Deterministic Decoupling of Global Features and its Application to Data Analysis

Eduardo Martínez-Enríquez, María del Mar González, and Javier Portilla

Abstract—We introduce a method for deterministic decoupling of global features and show its applicability to improve data analysis performance, as well as to open new venues for feature transfer. We propose a new formalism that is based on defining transformations on submanifolds, by following trajectories along the features' gradients. Through these transformations we define a *normalization* that, we demonstrate, allows for decoupling differentiable features. By applying this to sampling moments, we obtain a quasi-analytic solution for the *orthokurtosis*, a normalized version of the kurtosis that is not just decoupled from mean and variance, but also from skewness. We apply this method in the original data domain and at the output of a filter bank to regression and classification problems based on global descriptors, obtaining a consistent and significant improvement in performance as compared to using classical (non-decoupled) descriptors.

Index Terms—feature-based data analysis, feature redundancy, feature decoupling, nested normalization, feature transfer, local de-correlation, orthokurtosis, regression, classification.

1 Introduction

Data analysis relies on the statistical distribution of the observed samples. Usually, some *features* (i.e., real functions) are applied to extract relevant information from the observed data. In the Machine Learning field there are two basic scenarios for data analysis: (i) the *classical* one, where the set of features is chosen ad-hoc; (ii) the Deep Learning scenario, which involves automatically learning the features from the input data. In all cases, when features are used, *the observed dependencies among them come both from the statistical behavior of the data and from the features' joint algebraic structure*.

To illustrate this problem imagine that we analyze vectors representing 1-D signals, extracting some marginal sample moments and the sample auto-correlation. We will find a strong dependency between skewness and kurtosis, and also between consecutive correlation factors, e.g., $\rho(1)$ and $\rho(2)$, *even when the input data are i.i.d. samples*. The reason is that the two mentioned pairs of features, like many others, are *algebraically coupled*. As a consequence, their joint range is not just the outer product of their marginal ranges: some combinations of (independently) valid feature values are incompatible for the same input data. E.g., a skewness of 10 and a kurtosis of 3, or $\rho(1) = 0.9$ and $\rho(2) = 0$.

Feature coupling, thus, produces spurious redundancy, a sort of *feature entanglement*, regardless of (and in addition to) the redundancy derived from input data statistics. It causes difficulties for analysis, processing, and simulation. Having a joint feature range with a very intricate topology (full of "holes" and complex boundaries' structure) is an obstacle to interpreting the role of each feature separately from the others. It also complicates the manipulation of the samples, in case we wanted to average feature values [1], study the effect of modifying the value of a particular feature independently of the others, or simulating data by imposing some values onto their features [2]. An interesting example of these problems appeared in [3], which addressed the problem of simplifying the set of features used in [1] to visually describe texture samples.

Despite its large negative impact on data analysis and processing, much less effort has been devoted in the literature to study and reverse deterministic feature coupling compared to statistical data modeling. Note that, when algebraic coupling between features exists (as in the examples mentioned above), conventional techniques such as PCA, ICA, or even more advanced non-linear ICA (see, e.g., [4]), do not provide the right tools for disentangling the involved joint feature vector structures. Even in the ideal scenario of perfectly modeling all dependencies, a purely statistical approach applied to the observed features would mix up the two sources of redundancy, namely, statistical and algebraic. It is advantageous to address separately these two redundancy sources, as it is usual to apply the same type of features for diverse statistical distributions (even in ANNs, when doing *transfer learning* [5]). Therefore, many different real problems on different data distributions using the same features will benefit from their decoupling.

