



HAL
open science

Self-supervised learning with rotation-invariant kernels

Léon Zheng, Gilles Puy, Elisa Riccietti, Patrick Pérez, Rémi Gribonval

► **To cite this version:**

Léon Zheng, Gilles Puy, Elisa Riccietti, Patrick Pérez, Rémi Gribonval. Self-supervised learning with rotation-invariant kernels. 2022. hal-03738466v1

HAL Id: hal-03738466

<https://hal.science/hal-03738466v1>

Preprint submitted on 26 Jul 2022 (v1), last revised 6 Mar 2023 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Self-supervised learning with rotation-invariant kernels

Léon Zheng^{1,2} Gilles Puy¹ Elisa Riccietti² Patrick Pérez¹ Rémi Gribonval²

¹ valeo.ai, Paris, France

² Univ. de Lyon, ENS de Lyon, UCBL, CNRS, Inria, LIP, F-69342, Lyon Cedex 07, France

Abstract

A major paradigm for learning image representations in a self-supervised manner is to learn a model that is invariant to some predefined image transformations (cropping, blurring, color jittering, etc.), while regularizing the embedding distribution to avoid learning a degenerate solution. Our first contribution is to propose a general kernel framework to design a generic regularization loss that promotes the embedding distribution to be close to the uniform distribution on the hypersphere, with respect to the maximum mean discrepancy pseudometric. Our framework uses rotation-invariant kernels defined on the hypersphere, also known as dot-product kernels. Our second contribution is to show that this flexible kernel approach encompasses several existing self-supervised learning methods, including uniformity-based and information-maximization methods. Finally, by exploring empirically several kernel choices, our experiments demonstrate that using a truncated rotation-invariant kernel provides competitive results compared to state-of-the-art methods, and we show practical situations where our method benefits from the kernel trick to reduce computational complexity.

1 Introduction

Self-supervised learning is a promising approach for learning visual representations: recent methods [8; 9; 11; 22; 26; 31; 68] reach the performance of supervised pretraining in terms of quality for transfer learning [23; 49; 66] in many downstream tasks, like classification, object detection, semantic segmentation, etc. These methods rely on some prior knowledge on images: the semantic of an image is invariant [45] to some *small* transformations of the image, such as cropping, blurring, color jittering, etc. One way to design an objective function that encodes such an invariance property is to enforce two different augmentations of the same image to have a similar representation (or *embedding*) when they are encoded by the neural network [2; 9; 10; 26; 60; 68]. However, the main issue with this kind of objective function is to avoid an undesirable loss of information [35] where, e.g., the network learns to represent all images by the same constant representation. Hence, one of the main challenges in self-supervised learning is to propose an efficient way to *regularize* the embedding distribution in order to avoid such a *collapse*.

We propose a general kernel framework to design such a regularization. Intuitively, our regularization for self-supervision avoids collapse by promoting the embedding distribution to be close to the *uniform* distribution on the hypersphere, with respect to the *maximum mean discrepancy* (MMD) pseudometric [25], a distance on the space of probability measures based on the notion of embedding probabilities in a reproducing kernel Hilbert space (RKHS). Inspired by statistical tests for uniformity that are rotation-invariant [20], we choose to compute this pseudometric using *rotation-invariant*

Code [69]: <https://github.com/valeoai/sfrik>

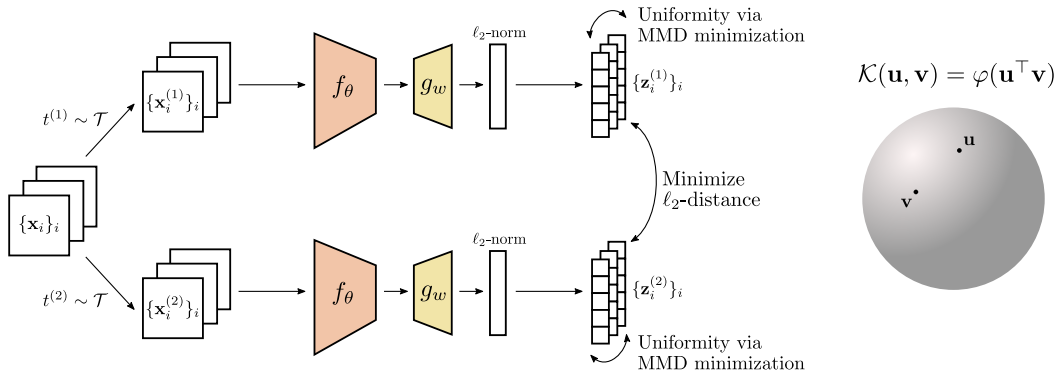


Figure 1: **Self-supervised learning with rotation-invariant kernels.** Our method has a Siamese architecture: the backbone f_θ followed by a projection head g_w with shared weights θ and w in both branches encodes two different views $\mathbf{x}_i^{(v)} = t^{(v)}(\mathbf{x}_i)$ of a same image \mathbf{x}_i (with $v \in \{1, 2\}$, where $t^{(v)} \sim \mathcal{T}$ is a random image augmentation) into two normalized embeddings $\mathbf{z}_i^{(1)}$ and $\mathbf{z}_i^{(2)}$. An invariance property on the learned representations is enforced by minimizing the ℓ_2 -distance between $\mathbf{z}_i^{(1)}$ and $\mathbf{z}_i^{(2)}$. To avoid collapse, the embedding distribution is regularized to be close to the uniform distribution on the hypersphere, in the sense of the maximum mean discrepancy associated to a rotation-invariant kernel $\mathcal{K}(\mathbf{u}, \mathbf{v}) = \varphi(\mathbf{u}^\top \mathbf{v})$ defined for \mathbf{u}, \mathbf{v} on the hypersphere.

kernels defined on the hypersphere (also known as dot-product kernels), i.e., kernels for which the evaluation for two vectors depends only on their inner product [56]. Our regularization of the embedding distribution works for *any* positive definite rotation-invariant kernel. Hence, we show that our kernel framework encompasses several regularization losses previously proposed in the literature. For instance, choosing the radial basis function kernel yields the uniformity loss in [60], and choosing a linear combination of the linear kernel and the quadratic kernel yields a regularization term that promotes the covariance matrix of the embedding distribution to be proportional to the identity matrix, like in information-maximization methods [2; 18].

Based on this framework, we explore empirically the quality of different rotation-invariant kernels defined on the hypersphere for self-supervision. Our experiments show that truncated kernels of the form $\mathcal{K}(\mathbf{u}, \mathbf{v}) = \sum_{\ell=0}^L b_\ell P_\ell(q; \mathbf{u}^\top \mathbf{v})$ work the best, with $L \in \{2, 3\}$, $b_\ell \geq 0$ for $\ell \in \{0, \dots, L\}$, where $P_\ell(q; \cdot)$ denotes the Legendre polynomial of order ℓ , dimension q . To our knowledge, this kernel choice has not been considered in previous self-supervised learning methods. Therefore, we introduce our self-supervised learning method SFRIK (Self-supervised learning with Rotation-Invariant Kernels, pronounced like “spheric”), which regularizes the embedding distribution to be close to the uniform distribution with respect to the MMD pseudometric associated to such a truncated kernel, as summarized in Figure 1. Our contributions are as follows:

- We propose a general kernel framework to design regularizers for self-supervised learning of visual representations, based on the MMD pseudometric and rotation-invariant kernels.
- We show that this general framework encompasses several previous self-supervised learning methods, like uniformity-based [60] and information-maximization [2] methods.
- We show experimentally that SFRIK with a truncated kernel is competitive compared to other self-supervised learning methods on several common evaluation protocols.
- We show that SFRIK benefits from large embedding dimensions without requiring a large batch size, and scales easily with the embedding dimension using the kernel trick.

Typically, while information-maximization losses [2; 68] have a quadratic complexity in the embedding dimension, the complexity of SFRIK’s loss is quadratic in the batch size and *linear* in the embedding dimension. Finally, our kernel approach opens perspectives for applying kernel approximation techniques (like random features [52] or Nyström approximations [62]) in self-supervised learning, to further reduce the computational complexity.

2 Related work

A major paradigm for self-supervised learning is to learn image representations that are *invariant* to image transformations (cropping, blurring, color jittering, etc.) [45]. We first describe two existing approaches for learning such an invariance property.

Instance-discrimination methods The first way is to rely on an *instance classification* approach where the representations are learned to discriminate image instances [63]. Typically, *contrastive learning* methods [10; 11; 31; 32; 33; 48] propose to discriminate instances within a batch of sampled images using the noise contrastive estimator (NCE) [27], by attracting embeddings of transformed images coming from the same image instance, and repulsing embeddings coming from different image instances. In practice, this estimator needs a large number of image representations in order to achieve good results, which requires either using a large batch size [10] or saving representations in a memory bank [31; 63]. In the limit of infinite samples, the contrastive loss is shown to behave asymptotically like an alignment and uniformity loss [60]. We show in fact that this uniformity loss can be derived from our general kernel framework for regularizing the embedding distribution.

Methods without instance-discrimination In distillation methods [9; 13; 21; 22; 26], a student network learns to predict the representations of a teacher network. In such methods, using various architecture tricks, like prediction head [26], stop-gradient [13], momentum encoder [9; 26], batch normalization [26] or centering [9], is shown empirically to be sufficient to avoid collapse without instance discrimination, even though it is not fully understood why these multiple factors induce a regularization during training [53; 58]. Instead of using these tricks, information-maximization methods [2; 18; 68] use a simple Siamese architecture and avoid collapse by maximizing the statistical information of a batch of embeddings. This is performed either with a whitening operation [18] or by adding explicitly a regularization term in the objective function to make the covariance [2] or the cross-correlation [68] matrix close to a scaled identity matrix. Our kernel method for self-supervision also adds an explicit regularization term in order to avoid using the above tricks. Moreover we show that SFRIK can also be interpreted as an information-maximization method.

We now discuss several works related to our proposed kernel framework for regularizing the embedding distribution on the hypersphere.

Uniformity on the hypersphere Our method is inspired from uniformity-based methods [3; 60] that avoid collapse by regularizing the embedding distribution to be as close as possible to the uniform distribution on the hypersphere, which has a high entropy. One way to perform this kind of regularization is to sample a fixed number of vectors uniformly at random on the hypersphere, and to align the learned representations to these vectors [3]. Another way is to maximize the average pairwise distance between embeddings using a radial basis function kernel [60], in the spirit of energy minimization methods that address the problem of distributing points evenly on the hypersphere [5; 28; 40]. Although high-entropy prior distributions other than the uniform distribution on the hypersphere –e.g., the uniform distribution on the hypercube or the normal distribution [12]– can be used for regularization, encoding images into ℓ_2 -normalized representations helps to stabilize training [14; 29; 39; 43; 50; 55; 59; 65].

Kernel methods In order to generalize previous uniformity losses like [60], our work proposes a kernel framework where we quantify proximity of the embedding distribution to the uniform distribution using the MMD pseudometric. The MMD was originally introduced for the two-sample problem [25]. It was also used to train generative models [7; 17; 38]. A natural choice of kernel for problems defined on the hypersphere is a rotation-invariant kernel (or dot-product, zonal kernel) [6; 16; 20; 42; 51; 56]. For instance, such kernels are typically used to design statistical tests for uniformity on the hypersphere [20], i.e., statistical tests for rejecting the null hypothesis where a given batch of normalized vectors is sampled from the uniform distribution on the hypersphere. Note that our kernel method for self-supervised learning is complementary to the kernel dependence maximization approach [37], in which the Hilbert-Schmidt independence criterion is used to maximize the dependency between image instances and their embedding.

3 Method description

Given an unlabeled dataset of images $\mathbf{x}_i \sim \mathbb{P}$, $i \in [N] := \{1, \dots, N\}$, sampled independently from a data distribution \mathbb{P} , the goal is to learn a *backbone* network f_θ parameterized by θ (e.g., a convolutional neural network) such that any new image $\mathbf{x} \sim \mathbb{P}$ is encoded by a good representation $f_\theta(\mathbf{x})$ whose quality is evaluated in several downstream tasks (see Section 4).

