Dependence Induced Representations

Xiangxiang Xu[1] and Lizhong Zheng[1]

***Abstract*— We study the problem of learning feature representations from a pair of random variables, where we focus on the representations that are induced by their dependence. We provide sufficient and necessary conditions for such dependence induced representations, and illustrate their connections to Hirschfeld–Gebelein–Rényi (HGR) maximal correlation functions and minimal sufficient statistics. We characterize a large family of loss functions that can learn dependence induced representations, including cross entropy, hinge loss, and their regularized variants. In particular, we show that the features learned from this family can be expressed as the composition of a loss-dependent function and the maximal correlation function, which reveals a key connection between representations learned from different losses. Our development also gives a statistical interpretation of the neural collapse phenomenon observed in deep classifiers. Finally, we present the learning design based on the feature separation, which allows hyperparameter tuning during inference.**

I. Introduction

Deep learning techniques [1] have demonstrated significant progress in extracting useful information from high-dimensional structured data. The learned feature representations can be easily adapted to different learning objectives, data modalities, or learning tasks. Compared with the practical successes, the understanding of the learned representations is rather limited: though it is widely acknowledged that such representations contain useful information for understanding the statistical behaviors of data [2], theoretical analyses of learned representations are still challenging. In particular, the difficulty comes from the unknown complicated structures behind high-dimensional data and the huge design space of practical learning algorithms. It is often difficult to obtain analytical solutions to the representations, except for a few cases with principled designs [3]. Theoretical analyses often rely on additional assumptions, making the results restricted to specific scenarios.

In contrast to the difficulty faced by theoretical characterizations, deep learning practices suggest the other extreme: despite the huge design space of loss functions and hyperparameter settings, the learned representations can still be reused or adapted to different setups; empirical studies also indicate that representations obtained from completely different methods can be highly similar [4]. A better understanding of this similarity can provide key insights on the learning mechanism and lead to more efficient learning designs.

*This work was supported in part by the Office of Naval Research (ONR) under grant N00014-19-1-2621.

[1]Xiangxiang Xu and Lizhong Zheng are with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 {xuxx, lizhong}@mit.edu

In this paper, we tackle the representation learning problem from a statistical perspective. We consider learning representations from a pair of variables X, Y and focus on the representations determined only by the $X - Y$ dependence. We refer to such representations as dependence induced representations, formalized as the representations invariant to transformations that preserve the dependence structure. We provide sufficient and necessary conditions for such representations and demonstrate their deep connections to the Hirschfeld–Gebelein–Rényi (HGR) maximal correlation functions and the notion of minimal sufficiency. Furthermore, we characterize a family of loss functions, termed D-losses, which can be used for learning dependence induced representations. Specifically, we demonstrate that the optimal feature is the composition of a loss-dependent term and the maximal correlation function, revealing the key connection between different feature representations. We also show that D-losses cover a large class of practical loss functions, including cross entropy and hinge loss as special cases. Our characterization also provides a statistical interpretation of the neural collapse behavior [5] in training deep classifiers. Based on our developments, we propose a design of feature adapters that allow hyperparameter tuning during inference.

II. Notations and Preliminaries

For a random variable Z, we use $\mathcal{Z}$ to denote the corresponding alphabet and use z to denote a specific value in $\mathcal{Z}$. We use P_Z to denote the probability distribution[1] of Z and denote the collection of probability distributions supported on $\mathcal{Z}$ by $\mathcal{P}^{\mathcal{Z}}$. For $n \geq 1$, we denote $[n] \triangleq \{1, 2, \ldots, n\}$.

A. Feature Space and Representation Learning

Given data Z with domain $\mathcal{Z}$, we refer to the mappings from $\mathcal{Z}$ to vector space as feature functions or feature mappings and refer to the mapped values as feature representations (or simply, features). In particular, we use $\mathcal{F}_{\mathcal{Z}} \triangleq \{\mathcal{Z} \to \mathbb{R}\}$ to denote the collection of one-dimensional feature functions, and denote the collection of k-dimensional features by $\mathcal{F}_{\mathcal{Z}}^k \triangleq (\mathcal{F}_{\mathcal{Z}})^k = \{\mathcal{Z} \to \mathbb{R}^k\}$ for each $k \geq 1$. We use $\mathcal{F}_{\mathcal{Z}}^* \triangleq \bigcup_{k \geq 1} \mathcal{F}_{\mathcal{Z}}^k$ to denote all such feature functions. For $f, h \in \mathcal{F}_{\mathcal{Z}}^*$, we denote $\Lambda_{f,h} \triangleq \mathbb{E}\left[f(Z)h^{\mathrm{T}}(Z)\right]$. Specifically, we use Λ_f to represent $\Lambda_{f,f} = \mathbb{E}\left[f(Z)f^{\mathrm{T}}(Z)\right]$.

1) Abstractions; Informational Equivalence: Given two random variables X and Y, following the convention of [6], we call Y an *abstraction* of X if there exists a function $h\colon \mathcal{X} \to \mathcal{Y}$ such that $Y = h(X)$. If each of X, Y is

[1]Throughout our development, we restrict to discrete random variables on finite alphabets with positive probability masses, i.e., $P_Z(z) > 0$ for all $z \in \mathcal{Z}$.

an abstraction of the other, we say that X and Y are *informationally equivalent*[2].

By definition, the conditional expectation $\mathbb{E}[Y|X]$ is an abstraction of X. Specifically, given $f \in \mathcal{F}_{\mathcal{Z}}^*$ and $s \in \mathcal{F}_{\mathcal{Z}}^*$, $f|_s(Z)$ is an abstraction of $s(Z)$, where we have defined

$$f|_s(z) \triangleq \mathbb{E}[f(Z)|s(Z) = s(z)], \qquad \text{for all } z \in \mathcal{Z}. \quad (1)$$

2) Representation Learning Algorithms: We restrict to the representation learning involving a pair of random variables $(X, Y) \sim P_{X,Y}$.We refer to each mapping of the form

$$(X, Y) \mapsto (f, g) \quad (2)$$

as a representation learning algorithm: it takes the random variable pair (X, Y) as the input and outputs feature function pair $f \in \mathcal{F}_{\mathcal{X}}^k$ and $g \in \mathcal{F}_{\mathcal{Y}}^k$ for a given[3] $k \geq 1$, which map original data X, Y to the k-dimensional feature representations $f(X), g(Y) \in \mathbb{R}^k$, respectively. Note that this is a mathematical simplification of the practical learning algorithms: we focus on the input-output relation, and omit the implementation details such as the computation.

B. Canonical Dependence Kernel and Maximal Correlation

Given $(X, Y) \sim P_{X,Y}$, we define the associated *canonical dependence kernel* (CDK) function [8] as a joint function $\mathfrak{i}_{X;Y} \in \mathcal{F}_{\mathcal{X}\times\mathcal{Y}}$, with

$$\mathfrak{i}_{X;Y}(x, y) \triangleq \frac{P_{X,Y}(x, y)}{P_X(y)P_Y(y)} - 1. \quad (3)$$

We can apply the modal decomposition [3], a singular value decomposition (SVD) in function space, to write the CDK as a superposition of rank-one singular modes:

$$\mathfrak{i}_{X;Y}(x, y) = \sum_{i=1}^{K} \sigma_i f_i^*(x) g_i^*(y), \quad (4)$$

where we have the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_K > 0$, and where $K \geq 0$ denotes the rank of $\mathfrak{i}_{X;Y}$. We also have the orthogonality relations of functions f_i^*, g_i^*: $\mathbb{E}\left[f_i^*(X)f_j^*(X)\right] = \mathbb{E}\left[g_i^*(Y)g_j^*(Y)\right] = \delta_{i,j}$, for all $i, j \in [K]$.

