2024 IEEE International Conference on Image Processing (ICIP) | 979-8-3503-4939-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICIP51287.2024.10648197

LEARNING ORTHONORMAL FEATURES IN SELF-SUPERVISED LEARNING USING FUNCTIONAL MAXIMAL CORRELATION

Bo Hu, Yuheng Bu, José C. Príncipe

Department of Electrical and Computer Engineering
University of Florida
{hubo, buyuheng}@ufl.edu principe@cnel.ufl.edu

ABSTRACT

This paper applies statistical dependence measures to interpret self-supervised learning (SSL). Conventional applications of measures like mutual information commonly use separate procedures for feature extraction and dependence estimation, where the relationship between optimal features and the strength of dependence is unclear. This causes limitations in tasks requiring multivariate feature representations, particularly in SSL. The recently introduced multivariate measure, functional maximal correlation, is a unified framework based on orthonormal decomposition of density ratios, wherein the spectrum and the bases become the measure and the features, respectively. This paper proposes that features in SSL can also be interpreted as basis functions of the density ratio. We introduce the Hierarchical Functional Maximal Correlation Algorithm (HFMCA), a theoretically justified approach that ensures faster convergence, enhanced stability, and prevents feature collapse by learning orthonormal bases as multivariate features.

1. INTRODUCTION

Measures of statistical dependence have been instrumental in learning features that maximize information transfer ([1, 2, 3, 4]), which has sparked a multitude of learning principles and algorithms in machine learning ([5, 6, 7, 8, 9]). Improving the diversity and interpretability of multivariate features is important in numerous machine learning tasks, such as one-shot learning, transfer learning, and especially self-supervised learning (SSL) [10, 11, 12]. In this paper, we explore the advances of using multivariate statistical dependence measures for these tasks.

Conventional dependence measures such as mutual information (MI) may encounter limitations in tasks that require multivariate properties. Methods based on these measures ([6, 13, 14, 15, 16, 17, 18]) typically involve a three-step iterative process of projection, estimation, and maximization: first, a network maps data to a feature space; second, a mutual information estimator is optimized to ensure tight variational bounds; and third, the feature extractor is optimized to maximize the estimated MI. The limitation lies in the gap between feature extraction and dependence estimation being two separate procedures, conducted by two separate models, which leaves the link between the optimal multivariate features and the strength of dependence obscure.

The Multivariate Statistical Dependence (MSD) is a recently introduced multivariate dependence measure based on the concept of orthonormal decomposition of density ratios [19, 20, 21]. The spectrum in this decomposition is the defined dependence measure, and the basis functions are the multivariate features. Together, they create the projection space associated with the density ratio. The measure is accompanied with the Functional Maximal Correlation Algorithm (FMCA), which uses neural networks and log-determinant costs to learn this decomposition directly from empirical data. This framework unifies dependence measurement and feature learning through density ratio decomposition, allowing the learning of multivariate features that are theoretically orthonormal.

This concept may provide a new theoretical approach for explaining SSL: the optimal multivariate features learned through SSL can be interpreted as the basis functions of the density ratio induced by a corresponding probabilistic system. Upon formulating the probabilistic system, we propose the Hierarchical Functional Maximal Correlation Algorithm (HFMCA), an algorithm for multiview self-supervised learning that explores the hierarchical relationship between data and their augmentations. HFMCA offers faster convergence, stability that prevents feature collapse, and a theoretical foundation for interpretability.

2. PRELIMINARY: DENSITY RATIO DECOMPOSITION

Spectrum, basis functions, and density ratios. A unique characteristic of the multivariate dependence measure is its direct application of spectral decomposition to the density ratio [21]. Given any two random processes $\mathbf{X}$ and $\mathbf{Y}$, with a joint distribution $p(X, Y)$ and the marginal product $p(X)p(Y)$, the statistical dependence measure is defined via an orthonormal decomposition:

$$\rho := \frac{p(X,Y)}{p(X)p(Y)} = \sum_{k=1}^{\infty} \sqrt{\sigma_k}\phi_k(X)\psi_k(Y),$$
$$\mathbb{E}_{\mathbf{X}}[\phi_k(\mathbf{X})\phi_{k'}(\mathbf{X})] = \mathbb{E}_{\mathbf{Y}}[\psi_k(\mathbf{Y})\psi_{k'}(\mathbf{Y})] = \begin{cases} 1, \ k = k' \\ 0, \ k \neq k' \end{cases}. \tag{1}$$

Each decomposition component has a unique role: the spectrum measures multivariate dependence, the bases serve as feature projectors, and the kernel-associated density ratio provides a metric distance. The task of modeling dependence thus becomes modeling this projection space, leading to the proposal of the Functional Maximal Correlation Algorithm (FMCA).