3.1 Invariance and uniformity for self-supervision

Our self-supervised learning method (see Figure 1) follows the principle of the recent methods [2; 8; 10; 26; 60; 68]. During self-supervised training, each image \mathbf{x}_i is augmented using two different random transformations $t^{(1)}$ and $t^{(2)}$ sampled from a distribution \mathcal{T} , which yields two views $\mathbf{x}_i^{(1)} := t^{(1)}(\mathbf{x}_i)$ and $\mathbf{x}_i^{(2)} := t^{(2)}(\mathbf{x}_i)$ of the image \mathbf{x}_i . Two representations $\mathbf{z}_i^{(v)}$ ($v = 1, 2$) are obtained by encoding each $\mathbf{x}_i^{(v)}$ with the backbone f_θ and ℓ_2 -normalizing the resulting feature vector. For a given subset of indices $I \subseteq [N]$, we write $\mathbf{Z}_I^{(v)} := \{\mathbf{z}_i^{(v)}\}_{i \in I}$. The backbone f_θ is trained by minimizing the total objective function:

$$\mathcal{L} = \mathbb{E}_{t^{(1)}, t^{(2)} \sim \mathcal{T}} \mathbb{E}_{I \subseteq [N]} \ell(\mathbf{Z}_I^{(1)}, \mathbf{Z}_I^{(2)}), \quad (1)$$

where batches I are drawn at random with a prescribed batch size, and the loss function ℓ is a weighted sum involving an alignment term ℓ_a and a uniformity term ℓ_u , in the spirit of [60]:

$$\ell(\mathbf{Z}_I^{(1)}, \mathbf{Z}_I^{(2)}) := \lambda \ell_a(\mathbf{Z}_I^{(1)}, \mathbf{Z}_I^{(2)}) + \mu (\ell_u(\mathbf{Z}_I^{(1)}) + \ell_u(\mathbf{Z}_I^{(2)})); \quad (2)$$

$\lambda, \mu > 0$ are hyperparameters, and we fix $\mu = 0.5$ without loss of generality. The alignment loss ℓ_a enforces the invariance property of the model, and is defined for a batch $I \subseteq [N]$ of cardinality $|I|$ as:

$$\ell_a(\mathbf{Z}_I^{(1)}, \mathbf{Z}_I^{(2)}) := \frac{1}{|I|} \sum_{i \in I} \|\mathbf{z}_i^{(1)} - \mathbf{z}_i^{(2)}\|_2^2. \quad (3)$$

Our main contribution is in the choice of the uniformity term ℓ_u , detailed in the rest of the section. Note that instead of applying the loss (2) to the output of f_θ (called image representation), we add a projection head g_w (a multi-layer perceptron) parameterized by w to the output of f_θ and apply the loss at the output of g_w (called image embedding). This is a common practice in self-supervision used, e.g., in [2; 8; 10; 26; 68], which improves the performance in the downstream tasks. Therefore, denoting \mathcal{S}^{q-1} the unit hypersphere in \mathbb{R}^q , the image embedding actually reads $\mathbf{z}_i^{(v)} := g_w \circ f_\theta(\mathbf{x}_i^{(v)}) / \|g_w \circ f_\theta(\mathbf{x}_i^{(v)})\|_2 \in \mathcal{S}^{q-1}$. Both g_w and f_θ are jointly trained without supervision by minimizing the objective function (1) using a stochastic mini-batch algorithm. After training, g_w is discarded and only f_θ is kept for the downstream tasks.

3.2 Uniformity loss via MMD minimization

We continue by explaining our generic kernel formulation of ℓ_u using the MMD pseudometric and rotation-invariant kernels. Then we provide examples of such kernels and describe our kernel choice.

3.2.1 MMD pseudometric and rotation-invariant kernels

Our uniformity loss ℓ_u relies on a pseudometric in the space of probability distributions based on a positive definite kernel \mathcal{K} defined on some space \mathcal{X} . Denoting \mathcal{H} the corresponding reproducing kernel Hilbert space (RKHS) with norm $\|\cdot\|_{\mathcal{H}}$, the MMD [4; 25] between two probability distributions $\mathbb{Q}_1, \mathbb{Q}_2$ on \mathcal{X} can be expressed as the distance in $\|\cdot\|_{\mathcal{H}}$ between the kernel mean embeddings [46]:

$$\text{MMD}(\mathbb{Q}_1, \mathbb{Q}_2) = \left\| \int_{\mathcal{X}} \mathcal{K}(\mathbf{u}, \cdot) d\mathbb{Q}_1(\mathbf{u}) - \int_{\mathcal{X}} \mathcal{K}(\mathbf{u}, \cdot) d\mathbb{Q}_2(\mathbf{u}) \right\|_{\mathcal{H}}. \quad (4)$$

We propose to use this pseudometric to measure the distance between the probability distribution of the embeddings $\mathbf{z}_i^{(v)}$ ($v = 1, 2$) and the uniform probability distribution on the hypersphere \mathcal{S}^{q-1} defined by $\mathbb{U} := \sigma_{q-1} / |\mathcal{S}^{q-1}|$, where σ_{q-1} denotes the normalized Hausdorff surface measure on \mathcal{S}^{q-1} , and $|\mathcal{S}^{q-1}| := \int_{\mathcal{S}^{q-1}} d\sigma_{q-1} = 2\pi^{\frac{q}{2}} / \Gamma(\frac{q}{2})$ is the surface area of \mathcal{S}^{q-1} , with Γ denoting the Gamma function. Intuitively, a good choice of kernel for measuring the distance (4) should distinguish

any probability distribution from the uniform distribution. Inspired by statistical tests for uniformity that are rotation-invariant [20], we propose to use *rotation-invariant* kernels, also known as *dot-product kernels* [56]. Rotation-invariant kernels on $\mathcal{X} := \mathcal{S}^{q-1}$ are of the form $\mathcal{K}(\mathbf{u}, \mathbf{v}) := \varphi(\mathbf{u}^\top \mathbf{v})$ with φ a continuous function defined on $[-1, 1]$ [56]. The following theorem characterizes the form of function φ that ensures that \mathcal{K} is positive definite, and thus that (4) is a valid pseudometric.

Theorem 1 ([54, Theorem 1]). *The kernel $\mathcal{K}(\mathbf{u}, \mathbf{v}) := \varphi(\mathbf{u}^\top \mathbf{v})$ on $\mathcal{X} := \mathcal{S}^{q-1}$ with φ continuous is positive definite if, and only if, the function φ admits an expansion:*

$$\varphi(t) = \sum_{\ell=0}^{+\infty} b_\ell P_\ell(q; t), \quad \text{with } b_\ell \geq 0, \quad (5)$$

where $P_\ell(q; t) := \ell! \Gamma\left(\frac{q-1}{2}\right) \sum_{k=0}^{\lfloor \frac{\ell}{2} \rfloor} \left(-\frac{1}{4}\right)^k \frac{(1-t^2)^k t^{\ell-2k}}{k! (\ell-2k)! \Gamma(k + \frac{q-1}{2})}$ is the Legendre (or Gegenbauer) polynomial of degree ℓ in dimension q [47, (2.32)].

As we are interested in measuring the distance between the embedding distribution and the uniform distribution on the hypersphere \mathbb{U} , we compute the kernel mean embedding of \mathbb{U} for a kernel satisfying the condition of Theorem 1 using the following known result (used, e.g., implicitly in [6]). As we could not locate a formal proof, we provide one in Appendix A.1.

Lemma 2. *Let $\mathcal{K}(\mathbf{u}, \mathbf{v}) := \varphi(\mathbf{u}^\top \mathbf{v})$ be a rotation-invariant kernel on $\mathcal{X} := \mathcal{S}^{q-1}$ where φ admits the expansion (5). The kernel mean embedding of the uniform distribution \mathbb{U} on \mathcal{S}^{q-1} is constant: $\int_{\mathcal{S}^{q-1}} \mathcal{K}(\mathbf{u}, \mathbf{v}) d\mathbb{U}(\mathbf{u}) = b_0 \in \mathbb{R}$ for all $\mathbf{v} \in \mathcal{S}^{q-1}$. The kernel mean embedding of any probability distribution \mathbb{Q} defined on the hypersphere satisfies: $\int_{\mathcal{S}^{q-1}} \mathcal{K}(\mathbf{u}, \cdot) d\mathbb{Q}(\mathbf{u}) = b_0 + \int_{\mathcal{S}^{q-1}} \tilde{\mathcal{K}}(\mathbf{u}, \cdot) d\mathbb{Q}(\mathbf{u})$, where $\tilde{\mathcal{K}}(\mathbf{u}, \mathbf{v}) := \tilde{\varphi}(\mathbf{u}^\top \mathbf{v})$ for any $\mathbf{u}, \mathbf{v} \in \mathcal{S}^{q-1}$ with $\tilde{\varphi} := \sum_{\ell=1}^{+\infty} b_\ell P_\ell(q; \cdot)$.*

Using Lemma 2 in (4) yields $\text{MMD}(\mathbb{Q}, \mathbb{U}) = \left\| \int_{\mathcal{S}^{q-1}} \tilde{\mathcal{K}}(\mathbf{u}, \cdot) d\mathbb{Q}(\mathbf{u}) \right\|_{\mathcal{H}}$ for any \mathbb{Q} on \mathcal{S}^{q-1} . Then, by the reproducing property in the RKHS \mathcal{H} , the squared MMD satisfies, for any rotation-invariant \mathcal{K} verifying the condition of Theorem 1:

$$\text{MMD}^2(\mathbb{Q}, \mathbb{U}) = \mathbb{E}_{\mathbf{z}, \mathbf{z}' \sim \mathbb{Q}} \left[\tilde{\mathcal{K}}(\mathbf{z}, \mathbf{z}') \right], \quad \text{with } \mathbf{z}, \mathbf{z}' \text{ i.i.d.} \quad (6)$$

3.2.2 Estimator of the squared MMD and kernel choices

Our uniformity loss ℓ_u for self-supervision is a biased estimator [7; 25] of $\text{MMD}^2(\mathbb{Q}, \mathbb{U})$ in (6). Given a batch $\mathbf{Z}_I := \{\mathbf{z}_i\}_{i \in I}$ sampled from \mathbb{Q} , the uniformity loss is defined as:

$$\ell_u(\mathbf{Z}_I) = \widehat{\text{MMD}}^2(\mathbb{Q}, \mathbb{U}; \{\mathbf{Z}_I\}) := \frac{1}{|I|^2} \sum_{i \in I} \sum_{i' \in I} \tilde{\mathcal{K}}(\mathbf{z}_i, \mathbf{z}_{i'}) = \frac{1}{|I|^2} \sum_{i \in I} \sum_{i' \in I} \tilde{\varphi}(\mathbf{z}_i^\top \mathbf{z}_{i'}). \quad (7)$$

In our framework, any rotation-invariant kernel satisfying the condition of Theorem 1 can be used to compute the uniformity term (7) and train a self-supervised model by minimizing (1). This uniformity term can be interpreted as an energy functional [6], where $\tilde{\mathcal{K}}$ quantifies the pairwise energy. Intuitively, minimizing such an energy functional tends to scatter evenly the embeddings on the hypersphere. We now give examples of kernels that can be used for our uniformity term. This illustrates that our framework offers a unification of several strategies for self-supervision.

Example 1: Radial basis function (RBF) kernel. Using an RBF kernel $\mathcal{K}(\mathbf{u}, \mathbf{v}) = e^{-t\|\mathbf{u}-\mathbf{v}\|_2^2}$ (with $t > 0$) in our uniformity term ℓ_u yields the regularization term from [60], with the only difference that [60] uses the logarithm of the energy functional as their uniformity loss.

Example 2: Generalized distance kernel [6]. This kernel is defined as $\mathcal{K}(\mathbf{u}, \mathbf{v}) := C - \|\mathbf{u} - \mathbf{v}\|_2^{2s-q+1}$ with $(q-1)/2 < s < (q+1)/2$ and a constant C sufficiently large. A variation of this choice of kernel is, e.g., used in [64] in a hard-contrastive loss for self-supervision on point clouds.