The functions f_i^*, g_i^* correspond to the maximally correlated functions in $\mathcal{F}_{\mathcal{X}}$ and $\mathcal{F}_{\mathcal{Y}}$, known as Hirschfeld–Gebelein–Rényi (HGR) maximal correlation functions [9], [10], [11]. To see the connection, let us denote the covariance of $f \in \mathcal{F}_{\mathcal{X}}$ and $g \in \mathcal{F}_{\mathcal{Y}}$ as

$$\operatorname{cov}(f, g) \triangleq \mathbb{E}_{P_{X,Y}}[f(X)g(Y)] - \mathbb{E}_{P_X P_Y}[f(X)g(Y)].$$

Then, for all $i = 1, \ldots, K$, we have $\sigma_i = \operatorname{cov}(f_i^*, g_i^*) = \mathbb{E}_{P_{X,Y}}[f_i^*(X)g_i^*(Y)]$ and $(f_i^*, g_i^*) = \arg\max_{f_i, g_i} \operatorname{cov}(f_i, g_i)$, where the maximization is taken over all $f_i \in \mathcal{F}_{\mathcal{X}}$ and $g_i \in \mathcal{F}_{\mathcal{Y}}$ such that $\mathbb{E}\left[f_i^2(X)\right] = \mathbb{E}\left[g_i^2(Y)\right] = 1$ and $\mathbb{E}\left[f_i(X)f_j^*(X)\right] = \mathbb{E}\left[g_i(Y)g_j^*(Y)\right] = 0$ for all $j < i$.

For convenience, we also define

$$f^* \triangleq (f_1^*, \ldots, f_K^*)^{\mathrm{T}}, \quad g^* \triangleq (g_1^*, \ldots, g_K^*)^{\mathrm{T}}. \quad (5)$$

C. Sufficient Statistics and Minimal Sufficiency

We start with the definition of sufficiency and minimal sufficiency. For more detailed discussions, see, e.g., [12, Section 2.9].

[2]The concept of informational equivalence was also introduced independently in later psychological studies, e.g., [7].

[3]Though the k is required to make the mapping (2) well-defined, we will only specify the choice when necessary.

Definition 1. *Given X, Y, $f(X)$ is a sufficient statistic of X for inferring Y if we have the Markov relation $X - f(X) - Y$. We call $f(X)$ a minimal sufficient statistic if $f(X)$ is sufficient, and for each given sufficient statistic T of X, $f(X)$ is an abstraction of T.*

By definition, all minimal sufficient statistics are informationally equivalent. Therefore, it suffices to specify one representative among the equivalent class, say, $S = s(X)$ from some s. Then, we can characterize the collection of all sufficient statistics as $\{f(X) \colon S \text{ is an abstraction of } f(X)\}$.

We can also express the sufficiency in terms of CDK functions. Specifically, let $\mathfrak{i}_{X;Y}(X, Y)$ indicate the random variable $\gamma(X, Y)$ where $\gamma = \mathfrak{i}_{X;Y}$. Then we have the following property. The proof is omitted.

Property 1. *$f(X)$ is a sufficient statistic if and only if $\mathfrak{i}_{X;Y}(X, Y)$ is an abstraction of $(f(X), Y)$.*

By symmetry, $g(Y)$ is sufficient if and only if $\mathfrak{i}_{X;Y}(X, Y)$ is an abstraction of $(X, g(Y))$. This motivates us to consider a symmetric notion of sufficiency: we say that $f(X)$ and $g(Y)$ are *jointly sufficient* if $\mathfrak{i}_{X;Y}(X, Y)$ is an abstraction of $(f(X), g(Y))$. In particular, we have the following characterization. The proof is omitted.

Property 2. *The following statements are equivalent:*

- *$f(X)$ and $g(Y)$ are jointly sufficient;*
- *Both $f(X)$ and $g(Y)$ are sufficient statistics;*
- *We have the Markov chain $X - f(X) - g(Y) - Y$.*

Remark 1. *We can replace $\mathfrak{i}_{X;Y}(X;Y)$ in these statements by its informationally equivalent transformations. For example, we can obtain the pointwise mutual information $\log(1 + \mathfrak{i}_{X;Y}(x, y)) = \left[\log \frac{P_{X,Y}}{P_X P_Y}\right](x, y)$ by considering the mapping $t \mapsto \log(1 + t)$.*

Remark 2. *We can have equivalent results expressed in terms of information measures. Specifically, the abstraction relation and informational equivalence were originally characterized by using conditional entropies [6], and the sufficiency can be characterized as some mutual information equalities [12, Section 2.9].*

III. Definition and Fundamental Properties

A. Dependence Preserving Transformations

We first formalize *dependence preserving transformations* as follows.

Definition 2. *Given $(X, Y) \sim P_{X,Y}$, a* dependence preserving transformation *generates transformed variables $(\hat{X}, \hat{Y})$, where $\hat{X} = \xi(X, Z)$ and $\hat{Y} = \eta(Y, W)$ are characterized by*

- *transformed alphabets $\hat{\mathcal{X}}, \hat{\mathcal{Y}}$ of $\hat{X}$ and $\hat{Y}$, respectively;*
- *random sources $Z \in \mathcal{Z}$, $W \in \mathcal{W}$ satisfying the Markov chain $Z - X - Y - W$, characterized by conditional distributions $P_{Z|X}, P_{W|Y}$;*
- *deterministic mappings $\xi \colon \mathcal{X} \times \mathcal{Z} \to \hat{\mathcal{X}}$, $\eta \colon \mathcal{Y} \times \mathcal{W} \to \hat{\mathcal{Y}}$ such that X and Y are abstractions of $\hat{X}$, $\hat{Y}$, respec-*

tively.[4]

With a slight abuse of notation, we use ξ^{-1} and η^{-1} to denote the mappings that decode X, Y from $\hat{X}$, $\hat{Y}$, respectively. This gives $X = \xi^{-1}(\hat{X})$ and $Y = \eta^{-1}(\hat{Y})$.

Remark 3. *A degenerated case of dependence preserving transformations is the one-to-one deterministic mapping, which can be obtained by setting both Z and W to constants.*

To see why the dependence is preserved, note that we can obtain $\hat{X} - X - Y - \hat{Y}$ from the Markov relation $Z - X - Y - W$, i.e., the transformations do not introduce additional information. Moreover, there is no loss of information: we have $X - \hat{X} - \hat{Y} - Y$ as X and Y are uniquely determined from $\hat{X}$ and $\hat{Y}$, respectively. We can formalize this dependence preserving property in terms of the CDK as follows. The proof is omitted.

Proposition 1. *Given (X, Y), let $\hat{X} \in \hat{\mathcal{X}}, \hat{Y} \in \hat{\mathcal{Y}}$ be the variables obtained from dependence preserving transformations ξ, η [cf. Definition 2]. Then, for all $(\hat{x}, \hat{y}) \in \hat{\mathcal{X}} \times \hat{\mathcal{Y}}$, we have $\mathfrak{i}_{\hat{X};\hat{Y}}(\hat{x}, \hat{y}) = \mathfrak{i}_{X;Y}(\xi^{-1}(\hat{x}), \eta^{-1}(\hat{y}))$.*

B. Invariance, Maximal Correlation & Minimal Sufficiency

We first introduce the following definition.

Definition 3. *Given (X, Y), we say a mathematical object*[5] *$\theta(X, Y)$ defined on (X, Y) is* dependence induced *if it is invariant to dependence preserving transformations, i.e., $\mathbb{P}\left\{\theta(X, Y) = \theta(\hat{X}, \hat{Y})\right\} = 1$, for all $(\hat{X}, \hat{Y})$ obtained from (X, Y) through dependence preserving transformations.*

Therefore, being dependence induced implies the object relies only on the dependence structure, regardless of the irrelevant information introduced during dependence preserving transformations, such as the Z, W in Definition 2.

