\text{sup})}) = \left\{\boldsymbol{D}\boldsymbol{Y}\boldsymbol{R}\boldsymbol{M}^{-1} \,\middle|\, \boldsymbol{D}\in\operatorname{diag}_+, \boldsymbol{R}\in O\right\},$$
$$\mathcal{S}_{\text{BT}}(\boldsymbol{G}^{(\text{sup})}) = \left\{\boldsymbol{Y}\boldsymbol{R}\boldsymbol{D} \,\middle|\, \boldsymbol{D}\in\operatorname{diag}_+, \boldsymbol{R}\in O\right\},$$

where $\boldsymbol{R} \in O$ means that $\boldsymbol{R}$ is a rotation matrix as defined for the VICReg loss, $\operatorname{diag}_+ = \operatorname{diag}(\mathbb{R}_+^N)$ are the set of diagonal matrix with positive entries, i.e. renormalization matrices, and $\boldsymbol{M}$ is a matrix that maps a deformation of the simplex into the canonical basis. Moreover, provided class templates, i.e. C data points associated with each of the C classes, $\boldsymbol{Y}$ is easily retrieved from any methods and $\boldsymbol{Z} \in \mathcal{S}_{\text{method}}$.

In essence, Theorem 2 states that SSL losses solve the supervised learning problem when the employed graph $\boldsymbol{G}$ is $\boldsymbol{G}^{(\text{sup})}$. Moreover, the matrix $\boldsymbol{D}$ appearing in Theorem 2 captures the fact that SimCLR solutions are invariant to rescaling logit and is akin to the cross-entropy loss, while BarlowTwin is invariant to column renormalization of $\boldsymbol{Z}$ and is akin to discriminant analysis. Lastly, VICReg might be thought of as a proxy for the least-squares loss. At a high-level, Theorem 2 suggests fruitful links between spectral embedding techniques captured in Theorem 1 and supervised learning. We let for future work the investigation of this link and translation of spectral embedding results in the realm of supervised learning.

While Theorem 2 describes what we have coined as the "interpolation optimum", i.e. solution in the interpolation regime with rich models, we ought to highlight that classical statistical learning literature analyzes losses under the light of "Bayes optimum", i.e. solutions in noisy context-free setting [4]. Those Bayes optima do not make as much sense for losses that intrinsically relate different inputs, yet for completeness we provide such a proposition on Bayes optimum in Appendix B.3.

4. PAL: Positive Active Learning

Now that we demonstrated how one should focus on the graph $\boldsymbol{G}$, rather than the (self-)supervised loss, we turn our focus into getting that graph $\boldsymbol{G}$. In particular, we propose an active learning framework that discovers $\boldsymbol{G}$ through efficient, low-cost queries.

4.1. One Framework to Rule Them All

From our understanding (Theorem 2), the difficulties of both supervised learning and SSL are the same: they need a correct graph $\boldsymbol{G}$, i.e they need to identify samples that are semantically similar, either through label annotations or through the right design of DA.

Algorithm 1: PAL framework with oracle

Data: $\boldsymbol{X} \in \mathbb{R}^{N\times D}$; unknown graph $\boldsymbol{G} = \boldsymbol{G}^{(\text{sup})}$.
Result: Embedding $f_\theta : \mathbb{R}^D \to \mathbb{R}^K$.
Initialization: weights θ_0, scheduler (γ_t); $T \in \mathbb{N}$;
for $t \in [T]$ **do**
  Collect $I_t, J_t \leftarrow$ from sampler;
  Collect $(\boldsymbol{G}_{ij} = \mathbf{1}_{\{y_i=y_j\}})_{(i,j)\in I_t}$ from labelers;
  Update $\theta_{t+1} \leftarrow \theta_t - \gamma_t \nabla_\theta \boldsymbol{L}(\theta_t; \boldsymbol{G}, I_t, J_t)$.