Example 3: Truncations of the Laplace-Fourier series [6]. A truncated kernel up to order L corresponds to a kernel $\mathcal{K}(\mathbf{u}, \mathbf{v}) = \sum_{\ell=0}^L b_\ell P_\ell(q; \mathbf{u}^\top \mathbf{v})$, with $b_\ell \geq 0$ for $\ell = 0, \dots, L$. It admits a closed-form expression given by the definition of Legendre polynomials $P_\ell(q, t)$ in Theorem 1. For instance: $P_1(q, t) = t$, $P_2(q, t) = (qt^2 - 1)/(q - 1)$ and $P_3(q, t) = ((q + 2)t^3 - 3t)/(q - 1)$. As a contribution, we explore numerically this choice of kernel for self-supervision in Section 4, since it has never been considered in previous self-supervised learning methods.

The expansion of φ in Legendre polynomials (5) for the RBF (Example 1) and the generalized distance kernel (Example 2) verifies $b_\ell > 0$ for each integer ℓ (see Appendix A.2). By [44, Theorem 10], this is a necessary and sufficient condition for a rotation-invariant kernel to be universal [44], and universality is a sufficient condition for injectivity of the corresponding kernel mean embedding mapping (i.e., the kernel is characteristic [19]). The benefit of this property is to guarantee that the uniform distribution \mathbb{U} is the unique solution to the minimization problem: $\min \{ \text{MMD}(\mathbb{Q}, \mathbb{U}) \mid \mathbb{Q} \text{ is a probability distribution on } \mathcal{S}^{q-1} \}$. In contrast, the truncated rotation-invariant kernel up to an order L (Example 3) is not universal. Yet, our experiments in Section 4 show that truncated kernels up to order $L = 2, 3$ provide better results than, e.g., [60] whose uniformity loss is based on the RBF kernel.

In summary, the uniformity loss in our method, called SFRIK, corresponds to (7) with a truncated kernel up to order $L = 3$ and satisfies:

$$\ell_u(\{\mathbf{z}_i\}_{i \in I}) = \frac{1}{|I|^2} \sum_{i \in I} \sum_{i' \in I} \left(b_1 \mathbf{z}_i^\top \mathbf{z}_{i'} + b_2 \frac{q(\mathbf{z}_i^\top \mathbf{z}_{i'})^2 - 1}{q-1} + b_3 \frac{(q+2)(\mathbf{z}_i^\top \mathbf{z}_{i'})^3 - 3\mathbf{z}_i^\top \mathbf{z}_{i'}}{q-1} \right), \quad (8)$$

where $b_\ell \geq 0$, $\ell = 1, 2, 3$, are hyperparameters, and q is the dimension of the image embedding \mathbf{z}_i .

3.3 Connection between SFRIK and VICReg [2]

In VICReg [2], the regularization loss minimized for self-supervision is a weighted sum of a *variance* term and a *covariance* term satisfying, respectively:

$$v(\mathbf{Z}_I) := \frac{1}{q} \sum_{j=1}^q \max(0, \gamma - \sqrt{\text{Var}(z^j) + \varepsilon}), \quad c(\mathbf{Z}_I) := \frac{1}{q} \sum_{1 \leq j \neq j' \leq q} [C(\mathbf{Z}_I)]_{j,j'}^2, \quad (9)$$

for a batch of image embeddings $\mathbf{Z}_I := \{\mathbf{z}_i\}_{i \in I}$, where z^j denotes the j -th coordinate of a (random) vector \mathbf{z} and ε is a fixed small scalar. The *variance* term $v(\mathbf{Z}_I)$ enforces the empirical variance $\text{Var}(z^j) := \frac{1}{|I|-1} \sum_{i \in I} (z_i^j - \bar{z}^j)^2$ in each coordinate $j = 1, \dots, q$ to be above a certain threshold $\gamma^2 > 0$ (here $\bar{\mathbf{z}}$ is the empirical mean of \mathbf{Z}_I). The *covariance* term $c(\cdot)$ enforces the non-diagonal entries of the empirical covariance matrix $C(\mathbf{Z}_I) := \frac{1}{|I|-1} \sum_{i \in I} (\mathbf{z}_i - \bar{\mathbf{z}})(\mathbf{z}_i - \bar{\mathbf{z}})^\top$ to be zero.

In order to connect VICReg and SFRIK, let us consider for simplicity a truncated kernel $\tilde{\mathcal{K}}(\mathbf{u}, \mathbf{v}) = \sum_{\ell=1}^L b_\ell P_\ell(q; \mathbf{u}^\top \mathbf{v})$ of order $L = 2$ (the reasoning would be the same if the kernel was not truncated), and assume $b_1, b_2 > 0$. By the addition theorem [47, Theorem 2, §1], there exists a feature map $\Phi : \mathcal{S}^{q-1} \rightarrow \mathbb{R}^m$ involving an orthonormal basis of spherical harmonics (homogeneous harmonic polynomials restricted to the hypersphere) of order 1 and 2 such that $\Phi(\mathbf{u})^\top \Phi(\mathbf{v}) = \tilde{\mathcal{K}}(\mathbf{u}, \mathbf{v})$. Hence, the kernel embedding of a distribution in the associated RKHS contains its first and second-order moments (see Appendix A.3). Therefore, denoting $N(q, \ell)$ the dimension of the space of spherical harmonics of order ℓ , dimension q , and defining $\Phi_\ell : \mathcal{S}^{q-1} \rightarrow \mathbb{R}^{N(q, \ell)}$, $\mathbf{z} \mapsto (Y_{\ell, k}(\mathbf{z}))_{k=1}^{N(q, \ell)}$ for $\ell \in \{1, 2\}$ with

$$\begin{aligned} \{Y_{1, k}\}_{k=1}^{N(q, 1)} &:= \{\mathbf{u} \mapsto u^j \mid 1 \leq j \leq q\}, \\ \{Y_{2, k}\}_{k=1}^{N(q, 2)} &:= \left\{ \mathbf{u} \mapsto u^j u^{j'} \mid 1 \leq j < j' \leq q \right\} \cup \left\{ \mathbf{u} \mapsto (u^j)^2 - \frac{1}{q} \mid 2 \leq j \leq q \right\}, \end{aligned} \quad (10)$$

it is possible to show (see Appendix A.3) that the squared MMD (6) can be written as

$$\text{MMD}^2(\mathbb{Q}, \mathbb{U}) = a_1 \|\mathbf{M}_1 \mathbb{E}_{\mathbf{z} \sim \mathbb{Q}}[\Phi_1(\mathbf{z})]\|_2^2 + a_2 \|\mathbf{M}_2 \mathbb{E}_{\mathbf{z} \sim \mathbb{Q}}[\Phi_2(\mathbf{z})]\|_2^2, \quad (11)$$

where $a_\ell := b_\ell |\mathcal{S}^{q-1}| / N(q, \ell)$ for $\ell \in \{1, 2\}$, and $\mathbf{M}_1, \mathbf{M}_2$ are two lower triangular matrices with nonzero diagonal entries. Hence, when \mathbb{Q} plays the role of the embedding distribution during self-supervised training, minimizing $\text{MMD}^2(\mathbb{Q}, \mathbb{U})$ in (11) as we propose for regularizing the embedding distribution promotes its expectation $\mathbb{E}_{\mathbf{z} \sim \mathbb{Q}}[\mathbf{z}]$ and its autocorrelation matrix $\mathbb{E}_{\mathbf{z} \sim \mathbb{Q}}[\mathbf{z}\mathbf{z}^\top]$ to be close to 0 and $q^{-1}\mathbf{I}_q$ respectively, where \mathbf{I}_q is the identity matrix. When $\text{MMD}^2(\mathbb{Q}, \mathbb{U}) = 0$, the covariance matrix is equal to $\mathbb{E}[(\mathbf{z} - \mathbb{E}[\mathbf{z}])(\mathbf{z} - \mathbb{E}[\mathbf{z}])^\top] = \mathbb{E}[\mathbf{z}\mathbf{z}^\top] - \mathbb{E}[\mathbf{z}]\mathbb{E}[\mathbf{z}]^\top = \frac{1}{q}\mathbf{I}_q$ because $b_1, b_2 > 0$ and the two terms on the right-hand side of (11) are null.

In conclusion, the regularization both in VICReg and SFRIK induces the embedding distribution to have a covariance matrix with zero non-diagonal entries. The diagonal entries of the covariance matrix are encouraged to be equal to $1/q$ in SFRIK, and greater than γ^2 in VICReg (we recall that the image embeddings $\{\mathbf{z}_i\}_{i \in I}$ are not ℓ_2 -normalized in VICReg). However, one difference in terms of regularization behavior is that SFRIK encourages the expectation of the embedding distribution to be zero, as shown in the first term of (11). This is not the case for VICReg, as we can see in (9).

Finally, the memory and computational complexities for computing the uniformity term (8) in SFRIK are $\mathcal{O}(|I|^2)$ and $\mathcal{O}(q|I|^2)$, as opposed to $\mathcal{O}(q^2)$ and $\mathcal{O}(q^2|I|)$ for the variance and covariance terms (9) in VICReg. In the setting where SFRIK and VICReg work best, i.e., larger dimension q and smaller batch size $|I|$, SFRIK has the lowest memory and computational complexities. This computational advantage is due to the kernel trick and it is illustrated in Section 4.

4 Experiments

We start by exploring empirically several choices of kernel weights for the uniformity term (8), and show in a controlled experimental setting that SFRIK’s regularization loss yields better results than existing alternatives. Then, we evaluate the quality of the image representations pretrained by SFRIK on ImageNet-1000 [15] on several downstream tasks.

4.1 Experimental setting

The backbone f_θ is either a ResNet-18 or a ResNet-50 [30], depending on the experiment. Following [2; 68], the projection head g_w is a three-layer MLP made of two hidden layers with ReLU activation and batch normalization [34], and a linear output layer. Unless otherwise specified, the size (number of neurons) of the two hidden layers is the same as the one, denoted q , of the output layer and the default value is $q = 8192$. The augmentations used for transforming images into views are the same as the ones used in [2]. The backbone and the projection head are trained with a LARS optimizer [67]. The weight decay is fixed at 10^{-6} . The learning rate scheduling starts with 10 warm-up epochs [24] with a linear increase from 0 to the “initial” learning rate $\text{initial_lr} = \text{base_lr} * \text{bs}/256$ (base_lr is called the base learning rate [24] and bs is the batch size), followed by a cosine decay [41] with a final learning rate 1000 times smaller than initial_lr .

For pretraining, we consider a subset of 20% of ImageNet [15] (denoted by IN20%), like in [22], and 100% of ImageNet. In IN20%, we keep all the 1000 classes but only 260 images per class. This reduced dataset allows for extensive hyperparameter tuning of all methods for fair comparisons in the shared experimental setting. We follow the usual evaluation protocol like, e.g., in [2; 9], by measuring the image representation quality by linear probing, transfer learning and semi-supervised learning.

4.2 Impact of different truncated rotation-invariant kernels on the regularization

We compare the performance of SFRIK with truncated rotation-invariant kernels of order $L = 1, 2, 3$ with other self-supervised methods that have the same Siamese architecture and the same form of training loss $\lambda \ell_a(\mathbf{Z}_I^{(1)}, \mathbf{Z}_I^{(2)}) + \mu \ell_r(\mathbf{Z}_I^{(1)}, \mathbf{Z}_I^{(2)})$ as SFRIK: $\ell_r(\mathbf{Z}_I^{(1)}, \mathbf{Z}_I^{(2)}) := \ell_u(\mathbf{Z}_I^{(1)}) + \ell_u(\mathbf{Z}_I^{(2)})$ in (2). These methods are SimCLR [10], Alignment & Uniformity on the Hypersphere (AUH) [60] and VICReg [2], for which we give the expression of ℓ_r in the Appendix A.4.

Protocol. To isolate the impact of the regularization loss ℓ_r on the quality of the learned image representations, we (re)implement all methods in the setting of Section 4.1, to get rid of the influence of other design choices, like image augmentations or projection head architecture. For fair comparisons, we fix the batch size at 2048, and tune the base learning rate and hyperparameters specific to each method’s loss. We also compare different embedding dimension $q \in \{1024, 2048, 4096, 8192\}$. We pretrain a ResNet-18 on IN20% for 100 epochs with all methods. For each method, we perform a grid search on its set of hyperparameters for a fair comparison. The hyperparameters are selected on a *separate* validation set that consists of *another* 20% subset of the ImageNet train set: we select the hyperparameter values that yield the highest top-1 accuracy obtained by weighted kNN-classification ($k = 20$) [9; 63] on this validation set. We finally report the evaluation results by linear probing on the usual ImageNet validation set, which is never seen during hyperparameter tuning. Note that the

Table 1: **Linear probing on IN20% (top-1 accuracy)** at different embedding dimensions q . All methods were pretrained on IN20% with a ResNet-18 for 100 epochs. We tuned all hyperparameters specific to each method and the learning rate. [†] indicates models that we retrained ourselves.