Theorem 1. *Given (X, Y), $(f^*(X), g^*(Y))$ is dependence induced, and we can construct a dependence preserving transformation to transform $(f^*(X), g^*(Y))$ into (X, Y), where f^*, g^* are the maximal correlation functions [cf.* (5)*].*

Proof. We first verify that $(f^*(X), g^*(Y))$ is dependence induced. To see this, we assume the $\hat{X}, \hat{Y}$ are transformed from X, Y, respectively, where $\hat{X} = \xi^{-1}(X), \hat{Y} = \eta^{-1}(Y)$ with probability one. Suppose $\mathfrak{i}_{X;Y}$ has the modal decomposition (4). Then, it follows from Proposition 1 that, for all $(\hat{x}, \hat{y}) \in \hat{\mathcal{X}} \times \hat{\mathcal{Y}}$,

$$\begin{aligned}\mathfrak{i}_{\hat{X};\hat{Y}}(\hat{x}, \hat{y}) &= \mathfrak{i}_{X;Y}(\xi^{-1}(\hat{x}), \eta^{-1}(\hat{y})) \\ &= \sum_{i \in [K]} \sigma_i \cdot f_i^*(\xi^{-1}(\hat{x})) \cdot g_i^*(\eta^{-1}(\hat{y})) \\ &= \sum_{i \in [K]} \sigma_i \cdot (f_i^* \circ \xi^{-1})(\hat{x}) \cdot (g_i^* \circ \eta^{-1})(\hat{y}),\end{aligned}$$

which is the modal decomposition of $\mathfrak{i}_{\hat{X};\hat{Y}}$. Therefore, the maximal correlation functions for $\hat{X}, \hat{Y}$ are $f^* \circ \xi^{-1}$, $g^* \circ \eta^{-1}$. This implies

$$\begin{aligned}&\mathbb{P}\left\{(f^* \circ \xi^{-1})(\hat{X}) = f^*(X)\right\} \\ &\quad = \mathbb{P}\left\{(g^* \circ \eta^{-1})(\hat{Y}) = g^*(Y)\right\} = 1.\end{aligned}$$

As a result, it holds with probability one that

$$\left((f^* \circ \xi^{-1})(\hat{X}), (g^* \circ \eta^{-1})(\hat{Y})\right) = (f^*(X), g^*(Y)).$$

Let $S = f^*(X), T = g^*(Y)$, we then construct a dependence preserving transformation that transforms (S, T) into $(\hat{S}, \hat{T}) = (X, Y)$ as follows. Due to the Markov relation $X - S - T - Y$, we can use $Z = X, W = Y$ as the random sources [cf. Definition 2], and set the mappings to $\xi\colon (s, z) \mapsto z$ and $\eta\colon (t, w) \mapsto w$. It can be verified that this transformation satisfies all conditions in Definition 2 and preserves the dependence. □

Remark 4. *Though the mapped results $f^*(X)$ and $g^*(Y)$ are dependence induced, the feature mappings f^*, g^* need to adapt to the dependence preserving transformations and are* not *dependence induced.*

We have the following corollary.

Corollary 1. *A mathematical object $\theta(X, Y)$ defined on (X, Y) is* dependence induced *if and only if $\theta(X, Y) = \theta(f^*(X), g^*(Y))$.*

Proof. Suppose $(\hat{X}, \hat{Y})$ is obtained from (X, Y) by applying a dependence preserving transformation. To see the sufficiency, note that from Theorem 1, $(f^*(X), g^*(Y))$ is dependence induced, and thus $\theta(\hat{X}, \hat{Y}) = \theta(f^*(X), g^*(Y))$. The necessity follows from the fact that (X, Y) can be transformed from $(f^*(X), g^*(Y))$ with a dependence preserving transformation [cf. Theorem 1]. □

Example 1. *the mutual information $I(X; Y) = I(f^*(X); g^*(Y))$ and the matrix $\operatorname{diag}(\sigma_1, \ldots, \sigma_K) = \mathbb{E}\left[f^*(X) g^{*\mathrm{T}}(Y)\right]$ are dependence induced.*

From the corollary, dependence induced mathematical objectives do not contain more information than $(f^*(X), g^*(Y))$. In addition, since $(f^*(X), g^*(Y))$ is dependence induced, we can characterize such objectives by considering only the input $(f^*(X), g^*(Y))$. This gives a canonical form of θ that is invariant to dependence preserving transformations.

Theorem 1 and the corollary also suggest the fundamental role of $f^*(X), g^*(Y)$ (and their informationally equivalent transformations) in characterizing the $X - Y$ dependence. It can be proved that $f^*(X), g^*(Y)$ are sufficient statistics, see, e.g., [8, Section 2]. We can also easily verify it with the notion of joint sufficiency: from (4), $f^*(X), g^*(Y)$ are joint sufficient, and are thus sufficient statistics [cf. Property 2]. Moreover, we can demonstrate that they are indeed minimal sufficient statistics.

Proposition 2. *Given (X, Y), the maximal correlation functions $f^*(X)$, $g^*(Y)$ [cf. 5] are minimal sufficient statistics.*

[4] This can be achieved when we have $P_{\hat{X}|X}(\hat{x}|x) P_{\hat{X}|X}(\hat{x}|x') = 0$ for all $\hat{x} \in \hat{\mathcal{X}}$ and $x, x' \in \mathcal{X}$ with $x \neq x'$, $P_{\hat{Y}|Y}(\hat{y}|y) P_{\hat{Y}|Y}(\hat{y}|y') = 0$ for all $\hat{y} \in \hat{\mathcal{Y}}$ and $y, y' \in \mathcal{Y}$ with $y \neq y'$.

[5] This includes scalars, matrices, functions, random variables, etc.

Proof. By symmetry, it suffices to establish the minimal sufficiency of $f^*(X)$. As $f^*(X)$ is a sufficient statistic, it remains only to prove the minimality. From the modal decomposition (4), we have $f_i^*(x) = \sigma_i^{-1}\mathbb{E}\left[g_i^*(Y)|X=x\right]$ for all $i \in [K]$. Therefore, for any given sufficient statistic $S = s(X)$, we have

$$f^*(x) = \Sigma^{-1}\mathbb{E}\left[g^*(Y)|X=x\right] = \Sigma^{-1}\mathbb{E}\left[g^*(Y)|S=s(x)\right],$$

where $\Sigma \triangleq \mathrm{diag}(\sigma_1, \ldots, \sigma_K)$. To obtain the second equality, we have used the fact that $P_{Y|X=x}$ is a function of $s(x)$, since $s(X)$ is sufficient. Therefore, $f^*(X)$ is an abstraction of any sufficient statistic S, which completes the proof. □

C. Dependence Induced Representations and Learning

We then consider the impacts of dependence preserving transformations on representation learning. Specifically, we focus on the learned representations that are dependence induced, i.e., invariant to such transformations. Formally, Let $\mathsf{alg}\colon (X,Y) \mapsto (f,g)$ be a representation learning algorithm that learns k-dimensional feature functions $f \in \mathcal{F}_{\mathcal{X}}^k, g \in \mathcal{F}_{\mathcal{Y}}^k$ from $(X,Y) \sim P_{X,Y}$. Suppose $(\hat{X},\hat{Y})$ are obtained by applying a dependence preserving transformation on (X,Y), via $\hat{X} = \xi(X,Z), \hat{Y} = \eta(Y,W)$, where Z, W, ξ, η satisfy the conditions in Definition 2. Applying the same algorithm to the transformed data gives feature functions

$$(f^{(\xi)}, g^{(\eta)}) = \mathsf{alg}(\hat{X},\hat{Y}),$$

where we have denoted the learned feature functions by $f^{(\xi)} \in \mathcal{F}_{\hat{\mathcal{X}}}^k$ and $g^{(\eta)} \in \mathcal{F}_{\hat{\mathcal{Y}}}^k$, with $\hat{\mathcal{X}}, \hat{\mathcal{Y}}$ denoting transformed alphabets. Note that the transformations ξ, η are not separately input to the algorithm.