Our framework suggests a generic way to proceed, having fixed the samples $\boldsymbol{X}$ in advance, and without much *a priori* knowledge on the similarity graph $\boldsymbol{G}$. In an active learning spirit, one would like to design a query strategy to discover $\boldsymbol{G}$, and an update rule for the learned parameter θ. To ground the discussion, let us focus on VICReg. The variance-covariance term can be rewritten with $R(a,b) = (a^\top b)^2 - \|a\|^2 - \|b\|^2$, this leads to the formula, proven in Appendix B.1.1,

$$\mathcal{L}_{\text{VIC}^2}(\theta;\boldsymbol{G},I,J) = \sum_{(i,j)\in I} \boldsymbol{G}_{i,j} \left\| f_\theta(\boldsymbol{x}_i) - f_\theta(\boldsymbol{x}_j)\right\|^2 \tag{9}$$
$$+ \sum_{(i,j)\in J} R(f_\theta(\boldsymbol{x}_i), f_\theta(\boldsymbol{x}_j)), \tag{10}$$

where $I = J = [N]^2$. An oracle would typically consider two small sets of indices $I, J \subset [N]^2$, asks labelers to provide $\boldsymbol{G}_{ij}$ for $i,j \in I$, and, given a step size γ_t, update the weights with

$$\theta_{t+1} = \theta_t - \gamma_t \nabla_\theta \boldsymbol{L}(\theta_t; \boldsymbol{G}, I, J), \tag{11}$$

which could be performed with the sole access to $(\boldsymbol{G}_{ij})_{(i,j)\in I}$. The pairs in J are used to avoid dimensional collapse, and in particular for the VICReg loss, to compute the variance-covariance regularization term. The complete picture leads to PAL, Algorithm 1. A particularly useful features of SGD for active learning is its robustness to labeling noise [10]. In other terms, *Algorithm 1 is robust to noise in the query answers.*

We will now dive more in-depth into two variants of oracles: passive and active ones. As we will see, passive oracles can recover traditional SSL as special cases, but will be much less efficient in learning good representation than active strategies.

4.2. Passive Oracles

Passive variations of the PAL algorithm consist in fixing the oracle behavior at iterations $t \in [T]$ before starting the training. This formulation, under which the oracle does not leverage any information collected along training, recovers both SSL and supervised learning, based on the different querying strategies.

Self-Supervised Oracle. Probably the simplest oracle to describe is the one corresponding to the SSL strategy. The original VICReg algorithm [3] is made of t gradient updates over $T = N_0 E$ iterations with $N = N_0 V E$ samples, where N_0 is the number of original samples, E is the number of epochs, V the number of views. At time $t \in [T]$, I_t is chosen as $\{(2t+1, 2(t+1))\}$, describing a positive pairs generated on the fly from one original sample $\boldsymbol{x}_i$ for $i = t \bmod. N_0$; and J_t is chosen as some mini-batch to estimate the covariance matrix of the features at the current epoch. Because it has been built to remove human feedback, SSL actually does not need to ask for labelers to query $\boldsymbol{G}_{s,s+1}$ (where $s = 2t+1$), since it is known by construction that those entries are going to be one.

Supervised Oracle. When it comes to a supervised learning oracle, the supervised learning loss provided in Theorem 1 –which is known to recover $\boldsymbol{Y}$ (given class templates) as per Theorem 2– is easily minimized with gradient descent based on (10). Hence a simple oracle to solve the supervised learning problem based on stochastic gradient descent: at time t, consider a random pair of indices (i_t, j_t) and set $I_t = J_t \leftarrow \{(i_t, j_t)\}$. The querying of $\boldsymbol{G}_{i_t,j_t}$ can either be done on the fly, or if the dataset is already annotated, it can be deduced from the knowledge of $\boldsymbol{G}_{i_t,j_t} = \mathbf{1}_{\{y_{i_t}=y_{j_t}\}}$.

Algorithm 2: Passive Oracle Specifications

SSL oracle:
- Sampler: $I_t = \left\{(i_{2t+1}, i_{2(t+1)})\right\}$, J_t a minibatch,
- Labeler: $\boldsymbol{G}_{2t+1,2(t+1)} = 1$.

Supervised oracle:
- Sampler: $I_t = J_t = \{(i_t, j_t)\}$ random in $[N]^2$,
- Labeler: $\boldsymbol{G}_{i,j} = \mathbf{1}_{\{y_i=y_j\}}$.