	SimCLR [†] [10]	AUH [†] [60]	VICReg [†] [2]	SFRIK		
				$L = 1$	$L = 2$	$L = 3$
$q = 1024$	45.2	45.3	40.6	-	45.2	-
$q = 2048$	45.8	45.9	44.0	-	45.9	-
$q = 4096$	46.0	46.7	44.9	-	46.9	-
$q = 8192$	46.1	46.8	46.0	27.7	47.0	47.5

learning rate for linear probing is also tuned on our separate validation set. See Appendix B.4 for more details on our hyperparameter tuning protocol.

Results. Table 1 shows that SFRIK at truncation order $L = 3$ surpasses SimCLR, AUH and VICReg by at least 0.7 points at $q = 8192$. The optimal truncation order in SFRIK is $L = 3$ at $q = 8192$. We did not try higher L as this requires much more hyperparameter tuning. During this tuning, we observed that SFRIK performs well when the weights b_2 and b_3 in (8) are larger than b_1 (e.g., $b_1 = 1, b_2 = 40, b_3 = 40$ for $q = 8192$). This contrasts with the RBF and the generalized distance kernel for which the weights b_ℓ decay polynomially with respect to ℓ (see Appendix A.2). This suggests that it is important to focus more on order 2, 3 than order 1 in the Legendre expansion (5) of the truncated kernel. We also remark that all methods benefit from a large embedding dimension q , including SimCLR which was originally introduced with a smaller dimension. In addition, we verified that increasing the batch size beyond 2048 does not improve the performance of SFRIK: the top-1 accuracy in linear probing at $L = 2$ drops by 0.7 points when the batch size increases to 4096.

4.3 Comparison with other self-supervised learning methods on ImageNet

We now pretrain a ResNet-50 with SFRIK on 100% of ImageNet [15]. We present in Table 2 the performance of SFRIK and several other methods on different downstream tasks. To isolate the impact of the training loss, we limit again the comparison to methods that have the same Siamese architecture and use the same image augmentations. In order to further explore the benefit of large embedding dimension, we consider a projection head with two hidden layers of size 8192, and an output layer of size $q \in \{8192, 16384, 32768\}$. Note that computing the covariance matrix for VICReg’s regularization loss (9) in these large dimensions requires respectively 0.3GB, 1.1GB and 4.3GB of memory when using float-32 precision, whereas SFRIK’s loss only requires 16MB of memory at a batch size of 2048.

Protocol. We follow the experimental setting of Section 4.1. For SFRIK, the batch size is fixed at 2048. The base learning rate is fixed at 0.4. To study the influence of the embedding dimension q and truncation order L , we compare four different settings: $(q, \lambda, b_1, b_2, b_3) = (8192, 4000, 1, 20, 0)$, $(q, \lambda, b_1, b_2, b_3) = (16384, 20000, 1, 40, 0)$, $(q, \lambda, b_1, b_2, b_3) = (32768, 40000, 1, 40, 0)$, and $(q, \lambda, b_1, b_2, b_3) = (32768, 40000, 1, 40, 40)$. The choice of hyperparameters is explained in the Appendix B.6. For comparison, we also pretrain a ResNet-50 with VICReg in the setting of Section 4.1. Similarly to the original paper [2], all the three layers of the projection head in VICReg are of size 8192, the alignment, variance and covariance weights are respectively 25, 25, 1, the base learning rate is 0.2, and the batch size is 2048. All pretrained backbones are then evaluated by: linear probing on ImageNet; linear probing on Places205 [70] in order to measure how the learned image representations generalize to an unseen dataset; semi-supervised learning on ImageNet as in [2] (the pretrained ResNet-50 backbones are fine-tuned during 20 epochs for the classification task on ImageNet using a subset of 1% or 10% of labeled images).

Results. Table 2 shows that SFRIK is on par with concurrent self-supervised learning methods on each downstream task, at the exception of the semi-supervised setting with 1% of labeled data. In particular, SFRIK has the best accuracy on linear probing on ImageNet. We confirm that using a truncation order $L = 3$ instead of $L = 2$ at dimension $q = 32768$ slightly improves the performance on the downstream tasks. We also observe that the performance of SFRIK on linear probing on

Table 2: **Linear probing on ImageNet and Places205, semi-supervised learning on ImageNet (top-1 and top-5 accuracy)**. All methods have been pretrained on ImageNet with a ResNet-50. We only include methods relying on a Siamese architecture with image augmentations limited to two views. The scores of the methods marked with * are taken from [13]. The score of VICReg[†] was obtained by retraining the model ourselves. For each downstream task, we highlight in bold the best score among all backbones pretrained on 200 epochs.

Method	Epochs	Linear probing		Semi-supervised	
		ImageNet	Places205	1% IN	10% IN
SimCLR* [10]	200	68.3 / -	-	-	-
SwAV* [8] (no multicrop)	200	69.1 / -	-	-	-
SimSiam [13]	200	70.0 / -	-	-	-
VICReg [†] [2] ($q = 8192$)	200	70.0 / 89.3	54.1 / 83.4	49.4 / 75.1	65.9 / 87.2
SFRIK ($L = 2, q = 8192$)	200	70.1 / 89.3	53.8 / 83.0	46.6 / 73.3	65.7 / 87.3
SFRIK ($L = 2, q = 16384$)	200	70.3 / 89.6	54.3 / 83.4	46.0 / 73.0	65.3 / 87.2
SFRIK ($L = 2, q = 32768$)	200	70.3 / 89.6	54.1 / 83.0	46.1 / 73.0	65.4 / 87.3
SFRIK ($L = 3, q = 32768$)	200	70.3 / 89.7	54.4 / 83.2	46.6 / 73.0	65.8 / 87.5
SFRIK ($L = 2, q = 8192$)	400	70.8 / 89.9	54.4 / 83.5	47.8 / 74.3	66.4 / 88.0

ImageNet increases with the embedding dimension q . This opens perspectives for studying alternative projection head architectures that can improve performance, possibly with very large hidden and output layers, and structured weight matrices [36] in order to further reduce memory cost.

Note that we did not include other methods using momentum encoder such as BYOL [26] or OBoW [22], or multiple image augmentations with cropping at different scales such as SwAV [8] in Table 2. These methods have a higher top-1 accuracy by linear probing on ImageNet than SFRIK for 200 pretraining epochs. Although we analyze our uniformity loss (8) in a Siamese architecture with only two views, SFRIK can be adapted to incorporate a momentum encoder and/or additional image augmentations. Such a study is left as future work.

5 Conclusion

We introduced SFRIK, a self-supervision method derived from our general framework to design regularization losses based on the MMD and rotation-invariant kernels. This framework encompasses existing uniformity-based [60] and information-maximization [2] methods. SFRIK with a truncated kernel surpasses all other alternatives in our controlled experiment setting with 20% of ImageNet. It also achieves results on par with other state-of-the-art methods when pretrained on ImageNet, while benefiting from a smaller computational complexity than VICReg [2] in a high-dimensional setting, thanks to the kernel trick.

Limitations. Future work can analyze how to avoid intensive hyperparameter tuning of the kernel weights b_ℓ in (8), which has negative environmental impact since it requires a lot of computing resources. To that end, alternative kernel choices for self-supervision can be explored, e.g., rotation-invariant kernel *without* restricting it on the hypersphere. To further reduce computational complexity in self-supervision, kernel approximation techniques can be applied to design well-adapted random features for rotation-invariant kernels, in the spirit of [42; 51; 52].

Acknowledgments and Disclosure of Funding

This project was funded by the CIFRE fellowship N°2020/1643 and supported in part by the AllegroAssai ANR project ANR-19-CHIA-0009. This work was granted access to the HPC resources of IDRIS under the allocation 20XX-AD011012940 made by GENCI.

References

- [1] Sheldon Axler, Paul Bourdon, and Ramey Wade. *Harmonic function theory*, volume 137. Springer Science & Business Media, 2013. 16
- [2] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022. 1, 2, 3, 4, 6, 7, 8, 9, 16, 17, 18, 19, 21, 22
- [3] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *ICML*, 2017. 3
- [4] Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006. 4
- [5] Sergiy V. Borodachov, Douglas P. Hardin, and Edward B. Saff. *Discrete energy on rectifiable sets*. Springer, 2019. 3
- [6] Johann Brauchart, Edward B. Saff, Ian H. Sloan, and Robert S. Womersley. QMC designs: Optimal order quasi Monte Carlo integration schemes on the sphere. *Mathematics of computation*, 83(290):2821–2851, 2014. 3, 5, 15
- [7] Francois-Xavier Briol, Alessandro Barp, Andrew B. Duncan, and Mark Girolami. Statistical inference for generative models with maximum mean discrepancy. *arXiv preprint arXiv:1906.05944*, June 2019. 3, 5
- [8] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 1, 4, 9, 22
- [9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 1, 3, 7, 19, 22
- [10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 3, 4, 7, 8, 9, 17, 21, 22
- [11] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020. 1, 3
- [12] Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. In *NeurIPS*, 2021. 3
- [13] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 3, 9
- [14] Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. In *UAI*, 2018. 3
- [15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 7, 8, 17, 22
- [16] Vincent Dutordoir, Nicolas Durrande, and James Hensman. Sparse Gaussian processes with spherical harmonic features. In *ICML*, 2020. 3
- [17] Gintare K. Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *UAI*, 2015. 3
- [18] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *ICML*, 2021. 2, 3
- [19] Kenji Fukumizu, Francis R. Bach, and Michael I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, 5(Jan):73–99, 2004. 6
- [20] Eduardo García-Portugués and Thomas Verdebout. An overview of uniformity tests on the hypersphere. *arXiv preprint arXiv:1804.00286*, 2018. 1, 3, 5
- [21] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Learning representations by predicting bags of visual words. In *CVPR*, 2020. 3
- [22] Spyros Gidaris, Andrei Bursuc, Gilles Puy, Nikos Komodakis, Matthieu Cord, and Patrick Pérez. Online bag-of-visual-words generation for self-supervised learning. In *CVPR*, 2021. 1, 3, 7, 9, 17, 18, 22
- [23] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1

- [24] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 7
- [25] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(1):723–773, 2012. 1, 3, 4, 5
- [26] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *NeurIPS*, 2020. 1, 3, 4, 9
- [27] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010. 3
- [28] Doug P. Hardin and Edward B. Saff. Minimal Riesz energy point configurations for rectifiable d-dimensional manifolds. *Advances in Mathematics*, 193(1):174–204, 2005. 3
- [29] Md. Abul Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, and Liming Chen. von Mises-Fisher mixture model-based deep learning: Application to face verification. *arXiv preprint arXiv:1706.04264*, 2017. 3
- [30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7
- [31] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 3
- [32] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *ICML*, 2020. 3
- [33] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019. 3
- [34] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 7
- [35] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *ICLR*, 2022. 1
- [36] Quoc Le, Tamás Sarlós, and Alex Smola. Fastfood-approximating kernel expansions in loglinear time. In *ICML*, 2013. 9
- [37] Yazhe Li, Roman Pogodin, Danica J. Sutherland, and Arthur Gretton. Self-supervised learning with kernel dependence maximization. In *NeurIPS*, 2021. 3
- [38] Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *ICML*, 2015. 3
- [39] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphreface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017. 3
- [40] Weiyang Liu, Rongmei Lin, Zhen Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song. Learning towards minimum hyperspherical energy. In *NeurIPS*, 2018. 3
- [41] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 7
- [42] Yueming Lyu. Spherical structured feature maps for kernel approximation. In *ICML*, 2017. 3, 9
- [43] Pascal Mettes, Elise van der Pol, and Cees Snoek. Hyperspherical prototype networks. In *NeurIPS*, 2019. 3
- [44] Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(12), 2006. 6
- [45] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020. 1, 3