From Definition 3, the learned representations are dependence induced if

$$\mathbb{P}\left\{f^{(\xi)}(\hat{X}) = f(X), g^{(\eta)}(\hat{Y}) = g(Y)\right\} = 1$$

for all possible generating processes of $\hat{X}$, $\hat{Y}$. We refer to the algorithm alg as a *dependence learning algorithm* if it learns dependence induced representations. Specifically, we have the following characterization of dependence induced representations and corresponding learning algorithms.

Theorem 2. *The collection $\mathscr{A}$ of dependence learning algorithms is given by*

$$\mathscr{A} \triangleq \{\mathsf{alg}\colon \mathsf{alg}(X,Y) = (\phi \circ f^*, \psi \circ g^*), \\ (\phi,\psi) = \mathsf{alg}(f^*(X), g^*(Y))\}, \quad (6)$$

where f^*, g^* are the maximal correlation functions [cf. (5)*].*

Proof. We first verify that each $\mathsf{alg} \in \mathscr{A}$ for $\mathscr{A}$ as defined in (6) is a dependence learning algorithm. It suffices to note that the representation pair learned from alg is $(\phi(f^*(X)), \psi(g^*(Y)))$. This is dependence induced since both $(f^*(X), g^*(Y))$ and the mapping pair (ϕ,ψ) are dependence induced [cf. Theorem 1, Corollary 1 and (6)].

To see the reverse direction, suppose alg is a dependence learning algorithm, and let $(\phi,\psi) = \mathsf{alg}(f^*(X), g^*(Y))$, $(\hat{f},\hat{g}) = \mathsf{alg}(X,Y)$ denote the learned feature functions. From Theorem 1, we can obtain (X,Y) from $(f^*(X), g^*(Y))$ by applying a dependence preserving transformation. As the representations are invariant to this transformation, we have $\phi(f^*(X)) = \hat{f}(X)$ and $\psi(g^*(Y)) = \hat{g}(Y)$ with probability one. This gives $\hat{f} = \phi \circ f^*, \hat{g} = \psi \circ g^*$, which implies that $\mathsf{alg} \in \mathscr{A}$. □

From Theorem 2, each dependence learning algorithm is uniquely specified by the function pair (ϕ,ψ). This gives a canonical form of the algorithm since the mappings ϕ,ψ are dependence induced (cf. (6)). The composition structure $(\phi \circ f^*, \psi \circ g^*)$ allows separate learning implementations of (ϕ,ψ) and (f^*, g^*), as we will detail in later discussions.

Moreover, all dependence induced representations are abstractions of $f^*(X)$, $g^*(Y)$, which are minimal sufficient statistics [cf. Proposition 2]. As a result, the dependence learning algorithms extract only dependence-related information. From the definition of minimal sufficiency, we can readily obtain the following corollary of Theorem 2. The proof is omitted.

Corollary 2. *Given (X,Y) and a dependence learning algorithm $\mathsf{alg} \in \mathscr{A}$, suppose $(f,g) = \mathsf{alg}(X,Y)$ are the output feature functions. Then,*

- *For any sufficient statistics $S = s(X)$ and $T = t(Y)$, $f(X)$, $g(Y)$ are abstractions of S and T, respectively;*
- *If $f(X)$ and $g(Y)$ are jointly sufficient, then $f(X)$ and $g(Y)$ are minimal sufficient statistics.*

In particular, the second claim of Corollary 2 provides a way to test the minimal sufficiency by constructing corresponding dependence learning algorithms. An example is the following generalization of Proposition 2.

Proposition 3. *Given (X,Y) and a one-to-one mapping $\tau\colon [-1,\infty) \to \mathbb{R}$, suppose $f \in \mathcal{F}_{\mathcal{X}}^k$ and $g \in \mathcal{F}_{\mathcal{Y}}^k$ factorize $\tau(\mathfrak{i}_{X;Y})$, i.e., $\tau(\mathfrak{i}_{X;Y}(x,y)) = f^{\mathrm{T}}(x)g(y)$ for all $x \in \mathcal{X}, y \in \mathcal{Y}$, and $\mathrm{rank}(\Lambda_f) = \mathrm{rank}(\Lambda_g) = k$. Then, $f(X)$ and $g(Y)$ are minimal sufficient statistics.*

Proof. Given (X,Y), consider the representation learning algorithm that outputs the solution to the optimization problem

$$\underset{f\in\mathcal{F}_{\mathcal{X}}^k, g\in\mathcal{F}_{\mathcal{Y}}^k}{\mathrm{minimize}}\ \mathbb{E}_{P_X P_Y}\left[\left|\tau(\mathfrak{i}_{X;Y}(X,Y)) - f^{\mathrm{T}}(X)g(Y)\right|^2\right].$$

It can be easily verified that the algorithm is a dependence learning algorithm. The solution is any f,g that factorize $\tau(\mathfrak{i}_{X;Y})$, i.e.,

$$f^{\mathrm{T}}(X)g(Y) = (\tau \circ \mathfrak{i}_{X;Y})(X,Y). \quad (7)$$

Since $\mathrm{rank}(\Lambda_f) = \mathrm{rank}(\Lambda_g) = k$, the choice of (f,g) is unique up to some invertible linear transformation pairs. In addition, (7) implies that $f(X)$ and $g(Y)$ are jointly sufficient. Finally, the minimal sufficiency follows from Corollary 2. □

We conclude this section by considering two extreme cases: independence and strict dependence, summarized as the following corollary. The proof is omitted.

Corollary 3. *Given (X,Y) and $\mathsf{alg} \in \mathscr{A}$, let $(f,g) = \mathsf{alg}(X,Y)$ be the learned feature functions. We have*

1) *if X and Y are independent, i.e., $P_{X,Y} = P_X P_Y$, then $f(X)$ and $g(Y)$ are constants and are independent of the marginal distributions P_X and P_Y;*

2) *if Y is an abstraction of X, then $f(X)$ is an abstraction of Y.*

Remark 5. *From our developments, we can also obtain results about information measures. For example, it follows from Theorem 2 that the dependence induced representations $f(X), g(Y)$ satisfy $H(f(X)) \leq H(f^*(X))$ and $H(g(Y)) \leq H(g^*(Y))$, where $H(\cdot)$ denotes the entropy.*

IV. D-Losses and Dependence Learning

Our discussions in the previous section have characterized the theoretical behaviors of dependence learning algorithms. It is, however, difficult to directly apply these results to practical deep representation learning, e.g., determining if a representation learning algorithm is a dependence learning algorithm or constructing dependence learning algorithms. To address these problems, it is necessary also to consider practical implementation and develop operable characterizations. To this end, we consider representation learning algorithms characterized by corresponding loss functions and characterize sufficient conditions on losses for constructing dependence learning algorithms. We investigate a family of loss functions, termed D-losses, and demonstrate their deep connections to learning practices.

A. D-Loss Family

Given $(X, Y) \sim P_{X,Y}$, let $\Gamma = \Gamma(f, g; P_{X,Y})$ denote a functional defined on $(f, g) \in \mathcal{F}_{\mathcal{X}}^k \times \mathcal{F}_{\mathcal{Y}}^k$. We will omit $P_{X,Y}$ and write $\Gamma(f, g)$ if the choice of $P_{X,Y}$ is clear from the context. Specifically, when we evaluate Γ on a dataset $\{(x_i, y_i)\}_{i=1}^n$, the $P_{X,Y}$ corresponds to the empirical distribution $\hat{P}_{X,Y}(x, y) \triangleq \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{\{x=x_i, y=y_i\}}$, where $\mathbb{1}_{\{\cdot\}}$ denotes the 0-1 indicator function.

We define the D-losses as follows.