Theoretical Remarks. Remarking that (10) is an unbiased formulation of VICReg, in the sense that

$$\mathcal{L}_{\mathrm{VIC}^2}(\boldsymbol{Z}) = \mathbb{E}_{I,J\sim\mathcal{U}([N]^2)}\left[\mathcal{L}_{\mathrm{VIC}^2}(\boldsymbol{Z}; I, J)\right].$$

As a consequence, when $\theta \mapsto \left\|f_\theta(\boldsymbol{X}) f_\theta(\boldsymbol{X})^\top - \boldsymbol{G}\right\|^2$ is strongly convex, Algorithm 3 with either the self-supervised or the supervised oracle will converge to the minimizer of the VICReg loss in $O(1/T)$ [8]. Moreover, while this results holds for the empirical loss with resampling, it is equally possible to get a similar result for the minimization of the infinite-data (aka population) version of the VICReg loss and the recovery of the ideal embedding representation, when performing a single pass over the data. In particular, by making sure that J only charges pairs (i, j) for i and j in two disjoint subsets of $[N]$, one can prove convergence rates in $O(1/N)$ (Theorem 3 in [12]).

Moreover, because the VICReg loss in Theorem 2 is nothing but a matrix factorization problem, one can directly translate results from this literature body into PAL. In particular, recent works have derived theoretical results regarding the matrix factorization problem based on toy models of neural networks, which might be plugged directly in here to claim theoretical results about the soundness of the PAL algorithm with neural networks [50, 22, 29]. Since those results hold for any graph $\boldsymbol{G}$, such results directly apply to both SSL and supervised learning, highlighting how PAL jointly derives results for SSL and supervised learning.

4.3. Active Oracles

Seen through the eyes of PAL, supervised and SSL –which employ passive querying– can be improved by refining the oracle to choose the next indices I_t and J_t to process at time t.

Low-Cost and Efficient Active Learning. A crucial point of this study is that the active learning framework stemming from PAL differs fundamentally from classic active learning. In the latter, at time t, one asks for a fresh label y_{i_t} for some chosen index i_t. Instead, PAL considers a batch of data I_t and asks for pairwise comparisons $\mathbf{1}_{\{y_i \sim y_j\}}$ for $(i, j) \in I$. Rather than asking labelers for fine-grained labels, such as "caracal" or "numbfish" on ImageNet, PAL would rather asks labelers if two images are related, or even to spot outliers in a set of images compared to a template, as illustrated on Fig. 1.[2] This is particularly useful when the cost of spotting a few outliers in a batch of M images is much less costly than annotating M data points. On such instances, Criteo engineers found that batches of 15 images was a sweet spot in terms of labeling efficiency [5]; while ImageNet was annotated by querying images on search engines, and spotting outliers among the results [21]. Meanwhile, reCaptcha (illustrated on Fig. 1) is said to have helped annotate millions of images [9]. We refer the curious reader to [43] and references therein regarding the design of efficient user interfaces for those labeling tasks.

Zoo of Active Learning Strategies. By introducing PAL, we open a way to match the practice of active learning and its theory through a grounded framework that encompasses current heuristics to annotate big chunks of data.

[2] This "spot-the-outliers" strategy is formalized with $I_t = \{(i_t, j) \mid j \in \tilde{I}_t\}$ for i_t representing the class template, and $\tilde{I}_t$ capturing the batch of data to spot outliers in.