- [46] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10 (1-2):1–141, 2017. 4
- [47] Claus Müller. *Analysis of spherical symmetries in Euclidean spaces*, volume 129. Springer Science & Business Media, 2012. 5, 6, 13, 14, 15, 16
- [48] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3
- [49] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. 1
- [50] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, 2015. 3
- [51] Jeffrey Pennington, Felix Xinnan X. Yu, and Sanjiv Kumar. Spherical random features for polynomial kernels. In *NeurIPS*, 2015. 3, 9
- [52] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NeurIPS*, 2007. 2, 9
- [53] Pierre H. Richemond, Jean-Bastien Grill, Florent Alché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel Smith, Soham De, Razvan Pascanu, Bilal Piot, and Michal Valko. BYOL works even without batch statistics. *arXiv preprint arXiv:2010.10241*, 2020. 3
- [54] Isaac J. Schoenberg. Positive definite functions on spheres. *Duke Mathematical Journal*, 9(1):96–108, 1942. 5
- [55] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 3
- [56] Alex Smola, Zoltán Ovári, and Robert C. Williamson. Regularization with dot-product kernels. In *NeurIPS*, 2000. 2, 3, 5
- [57] Murray R. Spiegel, Seymour Lipschutz, and John Liu. *Mathematical Handbook of Formulas and Tables*. Schaum’s Outlines, fourth edition, 2013. 15
- [58] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *ICML*, 2021. 3
- [59] Feng Wang, Xiang Xiang, Jian Cheng, and Alan L. Yuille. Normface: L2 hypersphere embedding for face verification. In *ACM Multimedia*, 2017. 3
- [60] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020. 1, 2, 3, 4, 5, 6, 7, 8, 9, 17, 21
- [61] Hermann Weyl. On the volume of tubes. *American Journal of Mathematics*, 61(2):461–472, 1939. 16
- [62] Christopher Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *NeurIPS*, 2000. 2
- [63] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 3, 7, 19
- [64] Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas Guibas, and Or Litany. PointContrast: Unsupervised pre-training for 3D point cloud understanding. In *ECCV*, 2020. 5
- [65] Jiacheng Xu and Greg Durrett. Spherical latent spaces for stable variational autoencoders. In *EMNLP*, 2018. 3
- [66] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NeurIPS*, 2014. 1
- [67] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 7
- [68] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021. 1, 2, 3, 4, 7
- [69] Léon Zheng, Gilles Puy, Elisa Riccietti, Patrick Pérez, and Rémi Gribonval. Code for reproducible research - Self-supervised learning with rotation-invariant kernels, July 2022. Code repository available at <https://hal.inria.fr/hal-03737572>, updates at <https://github.com/valeoai/sfrik>. 1
- [70] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using Places database. In *NeurIPS*, 2014. 8, 18, 22

A Theoretical results

We provide proofs and more details about the theoretical results in the main text.

A.1 Proof of Lemma 2

Consider a rotation-invariant kernel $\mathcal{K}(\mathbf{u}, \mathbf{v})$ defined on the hypersphere \mathcal{S}^{q-1} such that:

$$\mathcal{K}(\mathbf{u}, \mathbf{v}) = \sum_{\ell=0}^{+\infty} b_\ell P_\ell(q; \mathbf{u}^\top \mathbf{v}), \quad \forall \mathbf{u}, \mathbf{v} \in \mathcal{S}^{q-1}, \quad (12)$$

with weights $b_\ell \geq 0$ and $P_\ell(q; \cdot)$ the Legendre polynomial of order ℓ in dimension q . The proof of Lemma 2 relies on an orthonormal system of spherical harmonics. Let $\langle f, g \rangle_{(q)} := \int_{\mathcal{S}^{q-1}} f g d\sigma_{q-1}$ be the inner product in the space of continuous functions defined on \mathcal{S}^{q-1} and, for each $\ell \in \mathbb{N}$, consider $\{Y_{\ell,k} \mid k = 1, \dots, N(q, \ell)\}$ an orthonormal basis of spherical harmonics of order ℓ in dimension q (homogeneous harmonic polynomials in q variables restricted to \mathcal{S}^{q-1} , see e.g. [47] for more details), where $N(q, \ell)$ denotes the dimension of this space, which is by [47, Exercise 6, §3]:

$$N(q, \ell) = \begin{cases} q & \text{for } \ell = 1, \\ \frac{(2\ell+q-2)(\ell+q-3)!}{\ell!(q-2)!} & \text{for } \ell \geq 2. \end{cases} \quad (13)$$

By the addition theorem [47, Theorem 2, §1]:

$$\sum_{k=1}^{N(q, \ell)} Y_{\ell,k}(\mathbf{u}) Y_{\ell,k}(\mathbf{v}) = \frac{N(q, \ell)}{|\mathcal{S}^{q-1}|} P_\ell(q; \mathbf{u}^\top \mathbf{v}), \quad \mathbf{u}, \mathbf{v} \in \mathcal{S}^{q-1}. \quad (14)$$

Hence, the kernel $\mathcal{K}(\mathbf{u}, \mathbf{v})$ can be rewritten as:

$$\mathcal{K}(\mathbf{u}, \mathbf{v}) = \sum_{\ell=0}^{+\infty} \sum_{k=1}^{N(q, \ell)} \frac{b_\ell |\mathcal{S}^{q-1}|}{N(q, \ell)} Y_{\ell,k}(\mathbf{u}) Y_{\ell,k}(\mathbf{v}). \quad (15)$$

Since $\{Y_{\ell,k} \mid \ell = 0, \dots, +\infty, k = 1, \dots, N(q, \ell)\}$ is an orthonormal system for the inner product $\langle \cdot, \cdot \rangle_{(q)}$, and since $Y_{0,1}$ is constant on \mathcal{S}^{q-1} , we have for any integer ℓ and $k \in \{1, \dots, N(q, \ell)\}$ that:

$$\int_{\mathcal{S}^{q-1}} Y_{\ell,k} d\sigma_{q-1} = \frac{1}{Y_{0,1}} \langle Y_{\ell,k}, Y_{0,1} \rangle_{(q)} = \begin{cases} \frac{1}{Y_{0,1}} & \text{if } \ell = 0, k = 1 \\ 0 & \text{otherwise} \end{cases}. \quad (16)$$

Moreover $Y_{0,1} = 1/\sqrt{|\mathcal{S}^{q-1}|}$, because $1 = \langle Y_{0,1}, Y_{0,1} \rangle_{(q)} = \int_{\mathcal{S}^{q-1}} Y_{0,1}^2 d\sigma_{q-1} = Y_{0,1}^2 |\mathcal{S}^{q-1}|$. Therefore, the kernel mean embedding of the uniform distribution on the hypersphere $\mathbb{U} := \sigma_{q-1}/|\mathcal{S}^{q-1}|$ associated to the kernel \mathcal{K} is:

$$\begin{aligned} \forall \mathbf{v} \in \mathcal{S}^{q-1}, \int_{\mathcal{S}^{q-1}} \mathcal{K}(\mathbf{u}, \mathbf{v}) d\mathbb{U}(\mathbf{u}) &= \int_{\mathcal{S}^{q-1}} \sum_{\ell=0}^{+\infty} b_\ell P_\ell(q; \mathbf{u}^\top \mathbf{v}) \frac{d\sigma_{q-1}(\mathbf{u})}{|\mathcal{S}^{q-1}|} \\ &= \int_{\mathcal{S}^{q-1}} \sum_{\ell=0}^{+\infty} \sum_{k=1}^{N(q, \ell)} \frac{b_\ell}{N(q, \ell)} Y_{\ell,k}(\mathbf{u}) Y_{\ell,k}(\mathbf{v}) d\sigma_{q-1}(\mathbf{u}) \\ &= \sum_{\ell=0}^{+\infty} \frac{b_\ell}{N(q, \ell)} \sum_{k=1}^{N(q, \ell)} \left[\int_{\mathcal{S}^{q-1}} Y_{\ell,k}(\mathbf{u}) d\sigma_{q-1}(\mathbf{u}) \right] Y_{\ell,k}(\mathbf{v}) \\ &= b_0 \frac{1}{Y_{1,0}} Y_{1,0} = b_0, \end{aligned} \quad (17)$$

where we inverted series and integral in the second equation using the dominated convergence theorem: the series $\sum_{\ell=0}^{+\infty} b_\ell P_\ell(q; \mathbf{u}^\top \mathbf{v})$ converges for every \mathbf{u}, \mathbf{v} , and for any L , $|\sum_{\ell=0}^L b_\ell P_\ell(q; \mathbf{u}^\top \mathbf{v})| \leq \sum_{\ell=0}^L |b_\ell P_\ell(q; \mathbf{u}^\top \mathbf{v})| \leq \sum_{\ell=0}^{+\infty} b_\ell$ because $|P_\ell(q; \cdot)| \leq 1$ for all ℓ by [47, Lemma 2, §8], and

$\sum_{\ell=0}^{+\infty} b_\ell = \sum_{\ell=0}^{+\infty} b_\ell P_\ell(q; 1)$ is integrable on \mathcal{S}^{q-1} (we indeed have $P_\ell(q; 1) = 1$ for all ℓ by [47, §9]). This yields the first claim of Lemma 2.

Consider now any probability distribution \mathbb{Q} defined on the hypersphere. The kernel mean embedding of \mathbb{Q} is simply rewritten as:

$$\begin{aligned} \forall \mathbf{v} \in \mathcal{S}^{q-1}, \int_{\mathcal{S}^{q-1}} \mathcal{K}(\mathbf{u}, \mathbf{v}) d\mathbb{Q}(\mathbf{u}) &= \int_{\mathcal{S}^{q-1}} \sum_{\ell=0}^{+\infty} b_\ell P_\ell(q; \mathbf{u}^\top \mathbf{v}) d\mathbb{Q}(\mathbf{u}) \\ &= b_0 \int_{\mathcal{S}^{q-1}} P_0(q; \mathbf{u}^\top \mathbf{v}) d\mathbb{Q}(\mathbf{u}) + \int_{\mathcal{S}^{q-1}} \sum_{\ell=1}^{+\infty} b_\ell P_\ell(q; \mathbf{u}^\top \mathbf{v}) d\mathbb{Q}(\mathbf{u}) \\ &= b_0 + \int_{\mathcal{S}^{q-1}} \tilde{\mathcal{K}}(\mathbf{u}, \mathbf{v}) d\mathbb{Q}(\mathbf{u}), \end{aligned} \quad (18)$$

because the Legendre polynomial of order 0 is the constant function equal to 1 (see the closed form expression of $P_0(q; \cdot)$ in Theorem 1 in the main text) and $\int_{\mathcal{S}^{q-1}} d\mathbb{Q} = 1$. This ends the proof of Lemma 2.

A.2 Legendre expansion of rotation-invariant kernels

We show that the kernel weights b_ℓ in the Legendre expansion (5) of the RBF kernel and the generalized distance kernel decay with a rate at least polynomial with respect to ℓ .

