A. Active Learning Algorithms

NNCLR oracle. State-of-the-art the year of its release, [23] proposes a variation of SSL, that does not need human feedback but does update the similarity graph based on past observation. This furnishes strong evidence for the usefulness of active learning algorithms, even without human feedback. The NNCLR oracle consists in setting $I_t = J_t$ equals to some minibatch at time t, but to labels positive pairs in $\boldsymbol{G}_{ij}$ only for nearest neighbors, i.e.

$$\boldsymbol{G}_{ij}^{(t)} = \mathbf{1}_{\{\boldsymbol{z}_i \in \mathcal{N}(\boldsymbol{z}_j, I)\}} \triangleq \mathbf{1}_{\{\|\boldsymbol{z}_i - \boldsymbol{z}_j\| = \min_{j \in I_t} \|\boldsymbol{z}_i - \boldsymbol{z}_k\|\}}.$$

We describe the corresponding active oracle in Algorithm 4.

Algorithm 4: Active Oracle with No Human Feedback as per [23]

NNCLR oracle:
Sampler: $I_t = J_t$ is some minibatch,
Labeler: $\boldsymbol{G}_{i,j} = \mathbf{1}_{\{\boldsymbol{z}_i \in \mathcal{N}(\boldsymbol{z}_j; I)\}}$ where $\mathcal{N}(\boldsymbol{z}; I)$ design the nearest neighbor of $\boldsymbol{z}$ in the batch I.

The New Rise of Active Learning. Recently, active learning has become a focus of the machine learning community when training big models, as those models performance are known to depend on the order they process data [45], as well as in the percentage of data of different type they ingests [46]. In particular, the best paper award at NeurIPS last year [44] has suggested that the distance to the decision boundary could be leveraged smartly to reduce the number of data needed by current AI algorithms to train state of the art models (as compared to the scaling law of [30]). We describe their sampling oracle in Algorithm 5. Note that in the original paper, they query the exact labels of the selected points, while PAL only needs to query pairwise comparison. In the meantime, [17] suggested using ensemble active learning, while [1] suggested a model based uncertain predictions through gradient computation in deep learning models.

Algorithm 5: Active Oracle with Data Pruning as per [44]

Perform k-means clustering on $\boldsymbol{Z}_t = f_{\theta_t}$.
For each unlabeled points, compute cosine distance to its cluster center.
if *Few examples have been labels* **then**
  $I_t = J_t \leftarrow$ points near cluster centers,
else
  $I_t = J_t \leftarrow$ points far from cluster centers.

From Coarse to Fine-grained Query. While active learning usually assumes that the cost of answering any questions is constant, in practice, some queries might be easier to answer than others. For example, if a child has never seen some objects, such as a sophisticated designer chair, they might not easily provide pairwise comparison regarding those objects, e.g. they would be puzzled by the designer chair, and would hesitate to say that this is a chair. Similarly, it might be easier for human labelers to recognize attributes in an image, e.g. sandy fur, desert background, tufted ear, feline; rather than precise species, such as "caracal". This has been the basis for weakly supervised learning [18, 40]. It has also motivated some bandit models, such as [13, 25]. More generally, it suggests that one could efficiently learn by first querying weak, coarse-grained information, before refining queries to get precise, fine-grained feedback. We illustrate this high-level idea with Algorithm 6.

Algorithm 6: Active Oracle with Hierarchical Taxonomy

Samplers: Your favorite active learning sampler.
if f_{θ_t} *has not formed strong opinions on clusters* **then**
  Labelers: Human feedback for coarse-grained information (e.g. click on all animals with fur...),
else
  Labelers: Human feedback with fine-grained information (e.g. click on fishes that match a precise species).

B. Proofs

B.1. Proof of Theorem 1

B.1.1 VICReg Loss and Spectral Contrastive

This subsection will both identify the VICReg with the spectral contrastive one through their matrix formulation.

Let us begin by reformulating the invariance term in VICReg. For $\boldsymbol{Z}$ defined in (1), it is generalized to multiple pairs through

$$\begin{aligned}\mathcal{L}_{\text{VIC}-\text{INV}} &= \sum_{i,j\in[N]} \boldsymbol{G}_{ij} \left\|\boldsymbol{z}_i - \boldsymbol{z}_j\right\|^2 = \sum_{i,j\in[N]} 2\boldsymbol{G}_{ij} \left\|\boldsymbol{z}_i\right\|^2 - 2\boldsymbol{G}_{ij} \left\langle \boldsymbol{z}_i, \boldsymbol{z}_j \right\rangle \\ &= \sum_{i,j\in[N]} 2\boldsymbol{G}_{ij}[\boldsymbol{Z}\boldsymbol{Z}^\top]_{ii} - 2\boldsymbol{G}_{ij}[\boldsymbol{Z}\boldsymbol{Z}^\top]_{ji} = \sum_{i\in[N]} 2[\boldsymbol{D}\boldsymbol{Z}\boldsymbol{Z}^\top]_{ii} - 2[\boldsymbol{G}\boldsymbol{Z}\boldsymbol{Z}^\top]_{ii} \\ &= 2\operatorname{Tr}\left((\boldsymbol{D}-\boldsymbol{G})\boldsymbol{Z}\boldsymbol{Z}^\top\right) = 2\operatorname{Tr}\left(\boldsymbol{Z}^\top(\boldsymbol{D}-\boldsymbol{G})\boldsymbol{Z}\right)\end{aligned}$$

where $\boldsymbol{D}$ is the degree matrix defined as a diagonal matrix, with A the number of augmented samples per original input

$$\boldsymbol{D}_{ij} = \mathbf{1}_{\{i=j\}} \cdot \sum_{k\in[N]} \boldsymbol{G}_{ik} = A\mathbf{1}_{\{i=j\}}.$$