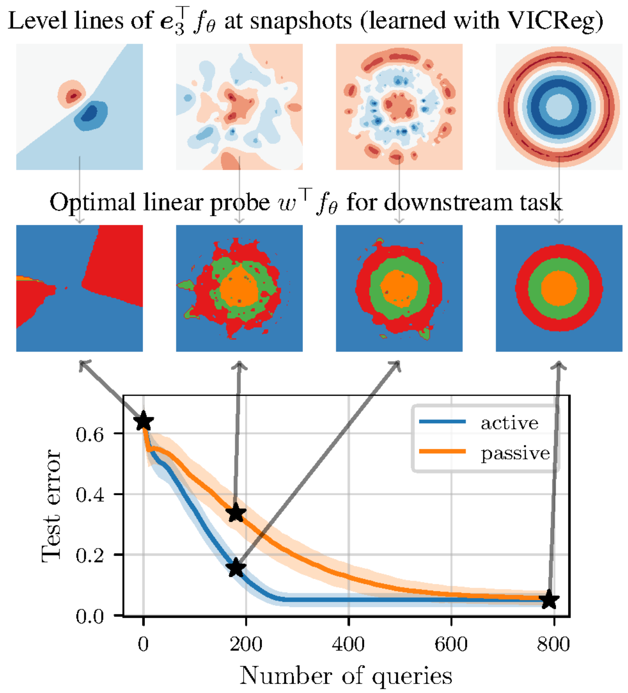

Figure 3. Comparison the active oracle of Algorithm 3 and the passive supervised one of Algorithm 2. Given q queries made, and the consequent reconstructed graph $\boldsymbol{G}_q$, we learn $f_{\theta_t} : \mathcal{X} \to \mathbb{R}^C$ by minimizing $\mathcal{L}_{\text{VIC}^2}$, and plot the downstream mean-square error of the optimal a linear classifier $w^\top f_{\theta_t}$ for the best $w \in \mathbb{R}^C$. Here $\mathcal{X} = \mathbb{R}^2$, and $y \in [4]$ spans four concentric circles (represented by the blue, red, green and orange classes), $N = 100$, query batches are chosen of size 10 and results are average over 100 trials (standard deviations being represented by the colorized regions). Snapshots at different points on the curve show the third coordinates of the reconstructed f_{θ_t}, and the ideal linear classifier that can be learned based on this embedding.

While the optimal oracle depends on the problem idiosyncrasies, as well as the labeling cost model, the vast literature body on active learning provides many heuristics to design sophisticated or intricate oracles under different structural assumptions. One could query information based on the least certain predictions [27, 1]; based on the distance to the decision boundaries [44]; by comparing predictions made by different methods in an ensemble [7, 17]; or by finding the queries whose answers are the most likely to modify the current guess for f_θ [49, 33, 31]. We refer the curious reader to Appendix A for further discussion on the matter. Throughout reviews, adaptations to PAL, ablation studies and comparisons on different benchmarks of those strategies is left for future work.

PAL *à la* Captcha. A natural and easy property to leverage in order to build active learning strategies is the fact that the N^2-entry matrix $\boldsymbol{G}$ is actually derived from the NC-entry matrix $\boldsymbol{Y}$. In particular, one can recover the full graph $\boldsymbol{G} = \boldsymbol{G}^{(\text{sup})}$ with less than NC pairwise queries, and in the best case only N queries –compare this to the N^2-entries that are queried by the supervised learning oracle. This idea is captured formally with the oracle described in Algorithm 3, where the matrix $\boldsymbol{Q}$ remembered past queries, and illustrated on Fig. 3. At time t, this oracle chooses to query against the class with the least numbers of known instances, and choose M data points, ask if they match this class, and update known labels as a consequence. An advantage of the query strategy of Algorithm 3 is that one can stop at any time t and have a balanced labeled dataset to learn with.

Algorithm 3: Oracle *à la* Captcha

Data: Class templates $(\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_C) \in \mathcal{X}^C$,
    $\boldsymbol{Q} \in \mathbb{R}_{N \times C}$ initialized at zero.
Choose the class with least known examples
  $j = \arg\min_j \mathbf{1}^\top \boldsymbol{Q}_t \boldsymbol{e}_j \in [C]$;
Collect pairwise comparison $\boldsymbol{Q}_{ij} \leftarrow \mathbf{1}_{\{\boldsymbol{x}_i \sim \mu_j\}}$ for i
  in a batch $B \subset [N] \setminus \boldsymbol{K}_t$ where $\boldsymbol{K}_t$ remove queries
  with known results based on $\boldsymbol{Q}_t$;
Sampler: $I_t = J_t$ all the new entries deduced in $\boldsymbol{G}$.
Labeler: Human feedback $\boldsymbol{Q}_{ij}$; deduction to fill $\boldsymbol{G}$.