Here we propose a mathematical and algorithmic framework for decoupling a set of given features, in the sense of finding another set of similar functions with their gradients mutually orthogonal everywhere in the domain - and, as a consequence, with their ranges decoupled. We study the mathematical conditions under which that is possible. We also study less favorable scenarios where only a limited and/or approximated decoupling is possible. We demon-

- *E. Martínez-Enríquez and J. Portilla are with the Instituto de Óptica, CSIC, Spain.*
  E-mails: eduardo.martinez@io.cfmac.csic.es, javier.portilla@csic.es
- *M. González is with Dept. of Mathematics, Universidad Autónoma de Madrid and ICMAT, Spain.*
  E-mail: mariamar.gonzalezn@uam.es

This work has been funded by the Spanish Government grants FIS2016-75891-P, PID2020-118071GB- I00 and PID2020-113596GB-I00. Additionally, Grant RED2018-102650-T funded by MCIN/AEI/ 10.13039/501100011033, and the "Severo Ochoa Programme for Centers of Excellence in R&D" (CEX2019-000904-S).

strate the practical application of the proposed method to several examples of data analysis, namely, statistical regression and textured image classification. Some of the seminal ideas and applied results presented has been published in three conference proceedings [2], [6], [7]. In this work we provide a solid framework, unifying, extending, and giving the necessary mathematical rigor to our previous results.

As concrete study cases, here we have focused on marginal moments and on the second-order moments at the output of a set of filters. Marginal moments are widely used in the signal processing and statistics literature, for analysis tasks such as estimation (e.g., the method of moments), detection, regression, classification, etc. [8], [9], [10], [11], [12], [13], and also for synthesis-by-analysis [1], [14], [15]. Typically they are used either implicitly, as empirical marginal histograms, or in their standardized form, and up to fourth order: sample mean, variance, skewness, and kurtosis. Whereas, as shown here, the first three standardized moments are already mutually decoupled, that is not the case for the skewness and kurtosis. The problem of the skewness-kurtosis coupling has been pointed out by several authors [16], [17], [18], but it had not been fully solved. In this respect, one of the main contributions of this paper is presenting a normalized version of the fourth-order sample moment, the *orthokurtosis*, which is not just decoupled from the sample mean and variance, but also from the skewness. This new statistical function is fully consistent with the previous standardized sample moments (mean, variance and skewness), that result from applying our decoupling technique to the first three raw moments. In addition, the orthokurtosis calculation has a modest computational cost. By using this new fourth-order feature, instead of the classical kurtosis, we obtain a dramatic accuracy gain in several regression problems (see Subsection 6.3). Furthermore, higher-than-four order moments have been very rarely used (see exceptions in, e.g., [19], [20]) because of their instability and mutual redundancy. By decoupling the marginal moments (exactly or approximately) here we are able to exploit very high order decoupled moments (up to 10th order) and demonstrate their positive impact for texture classification (see Subsection 6.4.1).

Banks of convolutional filters, on the other hand, are a classical tool in signal processing, with a huge field of application, including early human vision modelling and image/audio analysis, processing and synthesis. In addition, they have also been incorporated [21], [22] into artificial neural networks (ANNs) for signals having spatial dimensions (image, video, 3-D, etc.) with tremendous impact. In neural science, they have long been used to model the responses at early stages of animal and human visual and auditory systems [23], [24], [25], [26]. The latter image/audio representations share the feature of being redundant, thus avoiding the artifacts that plague critically-sampled linear transformations (e.g., orthogonal or bi-orthogonal wavelets [27]). However, redundancy in non-orthogonal linear representations demands paying a high price, namely, the algebraic coupling of undecimated sub-bands (outputs of the filters). Here we address the problem of deterministically decoupling the second-order moments at the output of a filter bank, with direct application, besides analysis, to transfer [28] and synthesis [2]. In Section 6 we show how the gradients of the resulting *decoupled* features are virtually orthogonal for white noise samples and very close to orthogonal for photographic textured image patches. Furthermore, we demonstrate how approximately decoupling not just variance, but also higher-order moments, at the output of a bank of filters, results in an important performance boost in texture classification (Subsection 6.4.2).