There are several important properties of the spectrum. First, all eigenvalues are bounded by 1. The largest eigenvalue is a constant 1 (i.e., $\sigma_1 = 1$), and the two variables are strictly independent if and only if all other eigenvalues are zero (i.e., $\sigma_2 = \sigma_3 = \cdots = 0$).

Neural networks implementation. When dealing with empirical data and lacking the knowledge of *pdf*, spectral decomposition can be achieved through optimization. The empirical studies suggest a log-determinant-based cost function, optimized via paired neural networks, offers superior stability. Using two neural networks, $\mathbf{f}_\theta : \mathcal{X} \to \mathbb{R}^K$ and $\mathbf{g}_\omega : \mathcal{Y} \to \mathbb{R}^K$ that map realizations of $\mathbf{X}$ and $\mathbf{Y}$ respectively, each to a K-dimensional output space, we compute the autocorrelation (ACFs) and crosscorrelation functions (CCFs) defined as follows:

$$\mathbf{R}_F = \mathbb{E}_{\mathbf{X}}[\mathbf{f}_\theta(\mathbf{X})\mathbf{f}_\theta^\intercal(\mathbf{X})], \ \mathbf{R}_G = \mathbb{E}_{\mathbf{Y}}[\mathbf{g}_\omega(\mathbf{Y})\mathbf{g}_\omega^\intercal(\mathbf{Y})],$$
$$\mathbf{P}_{FG} = \mathbb{E}_{\mathbf{X},\mathbf{Y}}[\mathbf{f}_\theta(\mathbf{X})\mathbf{g}_\omega^\intercal(\mathbf{Y})], \ \mathbf{R}_{FG} = \begin{bmatrix} \mathbf{R}_F & \mathbf{P}_{FG} \\ \mathbf{P}_{FG}^\intercal & \mathbf{R}_G \end{bmatrix}. \tag{2}$$

FMCA defines an optimization problem that minimizes the log-determinant of the marginal ACFs $\mathbf{R}_F$ and $\mathbf{R}_G$ for output orthonormality, while maximizing the log-determinant of the joint ACF $\mathbf{R}_{FG}$ to parallelize two projection spaces. The problem is formulated as follows:

$$\min_{\theta,\omega} \ r(\mathbf{f}_\theta, \mathbf{g}_\omega) := \log\det \mathbf{R}_{FG} - \log\det \mathbf{R}_F - \log\det \mathbf{R}_G. \tag{3}$$

Upon reaching optimality, normalization schemes are employed to network outputs. Theoretically, the objective function effectively captures the leading eigenvalues of the spectrum, while the neural networks, viewed as multivariate function approximators, approximate leading basis functions.

Linking dependence measurement and feature learning. Applying FMCA for feature learning is direct: formulate the joint density and marginal products, initiate nonlinear mappers, and minimize costs. This yields a multivariate dependence measure and a theoretically-grounded feature projector that together decompose the density ratio. These features naturally display orthonormality, ensuring diversity, which is vital for many learning tasks. This paper spotlights the potential of this property in self-supervised learning that exhibit *hierarchical structures*. We demonstrate that learning dependence structures from hierarchies is crucial for self-supervised learning and can be accomplished with the FMCA.

Costs, spectrum, and dependencies. The spectrum's eigenvalues range from 0 to 1. The optimal cost approximates their aggregation $r^* = \sum_{k=1}^{K} \log(1 - \sigma_k)$. Dependence can be evaluated using both the spectrum and cost, where a lower cost and higher eigenvalues indicate stronger dependence.

Normalizations after training. Note that it requires two additional normalization steps to map the outputs of two neural networks, $\mathbf{f}_\theta$ and $\mathbf{g}_\theta$, to the basis function, $\{\phi_k\}$ and $\{\psi_k\}$ of the density ratio. The first step is to enforce orthonormality, using the ACFs $\mathbf{R}_F$ and $\mathbf{R}_G$ of the marginal. After training, we simply normalize the outputs using

$$\overline{\mathbf{f}_\theta} = \mathbf{R}_F^{-\frac{1}{2}}\mathbf{f}_\theta, \ \ \overline{\mathbf{g}_\omega} = \mathbf{R}_G^{-\frac{1}{2}}\mathbf{g}_\omega. \tag{4}$$

which guarantees orthonormality of the functions $\overline{\mathbf{f}_\theta}$ and $\overline{\mathbf{g}_\omega}$.