Definition 4 (D-loss)**.** *Given $(X, Y) \sim P_{X,Y}$, a functional Γ defined on $(f, g) \in \mathcal{F}_{\mathcal{X}}^k \times \mathcal{F}_{\mathcal{Y}}^k$ is a D-loss if*

1) *suppose $\phi \circ \xi \in \mathcal{F}_{\mathcal{X}}^k$ and $\psi \circ \eta \in \mathcal{F}_{\mathcal{Y}}^k$, then we have*
$$\Gamma(\phi \circ \xi, \psi \circ \eta; P_{X,Y}) = \Gamma(\phi, \psi; P_{\xi(X),\eta(Y)}); \quad (8)$$
2) *for all $s \in \mathcal{F}_{\mathcal{X}}^*$ and $t \in \mathcal{F}_{\mathcal{Y}}^*$ that satisfy the Markov relation $X - s(X) - t(Y) - Y$, we have*
$$\Gamma(f|_s, g|_t) \leq \Gamma(f, g), \quad (9)$$
where $f|_s$ and $g|_t$ are as defined in (1)*.*

Specifically, we say a D-loss Γ is regular, if in (9), $\Gamma(f, g) = \Gamma(f|_s, g|_t) < \infty$ implies $(f, g) = (f|_s, g|_t)$.

We can obtain dependence learning algorithms from the D-loss family, demonstrated as follows.

Theorem 3. *Given $(X, Y) \sim P_{X,Y}$, suppose L defined on $(f, g) \in \mathcal{F}_{\mathcal{X}}^k \times \mathcal{F}_{\mathcal{Y}}^k$ is a D-loss and $v_k(X, Y) \triangleq \min_{(f,g)\in\mathcal{F}_{\mathcal{X}}^k \times \mathcal{F}_{\mathcal{Y}}^k} L(f, g; P_{X,Y})$ exists. Then,*

1) *$v_k(X, Y)$ is dependence induced;*
2) *there exist dependence induced mappings ϕ, ψ, such that $(f, g) = (\phi \circ f^*, \psi \circ g^*)$ achieves the minimum value v_k, where f^*, g^* are as defined in* (5)*;*
3) *if L is regular, then the representation learning algorithm $\big((X, Y) \mapsto \arg\min_{f,g} L(f, g; P_{X,Y})\big)$ is a dependence learning algorithm.*

Proof. We begin with the first two claims. Let $S = f^*(X), T = g^*(Y)$, and we use $\mathcal{S}, \mathcal{T}$ denote the corresponding alphabets. Suppose $(\hat{f}, \hat{g}) \in \mathcal{F}_{\mathcal{X}}^k \times \mathcal{F}_{\mathcal{Y}}^k$ achieves the minimum value $v_k(X, Y)$, and let $(\hat{\phi}, \hat{\psi}) \in \mathcal{F}_{\mathcal{S}}^k \times \mathcal{F}_{\mathcal{T}}^k$ be the mappings such that $\hat{f}|_{f^*} = \hat{\phi} \circ f^*, \hat{g}|_{g^*} = \hat{\psi} \circ g^*$. Then, from the Markov relation $X - S - T - Y$ and (9), we obtain
$$\begin{aligned} L(\hat{f}, \hat{g}; P_{X,Y}) &\geq L(\hat{f}|_{f^*}, \hat{g}|_{g^*}; P_{X,Y}) \\ &= L(\hat{\phi} \circ f^*, \hat{\psi} \circ g^*; P_{X,Y}) = L(\hat{\phi}, \hat{\psi}; P_{S,T}), \end{aligned}$$
where the last equality follows from (8). Note that since $(\hat{f}, \hat{g})$ achieves the minimum value of L, the inequality must hold with equality, so the minimum value v_k can be achieved by $(f, g) = (\hat{\phi} \circ f^*, \hat{\psi} \circ g^*)$. In addition, we have
$$(\hat{\phi}, \hat{\psi}) \in \underset{\phi\in\mathcal{F}_{\mathcal{S}}^k, \psi\in\mathcal{F}_{\mathcal{T}}^k}{\arg\min} \; L(\phi, \psi; P_{S,T}), \quad (10)$$
since otherwise we get an L value smaller than $L(\hat{f}, \hat{g}; P_{X;Y})$. From Corollary 1, both v_k and $(\hat{\phi}, \hat{\psi})$ and dependence induced. Finally, when L is regular, the equality $L(\hat{f}, \hat{g}) = L(\hat{f}|_{f^*}, \hat{g}|_{g^*})$ implies $(\hat{f}, \hat{g}) = (\hat{\phi} \circ f^*, \hat{\psi} \circ g^*)$. Therefore, it follows from (10) and Theorem 2 that $\big((X, Y) \mapsto \arg\min_{f,g} L(f, g; P_{X,Y})\big)$ is a dependence learning algorithm. □

Combining Theorem 3 and Theorem 2, if L is a regular D-loss, it suffices to solve $\text{minimize}_{\phi,\psi} L(\phi, \psi; P_{S,T})$ instead of the original problem $\text{minimize}_{f,g} L(f, g; P_{X,Y})$, where we have defined $S = f^*(X), T = g^*(Y)$. In particular, the optimal solution takes a composition form $(f, g) = (\phi \circ f^*, \psi \circ g^*)$, where only ϕ, ψ depend on the form of L. This separation between loss-dependent functions (ϕ, ψ) and loss-invariant (f^*, g^*) makes it possible to design efficient implementation for D-loss minimization problems, which we will detail in the next section.

Comparing the first claim with the second claim, a difference between D-losses and regular D-losses is that minimizing D-losses can give solutions not in the form $(\phi \circ f^*, \psi \circ g^*)$. However, from the first claim, restricting to such forms does not affect the optimality. Therefore, it is without loss of generality to restrict to such forms. Once we have this restriction, the D-loss minimization algorithm is also a dependence learning algorithm. Moreover, we can always convert a D-loss into a regular D-loss by introducing regularization terms. One example is demonstrated as follows. We omit its proof.

Property 3. *If Γ is a D-loss, then for all $\lambda > 0, \mu > 0$, $\big(\Gamma(f, g) + \lambda \cdot \mathbb{E}\left[\|f(X)\|^2\right] + \mu \cdot \mathbb{E}\left[\|g(Y)\|^2\right]\big)$ is a regular D-loss.*

We can interpret the regularization terms in Property 3 as a weight decay [13] on feature representations, which can be implicitly introduced in optimization, e.g., stochastic gradient descent. Due to these connections, we will consider general D-losses in the discussions that follow, and restrict to the optimal solutions of the form $(\phi \circ f^*, \psi \circ g^*)$. Specifically, we introduce several useful properties for constructing D-losses. The proofs can be obtained by applying Jensen's inequality and noting the Markov relation $X - S - T - Y$.

Property 4. *Suppose $\gamma\colon \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$ is convex with respect to both arguments, then $(f, g) \mapsto \mathbb{E}_{P_{X,Y}}\left[\gamma(f(X), g(Y))\right]$*

and $(f,g) \mapsto \mathbb{E}_{P_X P_Y}\left[\gamma(f(X), g(Y))\right]$ *are D-losses.*

Property 5. *Suppose* Γ *is a D-loss on* $(f,g) \in \mathcal{F}_{\mathcal{X}}^k \times \mathcal{F}_{\mathcal{Y}}^k$*, and* $\nu\colon \mathbb{R}\times\mathbb{R}^k\times\mathbb{R}^k\times\mathbb{R}^{k\times k} \to \mathbb{R}$ *is nondecreasing with respect to the first argument, then* $\nu\left(\Gamma(f,g), \mathbb{E}\left[f(X)\right], \mathbb{E}\left[g(Y)\right], \Lambda_{f,g}\right)$ *is also a D-loss, where* $\Lambda_{f,g} = \mathbb{E}\left[f(X)g^{\mathrm{T}}(Y)\right]$.

Given l losses, we can obtain a new loss by using a mapping $\mathbb{R}^l \to \mathbb{R}$ that aggregates the losses. We refer to the operation as a monotonic aggregation if the mapping is nondecreasing with respect to each argument.