RBF kernel The RBF kernel is defined as:

$$\mathcal{K}_{\text{RBF}}(\mathbf{u}, \mathbf{v}) = e^{-\sigma \|\mathbf{u} - \mathbf{v}\|_2^2} = e^{-2\sigma(1 - \mathbf{u}^\top \mathbf{v})} \quad \text{for } \mathbf{u}, \mathbf{v} \in \mathcal{S}^q, \quad (19)$$

where $\sigma > 0$ is the scale of the RBF kernel. Denote $\varphi(t) := e^{-2\sigma(1-t)}$ for $t \in [-1, 1]$. Since the RBF kernel is positive definite and rotation-invariant, by Theorem 1, there exist weights $b_\ell \geq 0$, $\ell = 0, \dots, +\infty$, such that:

$$\varphi(t) = e^{-2\sigma(1-t)} = \sum_{\ell=0}^{+\infty} b_\ell P_\ell(q; t) \quad \text{for } t \in [-1, 1]. \quad (20)$$

The Legendre polynomials $P_\ell(q; \cdot)$ are orthogonal on the interval $[-1, 1]$ with respect to the weight function $(1 - t^2)^{\frac{q-3}{2}}$, see e.g. [47]:

$$\int_{-1}^1 P_n(q; t) P_m(q; t) (1 - t^2)^{\frac{q-3}{2}} dt = 0 \quad \text{for } m \neq n. \quad (21)$$

Moreover, by [47, Exercise 3, §2]:

$$\int_{-1}^1 (P_n(q; t))^2 (1 - t^2)^{\frac{q-3}{2}} dt = \frac{|\mathcal{S}^{q-1}|}{|\mathcal{S}^{q-2}|} \frac{1}{N(q, n)} \quad \text{for any } n. \quad (22)$$

We multiply (20) by $P_\ell(q; t)(1 - t^2)^{\frac{q-3}{2}}$ and integrate the equation on $[-1, 1]$:

$$\begin{aligned} \int_{-1}^1 \varphi(t) P_\ell(q; t) (1 - t^2)^{\frac{q-3}{2}} dt &= \int_{-1}^1 \sum_{n=0}^{+\infty} b_n P_n(q; t) P_\ell(q; t) (1 - t^2)^{\frac{q-3}{2}} dt \\ &= \sum_{n=0}^{+\infty} b_n \int_{-1}^1 P_n(q; t) P_\ell(q; t) (1 - t^2)^{\frac{q-3}{2}} dt \\ &= b_\ell \frac{|\mathcal{S}^{q-1}|}{|\mathcal{S}^{q-2}|} \frac{1}{N(q, \ell)}, \end{aligned} \quad (23)$$

where the inversion between series and integral is justified by the dominated convergence theorem: the series $\sum_{n=0}^{+\infty} b_n P_n(q; t) P_\ell(q; t) (1 - t^2)^{\frac{q-3}{2}}$ converges for every t , and for any N ,

$|\sum_{n=0}^N b_n P_n(q; t) P_\ell(q; t) (1-t^2)^{\frac{q-3}{2}}| \leq \sum_{n=0}^{+\infty} b_n (1-t^2)^{\frac{q-3}{2}} := g(t)$ since $|P_n(q; \cdot)| \leq 1$ for any n by [47, Lemma 2, §8], and g is integrable on $[-1, 1]$. Hence:

$$b_\ell = N(q, \ell) \frac{|\mathcal{S}^{q-2}|}{|\mathcal{S}^{q-1}|} \int_{-1}^1 \varphi(t) P_\ell(q; t) (1-t^2)^{\frac{q-3}{2}} dt. \quad (24)$$

By the Rodrigues rule [47, Exercise 1, §2], since φ has continuous derivatives of all orders on $[-1, 1]$, we have:

$$b_\ell = N(q, \ell) \frac{|\mathcal{S}^{q-2}|}{|\mathcal{S}^{q-1}|} \frac{\Gamma(\frac{q-1}{2})}{2^\ell \Gamma(\ell + \frac{q-1}{2})} \int_{-1}^1 \varphi^{(\ell)}(t) (1-t^2)^{\ell + \frac{q-3}{2}} dt, \quad \ell \in \mathbb{N}, \quad (25)$$

where $\varphi^{(\ell)}$ is the ℓ -th derivative of φ , which is $\varphi^{(\ell)}(t) = e^{-2\sigma} (2\sigma)^\ell e^{2\sigma t}$. We now show that the weights b_ℓ decay very fast with respect to ℓ . We bound the integral:

$$\int_{-1}^1 \varphi^{(\ell)}(t) (1-t^2)^{\ell + \frac{q-3}{2}} dt = \int_{-1}^1 e^{-2\sigma} (2\sigma)^\ell e^{2\sigma t} (1-t^2)^{\ell + \frac{q-3}{2}} dt \leq \int_{-1}^1 (2\sigma)^\ell dt = 2(2\sigma)^\ell. \quad (26)$$

Hence:

$$b_\ell \leq 2N(q, \ell) \frac{|\mathcal{S}^{q-2}|}{|\mathcal{S}^{q-1}|} \frac{\Gamma(\frac{q-1}{2})}{2^\ell \Gamma(\ell + \frac{q-1}{2})} (2\sigma)^\ell. \quad (27)$$

Denote $(a)_n := \Gamma(n+a)/\Gamma(a)$ the Pochhammer symbol defined for any integer n and any scalar a . By the Stirling approximation of the Gamma function [57, (25.15)], the asymptotic behavior of $(a)_n$ when n goes to infinity is:

$$(a)_n \sim \frac{\sqrt{2\pi}}{\Gamma(a)} e^{-n} n^{a+n-1/2} \quad \text{as } n \rightarrow \infty. \quad (28)$$

Moreover, for a fixed dimension q , the asymptotic behavior of $N(q, \ell)$ defined by (13) when ℓ goes to infinity is:

$$N(q, \ell) \sim \frac{2}{(q-2)!} \ell^{q-2} \quad \text{as } \ell \rightarrow \infty. \quad (29)$$

Therefore, the asymptotic behavior of b_ℓ as ℓ goes to infinity is:

$$b_\ell = \mathcal{O}\left(\sigma^\ell e^\ell \ell^{q/2-1-\ell}\right) \quad \text{as } \ell \rightarrow +\infty. \quad (30)$$

Generalized distance kernel In Example 2 we defined the generalized distance kernel for \mathbf{u}, \mathbf{v} in the hypersphere \mathcal{S}^{q-1} as in [6, Section 5]. For $(q-1)/2 < s < (q+1)/2$, it is defined as:

$$\mathcal{K}_{\text{gd}}^{(s)}(\mathbf{u}, \mathbf{v}) := 2V_{q-1-2s}(\mathcal{S}^{q-1}) - \|\mathbf{u} - \mathbf{v}\|_2^{2s-q+1} \quad \text{for } \mathbf{u}, \mathbf{v} \in \mathcal{S}^{q-1}, \quad (31)$$

where

$$V_{q-1-2s}(\mathcal{S}^{q-1}) := \int_{\mathcal{S}^{q-1}} \int_{\mathcal{S}^{q-1}} \|\mathbf{u} - \mathbf{v}\|_2^{2s-q+1} d\sigma_{q-1}(\mathbf{u}) d\sigma_{q-1}(\mathbf{v}) = 2^{2s-1} \frac{\Gamma(q/2)\Gamma(s)}{\sqrt{\pi}\Gamma((q-1)/2+s)}. \quad (32)$$

Following [6, Section 5], the Legendre expansion of the generalized distance kernel $\mathcal{K}_{\text{gd}}^{(s)}$ is:

$$\mathcal{K}_{\text{gd}}^{(s)}(\mathbf{u}, \mathbf{v}) = V_{q-1-2s}(\mathcal{S}^{q-1}) + \sum_{\ell=1}^{+\infty} \alpha_\ell^{(s)} N(q, \ell) P_\ell(q; \mathbf{u}^\top \mathbf{v}), \quad (33)$$

$$\alpha_\ell^{(s)} := -V_{q-1-2s}(\mathcal{S}^{q-1}) \frac{((q-1)/2-s)_\ell}{((q-1)/2+s)_\ell}, \quad \ell \geq 1. \quad (34)$$

The kernel weights indeed decay polynomially with respect to ℓ , because according to [6, Section 5], the asymptotic behavior of the $\alpha_\ell^{(s)}$ is:

$$\alpha_\ell^{(s)} \sim 2^{2s-1} \frac{\Gamma(q/2)\Gamma(s)}{\sqrt{\pi}\Gamma((q-1)/2-s)} \ell^{-2s} \quad \text{as } \ell \rightarrow +\infty. \quad (35)$$

A.3 Connection between SFRIK and VICReg [2]

Consider a rotation-invariant kernel $\tilde{\mathcal{K}}(\mathbf{u}, \mathbf{v}) := \sum_{\ell=1}^{+\infty} b_\ell P_\ell(q; \mathbf{u}^\top \mathbf{v})$ defined on \mathcal{S}^{q-1} such that $b_\ell \geq 0$ for $\ell \in \{1, \dots, +\infty\}$, with $b_1, b_2 > 0$. To show the connection between SFRIK and VICReg, we construct a high-dimensional feature map $\Phi : \mathcal{S}^{q-1} \rightarrow \ell_2(\mathbb{N})$, where $\ell_2(\mathbb{N})$ denotes the space of square-summable sequences with its canonical inner product $\langle \cdot, \cdot \rangle_{\ell_2}$, such that $\langle \Phi(\mathbf{u}), \Phi(\mathbf{v}) \rangle_{\ell_2} = \tilde{\mathcal{K}}(\mathbf{u}, \mathbf{v})$ for any $\mathbf{u}, \mathbf{v} \in \mathcal{S}^{q-1}$.

One way to construct such a feature map is to consider an orthonormal system of spherical harmonics. For any integer ℓ , denote $\{Y_{\ell,k}\}_{k=1}^{N(q,\ell)}$ an orthonormal basis of spherical harmonics of order ℓ in dimension q . By the addition theorem [47, Theorem 2, §1] recalled in (14), the kernel $\tilde{\mathcal{K}}(\mathbf{u}, \mathbf{v})$ admits the decomposition:

$$\begin{aligned} \tilde{\mathcal{K}}(\mathbf{u}, \mathbf{v}) &= \sum_{\ell=1}^{+\infty} b_\ell P_\ell(q; \mathbf{u}^\top \mathbf{v}) = \sum_{\ell=1}^{+\infty} \sum_{k=1}^{N(q,\ell)} \frac{b_\ell |\mathcal{S}^{q-1}|}{N(q,\ell)} Y_{\ell,k}(\mathbf{u}) Y_{\ell,k}(\mathbf{v}) \\ &= \langle \Phi(\mathbf{u}), \Phi(\mathbf{v}) \rangle_{\ell_2}, \end{aligned} \quad (36)$$

where

$$\Phi := \left(\sqrt{\frac{b_\ell |\mathcal{S}^{q-1}|}{N(q,\ell)}} \Phi_\ell \right)_{\ell=1}^{+\infty} \quad \text{with} \quad \Phi_\ell : \begin{cases} \mathcal{S}^{q-1} \rightarrow \mathbb{R}^{N(q,\ell)} \\ \mathbf{u} \mapsto (Y_{\ell,k}(\mathbf{u}))_{k=1}^{N(q,\ell)} \end{cases} \quad \text{for } \ell \in \{1, \dots, +\infty\}. \quad (37)$$

Then, the MMD in (6) between any probability distribution \mathbb{Q} defined on the hypersphere and the uniform distribution \mathbb{U} on the hypersphere can be written as the norm in $\ell_2(\mathbb{N})$ of the generalized moment of \mathbb{Q} with the mapping Φ :

$$\begin{aligned} \text{MMD}^2(\mathbb{Q}, \mathbb{U}) &= \mathbb{E}_{\mathbf{z}, \mathbf{z}' \sim \mathbb{Q}} [\tilde{\mathcal{K}}(\mathbf{z}, \mathbf{z}')] \\ &= \mathbb{E}_{\mathbf{z}, \mathbf{z}' \sim \mathbb{Q}} [\langle \Phi(\mathbf{z}), \Phi(\mathbf{z}') \rangle_{\ell_2}] \\ &= \langle \mathbb{E}_{\mathbf{z} \sim \mathbb{Q}} [\Phi(\mathbf{z})], \mathbb{E}_{\mathbf{z}' \sim \mathbb{Q}} [\Phi(\mathbf{z}')] \rangle_{\ell_2} = \|\mathbb{E}_{\mathbf{z} \sim \mathbb{Q}} [\Phi(\mathbf{z})]\|_{\ell_2}^2 \\ &= \sum_{\ell=1}^{+\infty} \frac{b_\ell |\mathcal{S}^{q-1}|}{N(q,\ell)} \|\mathbb{E}_{\mathbf{z} \sim \mathbb{Q}} [\Phi_\ell(\mathbf{z})]\|_2^2. \end{aligned} \quad (38)$$

We now explain how to construct explicitly an orthonormal basis of spherical harmonics $\{Y_{\ell,k} \mid \ell = 1, \dots, +\infty; k = 1, \dots, N(q,\ell)\}$, based on the following theorem.