The second step is to enforce equilibrium, using the eigen-expansion:

$$\overline{\mathbf{P}}_{FG} = \mathbb{E}[\overline{\mathbf{f}_\theta}(\mathbf{X})\overline{\mathbf{g}_\omega}^\intercal(\mathbf{X})], \ \overline{\mathbf{P}}_{FG}\overline{\mathbf{P}}_{FG}^\intercal = \mathbf{Q}_F\mathbf{\Sigma}\mathbf{Q}_F{}^\intercal,$$
$$\overline{\mathbf{P}}_{FG}^\intercal\overline{\mathbf{P}}_{FG} = \mathbf{Q}_G\mathbf{\Sigma}\mathbf{Q}_G{}^\intercal, \ \mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_K \end{bmatrix}. \tag{5}$$

The obtained eigenvalues $\{\sigma_k\}$ will be the eigenvalues, and the normalized functions

$$\widehat{\mathbf{f}_\theta} = \mathbf{Q}_F^T\overline{\mathbf{f}_\theta}, \ \ \widehat{\mathbf{g}_\omega} = \mathbf{Q}_G^T\overline{\mathbf{g}_\omega}. \tag{6}$$

will match the eigenfunctions from the set $\{\phi_k\}$ and $\{\psi_k\}$. By minimizing the cost function, the goal of our algorithm is to identify the dominant eigenvalues in the eigenspectrum, such that the neural networks to converge to the leading eigenfunctions. As a result, an accurate approximation of the density ratio can be achieved when K and L are sufficiently large.

3. DENSITY RATIO DECOMPOSITION FOR SSL

We explore the potential of framing SSL as a statistical dependence measurement problem. Regardless of the specified protocols, augmentations of an image describe the common source object. This relationship implies statistical dependence.

Consider an unaugmented image $X \sim \mathbb{P}(\mathbf{X})$, with $\mathbb{P}(\mathbf{X})$ being the given data distribution prior to any augmentation, which is the source data distribution that we are presented with. The augmentation protocols can be modeled as a transformation function $\mathcal{T}(X; v)$, which takes an image $X \in \mathcal{X}$ and a positive integer index $v \in \mathcal{V}$ representing a specific protocol, and produce an augmented version of the image. The set $\mathcal{V}$ is a subset of positive integers $\mathcal{V} \subset \mathbb{Z}^+$, and its cardinality $|\mathcal{V}|$ is the total number of specified protocols. The augmented images will have a distribution $\mathbb{P}(\mathbf{Y})$, assuming they are modeled as a random process $\mathbf{Y} \in \mathcal{Y}$.

For simplicity, consider *pdf*s exist for all defined distributions. We propose that the augmentation procedure can be modeled as a conditional *pdf*

$$p(Y|\mathbf{X} = X) = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathbb{1}\{Y = \mathcal{T}(X; v)\}, \tag{7}$$

by applying a counting measure to all possible augmented versions of a given image X. Given $X \sim \mathbb{P}(\mathbf{X})$, augmentations of this image can be sampled from this conditional distribution as $Y \sim \mathbb{P}(\mathbf{Y}|\mathbf{X} = X)$. The marginal $p(\mathbf{Y})$ is obtained by marginalizing over all images in the dataset

$$\begin{aligned} p(\mathbf{X} = X, \mathbf{Y} = Y) &= p(X) \cdot \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathbb{1}\{Y = \mathcal{T}(X; v)\}, \\ p(\mathbf{Y} = Y) &= \int_{\mathcal{X}} p(\mathbf{X} = X, \mathbf{Y} = Y) dX. \end{aligned} \tag{8}$$

Simplify notations to be $p(X, Y)$, $p(X)$ and $p(Y)$. Naturally, we choose to decompose $\rho(X, Y) := \frac{p(X,Y)}{p(X)p(Y)} = \sum_{k=1}^{\infty} \sqrt{\sigma_k} \phi_k(X) \psi_k(Y)$, i.e., the density ratio induced by the joint distribution between the original image and its augmentations.

The remaining question becomes constructing the ACFs and CCFs in the log-determinant cost (3) and apply optimization. Assume that two neural networks $\mathbf{f}_\theta : \mathcal{X} \to \mathbb{R}^K$ and $\mathbf{g}_\omega : \mathcal{Y} \to \mathbb{R}^K$ are given. The two ACFs $\mathbf{R}_F = \mathbb{E}_{\mathbf{X}}[\mathbf{f}_\theta(\mathbf{X})\mathbf{f}_\theta^\intercal(\mathbf{X})]$ and $\mathbf{R}_G = \mathbb{E}_{\mathbf{Y}}[\mathbf{g}_\omega(\mathbf{Y})\mathbf{g}_\omega^\intercal(\mathbf{Y})]$ can be written and estimated empirically by their definitions.