This paper is organized as follows. Section 2 sets the mathematical foundations of the method, that allow, in favorable cases, to transform a given feature by decoupling it from a set of other given features. Section 3 proposes algorithms (based on the Nested Normalization concept, NeN) to obtain a hierarchically ordered set of mutually decoupled features and to transfer features from one observed sample to another. Section 4 addresses in detail two study cases of features for being decoupled, namely marginal moments, and the second-order moments at the output of a filter bank. Section 5 addresses analytically the local de-correlation effect of feature decoupling, and why this improves parameter discrimination, in regression problems. Section 6 is devoted to showing how the proposed method actually decouples the studied features, and its practical impact when it is applied to data analysis (regression and classification). Section 7 concludes the paper. In addition, some technical and/or very detailed contents have been encapsulated in appendices, for readability and reproducibility sake.

2 Deterministic Decoupling of Global Features

In this section we propose a method for, given a set of features $\mathcal{S}$ and another unrelated feature g, finding a transformed feature $\hat{g}$, identical to g on a high dimensional submanifold, that is *decoupled* to every feature in $\mathcal{S}$.

2.1 Preliminary Concepts

2.1.1 Global Shift-Invariant Features

In this paper we associate a finite discrete signal made of N samples with a vector $\mathbf{x} \in \mathbb{R}^N$, possibly lexico-graphically reordered, if the signal support is a multi-dimensional array. We will term a *feature* f of that vector $\mathbf{x}$ a differentiable real function $f : \overline{\Omega} \subset \mathbb{R}^N \rightarrow \mathbb{R}$ for a domain $\overline{\Omega}$. We define here *global feature* a feature that depends on all vector's coefficients.

In this paper we will focus on *shift-invariant* features, a special case of global features[1]. Within them, we will exemplify the application of our method to features of the form:

$$f(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} [\mathbf{m}(\mathbf{x})]_n, \tag{1}$$

with $\mathbf{m} : \overline{\Omega} \rightarrow \mathbb{R}^N$ being a differentiable shift-equivariant (i.e., commuting with shift operations), or shift-invariant, mapping, assuming a shift operation with boundary conditions (e.g., circular) has been defined. This kind of functions, being averages, play the role of sample statistics, like marginal moments, correlation coefficients, moments at the output of filters, etc.

1. The only non-global shift-invariant features are the trivial functions $f(\mathbf{x}) = c$, where c is a real constant.

2.1.2 Decoupled Features

We say that two features f_i and f_j are *algebraically decoupled* (from now on just *decoupled*) on a subset of $\overline{\Omega}$ iff

$$\nabla f_i(\mathbf{x}) \cdot \nabla f_j(\mathbf{x}) = 0, \quad \text{for all } \mathbf{x} \text{ in that subset.} \tag{2}$$

We extend this concept to a set of features $\mathcal{S} = \{f_j, j = 1 \dots M\}$ by terming that the features of a set are decoupled, iff they are mutually decoupled, i.e., iff all possible pairs of features within that set $\{(i,j) : i,j \in \{1..M\}, i \neq j\}$ are decoupled. Similarly, we say that a feature is decoupled to a (decoupled or not) set of features iff it is decoupled to each of the features in that set.

It is worth pointing out two special cases, namely, when features are trivially decoupled and when they are trivially coupled. We term *trivially decoupled* features those for which there exists at least one orthogonal basis where they have disjoint supports[2] (e.g. Fourier, orthogonal wavelets, etc.). Here we refer to support, in a given domain, as the subset of vector indices the feature depends on. On the other extreme, a feature map $\mathbf{f}$ is *trivially coupled* iff it exists at least one non-degenerate function $F : \mathbb{R}^M \to \mathbb{R}$ such that $F(\mathbf{f}(\mathbf{x})) = 0, \forall \mathbf{x} \in \overline{\Omega}$. In this paper, we present methods for decoupling features assuming none of those situations happens (for which decoupling is either unnecessary or impossible, respectively).