Theorem 3 ([1, Theorem 5.25]). *For any order $\ell \in \mathbb{N}$ and any dimension $q \geq 3$, the family*

$$\{Y'_{\ell,k}\}_{k=1}^{N(q,\ell)} := \left\{ \mathbf{u} \mapsto \partial_1^{\alpha_1} \partial_2^{\alpha_2} \dots \partial_q^{\alpha_q} \|\mathbf{u}\|_2^{2-q} \mid \alpha_1 + \alpha_2 + \dots + \alpha_q = \ell \text{ and } \alpha_1 \leq 1 \right\} \quad (39)$$

is a (non-orthonormal) basis of the space of spherical harmonics of order ℓ in dimension q , where α_j ($j = 1, \dots, q$) are nonnegative integers, and $\partial_j^{\alpha_j}$ denotes the α_j -th partial derivative with respect to the j -th coordinate.

Typically, we construct the orthonormal basis $\{Y_{\ell,k}\}_{k=1}^{N(q,\ell)}$ by orthonormalizing the basis $\{Y'_{\ell,k}\}_{k=1}^{N(q,\ell)}$ of Theorem 3 using, e.g., the Gram-Schmidt procedure. For $\ell = 1, \dots, +\infty$, denote:

$$\Phi'_\ell : \mathcal{S}^{q-1} \rightarrow \mathbb{R}^{N(q,\ell)}, \quad \mathbf{u} \mapsto (Y'_{\ell,k}(\mathbf{u}))_{k=1}^{N(q,\ell)}. \quad (40)$$

Then, for each $\ell = 1, \dots, +\infty$, there exists a lower triangular matrix \mathbf{M}_ℓ such that:

$$\Phi_\ell(\mathbf{u}) = \mathbf{M}_\ell \Phi'_\ell(\mathbf{u}), \quad \text{for all } \mathbf{u} \in \mathcal{S}^{q-1}. \quad (41)$$

Remark that it is possible to compute explicitly the entries of the matrices \mathbf{M}_ℓ , $\ell = 1, \dots, +\infty$, because there exists a closed-form expression for the inner product $\langle Y'_{\ell,k}, Y'_{\ell,k'} \rangle_{(q)}$ for any ℓ, k, k' : indeed, the function $Y'_{\ell,k}$ for any ℓ, k is a polynomial defined on the hypersphere, and the integral of any monomial with respect to the measure σ_{q-1} on the hypersphere \mathcal{S}^{q-1} admits a closed-form expression given by [61, Section 3].

By injecting (41) in (38), we obtain:

$$\text{MMD}^2(\mathbb{Q}, \mathbb{U}) = \sum_{\ell=1}^{+\infty} \frac{b_\ell |\mathcal{S}^{q-1}|}{N(q, \ell)} \|\mathbf{M}_\ell \mathbb{E}_{\mathbf{z} \sim \mathbb{Q}} [\Phi'_\ell(\mathbf{z})]\|_2^2. \quad (42)$$

This yields the claim of Section 3.3 by remarking with Theorem 3 that the families

$$\begin{aligned} \{Y'_{1,k}\}_{k=1}^{N(q,1)} &= \{\mathbf{u} \mapsto u^j \mid 1 \leq j \leq q\}, \\ \{Y'_{2,k}\}_{k=1}^{N(q,2)} &= \left\{ \mathbf{u} \mapsto u^j u^{j'} \mid 1 \leq j < j' \leq q \right\} \cup \left\{ \mathbf{u} \mapsto (u^j)^2 - \frac{1}{q} \mid 2 \leq j \leq q \right\} \end{aligned} \quad (43)$$

are bases of the space of spherical harmonics of order 1 and 2 in dimension q .

A.4 Regularization loss of SimCLR, AUH and VICReg

In SimCLR [10], Alignment & Uniformity on the Hypersphere (AUH) [60], VICReg [2] and SFRIK, each image \mathbf{x}_i in a batch $\{\mathbf{x}_i\}_{i \in I}$ is augmented into two different views $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$, which are encoded into two embeddings $\mathbf{z}_i^{(1)}$ and $\mathbf{z}_i^{(2)}$. These embeddings are normalized in SimCLR, AUH and SFRIK, but not in VICReg. This yields two batches of embeddings $\mathbf{Z}_I^{(v)} := \{\mathbf{z}_i^{(v)}\}_{i \in I}$ ($v = 1, 2$). The four methods share the same form of loss function:

$$\ell(\mathbf{Z}_I^{(1)}, \mathbf{Z}_I^{(2)}) := \lambda \ell_a(\mathbf{Z}_I^{(1)}, \mathbf{Z}_I^{(2)}) + \mu \ell_r(\mathbf{Z}_I^{(1)}, \mathbf{Z}_I^{(2)}), \quad (44)$$

where $\lambda, \mu > 0$ are hyperparameters, ℓ_a is the alignment loss defined by (3) (which is the same for all the four methods), and ℓ_r is the regularization loss specific to each method.

SimCLR The regularization loss in SimCLR is:

$$\ell_r(\mathbf{Z}_I^{(1)}, \mathbf{Z}_I^{(2)}) = \frac{1}{2|I|} \sum_{v=1}^2 \sum_{i \in I} \log \left(\sum_{v'=1}^2 \sum_{i' \in I} \mathbb{1}_{[(v,i) \neq (v',i')]} \exp \left(\mathbf{z}_i^{(v) \top} \mathbf{z}_{i'}^{(v')} / \tau \right) \right), \quad (45)$$

where $\tau > 0$ is a hyperparameter of the method called the temperature, and $\mathbb{1}_{[(v,i) \neq (v',i')]}$ is equal to 1 if $(v, i) \neq (v', i')$, and 0 otherwise.

Alignment & Uniformity The regularization loss in AUH is:

$$\ell_r(\mathbf{Z}_I^{(1)}, \mathbf{Z}_I^{(2)}) = \frac{1}{2|I|^2} \sum_{v=1}^2 \log \left(\sum_{i \in I} \sum_{i' \in I} \exp(-t \|\mathbf{z}_i^{(v)} - \mathbf{z}_{i'}^{(v)}\|_2^2) \right), \quad (46)$$

where $t > 0$ is a hyperparameter called the scale of the RBF kernel.

VICReg As introduced in Section 3.3, the regularization loss in VICReg is:

$$\ell_r(\mathbf{Z}_I^{(1)}, \mathbf{Z}_I^{(2)}) = \frac{1}{2} \left[v(\mathbf{Z}_I^{(1)}) + v(\mathbf{Z}_I^{(2)}) \right] + \frac{\nu}{2\mu} \left[c(\mathbf{Z}_I^{(1)}) + c(\mathbf{Z}_I^{(2)}) \right], \quad (47)$$

where $\nu > 0$ is a hyperparameter and μ the hyperparameter from (44). Here, $v(\cdot)$ and $c(\cdot)$ are respectively the variance and covariance terms defined by (9).

B Experimental setting

We give more details about the setting of our experiments presented in Section 4.

B.1 IN20% dataset description

The datasets used in our experiments include a subset of 20% of ImageNet-1000 [15] as in [22]. This reduced dataset, denoted IN20%, contains all the 1000 classes of ImageNet, but we keep only 260 images per class. The 260 images extracted are the same as those extracted in the official implementation of OBoW [22] (<https://github.com/valeoai/obow>). In Section 4.2, we also use another 20% subset of the ImageNet trainset as a separate validation set for hyperparameter tuning. See Appendix B.4 below for more details. The construction of this validation set is described in our implementation available at <https://github.com/valeoai/sfrik>.

B.2 Image augmentations

We follow the same image augmentation pipeline as in [2]. Our experiments include the following image augmentations implemented by PyTorch (`torchvision.transforms`):

- `RandomResizedCrop(224, scale=(0.08, 1.0))`: crop a random area of the image between 8% and 100% of the total area, and resize it to an image of size 224×224 ;
- `RandomHorizontalFlip()`: flip horizontally an image;
- `ColorJitter(brightness=0.4, contrast=0.4, saturation=0.2, hue=0.1)`: randomly change brightness, contrast, saturation and hue of an image by a factor randomly sampled in respectively $[0.6, 1.4]$, $[0.6, 1.4]$, $[0.8, 1.2]$ and $[-0.1, 0.1]$.
- `RandomGrayscale()`: convert an image into grayscale.

We also use image augmentations implemented by PIL, as in VICReg’s code available at <https://github.com/facebookresearch/vicreg>:

- `GaussianBlur()`: blur an image using a Gaussian kernel with standard deviation uniformly sampled in $[0.1, 2.0]$;
- `Solarization()`: randomly invert all pixel values above a threshold, which is 130.

In our experiments, the first image view is obtained by composing the following random augmentations: random cropping resized to 224×224 , random horizontal flip applied with probability 0.5, random color jittering applied with probability 0.8, random grayscale conversion applied with probability 0.2, random Gaussian blur applied with probability 0.1, and random solarization applied with probability 0.2. The second view is obtained by composing the same random augmentations as the first view, except that Gaussian blur is applied every time (probability 1), and solarization is never applied (probability 0).

B.3 Evaluation protocol

We describe the downstream tasks on which self-supervision methods are evaluated in our experiments of Section 4.

Linear probing on ImageNet Following, e.g., [2], the weights of the backbone (ResNet-18 or ResNet-50) are frozen and a linear layer followed by a softmax on top of the backbone is trained in a supervised setting on a training set. Then the model is evaluated on a test set. The training set is either IN20% or 100% of ImageNet, but with labels. The test set is the validation set of ImageNet. The linear layer is trained using an SGD optimizer with momentum parameter equal to 0.9 during 100 epochs. We apply a weight decay of 10^{-6} . The learning rate follows a cosine decay scheduling. The batch size is fixed at 256. Training images are augmented by composing a random cropping of an area between 8% and 100% of the total area resized to 224×224 , and a random horizontal flip of probability 0.5. Images at test time are resized to 256×256 , and cropped at the center with a size 224×224 . The initial learning rate is tuned as a hyperparameter, and we report the top-1 accuracy on the validation set of ImageNet obtained after the last training epoch, along with the corresponding top-5 accuracy. The code that we use for linear probing on ImageNet is the one of [2] available at <https://github.com/facebookresearch/vicreg>.

Linear probing on Places205 We use the code of [22], available at <https://github.com/valeoai/obow>, for the evaluation by linear probing on Places205 [70]. The weights of the backbone (ResNet-50) pretrained on ImageNet are frozen and a linear prediction layer is trained for the classification task on Places205. We note that a batch normalization layer with *non*-learnable scale and bias parameters is added at the output the backbone in [22]. The linear prediction layer is trained with an SGD optimizer with a 0.9 momentum parameter during 28 epochs. The weight decay is 10^{-4} . The batch size is 256. The learning rate decreases by a factor of 10 at epoch 10 and epoch 20. We use the same image augmentation pipeline for training and testing as in linear probing on ImageNet. The initial learning rate is tuned as a hyperparameter, and we report the top-1 accuracy on the validation set of Places205 obtained after the last training epoch, along with the corresponding top-5 accuracy.

Semi-supervised learning After pretraining a backbone by self-supervision (ResNet-18 or ResNet-50), we fine-tune this backbone and the linear classifier on the classification task of ImageNet with only

1% or 10% of the labeled data. The labeled images that are considered in these subsets are the ones used in SimCLR’s official code available at <https://github.com/google-research/simclr>. We use an SGD optimizer with momentum parameter equal to 0.9 during 20 epochs, without weight decay. The batch size is fixed at 256. The learning rates of the backbone and the linear classifier follow a cosine decay scheduling with different initial learning rates. These initial learning rates are tuned as hyperparameters. We report the top-1 accuracy on the validation set of ImageNet obtained after the last training epoch, along with the corresponding top-5 accuracy. We use the same image augmentation pipeline for training and testing as in linear probing on ImageNet. The code that we use for semi-supervised learning on ImageNet is the one of [2] available at <https://github.com/facebookresearch/vicreg>.