Multiview system. Observe that sampling from this joint is to first sample an image in the dataset, then sample an augmentation from the conditional, indicating that the CCF can also be estimated in a similar way, in terms of the conditional:

$$\begin{aligned} \mathbf{P}_{FG} &= \mathbb{E}_{\mathbf{X},\mathbf{Y}}[\mathbf{f}_\theta(\mathbf{X})\mathbf{g}_\omega^\intercal(\mathbf{Y})] \\ &= \iint p(X) p(Y|\mathbf{X} = X) \mathbf{f}_\theta(X) \mathbf{g}_\omega^\intercal(Y) dX dY \\ &= \int p(X) \mathbf{f}_\theta(X) \mathbb{E}_{\mathbf{Y}}[\mathbf{g}_\omega^\intercal(\mathbf{Y})|\mathbf{X} = X] dX. \end{aligned} \tag{9}$$

Therefore, a proper approach is to first estimate the conditional expectation $\mathbb{E}_{\mathbf{Y}}[\mathbf{g}_\omega^\intercal(\mathbf{Y})|\mathbf{X} = X]$, by averaging over multiple augmentations of each individual image, then estimate the CCF by averaging over all images. This estimation of the conditional mean, which uses multiple views of an image similar to [22, 23], differs from the conventional contrastive learning approach that uses only two views.

To frame this formally, we introduce a series of L i.i.d. categorical r.v., $\mathbf{V} = \{\mathbf{v}_1, \cdots, \mathbf{v}_L\}$, denoting the execution of L augmentations. For each image, a set of indices $\{v_1, \cdots, v_L\} \subset \mathcal{V}$ is sampled, generating L views $\mathcal{T}(X; V) = \{\mathcal{T}(X; v), v \in v_1, \cdots, v_L\}$. Then the conditional mean can be estimated by averaging over these L views: $\mathbb{E}_{\mathbf{Y}}[\mathbf{g}_\omega(\mathbf{Y})|\mathbf{X} = X] \approx \frac{1}{L} \sum_{l=1}^{L} \mathbf{g}_\omega(\mathcal{T}(X; v_l))$. Then the CCF is estimated by averaging over all images.

Hierarchical structure. The second observation is that since augmentations, such as patches, are often part of the original image, this implies a hierarchical relationship that allows the two parameterized networks $\mathbf{f}_\theta$ and $\mathbf{g}_\omega$ to have shared structures. Instead of using two separate networks, the model topology can be simplified to be a cascade of the backbone $\mathbf{f}_\theta^{(1)}$ and the projection head $\mathbf{f}_\theta^{(2)}$ as approximators for basis functions. The backbone $\mathbf{f}_\theta^{(1)}$ is first applied to the L augmentations of an image, extracting L low-level features $\mathbf{Z}_l^{(1)}$, each with K dimensions. These L features are then concatenated in the feature channel, acting as inputs to the projection head, and producing the high-level feature $\mathbf{Z}^{(2)}$. The mapping to $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$ will be considered function approximators $\mathbf{f}_\theta$ and $\mathbf{g}_\omega$.

Combining the two modifications, we introduce HFMCA for SSL as follows.

Proposition 1. *Denote the feature maps produced by the backbone and the projection head as $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$, respectively. Also assign L auxiliary indices to $\mathbf{Z}^{(1)}$ as $\{\mathbf{Z}_l^{(1)}, l = 1, \cdots, L\}$, corresponding to L augmentations of an image. HFMCA solves the optimization problem:*

$$\begin{gathered} \mathbf{R}_1 = \mathbb{E}[\mathbf{Z}^{(1)}\mathbf{Z}^{(1)\intercal}], \ \mathbf{R}_2 = \mathbb{E}[\mathbf{Z}^{(2)}\mathbf{Z}^{(2)\intercal}], \\ \mathbf{P}_{1,2} = \frac{1}{L}\mathbb{E}[\sum_{l=1}^{L} \mathbf{Z}_l^{(1)}\mathbf{Z}^{(2)\intercal}], \ \mathbf{R}_{1,2} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{P}_{1,2} \\ \mathbf{P}_{1,2}^\intercal & \mathbf{R}_2 \end{bmatrix}, \\ \min_\theta r_H := \log\det \mathbf{R}_{1,2} - \log\det \mathbf{R}_1 - \log\det \mathbf{R}_2. \end{gathered} \tag{10}$$

By the theory of FMCA, the objective function reaches the leading eigenvalues of the density ratio, with neural networks reaching the leading orthonormal basis functions upon normalizations.

This proposition described the cost and optimization problem for our algorithm, which involves constructing the cost function $r_H(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)})$ and employing the HFMCA for minimization. This procedure measures statistical dependence between two hierarchical levels and, most importantly, extracts multivariate features that are theoretically orthonormal.