The variance-covariance term can be simplified by replacing the the Hinge loss for the variance by a squared norm [32], by setting $\beta = \alpha = 1$ and replacing A by N in $\boldsymbol{D}$ to regularize diagonal terms a bit more. Those simplifications lead to a more principled regularization term that enforces orthogonality over the dataset of the different features learned by the network f_θ [11]. The consequent regularization reads $\left\|\boldsymbol{Z}^\top\boldsymbol{Z}/N - \boldsymbol{I}_K\right\|^2$. As a consequence, VICReg can be understood as solving for

$$N\mathcal{L}_{\text{VIC}} \approx \left\|\boldsymbol{Z}^\top\boldsymbol{Z} - N\boldsymbol{I}_N\right\|^2 + 2\operatorname{Tr}(\boldsymbol{Z}^\top(N\boldsymbol{I}_N - \boldsymbol{G})\boldsymbol{Z}).$$

For the spectral contrastive Loss, it is useful to incorporate negative pairs that are sampled for the same augmentations for two different samples $\left\langle \boldsymbol{Z}_i^{(1)}, \boldsymbol{Z}_j^{(1)} \right\rangle$ in the repulsive term. Moreover adding $\left\|\boldsymbol{Z}_i^{(v)}\right\|$ on both the positive part and the negative part will not change much since $-2x + x^2$ is minimized for $x = 1$. Those modifications lead to

$$\begin{aligned}\mathcal{L}_{\text{VIC}^2} &= -2\sum_{i,j\in[N]} \boldsymbol{G}_{ij}\boldsymbol{z}_i^\top\boldsymbol{z}_j + \sum_{i,j\in[N]} (\boldsymbol{z}_i^\top\boldsymbol{z}_j)^2 = -2\sum_{i,j\in[N]} \boldsymbol{G}_{ij}[\boldsymbol{Z}\boldsymbol{Z}^\top]_{ij} + \sum_{i,j\in[N]} [\boldsymbol{Z}\boldsymbol{Z}^\top]_{ij}^2 \\ &= -2\operatorname{Tr}(\boldsymbol{G}\boldsymbol{Z}\boldsymbol{Z}^\top) + \operatorname{Tr}(\boldsymbol{Z}\boldsymbol{Z}^\top\boldsymbol{Z}\boldsymbol{Z}^\top) = \operatorname{Tr}(\boldsymbol{Z}\boldsymbol{Z}^\top\boldsymbol{Z}\boldsymbol{Z}^\top - 2\boldsymbol{G}\boldsymbol{Z}\boldsymbol{Z}^\top + \boldsymbol{G}^2) - \operatorname{Tr}(\boldsymbol{G}^2) \\ &= \operatorname{Tr}((\boldsymbol{Z}\boldsymbol{Z}^\top - \boldsymbol{G})^2) - \operatorname{Tr}(\boldsymbol{G}^2) = \left\|\boldsymbol{Z}\boldsymbol{Z}^\top - \boldsymbol{G}\right\|_F^2 + \text{cst}.\end{aligned}$$

The last term being a finite constant, it can be removed from the loss.

Indeed, one can relate both the spectral contrastive loss and VICReg, by remarking that

$$\begin{aligned}&\left\|\boldsymbol{Z}^\top\boldsymbol{Z} - N\boldsymbol{I}_N\right\|^2 + 2\operatorname{Tr}(\boldsymbol{Z}^\top(N\boldsymbol{I}_N - \boldsymbol{G})\boldsymbol{Z}) \\ &\quad= \operatorname{Tr}(\boldsymbol{Z}\boldsymbol{Z}^\top\boldsymbol{Z}\boldsymbol{Z}^\top - 2N\boldsymbol{I}_N\boldsymbol{Z}^\top\boldsymbol{Z} + N^2\boldsymbol{I}_N) + 2\operatorname{Tr}(\boldsymbol{Z}^\top(N\boldsymbol{I}_N - \boldsymbol{G})\boldsymbol{Z}) \\ &\quad= \operatorname{Tr}(\boldsymbol{Z}\boldsymbol{Z}^\top\boldsymbol{Z}\boldsymbol{Z}^\top - 2\boldsymbol{G}\boldsymbol{Z}\boldsymbol{Z}^\top) + N^3 = \operatorname{Tr}((\boldsymbol{Z}\boldsymbol{Z}^\top - \boldsymbol{G})^2) - \boldsymbol{G}^2) + N^3 \\ &\quad= \left\|\boldsymbol{Z}\boldsymbol{Z}^\top - \boldsymbol{G}\right\|_F^2 + \text{cst}.\end{aligned}$$

Finally, the variance covariance term can be written as

$$\begin{aligned}\left\|\boldsymbol{Z}\boldsymbol{Z}^\top/N - I\right\|^2 &= \left\|\sum_{i\in[N]} \boldsymbol{z}_i\boldsymbol{z}_i^\top - I\right\|^2 = \operatorname{Tr}\left(\sum_{i,j\in[N]} \boldsymbol{z}_j\boldsymbol{z}_j^\top\boldsymbol{z}_i\boldsymbol{z}_i^\top - 2\sum_{i\in[N]} \boldsymbol{z}_i\boldsymbol{z}_i^\top + I\right) \\ &= \sum_{i,j\in[N]} (\boldsymbol{z}_j^\top\boldsymbol{z}_i)^2 - \sum_{i,j\in[N]} (\boldsymbol{z}_i^\top\boldsymbol{z}_i + \boldsymbol{z}_j^\top\boldsymbol{z}_j) + \text{cst} = \sum_{i,j\in[N]} R(\boldsymbol{z}_i, \boldsymbol{z}_j) + \text{cst},\end{aligned}$$

where $R(a, b) = (a^\top b)^\top - \|a\|^2 - \|b\|^2$.