Property 6. *A monotonic aggregation of D-losses is a D-loss.*

B. D-losses in Learning Practices

We demonstrate the connection between D-losses and learning practices, including loss functions and learning techniques such as regularization. For convenience, we adopt extended-value extension [cf. [14, Section 3.1.2]], which allows the value of Γ to be infinity. In particular, we define the characteristic function (0-infinity indicator function) of a set $\mathcal{D} \subset \mathbb{R}^m$, such that for all $\underline{v} \in \mathbb{R}^m$,

$$\mathbb{I}_{\mathcal{D}}(\underline{v}) \triangleq \begin{cases} 0, & \underline{v} \in \mathcal{D} \\ \infty, & \underline{v} \notin \mathcal{D} \end{cases}. \tag{11}$$

1) Log Loss (Softmax & Cross Entropy): We demonstrate the log loss minimization problem is equivalent to a D-loss minimization problem. Our development will use the following fact.

Fact 1. *The functional* $\mathbb{E}_{P_X}\left[\log \mathbb{E}_{P_Y}\left[\exp(f^{\mathrm{T}}(X)g(Y))\right]\right]$ *defined on* $(f,g) \in \mathcal{F}_{\mathcal{X}}^k \times \mathcal{F}_{\mathcal{Y}}^k$ *is a D-loss, where the inner expectation is taken w.r.t.* $Y \sim P_Y$*, and the outer expectation is taken w.r.t.* $X \sim P_X$.

In particular, let X and Y denote the data variable and categorical label, respectively. We use $h(x) \in \mathbb{R}^d$ to denote the feature representation, which corresponds to the feature in the last hidden layer, and denote the weight and bias term corresponding to $Y = y$ as $w(y) \in \mathbb{R}^d$ and $b(y) \in \mathbb{R}$, respectively. By applying the softmax function, we get a posterior parameterized by h, w, b:

$$\tilde{P}_{Y|X}^{(h,w,b)}(y|x) = \frac{\exp(h^{\mathrm{T}}(x)w(y) + b(y))}{\sum_{y' \in \mathcal{Y}} \exp(h^{\mathrm{T}}(x)w(y') + b(y'))}. \tag{12}$$

Then, the log loss is given by the expected value of $-\log \tilde{P}_{Y|X}^{(h,w,b)}$, i.e., $\mathbb{E}_{(\hat{X},\hat{Y})\sim P_{X,Y}}\left[-\log \tilde{P}_{Y|X}^{(h,w,b)}(\hat{Y}|\hat{X})\right]$. Let $k = d+1$, and we define the calibrated bias term $\beta(y) \triangleq b(y) - \log P_Y(y)$, and features

$$f = \begin{bmatrix} 1 \\ h \end{bmatrix} \in \mathcal{F}_{\mathcal{X}}^k, \quad g = \begin{bmatrix} \beta \\ w \end{bmatrix} \in \mathcal{F}_{\mathcal{Y}}^k. \tag{13}$$

Then, we can rewrite the log loss as $-\mathbb{E}\left[f^{\mathrm{T}}(X)g(Y)\right] + \mathbb{E}_{P_X}\left[\log \mathbb{E}_{P_Y}\left[\exp(f^{\mathrm{T}}(X)g(Y))\right]\right] + H(Y)$, where $H(Y)$ denotes the entropy of P_Y. Therefore, minimizing log loss is equivalent to the minimization of the following functional defined on $(f,g) \in \mathcal{F}_{\mathcal{X}}^k \times \mathcal{F}_{\mathcal{Y}}^k$:

$$\begin{aligned} &-\mathbb{E}\left[f^{\mathrm{T}}(X)g(Y)\right] + \mathbb{E}_{P_X}\left[\log \mathbb{E}_{P_Y}\left[\exp(f^{\mathrm{T}}(X)g(Y))\right]\right] \\ &\quad + \mathbb{E}\left[\mathbb{I}_{\{1\}}(f_1(X))\right], \end{aligned} \tag{14}$$

where f_1 denotes the first dimension of f, and where $\mathbb{I}$ is the characteristic function as defined in (11). Note that $\mathbb{E}\left[\mathbb{I}_{\{1\}}(f_1(X))\right] < \infty$ if and only if $\mathbb{P}\left\{f_1(X) = 1\right\} = 1$, which guarantees that the optimal f takes the form of (13).

From Fact 1 and Property 4, Property 5, Property 6, the extended log loss (14) is a D-loss. Therefore, for a properly regularized log loss (e.g., Property 3), the optimal $h(X)$ and $w(Y)$ are dependence induced. It is worth mentioning that though $b(Y)$ is not dependence induced in general, the calibrated bias $\beta(Y)$ is dependence induced.

2) Support Vector Machine [15]: We then consider the loss applied in support vector machine (SVM). In particular, we consider a binary classification problem on data X, where the label $Y \in \mathcal{Y} = \{-1, 1\}$ is balanced. Suppose our goal is to find the optimal feature $h(x) \in \mathbb{R}^d$, weight $\underline{w} \in \mathbb{R}^d$, and bias $b \in \mathbb{R}$. The SVM loss for the separating hyperplane $\langle \underline{w}, h(x)\rangle + b = 0$ can be written as [13]

$$\mathbb{E}_{P_{X,Y}}\left[\ell_{\text{hinge}}(Y, \langle \underline{w}, h(X)\rangle + b)\right] + \lambda \cdot \|\underline{w}\|^2, \tag{15}$$

where $\ell_{\text{hinge}}\colon \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$ denotes the hinge loss, defined as $\ell_{\text{hinge}}(y,z) \triangleq (1 - yz)^+$ with $x^+ \triangleq \max\{0, x\}$, and where $\lambda > 0$ is a hyperparameter. Let $k = d+1$ and define feature functions $f \in \mathcal{F}_{\mathcal{X}}^k, g \in \mathcal{F}_{\mathcal{Y}}^k$ as

$$f(x) = \begin{bmatrix} h(x) \\ 1 \end{bmatrix}, \quad g(y) = y \cdot \begin{bmatrix} \underline{w} \\ b \end{bmatrix}. \tag{16}$$

Then, we have $\ell_{\text{hinge}}(Y, \langle \underline{w}, h(X)\rangle + b) = (1 - \langle \underline{w}Y, h(X)\rangle - bY)^+ = (1 - \langle f(X), g(Y)\rangle)^+$ and $\|\underline{w}\|^2 = \mathbb{E}\left[\|g_{[d]}(Y)\|^2\right]$ where we have defined $g_{[d]} \triangleq (g_1, \ldots, g_d)^{\mathrm{T}}$ and thus $g_{[d]}(Y) = Y \cdot \underline{w}$.

Therefore, the minimization of (15) is equivalent to minimizing the functional

$$\begin{aligned} &\mathbb{E}\left[\left(1 - \langle f(X), g(Y)\rangle\right)^+\right] + \lambda \cdot \mathbb{E}\left[\|g_{[d]}(Y)\|^2\right] \\ &\quad + \mathbb{E}\left[\mathbb{I}_{\{1\}}(f_k(X))\right] + \mathbb{I}_{\{0\}}(\mathbb{E}\left[g(Y)\right]), \end{aligned} \tag{17}$$

where $f_k(X)$ denotes the k-th dimension of f. To see this, note that $\mathbb{E}\left[\mathbb{I}_{\{1\}}(f_k(X))\right] < \infty$ if and only if $\mathbb{P}\left\{f_k(X) = 1\right\} = 1$, and $\mathbb{I}_{\{0\}}(\mathbb{E}\left[g(Y)\right]) < \infty$ if and only if $\mathbb{E}\left[g(Y)\right] = 0$, i.e., $g(-1) = -g(1)$. As a result, the optimal solution (f,g) that minimizes (17) must take the form (16). Note that from Property 4, the first three terms of (17) are D-losses. Then, by combining Property 5 and Property 6, we can verify (17) is a D-loss.