Full algorithm of HFMCA. Now that we have introduced the cost function, we now describe the full algorithm. For SSL, we minimize the cost $r_H(\mathcal{T}(\mathbf{X}; \mathbf{v}), \mathcal{T}(\mathbf{X}; \mathbf{V}))$ to learn diverse features. Unlike conventional methods optimizing similarity measures between augmentation pairs, HFMCA uses an additional network after the backbone. The backbone CNN is first applied to the L augmentations of a source image, extracting L feature maps $\mathbf{Z}_l^{(1)}$ of K dimensions. These L feature

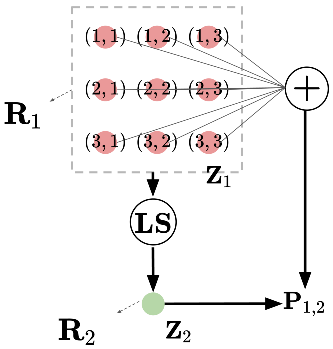

Figure 1: Illustration of HFMCA's cost: Each element of $\mathbf{Z}_1$ is a feature of an augmentation, produced by the backbone. These features are concatenated as inputs to the projection head (**LS**), producing the feature $\mathbf{Z}_2$. Then, $\mathbf{R}_1$, $\mathbf{R}_2$, and $\mathbf{P}_{1,2}$ are constructed, and HFMCA is applied.

maps are then concatenated in the feature channel, serving as inputs to the additional network and yielding the higher-level features $\mathbf{Z}^{(2)}$. Minimizing the log-determinants of marginal ACFs, $\mathbf{R}_1$ and $\mathbf{R}_2$, ensures orthonormality. Meanwhile, maximizing the joint ACF $\mathbf{R}_{1,2}$ align the two sets of bases as parallel as possible. Solving this min-max problem effectively extracts shared information between two levels. HFMCA offers orthonormal features as the density ratio's basis functions, ensuring diversity. The algorithm's diagram is in Figure 1, and its pseudocode can be found in Algorithm 1.

Post-training normalization. As discussed in the preliminary, extra normalization steps need to be applied to the network outputs to obtain the eigenfunctions. For simplicity, denote the mapping from an augmentation to the outputs of the backbone as $\mathbf{f}$, and the mapping from the original image to the outputs of the projection head as $\mathbf{g}$. After training, the first step is to enforce the orthonormality with $\overline{\mathbf{f}} = \mathbf{R}_1^{-\frac{1}{2}}\mathbf{f}, \overline{\mathbf{g}} = \mathbf{R}_2^{-\frac{1}{2}}\mathbf{g}$. The second step is to apply the singular-value decomposition such that the functions are invariant to the conditional mean operator, which follows

$$\begin{gathered}\mathbb{E}[\overline{\mathbf{f}}\ \overline{\mathbf{g}}^\intercal] = \widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}^{\frac{1}{2}}\widehat{\mathbf{V}}^\intercal,\ \widehat{\mathbf{\Sigma}} = \mathrm{diag}([\widehat{\sigma_1}, \cdots, \widehat{\sigma_K}]), \\ \widehat{\mathbf{f}} = \widehat{\mathbf{U}}\overline{\mathbf{f}},\ \widehat{\mathbf{g}} = \widehat{\mathbf{V}}^\intercal\overline{\mathbf{g}}.\end{gathered} \tag{11}$$

After the normalization, we obtain the leading K eigenvalues $\{\widehat{\sigma_k}\}$ and the corresponding basis functions $\{\widehat{\mathbf{f}}, \widehat{\mathbf{g}}\}$. Since they form a decomposition of the density ratio, the approximated density ratio has the form

$$\widehat{\rho_{1,2}} = \widehat{\mathbf{f}}^\intercal\widehat{\mathbf{\Sigma}}^{\frac{1}{2}}\widehat{\mathbf{g}} \approx \frac{p(X, Y)}{p(X)p(Y)}, \tag{12}$$

where X and Y are defined with the augmentation procedure discussed before. The full procedure will produce the spectrum, the basis functions, and the approximated density ratio $\widehat{\rho_{1,2}}$. The algorithm for the test is illustrated in Algorithm 2.

Gradient estimation. In our implementation, an adaptive filter can be added for gradient estimation, similar to a conventional Adam optimizer. Note that the gradient of r_H has the form

$$\begin{aligned}\frac{\partial r_H}{\partial \theta} = \mathbf{Tr}((\mathbf{R}_{1,2})^{-1}\frac{\partial \mathbf{R}_{1,2}}{\partial \theta}) - \mathbf{Tr}((\mathbf{R}_1)^{-1}\frac{\partial \mathbf{R}_1}{\partial \theta}) \\ -\mathbf{Tr}((\mathbf{R}_2)^{-1}\frac{\partial \mathbf{R}_2}{\partial \theta}).\end{aligned} \tag{13}$$

Thus, we can use adaptive filters to estimate the three ACFs, and substitute the argument within the inverse function with these estimated values. With this, we have introduced the final algorithm of using HFMCA for SSL.