The basic Algorithm 3 can be improved in several ways. First, class templates can be deduced based on initial queries: the first data point $\mu_1 = \boldsymbol{x}_1$ provides a first class template; after querying $\mathbf{1}_{\{\boldsymbol{x}_2 \sim \boldsymbol{x}_1\}}$ if the answer is negative, $\mu_2 = \boldsymbol{x}_2$ provides a second class template (otherwise it is part of class one); so forth and so on (if $\mathbf{1}_{\{\boldsymbol{x}=\mu_1\}} = \cdots = \mathbf{1}_{\{\boldsymbol{x}=\mu_k\}} = 0$, set $\mu_{k+1} = \boldsymbol{x}$). Those templates could be refined during training by defining the templates as the example the most at the center of the classes examples with some well-thought notion of distance (either in the input space or the embedding space). Second, when classes are unbalanced and class probabilities are roughly known, one should rather choose $y(t)$ to be the class that minimizes the number of known examples in this class divided by the probability of this class. Third, if C the number of classes is small, random sampling of the batch B will work well enough. Yet, when C is big, random sampling will mainly lead to negative observations and too few positive ones. In this situation, the algorithm is improved by training a classifier based on known labels at time t (eventually incorporating unlabeled data with semi-supervised learning techniques), and querying labels that were classified as the same class. Finally, to avoid only getting negative pairs on datasets with many classes, one could leverage hierarchy in the labels: if dealing with the classes of ImageNet, one can first ask reviewers coarse-grained information, e.g. flag pictures that are not fishes; before going deeper in the taxonomy.

Modality	Oracle	1 accuracy	5 accuracy
Passive	SimCLR [15]	71.7	-
Passive	VICReg [3]	73.2	91.1
Active	NNCLR [23]	**75.6**	**92.4**

Table 1. Best known performance on ImageNet for state-of-the-art SSL methods. Notice how NNCLR [23] derives states of the art performance on ImageNet thanks to an active rule for labelers in Algorithm 1, which consists in defining positive pairs as nearest neighbors in the embedding space as detailed in Algorithm 4. This rule allows to beat the passive strategy that are provided by SimCLR and VICReg.

5. Experiments

This section provides experimental details to validate the various theoretical results derived in previous sections. In order to remove confounding effects linked with architecture, optimization, data curation and other design choices that might impact the different empirical validation we focus here on closed-form solution based on kernel methods with synthetic dataset. Further real-world empirical validations are provided in Appendix C. In particular, Table 1 reminds the reader how NNCLR [23] succeed to beat state-of-the-art SSL methods on ImageNet thanks to an active labeler oracle, which defines positive pairs as nearest neighbors in the embedding space $f_{\theta_t}(\mathcal{X})$.

Kernel methods are rich "linear" parametric models defined as $f_\theta = \theta^\top \phi(\boldsymbol{x})$, for $\phi(\boldsymbol{x})$ and θ belonging to a separable Hilbert space $\mathcal{H}$. Because those model can approximate any function [36], it is important to regularize θ in practice, either with early stopping in SGD, or with Tikhonov regularization, which can be written as $\lambda \operatorname{Tr}(\boldsymbol{Z}^\top \boldsymbol{K}^{-1} \boldsymbol{Z})$ where $\lambda > 0$ is a regularization parameter and $\boldsymbol{K} \in \mathbb{R}^{N\times N}$ is the kernel matrix defined as $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)$. In this setting, rather than matching the top of the spectral decomposition of $\boldsymbol{G}$, the solution recovered by VICReg amounts to the top spectral decomposition of $\boldsymbol{G} - \lambda \boldsymbol{K}^{-1}$ [12]. This allows to compute the ideal representation of f_θ in closed-form given any graph $\boldsymbol{G}$ based on the regularized kernel model $f_\theta = \theta^\top \phi(\boldsymbol{x})$, hence ablating the effects that are unrelated to the theory described in this study. In this controlled setting, the superiority of active algorithms is undeniable, and illustrated on Fig. 3, where we illustrate the optimal downstream error one can achieve with linear probing of the minimizer f_θ of the VICReg loss. Experimental details and more extensive validations are provided in Appendix C: in particular, the use of non-contrastive versus contrastive graphs, i.e. that set $\boldsymbol{G}_{ij} = -1$ when $y_i \neq y_j$, is studied on Fig. 7; the ability to incorporate label knowledge in SSL methods is the object of Fig. 4; robustness to noise is shown on Fig. 8; and relations between test error and the number of connected components of the reconstructed $\boldsymbol{G}$ is analyzed on Fig. 9.