3) Variational Forms of φ-Divergences: From Property 4, $\left(-\mathbb{E}_{P_{X,Y}}\left[f^{\mathrm{T}}(X)g(Y)\right] + \mathbb{E}_{P_X P_Y}\left[u(f^{\mathrm{T}}(X)g(Y))\right]\right)$ is a D-loss for all convex function $u\colon \mathbb{R} \to \mathbb{R}$. Such functionals appear in characterizing variational forms of φ-divergences [16], where $u = \varphi^\star$ is the convex conjugate of some convex function $\varphi\colon [0,\infty) \to \mathbb{R}$ with $\varphi(1) = 0$.

4) Learning Techniques: Regularization, Constrained Features, and Feature Nesting: We consider several learning techniques that preserve the D-loss structure. We first consider the regularization. Given a D-loss $L(f,g)$, we can construct the regularized loss $L(f,g) + \lambda \cdot R(f,g)$, where $R(f,g)$ is the regularization term and $\lambda > 0$ is the hyperparameter. Then, from Property 6, the regularized loss is a D-loss if $R(f,g)$ is a D-loss. From Property 4, the squared distance $\mathbb{E}\left[\|f(X) - g(Y)\|^2\right]$, and generally $\mathbb{E}\left[\|f(X) - g(Y)\|_p\right]$ for $p \geq 1$ are D-losses, where $\|\underline{v}\|_p \triangleq (\sum_i v_i^p)^{\frac{1}{p}}$ denotes the p-norm. Similarly, $\mathbb{E}\left[\|f(X)\|_p\right]$, $\mathbb{E}\left[\|g(Y)\|_p\right]$, $p \geq 1$ and $\mathbb{E}\left[\|f(X)\|^2\right]$, $\mathbb{E}\left[\|g(Y)\|^2\right]$ are also D-losses. During imple-

mentation, the expectation is estimated by the corresponding empirical averages over each mini-batch.

In contrast to the batch-wise regularization terms, another common practice is directly introducing constraints on each instance of feature representations, e.g., force features to be nonnegative. With the characteristic function (11), we can use an extended loss to characterize such constraints. For example, suppose the original loss $L(f, g)$ is a D-loss and $f(X)$ is constrained to be within a set $\mathcal{C} \subset \mathbb{R}^k$. Then we can consider the extended loss $L(f, g) + \mathbb{E}\left[\mathbb{I}_{\mathcal{C}}(f(X))\right]$, which leads to the same optimal solution. From Property 4, the extended loss is a D-loss if $\mathcal{C}$ is convex. Two important examples of such convex sets are: 1) p-norm balls, e.g., $\mathcal{C} = \{\underline{v} \in \mathbb{R}^k \colon \|\underline{v}\|_p \leq 1\}$, where each feature instance is projected onto the norm ball; 2) nonnegative orthant, i.e., $\mathcal{C} = \{\underline{v} \in \mathbb{R}^k \colon v_i \geq 0\}$, corresponding to learning nonnegative features.

The regularization methods are also used to combine multiple training objectives by optimizing their weighted sum. From Property 6, if each objective is a D-loss, the weighted sum will also be a D-loss. A special example is *Feature Nesting*, which constructs the weighted sum of the same loss applied on a set of nested features. Specifically, suppose the loss L is defined for all d-dimensional feature function pair (f, g), e.g., $\mathbb{E}\left[f^{\mathrm{T}}(X)g(Y)\right]$. Then, give $(f, g) \in \mathcal{F}_{\mathcal{X}}^k \times \mathcal{F}_{\mathcal{Y}}^k$, we consider the nested loss

$$\sum_{i=1}^{k} c_i \cdot L(f_{[i]}, g_{[i]}), \tag{18}$$

where $c_i, i = 1, \ldots, k$ are non-negative weights, and where for each $i \in [k]$, we have defined $f_{[i]} \in \mathcal{F}_{\mathcal{X}}^i, g_{[i]} \in \mathcal{F}_{\mathcal{Y}}^i$ as

$$f_{[i]} \triangleq (f_1, \ldots, f_i)^{\mathrm{T}}, \qquad g_{[i]} \triangleq (g_1, \ldots, g_i)^{\mathrm{T}}. \tag{19}$$

Such construction is applied to induce structures on the representations f, g, referred to as the *nesting technique* [3, Section 4], also called *matryoshka representation learning* [17]. The nested loss (18) is a D-loss if L is a D-loss.

C. Representations Induced by Strict Dependence

We conclude this section by discussing a special case where X and Y are strictly dependent, and we have the following characterization [cf. Corollary 3].

Proposition 4. *Given a regular D-loss $L(f, g; P_{X,Y})$ defined for $f \in \mathcal{F}_{\mathcal{X}}^k, g \in \mathcal{F}_{\mathcal{Y}}^k$, let us define the representation learning algorithm* $\mathsf{alg}\colon (X, Y) \mapsto \arg\min_{f,g} L(f, g; P_{X,Y})$. *Suppose $Y = \mathtt{y}(X)$ for a mapping $\mathtt{y}\colon \mathcal{X} \to \mathcal{Y}$. Then, we have*

$$\mathsf{alg}(X, Y) = \{(\phi \circ \mathtt{y}, \psi) \colon (\phi, \psi) \in \mathsf{alg}(Y, Y)\}. \tag{20}$$

Note that a specific case of Proposition 4 is the classification scenario, where Y corresponds to the label of data X. The mapping $\mathtt{y}$ corresponds to the labeling function, and $Y = \mathtt{y}(X)$ indicates that we can obtain such a label from data without any ambiguities. This happens when $P_{X,Y}$ is the empirical distribution of a classification dataset $\{(x_i, y_i)\}_{i=1}^n$ with distinct X samples, i.e., $x_i \neq x_j$ for all $i \neq j$. For such datasets, from (20), the optimal feature function f satisfies $f(x) = \phi(\mathtt{y}(x))$, i.e., all X samples with the same label $\mathtt{y}(X)$ share the same representation.

This phenomenon has been observed in training classification deep neural networks with cross entropy [5], referred to as the "neural collapse"[6], where theoretical characterizations were discussed therein and also in subsequent works [18], [19]. Our development demonstrates that this phenomenon is essentially a consequence of the special dependence structure of training data and is shared by a large collection of losses.

V. Learning With Feature Adapters

The composition structure of dependence induced representations provides a structured implementation of dependence learning algorithms, where we use $S = f^*(X), T = g^*(Y)$ as an intermediate interface for feature representations. As shown in Fig. 1, when S, T are given, the minimization of D-loss $L(f, g; P_{X,Y})$ can be solved by minimizing $L(\phi, \psi; P_{S,T})$ over (ϕ, ψ), and we can retrieve the optimal representations via $f(X) = \phi(S), g(Y) = \psi(T)$ [cf. Theorem 3]. We call such ϕ, ψ as *feature adapters*. Therefore, we can obtain solutions to different D-losses by training only the feature adapters ϕ, ψ, without changing f^*, g^*. The maximal correlation functions f^*, g^* can be learned separately, e.g., as pre-trained deep neural networks, where the training details will be discussed later.

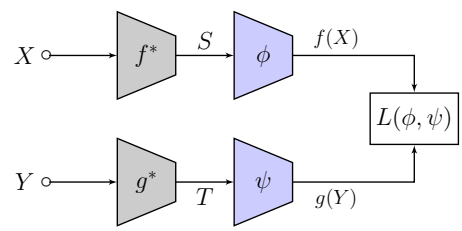

Fig. 1. Feature learning by training adapters ϕ, ψ. The feature extractors f^*, g^* can be frozen pre-trained networks or non-trainable modules.

The separation enables more efficient implementation of the learning procedures. In particular, when S, T have much simpler structures compared with original X, Y, we can use lightweight neural networks as the adapters ϕ, ψ, and the adapter training does not require back-propagating learning errors [20] to the f^*, g^* modules.

Note that the representation interface S, T are both informationally and computationally efficient: from the minimal sufficiency, S, T contain necessary information without introducing redundancies; even without adaptations, we can use f^*, g^* to conduct inference tasks by some linear assembling processes, as discussed in [3].