2.1.3 Normalization map

The construction of a normalization will be key in our decoupling process.

Let $\mathcal{S} = \{f_j : \overline{\Omega} \to \mathbb{R}, j = 1 \dots M\}$ be a set of features, Ω a subset of $\overline{\Omega}$, $\hat{\mathbf{x}}_{\mathcal{S}}(\mathbf{x}) : \Omega \to \Omega$ be a continuous non-constant mapping, and $\mathbf{v}^{ref}$ a vector made of $\{v_j^{ref}, j = 1 \dots M\}$ (the *reference values*), some jointly compatible given reference values for the features in $\mathcal{S}$. We say that $\hat{\mathbf{x}}_{\mathcal{S}}(\mathbf{x}; \mathbf{v}^{ref})$ is a *normalization w.r.t.* $\mathcal{S}$ and $\mathbf{v}^{ref}$ in Ω iff it holds that

(i) $\{f_j(\hat{\mathbf{x}}_{\mathcal{S}}(\mathbf{x}; \mathbf{v}^{ref})) = v_j^{ref}\}_{j=1}^M$;
(ii) if $\{f_j(\mathbf{x}) = v_j^{ref}\}_{j=1}^M$ then $\hat{\mathbf{x}}_{\mathcal{S}}(\mathbf{x}; \mathbf{v}^{ref}) = \mathbf{x}$.

Note that previous conditions imply that every normalization is idempotent:

$$\hat{\mathbf{x}}_{\mathcal{S}}(\hat{\mathbf{x}}_{\mathcal{S}}(\mathbf{x}; \mathbf{v}^{ref}); \mathbf{v}^{ref}) = \hat{\mathbf{x}}_{\mathcal{S}}(\mathbf{x}; \mathbf{v}^{ref}). \tag{3}$$

We now set up some notation for this set of reference values which will be useful in our exposition. Let $\mathbf{f}_{\mathcal{S}} : \overline{\Omega} \to \mathbb{R}^M$ be the vector transformation made of the ordered features in $\mathcal{S}$, $[f_1(\mathbf{x}) \dots f_M(\mathbf{x})]$, and set $\mathbf{v}^{ref}$ to be an M-dimensional vector. We define a *reference manifold* as $\mathcal{R}_{\mathcal{S}}(\mathbf{v}^{ref}) = \mathbf{f}_{\mathcal{S}}^{-1}(\mathbf{v}^{ref})$, i.e., the set of vectors $\mathbf{x}$ such that $\mathbf{f}_{\mathcal{S}}(\mathbf{x}) = \mathbf{v}^{ref}$.

For being a valid set of reference values for the features, $\mathbf{v}^{ref}$ must be made of jointly compatible values of the functions in an algebraic sense, i.e., $\{\mathbf{x} : \mathbf{f}_{\mathcal{S}}(\mathbf{x}) = \mathbf{v}^{ref}\} \neq \emptyset$. In addition, we will assume a *non-degeneracy* hypothesis, under which all feature gradients are linearly independent at every point (this will be condition C1 in Subsection 2.3.1). This is a stronger condition than the set of features not being trivially coupled, and it implies that the dimension of the reference manifold is *everywhere* $N - M$.

2. Note that they can not have disjoint supports in the original domain if they are both global.

2.2 Motivating example: decoupling two features

In order to motivate the general algorithm, let us explain the method in the case of two features.

2.2.1 Gradient systems

Let us fix one feature f, defined in a connected open set $\overline{\Omega}$. We study the trajectories $\mathbf{x}(t)$ of the initial value problem

$$\begin{cases} \dfrac{d\mathbf{x}}{dt} = -\nabla f(\mathbf{x}), \\ \mathbf{x}(0) = \mathbf{x}_0. \end{cases} \tag{4}$$

This ODE is known as a gradient system. It is clear that moving along a (non-constant) trajectory in the $+t$ (resp. $-t$) direction will strictly decrease (resp. increase) the value of the function f until it reaches the minimum (resp. maximum) value or stabilize at a critical point of f (see the reference [29, Section 9.3] for a discussion of this type of systems).