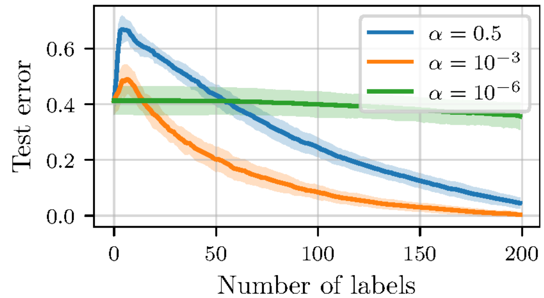

Figure 4. A major motivation of this paper is to be able to add prior information on sample relationships in SSL methods, and more in particular, to have a simple way to leverage known labels. We do by considering $\boldsymbol{Y}$ containing one-hot encoding of known labels, and rows being zero otherwise and the mixed graph $\boldsymbol{G} = (1-\alpha)\cdot \boldsymbol{G}^{(\mathrm{ssl})} + \alpha \cdot \hat{\boldsymbol{Y}}\hat{\boldsymbol{Y}}^\top$. The setting is the same as Fig. 5 with $N = 200$ and two augmentations per sample. When zero labels are known (left of the plot), we are in the full SSL regime, while when all the 200 labels are known (right of the plot), we recover supervised learning performance. When few labels are given the effect of the supervised graph can be counterproductive if the mixing coefficient α is too big. However, when mixed properly, adding prior label information in SSL methods allows to improve performance.

6. Conclusions

This work introduces PAL, a learning framework that revolves around the central concept of similarity graph. We first showed how similarity graphs are the implicit backbone of self-supervised learning methods, and how this concept extends to tackle supervised learning problems. This observation does not solely unveil a rich learning framework, but also provides a single algorithm based on a querying oracle that can describe both SSL and supervised learning techniques, opening the way to new oracles that benefit from techniques stemming from both the supervised and self-supervised learning literature. Finally, PAL leads to an efficient formalization of active learning as performed in practice to annotate large datasets, potentially enabling fruitful exchanges between the practice and the theory of active learning. Promising directions for future works include empirical validations on large-scale datasets, as well as theoretical study of the newly introduced active learning framework.

References

[1] Jordan Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *ICLR*, 2020. 7, 11

[2] Randall Balestriero and Yann LeCun. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. *arXiv preprint arXiv:2205.11508*, 2022. 4

[3] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 3, 6, 8

[4] Peter Bartlett, Michael Jordan, and Jon McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 2006. 5

[5] Renaud Bauvin. Lessons learned from annotating 5 million images, 2019. Medium. 6

[6] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19, 2006. 1

[7] Mustafa Bilgic, Lilyana Mihalkova, and Lise Getoor. Active learning for networked data. In *ICML*, 2010. 7

[8] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, pages 231–357, 2015. 6

[9] ByteBridge. Data annotation: By typing captcha, you are actually helping ai model training, 2021. Medium. 6

[10] Vivien Cabannes, Francis Bach, Vianney Perchet, and Alessandro Rudi. Active labeling: streaming stochastic gradients. In *NeurIPS*, 2022. 5

[11] Vivien Cabannes, Alberto Bietti, and Randall Balestriero. On minimal variations for unsupervised representation learning. In *ICASSP*, 2023. 12

[12] Vivien Cabannes, Bobak T Kiani, Randall Balestriero, Yann LeCun, and Alberto Bietti. The SSL interplay: Augmentations, inductive bias, and generalization. *arXiv preprint arXiv:2302.02774*, 2023. 3, 6, 8

[13] Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. Incremental algorithms for hierarchical classification. *Journal of Machine Learning Research*, 2006. 11

[14] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 3

[15] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020. 8

[16] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020. 1

[17] Kashyap Chitta, Jose Alvarez, Elmar Haussmann, and Clement Farabet. Training data subset search with ensemble active learning. *IEEE Transactions on Intelligent Transportation Systems*, 2021. 7, 11