Weighted kNN classification We follow the usual protocol of [9; 63]. We compute the normalized representations $f_\theta(\mathbf{x}_i)$ of the images \mathbf{x}_i , $i \in [N]$, in the training set. The label of an image \mathbf{x}_{test} in the test set is predicted by a weighted vote of its k nearest neighbors \mathcal{N}_k in the representation space: the class c gets a score of $w_c := \sum_{i \in \mathcal{N}_k} \exp(f_\theta(\mathbf{x}_i)^\top f_\theta(\mathbf{x}_{\text{test}})/0.07) \mathbb{1}_{[c_i=c]}$ where $f_\theta(\mathbf{x}_{\text{test}})$ is normalized, c_i is the class of \mathbf{x}_i , and $\mathbb{1}_{[c_i=c]}$ is equal to 1 if $c_i = c$, and 0 otherwise. We report the kNN classification top-1 accuracy for $k = 20$. The image augmentation pipeline for both training and testing is the following one: images are resized to 256×256 , and cropped at the center with a size 224×224 . The code that we use for kNN classification is the one of [9] available at <https://github.com/facebookresearch/dino>.

B.4 Hyperparameters for experiments on IN20%

We describe in detail our protocol for hyperparameter tuning for the experiments in Section 4.2. For a rigorous tuning, it is important that the dataset used for the final evaluation remains unseen during training and hyperparameter tuning. For each pretraining method, we pretrain on the IN20% training set (blue subset in Figure 2) a backbone for each choice of hyperparameters. These backbones are then evaluated by weighted kNN classification on a separate validation set, which is another 20% subset of the ImageNet train set (purple subset in Figure 2), and we select the hyperparameters that yield the highest top-1 accuracy on this kNN evaluation. Then, we tune the learning rate for the linear probing evaluation, again on our separate validation set (purple subset in Figure 2). Finally, we use the model trained with the best learning rate discovered for linear probing evaluation on the usual ImageNet validation set (red subset in Figure 2), which has never been seen during hyperparameter tuning.

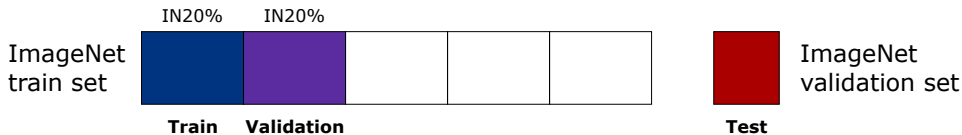


Figure 2: ImageNet dataset split for hyperparameter tuning in experiments of Section 4.2 on IN20%.

For the interest of reproducible research, we report the values of the optimal hyperparameters found after our hyperparameter tuning on a separate validation set, for each pretraining experiment on IN20% with a ResNet-18 presented in Section 4.2. These hyperparameters yield the evaluation results reported in Section 4.2 for linear probing on the usual ImageNet validation set. We recall that the hyperparameters specific to each self-supervision method have been introduced in Appendix A.4.

SimCLR For each embedding dimension, we fix the batch size at 2048, and we tune the temperature τ and the base learning rate `base_lr` for pretraining with SimCLR. Then, we tune the initial learning rate `lr_head` for linear probing on IN20%. The optimal hyperparameters are shown in Table 3.

AUH For each embedding dimension, we fix the batch size at 2048, and we tune the alignment weight λ , the scale of the RBF kernel t , and the base learning rate `base_lr` for pretraining with AUH. Without loss of generality, the uniformity weight is fixed at $\mu = 1$. Then, we tune the initial learning rate `lr_head` for linear probing on IN20%. The optimal hyperparameters are shown in Table 4.

VICReg For each embedding dimension, we fix the batch size at 2048, and we tune the alignment weight λ , the variance weight μ , and the base learning rate `base_lr` for pretraining with VICReg.

Without loss of generality, the covariance weight is fixed at $\nu = 1$. Then, we tune the initial learning rate `lr_head` for linear probing on IN20%. The optimal hyperparameters are shown in Table 5.

SFRIK, batch size 2048 For each embedding dimension, we fix the batch size at 2048, and we tune the alignment weight λ in (2), the kernel weights b_ℓ ($\ell \in \{1, 2, 3\}$) in (8), and the base learning rate `base_lr` for pretraining with SFRIK. Without loss of generality, the uniformity weight in (2) is fixed at $\mu = 0.5$. Then, we tune the initial learning rate `lr_head` for linear probing on IN20%. The optimal hyperparameters are shown in Table 6.

SFRIK, batch size 4096 We fix the dimension at $q = 8192$, and the batch size at 4096. We tune the alignment weight λ in (2), the kernel weights b_ℓ ($\ell \in \{1, 2, 3\}$) in (8), and the base learning rate `base_lr` for pretraining with SFRIK. Without loss of generality, the uniformity weight in (2) is fixed at $\mu = 0.5$. Then, we tune the initial learning rate `lr_head` for linear probing on IN20%. The optimal hyperparameters are: $\lambda = 4000$, $(b_1, b_2, b_3) = (1, 20, 0)$, `base_lr` = 0.8, `lr_head` = 1.0. This yields a top-1 accuracy of 46.3 for linear probing on IN20%.

Table 3: Hyperparameter choice for SimCLR pretrained on IN20% with a ResNet-18 during 100 epochs, evaluated by linear probing on IN20%.

Dimension	Temperature	base_lr	lr_head
$q = 1024$	0.15	1.0	0.2
$q = 2048$	0.15	1.0	0.2
$q = 4096$	0.15	1.0	0.2
$q = 8192$	0.15	0.8	0.2

Table 4: Hyperparameter choice for AUH pretrained on IN20% with a ResNet-18 during 100 epochs, evaluated by linear probing on IN20%.

Dimension	Alignment weight λ	Scale t	base_lr	lr_head
$q = 1024$	400	2.5	1.0	10.0
$q = 2048$	1000	2.5	1.0	4.0
$q = 4096$	2000	2.5	1.0	1.0
$q = 8192$	3000	2.5	1.0	2.0

Table 5: Hyperparameter choice for VICReg pretrained on IN20% with a ResNet-18 during 100 epochs, evaluated by linear probing on IN20%.

Dimension	Alignment weight λ	Variance weight μ	base_lr	lr_head
$q = 1024$	4	10	0.4	0.2
$q = 2048$	4	4	0.7	1.0
$q = 4096$	10	10	0.6	0.2
$q = 8192$	10	10	0.7	0.2

Table 6: Hyperparameter choice for SFRIK pretrained on IN20% with a ResNet-18 during 100 epochs, evaluated by linear probing on IN20%.

Order	Dimension	Alignment weight λ	Kernel weights (b_1, b_2, b_3)	base_lr	lr_head
$L = 1$	$q = 8192$	10000	(1, 0, 0)	0.4	10.0
$L = 2$	$q = 1024$	400	(1, 40, 0)	1.0	2.0
	$q = 2048$	400	(1, 40, 0)	1.0	4.0
	$q = 4096$	1000	(1, 40, 0)	1.0	4.0
	$q = 8192$	2000	(1, 20, 0)	1.0	10.0
$L = 3$	$q = 8192$	4000	(1, 40, 40)	1.2	1.0

Table 7: **Linear probing on IN20% (top-1 accuracy)** at different embedding dimensions q . All methods were pretrained on IN20% with a ResNet-18 for 100 epochs. Hyperparameters specific to each method and the learning rate are tuned on the *same* dataset as the one used for evaluation in linear probing, which is less rigorous than tuning the hyperparameters on a separate validation set as described in Appendix B.4. † indicates models that we retrained ourselves.

	SimCLR† [10]	AUH† [60]	VICReg† [2]	SFRIK		
				$L = 1$	$L = 2$	$L = 3$
$q = 1024$	45.2	45.2	40.8	-	44.2	-
$q = 2048$	45.8	45.6	44.1	-	45.5	-
$q = 4096$	46.3	46.8	44.9	-	47.0	-
$q = 8192$	46.2	46.8	46.0	27.5	47.0	47.6

B.5 Hyperparameter tuning without a separate validation set

A common practice in the self-supervised learning literature, e.g., [2; 10], is to select the hyperparameters by measuring the performance on the validation set of ImageNet (red dataset in Figure 2) instead of a separate validation dataset (purple dataset in Figure 2). In this paragraph, we verify whether this less rigorous practice changes the conclusion of the experiments in Section 4.2. In Table 7, we report the evaluation of the different backbones pretrained on IN20% after tuning each method *directly on the validation set* of ImageNet, which is the same dataset used for evaluation in linear probing. By comparison with Table 1, which follows the more rigorous hyperparameter tuning protocol described in Appendix B.4, we observe that although the absolute figures of merit slightly vary if we use the less rigorous protocol instead of the more rigorous one, the conclusion of the experiments in Section 4.2 does not change. This gives an empirical justification to this common practice.

B.6 Hyperparameters for experiments on 100% of ImageNet

We explain our choice of hyperparameters of Section 4.3, where we pretrained a ResNet-50 on full ImageNet with SFRIK during 200 or 400 epochs. Since hyperparameter tuning is costly on full ImageNet, we pretrain several ResNet-50 for different values of kernel weights b_ℓ , alignment weight λ and base learning rate `base_lr`, and pause the pretraining after 50 epochs. We evaluate the obtained backbones on kNN classification (top-1 accuracy, $k = 20$), and select the best performing backbones. Then we continue pretraining these selected backbones until reaching epoch 200 or 400. Finally we choose the set of hyperparameters that yields the highest top-1 accuracy for linear probing on ImageNet after 200 or 400 epochs of pretraining. Because of the conclusions of Appendix B.5, our hyperparameter tuning follows the common practice in the self-supervised learning literature [2; 10] where hyperparameters are selected by measuring the performance on the validation set of ImageNet.

In Tables 8 and 9, we give the optimal hyperparameters found for linear probing on ImageNet, linear probing on Places205, and semi-supervised learning on ImageNet when evaluating pretrained ResNet-50 backbones with SFRIK and VICReg. The hyperparameters that are tuned for evaluation are: the initial learning rate `lr_head` of the linear layer in linear probing; and the initial learning rate `lr_backbone` and `lr_head` for respectively the backbone and the linear layer in semi-supervised learning. These hyperparameters yield the evaluation results reported in Section 4.3.

B.7 Computational resources

Pretrainings on IN20% with a ResNet-18 backbone and a batch size of 2048 (respectively 4096) are performed with 4 (respectively 8) NVIDIA Tesla V100 GPUs with 32GB of memory. Pretrainings on 100% of ImageNet with a ResNet-50 backbone and a batch size of 4096 are performed with 8 NVIDIA Tesla V100 GPUs with 32GB of memory. The total amount of compute used for this work is around 35,000 GPU hours.

Table 8: Hyperparameter tuning for linear probing on ImageNet and Places205 for SFRIK and VICReg pretrained on ImageNet with a ResNet-50.

Method	Epochs	lr_head	
		ImageNet	Places205
VICReg [†] [2] ($q = 8192$)	200	0.02	0.01
SFRIK ($L = 2, q = 8192$)	200	1.0	0.01
SFRIK ($L = 2, q = 16384$)	200	0.3	0.01
SFRIK ($L = 2, q = 32768$)	200	1.0	0.01
SFRIK ($L = 3, q = 32768$)	200	0.4	0.01
SFRIK ($L = 2, q = 8192$)	400	2.0	0.01

Table 9: Hyperparameter tuning for semi-supervised learning on ImageNet for SFRIK and VICReg pretrained on ImageNet with a ResNet-50.

Method	Epochs	Semi-supervised, 1% IN		Semi-supervised, 10% IN	
		lr_backbone	lr_head	lr_backbone	lr_head
VICReg [†] [2] ($q = 8192$)	200	0.02	0.2	0.2	0.04
SFRIK ($L = 2, q = 8192$)	200	0.004	1.6	0.02	0.4
SFRIK ($L = 2, q = 16384$)	200	0.004	1.4	0.04	0.2
SFRIK ($L = 2, q = 32768$)	200	0.004	1.0	0.04	0.1
SFRIK ($L = 3, q = 32768$)	200	0.004	1.0	0.02	0.1
SFRIK ($L = 2, q = 8192$)	400	0.004	1.4	0.02	0.2

B.8 Public resources

We acknowledge the use of the following public resources, during the course of this work:

- VICReg official code [2] MIT License
- DINO official code [9] Apache License 2.0
- OBoW official code [22] Apache License 2.0
- SwAV official code [8] CC BY-NC 4.0
- SimCLR official code [10] Apache License 2.0
- ImageNet dataset [15]
- Places 205 dataset [70] Attribution CC BY