B.1.2 The SimCLR Loss

SimCLR can be seen as a generalized linear model, where two variables A, B are observed and the probability observing B knowing A is given by

$$p_{ij} = \mathbb{P}(B = j \mid A = i) \propto \exp\left(\frac{\boldsymbol{z}_i^\top \boldsymbol{z}_j}{\|\boldsymbol{z}_i\| \, \|\boldsymbol{z}_j\|}\right).$$

For simplicity, let us define $\tilde{\boldsymbol{z}} = \boldsymbol{z}/\|\boldsymbol{z}\|$. SimCLR tries to maximize the likelihood of (A, B) denoting random pairs coming from the same augmentations based on the observation of the graph

$$\prod_{ij \in [N]} p_{ij}^{\boldsymbol{G}_{ij}} = \exp\left(\sum_{ij \in [N]} \boldsymbol{G}_{ij} \log\left(\frac{\exp\left(\tilde{\boldsymbol{z}}_i^\top \tilde{\boldsymbol{z}}_j\right)}{\sum_{k \in [N]} \exp(\tilde{\boldsymbol{z}}_i^\top \tilde{\boldsymbol{z}}_k)}\right)\right).$$

The SimCLR loss is nothing but the inverse of the log likelihood.

$$\mathcal{L}_{\text{SimCLR}} = -\sum_{ij \in [N]} \boldsymbol{G}_{ij} \log\left(\frac{\exp\left(\tilde{\boldsymbol{z}}_i^\top \tilde{\boldsymbol{z}}_j\right)}{\sum_{k \in [N]} \exp(\tilde{\boldsymbol{z}}_i^\top \tilde{\boldsymbol{z}}_k)}\right).$$

B.1.3 Barlow Twins

When $\lambda = 1$, which we will consider for simplicity, the BarlowTwins loss simplifies as

$$\mathcal{L}_{\text{BT}} = \sum_i (1 - \boldsymbol{C}_{ii})^2 + \sum_{i \neq j} \boldsymbol{C}_{ij}^2 = \|\boldsymbol{C} - \boldsymbol{I}_k\|_F^2 .$$

Because cross-correlations are normalized cross-covariances, it is useful to introduce $\tilde{\boldsymbol{Z}}$ the column normalized version of $\boldsymbol{Z}$. Formally written in normalized matrix with the Hadamard product notation as

$$\tilde{\boldsymbol{Z}}_{ij} = \frac{z_{ij}}{\sqrt{\sum_k z_{kj}^2}} = \frac{z_{ij}}{[(\boldsymbol{Z} \otimes \boldsymbol{Z})\boldsymbol{1}]_j^{1/2}} \qquad \text{i.e.} \qquad \tilde{\boldsymbol{Z}} = \boldsymbol{Z} \operatorname{diag}(\boldsymbol{Z}^{\otimes 2}\boldsymbol{1})^{-1/2}$$

The way the cross-correlation is built can be generalized to multiple positive pairs as

$$\boldsymbol{C}_{ij} = \frac{\sum_{kl} \boldsymbol{G}_{kl} z_{ki} z_{lj}}{\sqrt{\sum_k z_{ki}^2}\sqrt{\sum_l z_{lj}^2}} = \sum_{kl} \boldsymbol{G}_{kl} \tilde{z}_{ki} \tilde{z}_{lj} = [\tilde{\boldsymbol{Z}}^\top \boldsymbol{G} \tilde{\boldsymbol{Z}}]_{ij}.$$

As a consequence, the BarlowTwins loss can be rewritten with the sole use of $\boldsymbol{G}$ as

$$\mathcal{L}_{\text{BT}} = \left\| \tilde{\boldsymbol{Z}}^\top \boldsymbol{G} \boldsymbol{Z} - \boldsymbol{I}_K \right\|_F^2 .$$

B.2. The SSL Losses for Supervised Learning

This subsection is devoted to the proof of Theorem 2.

B.2.1 Recovery Lemma

The backbone of Theorem 2 is following Lemma.

Lemma 1 (Equivalence between $\boldsymbol{Y}$ and $\boldsymbol{G}$)**.** *Given any supervised classification similarity matrix $\boldsymbol{G} = \boldsymbol{Y}\boldsymbol{Y}^\top$ (6), one can recover the corresponding one-hot label encoding $\boldsymbol{Y}$, up to an orthogonal transformation $\boldsymbol{R}$, as*

$$\exists \, \boldsymbol{R} \in O(C), \quad \textit{s.t.} \quad \boldsymbol{Y} = \boldsymbol{P}\sqrt{\boldsymbol{D}}\boldsymbol{R},$$

where $\boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^T$ is the eigenvalue decomposition of the adjacency matrix $\boldsymbol{G}(\boldsymbol{Y})$. Moreover the rotation $\boldsymbol{R}$ is easily recovered by specifying the labels of C samples associated with each of the C different classes.

Proof. Lemma 1 follows from the fact that $\boldsymbol{G} = \boldsymbol{Y}\boldsymbol{Y}^\top$ so that $\boldsymbol{Y}$ is a square root of $\boldsymbol{G}$, and that any two square roots of a matrix are isometric. In particular, if the SVD of $\boldsymbol{Y}$ is written as

$$\boldsymbol{Y} = \boldsymbol{P}\sqrt{\boldsymbol{D}}\boldsymbol{R}, \qquad \boldsymbol{P} \in O(N), \boldsymbol{R} \in O(C), \boldsymbol{D} = \begin{bmatrix} \boldsymbol{D}_1 & 0 \end{bmatrix} \in \mathbb{R}^{N \times C}, \boldsymbol{D}_1 = \operatorname{diag}(\sigma_1^2, \cdots \sigma_C^2),$$

the decomposition $\boldsymbol{G} = PDP^\top$ is an eigenvalue decomposition of $\boldsymbol{G}$. The part $P\sqrt{D}$ is unique up to the application of a rotation on the right, which could be absorbed in R.