Algorithm 1 HFMCA for SSL - Training Procedure

```
Choose the number of views $L$, the learning rate $\eta$, and initialize CNN $\mathbf{f}_\theta^{(1)}, \mathbf{f}_\theta^{(2)}$.
while convergence is not reached do
    Sample a batch of images $\mathbf{X}$.
    Create $L$ augmentations of $\mathbf{X}$, generating $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_L$.
    Calculate $\mathbf{Z}_l^{(1)} = \mathbf{f}_\theta^{(1)}(\mathbf{X}_l)$ for all $l$, using the backbone.
    Concatenate feature maps as inputs to the projection head: $\mathbf{Z}^{(2)} = \mathbf{f}_\theta^{(2)}([\mathbf{Z}_1^{(1)}, \mathbf{Z}_2^{(1)}, \cdots, \mathbf{Z}_L^{(1)}]^\intercal)$.
    Construct $\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_{1,2}$ using Equation (10).
    Apply the adaptive filter, if needed.
    Compute the gradient using Equation (13).
    Update using SGD: $\theta \leftarrow \theta - \eta \cdot \partial r_H / \partial \theta$.
end while
```

Algorithm 2 HFMCA for SSL - Test Procedure

```
Denote $\mathbf{f}$ and $\mathbf{g}$ as mappings from the original image to projection head outputs and from augmentation to backbone outputs, respectively.
Re-estimate $\mathbf{R}_1, \mathbf{R}_2, \mathbf{P}_{1,2}, \mathbf{R}_{1,2}$ with the full training set samples.
Perform eigendecomposition: $\mathbf{R}_1^{-\frac{1}{2}}\mathbf{P}_{1,2}\mathbf{R}_2^{-\frac{1}{2}} = \widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}^{\frac{1}{2}}\widehat{\mathbf{V}}^\intercal$.
Normalize outputs: $\overline{\mathbf{f}} = \mathbf{R}_1^{-\frac{1}{2}}\mathbf{f}, \overline{\mathbf{g}} = \mathbf{R}_2^{-\frac{1}{2}}\mathbf{g}$.
Normalize outputs: $\widehat{\mathbf{f}} = \widehat{\mathbf{U}}\overline{\mathbf{f}},\ \widehat{\mathbf{g}} = \widehat{\mathbf{V}}^\intercal\overline{\mathbf{g}}$.
Obtain eigenvalues: $\widehat{\mathbf{\Sigma}}$.
Obtain eigenfunctions: $\widehat{\mathbf{f}}$ and $\widehat{\mathbf{g}}$.
Calculate density ratio: $\widehat{\rho_{1,2}} = \widehat{\mathbf{f}}^\intercal\widehat{\mathbf{\Sigma}}^{\frac{1}{2}}\widehat{\mathbf{g}}$.
For test samples, use $\widehat{\mathbf{f}}$ as feature and perform K-Nearest Neighbors (KNN).
```

4. EXPERIMENTS

In this section, we show that HFMCA exhibits faster convergence, higher accuracy, and improved stability in SSL. Our dependence measure can also serve as a quality indicator for the augmentation protocol, independently of classification accuracy.

Regularization hyperparameter. Each time we compute the inverse of any ACFs (e.g., gradient estimation in Equation (13)), similar to the pseudo-inverse, we add a small diagonal matrix, scaled by a regularization parameter, denoted as

Method	**Heads**	**CIFAR10**			**CIFAR100**		
		Epoch 20	Epoch 200	Epoch 800	Epoch 20	Epoch 200	Epoch 800
Methods with two views							
MoCo [24]	128	57.2	83.8	90.0	22.3	45.7	69.8
SimCLR [11]	128	46.7	82.2	87.5	19.6	43.9	65.7
Barlow Twins [25]	2048	45.7	83.5	85.7	28.1	47.1	**70.9**
SimSiam [26]	2048	50.5	83.7	90.0	22.5	39.9	66.0
VICReg [27]	2048	44.8	81.2	90.2	20.3	37.8	68.5
VICRegL [12]	2048	43.2	78.7	89.7	21.5	41.2	67.3
Methods with multiple views							
FastSiam [22]	2048	76.8	87.9	90.1	45.8	62.2	69.9
HFMCA	128	**81.8**	**89.3**	**90.7**	**47.5**	**67.6**	70.3

Table 1: Classification accuracy on CIFAR10 and CIFAR100 highlights HFMCA's effectiveness. HFMCA converges fastest among all methods, retaining near-optimal accuracy.