Thus, in order to study the set of values that f may take along a trajectory, one needs to impose some constraints on its equilibrium points. The precise set of conditions are given in Appendix A, and can be summarized in:

B1. Maxima and minima are all global extrema, not just local.
B2. The set made of the basins of attraction of all saddle points, denoted by Λ, is of lower dimension.

The first condition ensures that the trajectory will not stop at a local non-global extreme. The second condition guarantees that saddle points are non-degenerate and thus, essentially unstable, so it allows to circumvent them by adding small perturbations on the initial condition $\mathbf{x}_0$ (see Section 3.3.2).

We denote the trajectory that passes through the point $\mathbf{x}_0$, or equivalently, the integral manifold of the system (4), by $\mathcal{I}(\mathbf{x}_0, f)$. The main property we will need is that all possible f values are reachable from any initial point $\mathbf{x}_0$ by moving along the gradient. As a consequence, fixed a reference value v_f^{ref} for f, conditions B1-B2 guarantee that, for $\mathbf{x} \notin \Lambda$, each trajectory $\mathcal{I}(\mathbf{x}, f)$ reaches (only once) this value. See Appendix A for a technical discussion.

2.2.2 Decoupling via normalization

Assume that we are given two features $\{f, g\}$. We would like to replace g by a "similar" feature $\hat{g}$ that is decoupled from f in the sense given by (2), via normalization.

More precisely, fixed one feature f, we would like to construct a normalization map $\hat{\mathbf{x}}_f(\mathbf{x})$ as defined in Section 2.1.3. A possibility is to choose a point in the trajectory $\mathcal{I}(\mathbf{x}, f)$ that attains some reference value v^{ref} of the feature f. Thus it is natural to define the normalization by $\hat{\mathbf{x}}_f(\mathbf{x}; v^{ref})$, this is, as the map that sends a point $\mathbf{x}$ to the point where the trajectory $\mathcal{I}(\mathbf{x}, f)$ crosses the reference manifold $\mathcal{R}_f(v^{ref})$, which is unique by our previous discussion on gradient systems.

Now, given another feature g, we define $\hat{g}$ by

$$\hat{g}(\mathbf{x}) = g(\hat{\mathbf{x}}_f(\mathbf{x})). \tag{5}$$

Then f and $\hat{g}$ are decoupled, this is, their gradients are orthogonal. To show this, apply the chain rule to Eq. (5), to obtain $\nabla \hat{g}_f(\mathbf{x}) = \mathbf{J}_{\hat{\mathbf{x}}_f}^T(\mathbf{x}) \nabla g(\hat{\mathbf{x}}_f(\mathbf{x}))$, where $\mathbf{J}_{\hat{\mathbf{x}}_f}$ represents

the Jacobian matrix of the map $\hat{\mathbf{x}}_f$. Now, by pre-multiplying both terms by $\nabla f(\mathbf{x})^T$, we obtain:

$$\nabla f(\mathbf{x}) \cdot \nabla \hat{g}(\mathbf{x}) = (\mathbf{J}_{\hat{\mathbf{x}}_f}(\mathbf{x}) \nabla f(\mathbf{x}))^T \nabla g(\hat{\mathbf{x}}_f(\mathbf{x})). \quad (6)$$

Finally, note that

$$\mathbf{J}_{\hat{\mathbf{x}}_f}(\mathbf{x}) \nabla f(\mathbf{x}) = \frac{d}{dt}\Big|_{t=0} \hat{\mathbf{x}}_f(\alpha(t)),$$

where $\alpha(t)$ is the integral curve of (4) starting at the point $\mathbf{x}$. By construction, the map $\hat{\mathbf{x}}_f$ is constant along this curve and, thus, the above expression vanishes. This shows that the two gradients in expression (6) are orthogonal, as desired. Figure 1 graphically illustrates these concepts for two given features f_1, f_2.