[18] Timothée Cour, Benjamin Sapp, and Ben Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 2011. 11

[19] Thomas Cover and Joy Thomas. *Elements of Information Theory*. Wiley, 1991. 18

[20] Sanjoy Dasgupta. Two faces of active learning. *Theoretical Computer Science*, 412(19):1767–1781, 2011. 1

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, 2009. 6

[22] Simon Du, Wei Hu, and Jason Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *NeurIPS*, 2018. 6

[23] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021. 8, 11

[24] Nanyi Fei, Zhiwu Lu, Yizhao Gao, Guoxing Yang, Yuqi Huo, Jingyuan Wen, Haoyu Lu, Ruihua Song, Xin Gao, Tao Xiang, Hao Sun, and Ji-Rong Wen. Towards artificial general intelligence via a multimodal foundation model. *Nature Communications*, 2022. 1

[25] Tanner Fiez, Lalit Jain, Kevin G Jamieson, and Lillian Ratliff. Sequential Experimental Design for Transductive Linear Bandits. In *Advances in Neural Information Processing Systems*, 2019. 11

[26] Zhe Gan, Linjie Li, Chunyuan Li, Lijuan Wang, Zicheng Liu, and Jianfeng Gao. Vision-language pre-training: Basics, recent advances, and future trends, 2022. 1

[27] Steve Hanneke. Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning*, pages 131–309, 2014. 1, 7

[28] Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[29] Liwei Jiang, Yudong Chen, and Lijun Ding. Algorithmic regularization in model-free overparametrized asymmetric matrix factorization. In *ArXiv*, 2022. 6

[30] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. In *ArXiv*, 2020. 11

[31] Mina Karzand and Robert Nowak. Maximin active learning in overparameterized model classes. *IEEE Journal on Selected Areas in Information Theory*, 2020. 1, 7

[32] Bobak T Kiani, Randall Balestriero, Yubei Chen, Seth Lloyd, and Yann LeCun. Joint embedding self-supervised learning in the kernel regime. *arXiv preprint arXiv:2209.14884*, 2022. 12

[33] Donald Knuth. The computer as master mind. *Journal of Recreational Mathematics*, 1977. 7

[34] Yann LeCun. A path towards autonomous machine intelligence. In *ArXiv*, 2022. 1

[35] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 1

[36] Charles Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 2006. 8

[37] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020. 1

[38] Alex Nowak-Vila, Francis Bach, and Alessandro Rudi. Sharp analysis of learning with discrete losses. In *Artificial Intelligence and Statistics*, 2019. 14

[39] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010. 1

[40] Alexander Ratner, Stephen Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: rapid training data creation with weak supervision. *The VLDB Journal*, 2020. 11

[41] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison, 2010. 1

[42] Burr Settles. From theories to queries: Active learning in practice. In *Active learning and experimental design workshop in conjunction with AISTATS 2010*, pages 1–18. JMLR Workshop and Conference Proceedings, 2011. 1

[43] Patrice Simard, Saleema Amershi, David Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, and John Wernsing. Machine teaching: A new paradigm for building machine learning systems. In *ArXiv*, 2017. 6

[44] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In *NeurIPS*, 2022. 7, 11

[45] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, 2022. 11

[46] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. In *ArXiv*, 2023. 11

[47] Rom Varshamov. Estimate of the number of signals in error correcting codes. *Doklady Akademii Nauk SSSR*, 1957. 18

[48] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 1

[49] Zhilin Yang, William Cohen, , and Ruslan Salakhutdinov. Semi-supervised learning with graph embeddings. In *ICML*, 2016. 7

[50] Tian Ye and Simon Du. Global convergence of gradient descent for asymmetric low-rank matrix factorization. In *ArXiv*, 2021. 6

[51] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021. 3

[52] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1476–1485, 2019. 1

[53] Evgenii Zheltonozhskii, Chaim Baskin, Avi Mendelson, Alex M Bronstein, and Or Litany. Contrast to divide: Self-supervised pre-training for learning with noisy labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1657–1667, 2022. 1