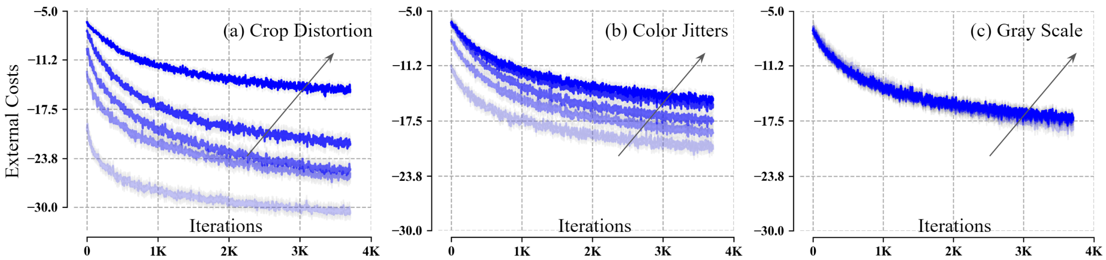

Fig. 2: The learning dynamics of costs (dependence level) are displayed for five distortion strengths across three protocols. The arrow's direction indicates an increase in distortion strength. A lower cost value implies a higher dependence level. The figure implies that as distortion strength increases, the dependence level decreases. Even in extreme cases, a consistent level of dependence remains, intrinsic to the dataset.

Strength	A (%)	C
0	24.8	-30.4
0.25	54.5	-26.1
0.5	67.7	-25.4
0.75	71.2	-21.9
1	69.9	-15.1

(a) Crop Distortion

Strength	A (%)	C
0	49.0	-20.4
0.25	69.6	-18.8
0.5	71.0	-17.5
0.75	70.6	-15.7
1	70.8	-15.0

(b) Color Jitters

Strength	A (%)	C
0	61.2	-18.1
0.25	71.2	-17.4
0.5	70.4	-17.3
0.75	71.4	-17.2
1	70.7	-17.1

(c) Grey Scale

Table 2: A comparison of classification accuracy (A) and costs (C) across three protocols. An increase in distortion strength leads to decreased dependence but improves classification accuracy. The external costs never retain zero in all scenarios, suggesting an intrinsic level of dependence within the dataset.

$\lambda\mathbf{I}$. This ensures the invertibility of the matrices. We found this constant to be important, and it may impact the learned spectrum. To achieve optimal performance in SSL, we select $\lambda = 0.1$.

Fast convergence in self-supervised learning. Our HFMCA model exhibits faster convergence and superior accuracy in SSL, as shown in Table 1. We compared its performance with multiple benchmark models on CIFAR10 and CIFAR100, with the max accuracy achieved over 20, 200, and 800 epochs reported, where HFMCA consistently outperformed them. All experiments use a consistent setup: a ResNet-18 backbone, batch size of 64, SGD optimizer, a learning rate of 0.06, and momentum of 0.9, following benchmark settings. We use standard SimCLR protocols [11] for augmentation and apply a KNN to embedded training images.

In HFMCA, for a batch of 64 images, we generate 128-dimensional feature maps for $L = 9$ distinct augmentations per image using a ResNet-18 backbone. These feature maps are then reshaped into a 3×3 grid, forming a tensor of size $(64, 128, 3, 3)$ which is fed into a 3-layer CNN, creating a 128-dimensional feature per source image. The cost is constructed following Proposition 1 and Fig. 1, while accuracy is evaluated via KNN on the final layer of the ResNet-18 backbone. We compare HFMCA with a varity of standard baselines, and HFMCA offers the fastest convergence and highest accuracy.

Table 1 highlights the benefits of shifting from the conventional similarity-contrastivity model to HFMCA's dependence-orthonormality framework. HFMCA promotes feature diversity via log-determinant-based cost functions, supported by orthonormal decompositions. Basically, it reformulates the task from contrasting two views to the measurement of statistical dependence among L distinct views, and results in more efficient training.

Dependencies versus augmentation protocols. HFMCA's strength as a statistic dependence measure is demonstrated through varying augmentation protocols. Our first observation is the impressive stability of HFMCA across all tests. We observe no occurrence of feature collapse, even under extreme augmentations described later. Second, the dependence measurement provides a novel indication for evaluating the quality of augmentation protocols, independent of classification accuracy.

Notably, even with strong augmentation, some level of dependence among views persists. We consistently observe a decrease in dependence level as we enrich the augmentation. The default augmentations for CIFAR10 [11] include random crops, color jitters, and gray scales. We test five distortion strengths across these three protocols. Each protocol is tested individually, keeping the other two at default values. Random crop strength varies from no cropping at all to the sampling and resizing of any patch from 1×1 to 32×32 as inputs. Color jitter strength refers to the intensity of distortions in brightness, contrast, saturation, and hue. Gray scale strength is the likelihood of images converting to gray scales, with the maximum strength making all images colorless.

Fig. 2 shows training dynamics of the external cost as the dependence level, indicating that random crops impact the most, followed by color jitters, and gray scale. Increased distortion strength reduces dependence level (increases costs), but never reaches strict independence. Intriguingly, even in extreme cases, the learning settles at a certain level of dependence intrinsic to the dataset, which can be interpreted as the intrinsic dimension of the data set. Modeling this intrinsic level of dependence, which is unaffected by the augmentation's richness, can be fundamental to self-supervised learning.