In order to recover $\boldsymbol{Y}$ from $\boldsymbol{G}$, notice that up to a permutation of lines and columns, $\boldsymbol{G}$ has a block diagonal structure where each block corresponds to one label. If each one label is given to each block, this allows to retrieve exactly $\boldsymbol{Y}$ hence to identify $\boldsymbol{R}$ afterwards by solving for $\boldsymbol{R} = (\boldsymbol{P}\sqrt{mD})^{-1}\boldsymbol{Y}$. □

While lemma 1 describes the classification case, in the generic case, if the y are categorical, yet the loss $\ell(y, z)$ is not the zero-one loss, it is natural to define the similarity matrix as

$$\boldsymbol{G} \triangleq (-\ell(y_i, y_j))_{i,j \in [N]} \in \mathbb{R}^{C \times C}. \tag{12}$$

For example, y_i could be rankings modeled with $y_i \in \mathfrak{S}_m$ where $m! = C$, i.e. $m = \Gamma^{-1}(C) - 1$, and ℓ could be the Kendall loss. In this setting,

$$\boldsymbol{G} = \boldsymbol{Y}\boldsymbol{L}\boldsymbol{Y}, \qquad \text{where} \qquad \boldsymbol{L} \triangleq (-\ell(y, z))_{y,z \in [C]} \in \mathbb{R}^{C \times C},$$

and $\boldsymbol{Y}$ is retrieved through $\boldsymbol{Y} = \boldsymbol{P}\sqrt{\boldsymbol{D}}\boldsymbol{R}\boldsymbol{L}^{1/2}$, where $\boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^\top$ is the eigenvalue decomposition of $\boldsymbol{G}$, and $\boldsymbol{R} \in \mathbb{R}^{C \times C}$ is an unknown rotation matrix, that might be identified by specifying at most C labels associated with each of the C different classes, but might be identified with a smaller number of samples if ℓ has a strong structure implying that $\boldsymbol{L}$ is low-rank (see Eq. (11) in [38]). Indeed, the fact that compared to (6), the graph (12) could be much lower rank, could lead to more efficient algorithm to image it in the active learning framework. In essence, it would better leverage the structure encoded by the loss ℓ.

Finally, in the regression setting, one can choose $\boldsymbol{G}_{ij} = -y_i^\top y_j$.

B.2.2 The VICReg Loss

The VICReg loss is characterized as

$$\mathcal{L}_{\text{VIC}-2} = \left\| \boldsymbol{Z}\boldsymbol{Z}^\top - \boldsymbol{G} \right\|_F^2 + \text{cst}.$$

So, it is minimized for $\boldsymbol{Z}$ being a square root of the matrix $\boldsymbol{G}$. This is possible when the rank of $\boldsymbol{G}$ which is at most C since $\boldsymbol{G} = \boldsymbol{Y}\boldsymbol{Y}^\top$ is less than the rank of $\boldsymbol{Z}$ which is K. In this setting, since $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are two square roots of $\boldsymbol{G}$, we get

$$\exists\, \boldsymbol{R} \in O(C, K), \qquad \boldsymbol{Z} = \boldsymbol{Y}\boldsymbol{R},$$

where we define the rotation $O(C, K)$ as

$$O(C, K) = \left\{ \boldsymbol{R} \in \mathbb{R}^{C \times K} \,\middle|\, \boldsymbol{R}\boldsymbol{R}^\top = \boldsymbol{I}_C \right\}. \tag{13}$$

B.2.3 The SimCLR Loss

The probabilistic interpretation of SimCLR states that the SimCLR losses tries to maximize the likelihood of the events

$$\cup_{ij \in [N]} \left\{ \boldsymbol{G}_{ij} = 1 \right\} \cap \left\{ Y = i \;\&\; X = j \right\},$$

which translate as a loss in the minimization of

$$\mathcal{L}_{\text{SimCLR}} = -\sum_{ij \in [N]} \boldsymbol{G}_{ij} \log \left(\frac{\exp\left(\tilde{\boldsymbol{z}}_i^\top \tilde{\boldsymbol{z}}_j\right)}{\sum_{k \in [N]} \exp(\tilde{\boldsymbol{z}}_i^\top \tilde{\boldsymbol{z}}_k)} \right).$$

This is the cross entropy between $\boldsymbol{G}_{ij}$ and p_{ij} defined in the proof of the characterization of SimCLR. If the minimization with respect to p_{ij} was unconstrained, then one should match $p_{ij} \propto \boldsymbol{G}_{ij}$. Yet, the form of $p_{ij} \in [\exp(-1), \exp(1)]$ constraints it to go for a slightly different solution.

Remark that for two $\tilde{\boldsymbol{z}}_i, \tilde{\boldsymbol{z}}_j$ whose index i and j belongs to different clusters defined by the graph $\boldsymbol{G}$, the loss is a increasing function of the quantity $\exp(\tilde{\boldsymbol{z}}_i \tilde{\boldsymbol{z}}_j)$. By symmetry, we deduce that all the $\tilde{\boldsymbol{z}}_i \tilde{\boldsymbol{z}}_j = \cos(\boldsymbol{z}_i, \boldsymbol{z}_j)$ should be one for all (i, j) such that $\boldsymbol{G}_{ij} = 1$. On the other hand, the loss is a decreasing function of the $\exp(\tilde{\boldsymbol{z}}_i^\top \tilde{\boldsymbol{z}}_j)$ when $\boldsymbol{G}_{ij} = 1$. When the number of sample per class is constant, we deduce by symmetry that the different anchors for the different classes should be put at the extremity of the simplex with C vertices centered at the origin and rotate with an arbitrary matrix $\boldsymbol{R} \in O(C-1)$, which allow to recover the different classes (without their explicit labels if not provided). When the different class have different number of samples N_i with $\sum_{i\in[C]} N_i = N$, and their anchor in the output space is $\boldsymbol{c} \in \mathbb{R}^K$, we are trying to minimize

$$\sum_{j\in[C]} N_j \log\left(\sum_{i\in[C]} N_i \exp(\boldsymbol{c}_i^\top \boldsymbol{c}_j)\right), \tag{14}$$

which will deform the simplex to have bigger angles between classes that are highly represented. For example, when $N_1 = N_2 \approx N/2$, we will have $\boldsymbol{c}_1 \approx -\boldsymbol{c}_2$ while the other anchors are orthogonal to one another and to $\boldsymbol{c}_1$. Denoting $\boldsymbol{M} \in \mathbb{R}^C$ the matrix that maps the anchor of the class i for one solution of (14) to the i-th element of the canonical basis $\boldsymbol{e}_i$ as $\boldsymbol{v}_i \boldsymbol{M} = \boldsymbol{e}_i$, we get that the solution

$$\boldsymbol{Z} = \boldsymbol{D}\boldsymbol{Y}\boldsymbol{R}\boldsymbol{M}^{-1}, \qquad \text{with} \qquad \boldsymbol{D} \in \operatorname{diag}(\mathbb{R}_+^N); \boldsymbol{R} \in O(K, C).$$