A. Feature Adapters for Dependence Learning

The feature adapters allow us to adapt to different D-losses by training only the adapters. In particular, we can use feature adapters to implement constrained feature learning and inference-time hyperparameter tuning.

[6]The Neural Collapse (NC) [5] was defined as the collection of four interconnected phenomena, NC1-NC4, where NC1 corresponds to the phenomenon of $f(X)$ being an abstraction of Y, also referred to as the "variability collapse."

1) Constrained Feature Learning With Adapters: When the training loss is a D-loss, we can implement constrained feature learning with the architectures shown in Fig. 1. Specifically, from our discussions in Section IV-B.4, we can effectively train the adapters ϕ, ψ to learn features restricted to p-norm balls or non-negative features f, g.

2) Inference-time Hyperparameter Tuning: Suppose we have a class of D-losses $L_\lambda(f, g; P_{X,Y})$, where $\lambda \in \mathcal{I}$ is the hyperparameter, e.g., the weights for regularization terms. The optimal feature functions that minimizes L_λ depend on λ, which we denote by $(f^{(\lambda)}, g^{(\lambda)})$. The values of λ are often determined by comparing performances of different λ, which requires learning $(f^{(\lambda)}, g^{(\lambda)})$ for a set of different values.

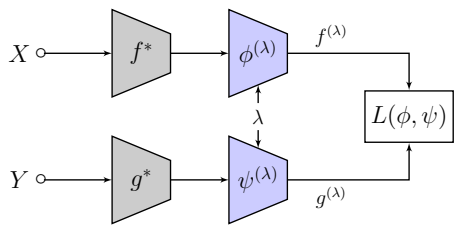

Fig. 2. The feature adapters $\phi^{(\lambda)}, \psi^{(\lambda)}$ are parameterized by the hyperparameter λ, which is tunable during inference.

From Theorem 3, we have $(f^{(\lambda)}, g^{(\lambda)}) = (\phi^{(\lambda)} \circ f^*, \psi^{(\lambda)} \circ g^*)$. Note that only the adapters $(\phi^{(\lambda)}, \psi^{(\lambda)})$ depend on the λ, and can be learned by minimizing $L_\lambda(\phi^{(\lambda)}, \psi^{(\lambda)}; P_{S,T})$. Therefore, it suffices to learn feature adapters parameterized by λ. Specifically, we define the loss $L(\phi, \psi; P_{S,T}) \triangleq \mathbb{E}_{\lambda \sim P_\lambda}\left[L_\lambda(\phi^{(\lambda)}, \psi^{(\lambda)}; P_{S,T})\right]$, where P_λ is a distribution supported on $\mathcal{I}$. Then, minimizing $L(\phi, \psi; P_{S,T})$ leads to parameterized $\phi^{(\lambda)}, \psi^{(\lambda)}$, which give parameterized representations $f^{(\lambda)}(X), g^{(\lambda)}(Y)$, as shown in Fig. 2.

In contrast to the common practice, this design allows us to tune the hyperparameter λ over a continuous range, without retraining or requiring validation samples. This can be useful in learning scenarios where we have many different downstream tasks but few validation samples.

B. Learning Maximal Correlation Functions

The maximal correlation functions can be effectively learned from data via maximizing the nested H-score [3]. Specifically, for $f \in \mathcal{F}_{\mathcal{X}}^k$, $g \in \mathcal{F}_{\mathcal{Y}}^k$, we define the nested H-score $\mathscr{H}_{\mathsf{nest}}(f, g)$ as[7]

$$\mathscr{H}_{\mathsf{nest}}(f, g) \triangleq \sum_{i=1}^{k} \mathscr{H}(f_{[i]}, g_{[i]}), \tag{21}$$

where $f_{[i]}, g_{[i]}$ are as defined in (19), and where $\mathscr{H}(f, g)$ is the H-score, defined as

$$\mathscr{H}(f, g) \triangleq \mathbb{E}\left[f^{\mathrm{T}}(X) g(Y)\right] - \left(\mathbb{E}\left[f(X)\right]\right)^{\mathrm{T}} \mathbb{E}\left[g(Y)\right] - \frac{1}{2} \cdot \operatorname{tr}\left(\Lambda_f \Lambda_g\right), \tag{22}$$

where $\Lambda_f = \mathbb{E}\left[f(X) f^{\mathrm{T}}(X)\right]$, $\Lambda_g = \mathbb{E}\left[g(Y) g^{\mathrm{T}}(Y)\right]$. It can be verified that $-\mathscr{H}_{\mathsf{nest}}(f, g)$ is a regular D-loss.

[7]The form of the nested H-score is nonunique; for example, the weights before each H-score can be arbitrary positive numbers. See [3, Section 4.1] for detailed discussions.

Note that since $f^*(X), g^*(Y)$ are minimal sufficient statistics, it is possible to learn $f^*(X), g^*(Y)$ by adapting existing sufficient statistics, which can significantly reduce the computation costs. Such sufficient statistics can be obtained from pretrained models, e.g., deep classifiers. In engineering applications, we can also obtain analytical forms of sufficient statistics from corresponding physical system models.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] L. Breiman, "Statistical modeling: The two cultures (with comments and a rejoinder by the author)," *Statistical science*, vol. 16, no. 3, pp. 199–231, 2001.

[3] X. Xu and L. Zheng, "Neural feature learning in function space," *Journal of Machine Learning Research*, vol. 25, no. 142, pp. 1–76, 2024.

[4] M. Huh, B. Cheung, T. Wang, and P. Isola, "Position: The Platonic Representation Hypothesis," in *Forty-first International Conference on Machine Learning*.

[5] V. Papyan, X. Han, and D. L. Donoho, "Prevalence of neural collapse during the terminal phase of deep learning training," *Proceedings of the National Academy of Sciences*, vol. 117, no. 40, pp. 24 652–24 663, 2020.

[6] C. Shannon, "The lattice theory of information," *Transactions of the IRE professional Group on Information Theory*, vol. 1, no. 1, pp. 105–107, 1953.

[7] H. A. Simon, "On the forms of mental representation," in *Perception and Cognition*, W. Savage, Ed. University of Minnesota Press, 1978, pp. 3–18.

[8] S.-L. Huang, A. Makur, G. W. Wornell, and L. Zheng, "Universal features for high-dimensional learning and inference," *Foundations and Trends® in Communications and Information Theory*, vol. 21, no. 1-2, pp. 1–299, 2024. [Online]. Available: http://dx.doi.org/10.1561/0100000107

[9] H. O. Hirschfeld, "A connection between correlation and contingency," in *Proceedings of the Cambridge Philosophical Society*, vol. 31, no. 4, 1935, pp. 520–524.

[10] H. Gebelein, "Das statistische problem der korrelation als variations- und eigenwertproblem und sein zusammenhang mit der ausgleichsrechnung," *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 21, no. 6, pp. 364–379, 1941.

[11] A. Rényi, "On measures of dependence," *Acta Mathematica Academiae Scientiarum Hungarica*, vol. 10, no. 3-4, pp. 441–451, 1959.

[12] T. M. Cover and J. A. Thomas, "Elements of information theory (wiley series in telecommunications and signal processing)," 2006.

[13] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[14] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[15] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[16] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Estimating divergence functionals and the likelihood ratio by convex risk minimization," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5847–5861, 2010.

[17] A. Kusupati, G. Bhatt, A. Rege, M. Wallingford, A. Sinha, V. Ramanujan, W. Howard-Snyder, K. Chen, S. Kakade, P. Jain *et al.*, "Matryoshka representation learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 30 233–30 249, 2022.

[18] D. G. Mixon, H. Parshall, and J. Pi, "Neural collapse with unconstrained features," *Sampling Theory, Signal Processing, and Data Analysis*, vol. 20, no. 2, p. 11, 2022.

[19] Z. Zhu, T. Ding, J. Zhou, X. Li, C. You, J. Sulam, and Q. Qu, "A geometric analysis of neural collapse with unconstrained features," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 820–29 834, 2021.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.