Table 2 further supports our argument by showing the classification accuracy (A) and external costs (EC) for these experiments. The results further support HFMCA's robustness, showing no major accuracy drop or feature collapse with increased distortion. A decrease in external costs corresponds to an increase in classification accuracy. This consistency suggests our dependence measure's potential for evaluating the quality of different augmentation protocols.

Learning dynamics of the eigenspectrum. Finally, we visualize the eigenspectrum's learning curve in the following figure. The displayed learning dynamics reveal additional insights. Note that in the heatmap, color intensity represents eigenvalue magnitude. Each row of the heatmap represents one of the 128 eigenvalues, showing how each eigenvalue is being maximized. Upon observing the spectrum, we notice several properties:

- The eigenvalues are all bounded by 1, with the largest eigenvalue being 1, matching the theoretical property and showcasing the stability of HFMCA;
- The eigenvalues appear to converge sequentially, from the largest to the smallest, with an eigenvalue starting to maximize only after the previous one has almost converged and stabilized;
- The number of positive eigenvalues may indicate the dataset's intrinsic complexity. The figure suggests potential redundancy in the feature dimensions, and this spectrum could guide the selection of optimal network output dimensions.

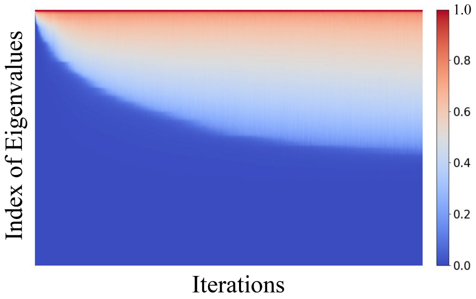

Fig. 3: Visualizing the eigenspectrum learning dynamics: The heatmap's color intensity represents the eigenvalue magnitude. Each row shows the maximization of one of the 128 eigenvalues, and each column displays one of the 10^4 iterations. All eigenvalues in the spectrum are bounded by 1 and converge sequentially from largest to smallest. The spectrum potentially represents the dataset's intrinsic dimensions.

5. ACKNOWLEDGMENT

The authors gratefully acknowledge partial support by the Office of Naval Research (N00014-21-1-2324), (N00014-21-1-2295) and (N00014-21-1-2345).

6. DISCUSSION

This paper provides a theoretical interpretation of SSL features as orthonormal basis functions of the density ratio. We propose HFMCA for learning SSL features with fast convergence and enhanced stability. In the appendix, we further discuss how this analysis extends to internal features to provide model interpretabilities. Our study has not yet incorporated local-level supervision, such as patch augmentation [23], which can be explored in future work.

References

[1] Alfréd Rényi. On measures of dependence. *Acta mathematica hungarica*, 10(3-4):441–451, 1959.

[2] Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.

[3] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6):066138, 2004.

[4] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2 edition, 2006.

[5] Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

[6] Xiao Wang and Maya R. Gupta. Deep information bottleneck. In *International Conference on Learning Representations (ICLR)*, 2016.

[7] 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 *International Conference on Learning Representations (ICLR)*, 2019.

[8] Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019.

[9] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations (ICLR)*, 2017.

[10] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607, 2020.

[12] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicregl: Self-supervised learning of local visual features. *arXiv preprint arXiv:2210.01571*, 2022.

[13] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[14] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R. Devon Hjelm. Mine: Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, volume 80, pages 531–540, 2018.

[15] 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. *arXiv preprint arXiv:1808.06670*, 2018.

[16] Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*, 2020.

[17] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

[18] David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 875–884. PMLR, 2020.

[19] Shao-Lun Huang, Gregory W Wornell, and Lizhong Zheng. Gaussian universal features, canonical correlations, and common information. In *2018 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2018.

[20] Shao-Lun Huang, Anuran Makur, Gregory W Wornell, and Lizhong Zheng. On universal features for high-dimensional learning and inference. *arXiv preprint arXiv:1911.09105*, 2019.

[21] Bo Hu and Jose C Principe. The cross density kernel function: A novel framework to quantify statistical dependence for random processes. *arXiv preprint arXiv:2212.04631*, 2022.

[22] Daniel Pototzky, Azhar Sultan, and Lars Schmidt-Thieme. Fastsiam: Resource-efficient self-supervised learning on a single gpu. In *Pattern Recognition: 44th DAGM German Conference, DAGM GCPR 2022, Konstanz, Germany, September 27–30, 2022, Proceedings*, pages 53–67. Springer, 2022.

[23] Shengbang Tong, Yubei Chen, Yi Ma, and Yann Lecun. Empssl: Towards self-supervised learning in one training epoch. *arXiv preprint arXiv:2304.03977*, 2023.

[24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2020.

[25] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[26] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, 2021.

[27] Shang Wang, Zhixuan Liao, Mathilde Caron, and Piotr Bojanowski. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.