The fact that $\boldsymbol{Z}$ is invariant by scaling each vector $\boldsymbol{z}$ reminds us of implementation of the cross-entropy, where to avoid divergence to infinity (since the sigmoid is optimized at infinity) one has to normalize the solution. The SimCLR loss is actually built on the same generalized linear model as the cross-entropy, and one can roughly think of SimCLR as the SSL version of the cross-entropy.

B.2.4 BarlowTwins

To minimize the BarlowTwins loss

$$\mathcal{L}_{\mathrm{BT}} = \left\| \tilde{\boldsymbol{Z}}^\top \boldsymbol{G} \tilde{\boldsymbol{Z}} - I_K \right\|_F^2,$$

we want $\tilde{\boldsymbol{Z}}$ to be a square root of the inverse of $\boldsymbol{G}$. To be more precise, introduce the eigenvalue decomposition of $\boldsymbol{G}$ as $\boldsymbol{G} = \boldsymbol{P}\boldsymbol{S}\boldsymbol{P}^\top$ where $\boldsymbol{P} \in \mathbb{R}^{N\times C}$ and $\boldsymbol{S} \in \mathbb{R}^{C\times C}$ since $\boldsymbol{G}$ is at most of rank C. The minimizer of BarlowTwins is $\tilde{\boldsymbol{Z}} = [\boldsymbol{P}\boldsymbol{S}^{-1/2}, 0_{N\times(K-C)}]$. Since $\tilde{\boldsymbol{Z}}$ is the column normalized version of $\boldsymbol{Z}$, $\boldsymbol{Z}$ can be reconstructed for any diagonal matrix $\boldsymbol{D} \in \operatorname{diag}(\mathbb{R}_+^K)$ as $\boldsymbol{Z} = \tilde{\boldsymbol{Z}}\boldsymbol{D}$. Incorporating $[\boldsymbol{S}^{-1}, 0]$ in $\boldsymbol{D}$, we get that the minimizer of the BarlowTwins loss are exactly the matrices $\boldsymbol{Z} = \boldsymbol{P}\boldsymbol{S}^{1/2}[\boldsymbol{D}, 0_{K-C}]$ for $\boldsymbol{D} \in \operatorname{diag}\mathbb{R}_+^C$. Moreover, since both $\boldsymbol{P}\boldsymbol{S}^{1/2}$ and $\boldsymbol{Y}$ are square root of $\boldsymbol{G}$, we know that the exists a rotation matrix $\boldsymbol{R} \in O(K)$ such that $\boldsymbol{P}\boldsymbol{S}^{1/2} = \boldsymbol{Y}\boldsymbol{R}$. All together, we get that the minimizer of the BarlowTwins loss are exactly the

$$\boldsymbol{Z} = \boldsymbol{Y}\boldsymbol{R}\boldsymbol{D}, \qquad \text{for} \quad \boldsymbol{R} \in O^{C,K},\ \boldsymbol{D} \in \operatorname{diag}(\mathbb{R}_+^C)$$

The fact that BarlowTwins do not care about the amplitude of the solution $\boldsymbol{Z}$ reminds us of discriminant analysis that learns classifiers by optimizing ratio and angles.

B.3. Bayes optimum

For completeness, we now state a Bayes optimum proposition regarding the VICReg loss of the paper.

Proposition 1 (Bayes optimum)**.** *When $K \geq C$ and there is no context, i.e. $\boldsymbol{x}_i = \boldsymbol{x}_0$ for all $i \in [N]$, and y_i are sampled according to a noisy distribution $(y \mid \boldsymbol{x} = \boldsymbol{x}_0)$, the naive study of the VICReg Bayes optimum is meaningless, since*

$$\operatorname*{arg\,min}_{\boldsymbol{z}\in\mathbb{R}^K} \mathcal{L}_{\mathrm{VIC}-2}\left(\begin{bmatrix}\boldsymbol{z}\\ \vdots \\ \boldsymbol{z}\end{bmatrix}; \boldsymbol{G}\right) = \left\{\boldsymbol{z} \in \mathbb{R}^K \;\middle|\; \|\boldsymbol{z}\| = 1\right\}.$$

Yet, if one free the variable $\boldsymbol{Z} \in \mathbb{R}^{N\times K}$, we have

$$\operatorname*{arg\,min}_{\boldsymbol{Z}\in\mathbb{R}^{N\times K}} \mathcal{L}_{\mathrm{VIC}-2}(\boldsymbol{Z}; \boldsymbol{G}) = N^{1/2} \cdot \left\{(\mathbb{P}(Y=i)^{1/2} e_i)_{i\in[C]} \cdot \boldsymbol{R} \;\middle|\; \boldsymbol{R} \in O(C, K)\right\},$$

where $(e_i)_{i\in[K]}$ is the canonical basis of $\mathbb{R}^K$.

Proof. For the first part of the proof, remark that the invariance term in VICReg will be zero for any $\boldsymbol{z}$, so VICReg loss is minimized for any vector that minimized the variance-covariance term $\left\|\boldsymbol{z}\boldsymbol{z}^\top - I\right\|^2$, which is done for any unit vector.

For the second part, remark that $\boldsymbol{G} = \boldsymbol{Y}\boldsymbol{Y}^\top$ has C connected components, that are all full cliques, i.e. the adjacency is filled with one. As a consequence, the eigenvectors of $\boldsymbol{G}$ associated with non-zeros elements are exactly the $(\mathbf{1}_{\{}{}y_i = y)_{i\in[N]}$ for $y \in [C]$, and the corresponding eigenvalues are N_y where $N_y = \sum_{i\in[N]} \mathbf{1}_{\{y_i=y\}} = N\,\mathbb{P}(Y = i)$ are the number of element in the class $i \in [C]$. As a consequence, a square root of $\boldsymbol{G}$ is $N^{1/2}(\mathbb{P}(Y = i)^{1/2}\delta_{ij})_{i\in[C],j\in[K]}$, hence the proposition following the fact that all the square root of $\boldsymbol{G}$ are isomorphic. □

C. Additional experimental details

C.1. Essential Code

SSL Graph

```
G = torch.zeros(N * V, N * V)                                        #  X in R^{Np x D}, V views
i = torch.arange(0, N * V).repeat_interleave(V - 1)                  # row indices
j = (i + torch.arange(1, V).repeat(N * V) * N).remainder(p * V)      # column indices
G[i,j] = 1                                                           # unweighted graph
```

Sup Graph

```
Y = torch.nn.functional.one_hot(labels, num_classes=num_classes).float()
G = Y @ Y.T
```

VICReg.

```
C = torch.cov(Z.t())                                    # Z in R^{N x K}
reg_loss = torch.nn.functional.mse_loss(C, torch.eye(K))
reg_loss *= out_dim ** 2                                # correct for mean vs sum
i,j = G.nonzero(as_tuple=True)
inv_loss = torch.nn.functional.mse_loss(Z[i], Z[j])     # pairwise L2 weighted by G_{i,j}
inv_loss *= out_dim
loss = beta * inv_loss + reg_loss
```

SimCLR

```
Z_renorm = torch.nn.functional.normalize(Z, dim=1)          # Z \in \mathbb{R}^{N \times K}
cosim = Z_renorm @ Z_renorm.t() / tau                       # N x N matrix, tau is the temperature
mask = 1 - torch.eye(N, N, device=Z.device, dtype=Z.dtype)
loss = (G * (torch.logsumexp(cosim*mask, dim=1, keepdim=True) - cosim)).mean()
```

SCL

```
Z = torch.nn.functional.normalize(Z, dim=1)
loss = torch.nn.functional.mse_loss(G, Z@Z.T)
```

C.2. Controlled experiments

C.2.1 Setup

The train and test set of Figure 3 is shown on Fig. 5. The similarity graphs corresponding to the different snapshots on Fig. 3 are shown on Fig. 6. In all the experiments, we consider $K = C + 1 = 5$.

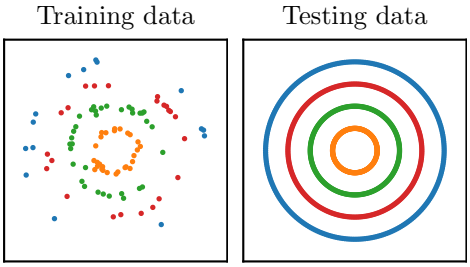

Figure 5. Setup for the controlled experiments of Fig. 3. The dataset is made of four concentric circles that corresponds to four different classes represented by different colors. The training dataset is made of one hundred random points, with some noise.

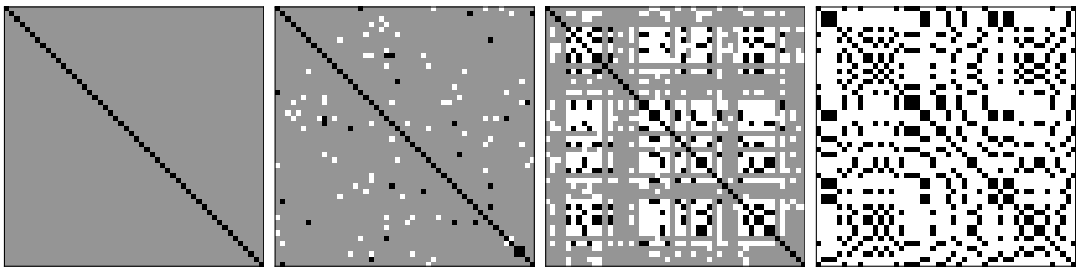

Figure 6. Graphs $\boldsymbol{G}$ corresponding to the different snapshot taken on Fig. 3. Grey indicates zeros, white indicates negative observations, and black means positive ones. The main strength of the active strategy in Algorithm 3 is that, by leveraging the underlying structure of the graph, is able to deduce much faster the full graph $\boldsymbol{G}$ than the naive passive implementation that only asks for random query pairs. Basically a positive observation is turned into many negative observations.

C.2.2 Contrastive vs. Non-Contrastive

Intuitively, it is useful to distinguish more explicitly between positive, negative and unknown relations, which we test on Fig. 7. To do so, the graph $\boldsymbol{G}$ is modified to encode semantically similar elements as positive edges, dissimilar ones as negative edges, while unknown relationships are going to be represented by zeros.

$$\boldsymbol{G}_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \sim \boldsymbol{x}_j \text{ has been observed,} \\ -1 & \text{if } \boldsymbol{x}_i \not\sim \boldsymbol{x}_j \text{ has been observed,} \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

One might wonder if this really improves performance. The comparison is the object of Fig. 7

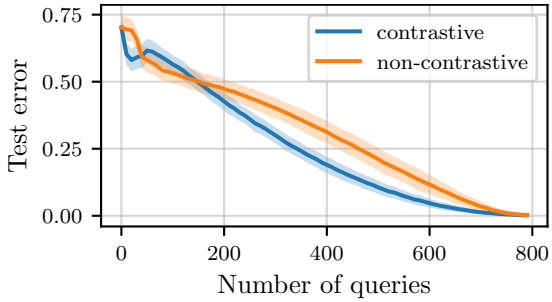

Figure 7. Comparison of contrastive ($\boldsymbol{G}_{ij} \in \{-1, 0, 1\}$) and non-contrastive ($\boldsymbol{G}_{ij} \in \{0, 1\}$) variation of VICReg with $N = 300$. The setting is the same as Fig. 3 with Algorithm 3. We remark the usefulness to distinguish between negative pairs and unknown pairs, although some instability issues seem to appear when few entries are known for the contrastive method.

C.2.3 The Benefits of Incorporating Known Labels

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. Let us denote by $\hat{\boldsymbol{Y}} \in \mathbb{R}^{N \times D}$ the one-hot matrix $(\boldsymbol{y}_i)_{i \in [N]}$ where $\boldsymbol{y}_i$ is the one-hot vector of the label y_i, such that if y_i is unknown $\boldsymbol{y}_i = 0$. The knowledge of some coefficients of the real $\boldsymbol{Y}$, leads to the knowledge of a few coefficient of $\boldsymbol{G}^{(\text{sup})} = \boldsymbol{Y}\boldsymbol{Y}^\top$, those could be added to the SSL graph to add useful connection deduced from the labels, leading to

$$\boldsymbol{G} = (1 - \alpha) \cdot \boldsymbol{G}^{(\text{ssl})} + \alpha \cdot \hat{\boldsymbol{Y}}\hat{\boldsymbol{Y}}^\top,$$

where $\alpha \in [0, 1]$ is a mixing coefficient stating how much the supervised information should weigh in the similarity matrix. Naively, we could set $\alpha = 1/2$, yet when only few labels are given this would destabilize the spectral decomposition of $\boldsymbol{G}$ too much, and we observe on Figure 4 that a small mixing coefficient is better. An explanation could be that the relations encoded by SSL are quite local and subtle, while the connections suggested by supervised learning are quite global and brutal on it suggested to fold the input space, hence need to be dampened when mixing the SSL and supervised graphs.

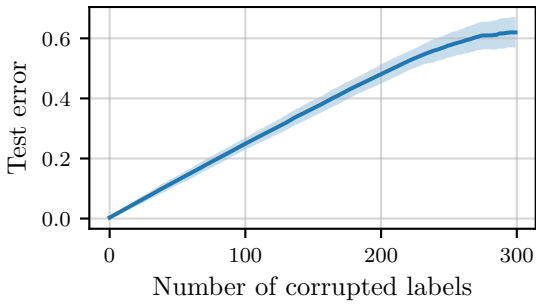

Figure 8. Study of the effect of labeling noise. The setup is the same as Fig. 3 yet with $N = 300$ points. We consider having full access to $\boldsymbol{Y}$ thus to $\boldsymbol{G} = \boldsymbol{Y}\boldsymbol{Y}^\top$ yet we assume that a certain number of labels y_i are corrupted. We see that the algorithm is somewhat robust to noise.

C.2.4 Robustness to noise

As mentioned in the last part of the paper, depending on the algorithm used, the effect of noise in queries answers might lead to dramatic performance loss. In the main text, we were careful to describe algorithms that are robust to noise. The effect of noise in the labels for Algorithm 3 is studied in Fig. 8. Because of its structure, noise in the query for Algorithm 3 is equivalent to noise in the label $\boldsymbol{y}$. This explains the setup of the figure.

C.2.5 The Importance to Recover Connected Components

An interesting experiments is provided by Fig. 9, which compares the test error and the number of connected components of the graph $\boldsymbol{G}$ as a function of the number of missing entries of $\boldsymbol{G} = \boldsymbol{G}^{(\text{sup})}$. In our synthetic experiment, $\boldsymbol{G}^{(\text{sup})}$ has four connected components corresponding to the four classes in the dataset, e.g. Fig. 2. Typically, based on transitivity of the similarity relation $\sim$, one can hope to only need $O(1/N) = O(NC/N^2)$ queries, i.e. reconstructed entries of $\boldsymbol{G}$, to have a good sense of the global $\boldsymbol{G}$, hence to learn f_θ. Moreover, on Fig. 9, the test error can be relatively well-predicted by the number of connected components of the graph $\boldsymbol{G}$. This suggests creative ways to design active learning strategies based on search to optimize the number of connected components of $\boldsymbol{G}$. However, leveraging transitivity of the similarity relationship to fill $\boldsymbol{G}$ efficiently might be limited when queries answers are noisy, although literature on error correcting codes might be useful [47, 19]. Moreover, the binary (and transitive) nature of similarity can be questioned when SSL sometimes uses DA that provides iconoclast unrealistic images, and one might prefer to assign similarity scores. Problems that do not occur with the transitivity agnostic Algorithm 3.

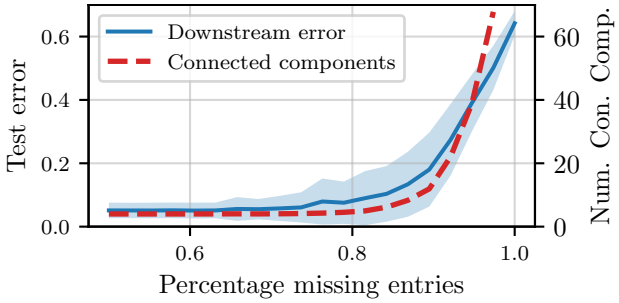

Figure 9. Comparison between the test error and number of connected components in the graph $\boldsymbol{G}$ as a function of the percentage of missing entries in $\boldsymbol{G}$. The test error is reported as in Fig. 3, but it is reported as a function of missing entries of the supervised learning graph $\boldsymbol{G}^{(\text{sup})}$. The standard deviation for the red curve is not represented here as the number of connected components is highly concentrated around its mean.

C.2.6 Mixture of Gaussian

One can question if the findings presented so far are specific to the concentric circles datasets. In order to assert the validity of those findings, we consider a second dataset, made of mixture of Gaussian, formally

$$\boldsymbol{X} = \boldsymbol{Y} + \sigma \boldsymbol{E}, \qquad \text{where} \qquad \boldsymbol{E}_{ij} \sim \mathcal{N}(0, 1),$$

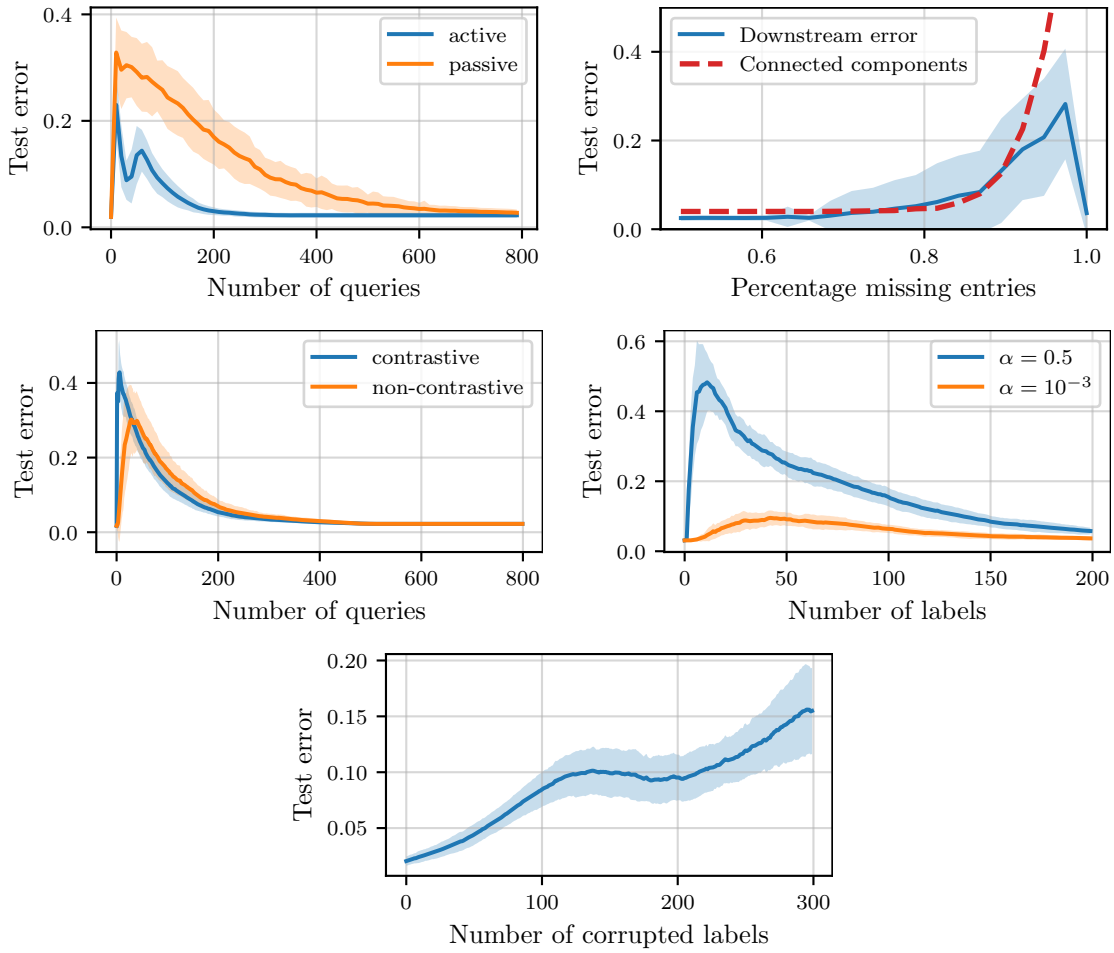

Figure 10. Same figures as before with a mixture of Gaussians dataset. The mixture dataset has the particularity that the downstream task can be solved with any orthogonal basis of $\mathbb{R}^C$. When no queries has been made, $\boldsymbol{G} = I_N$, and the spectral decomposition of this graph will lead to a representation that can solve the downstream task, explaining why when no queries have been made, or when all the entries of $\boldsymbol{G}$ are removed, the downstream task can be solved.

given a label $y \in [C]$, $\boldsymbol{x}$ is generated according to $\mathcal{N}(\boldsymbol{e}_y, \sigma I_C)$. The results are reported on Fig. 10 with $\sigma = .3$.

C.3. Real-world experiment

While it is hard to control all the factors that come into play when training a neural network on real data, our experiments suggested that what we have seen in controlled experiments transfer to real-world problems. In particular, we consider the CIFAR-10 dataset, with a resnet 18 architecture. A first stage was representation learning, where we used the VICReg loss to learn representation with the CIFAR-10 training set. In particular, we removed the classifier head of the resnet and replaced it with two fully connected layers with batch norms. The number of output dimensions was set to $K = 16$, and the number of hidden neurons was set to $4K$. After the representation was learned, we replaced the classifier head by a linear layer with $K = C$ output dimension and fit this last layer on the CIFAR-10 training set. The resulting network was then tested on the CIFAR-10 testing set. Regarding hyperparameters (network, DA, optimizer), we fixed them in accordance with tutorial online (in particular the pytorch-lightning tutorial) in order to achieve high performance results on CIFAR with SSL. In our first experiments, we stopped after two epochs of training for pretraining (since the output dimension is quite small, there is no need to go really far away in training), and twenty epochs downstream. The pretraining task consisted in all the training data of CIFAR-10 tackle, We found that the representation learned with SSL was achieving 28 % accuracy on CIFAR-10 with linear probing, while the representation learned directly with the supervised graph was achieving 63 % accuracy. In the meanwhile, training a resnet with classifier head to be made of 60 hidden neurons and 10 output dimensions with the ground truth labels and the mean-square error in the exact same setting leads to a performance of 63% too. In other terms, in these simple

experiments, one can use the VICReg technique we derived here, or the MSE loss and get the same performance. Training for tens epochs for the upstream task (the minimization of the VICReg loss), and one hundred for the downstream one (the linear head fitting), we improved performance to 62% for SSL and 66% for the supervised learning graph. Furthermore, we did not perform extensive hyperparameter tuning, which suggests that the supervised learning performance could be even more competitive, since we took parameters that are known to be good for the self-supervised learning techniques. All the code is available to reproduce our experiments.