2.3 Decoupling features from a given set: multi-feature normalization

Now we consider the problem of decoupling a feature g from a given set $\mathcal{S} = \{f_i : \overline{\Omega} \subset \mathbb{R}^N \to \mathbb{R}, i = 1 \dots M\}$ of M features. The method follows the normalization scheme explained in the two-feature case, by following trajectories given by the gradients of all the f_i. Unlike the simple case of gradient systems, in order to build integral manifolds from multiple feature gradients, additional conditions must be fulfilled. Indeed, we will give a necessary and sufficient condition for this method to apply (Proposition 2.2 below).

It is also clear that, in order to construct a normalization, we need to restrict to a subdomain Ω, obtained from $\overline{\Omega}$ by removing its critical points corresponding to the set of given features $\mathcal{S} = \{f_j, j = 1...M\}$.

2.3.1 Invariant mapping with respect to a set of features

We first introduce the notion of a mapping being invariant with respect to a set of features $\mathcal{S} = \{f_i : \overline{\Omega} \subset \mathbb{R}^N \to \mathbb{R}, i = 1 \dots M\}$, a concept that will greatly facilitate the decoupling of an arbitrary feature g from this set.

We say that a non-constant, differentiable mapping $\mathbf{y}_\mathcal{S} : \Omega \to \Omega$ is *invariant* w.r.t. $\mathcal{S}$ iff

$$\mathbf{J}_{\mathbf{y}_\mathcal{S}}(\mathbf{x}) \nabla f_i(\mathbf{x}) = \mathbf{0}, \forall f_i \in \mathcal{S}, \forall \mathbf{x} \in \Omega, \quad (7)$$

where $\mathbf{J}_{\mathbf{y}_\mathcal{S}}$ represents the Jacobian matrix of $\mathbf{y}_\mathcal{S}$.

Proposition 2.1. Obtaining decoupled features from invariant mappings. *Let g be an arbitrary feature $g : \overline{\Omega} \to \mathbb{R}$, and $\mathbf{y}_\mathcal{S}$ an invariant mapping w.r.t. a set of features $\mathcal{S}$. From them we construct a new feature:*

$$\hat{g}_\mathcal{S}(\mathbf{x}) = g(\mathbf{y}_\mathcal{S}(\mathbf{x})). \quad (8)$$

Then it holds that $\nabla \hat{g}_\mathcal{S}(\mathbf{x}) \cdot \nabla f_i(\mathbf{x}) = 0$, $\forall f_i \in \mathcal{S}, \forall \mathbf{x} \in \Omega$, i.e., the new feature $\hat{g}_\mathcal{S}$ is decoupled from all features in $\mathcal{S}$.

Proof. Mimicking the calculation in Eq. (6), we have

$$\begin{aligned} \nabla f_i(\mathbf{x}) \cdot \nabla \hat{g}_\mathcal{S}(\mathbf{x}) &= (\mathbf{J}_{\mathbf{y}_\mathcal{S}}(\mathbf{x}) \nabla f_i(\mathbf{x}))^T \nabla g(\mathbf{y}_\mathcal{S}(\mathbf{x})) \\ &= 0, \forall f_i \in \mathcal{S}, \forall \mathbf{x} \in \Omega \end{aligned} \quad (9)$$

where the last equality holds because of Eq. (7). □

Now that the significance of having an invariant mapping has been established, let us consider the problem of existence. We will need to assume that:

C1. The gradients $\{\nabla f_i(\mathbf{x})\}$ are linearly independent at every point $\mathbf{x}$; and

C2. they satisfy the Frobenius condition, which is an integrability condition for several gradients ∇f_i. The related technicalities are addressed in Appendix B.

We will always assume condition C1 in order to avoid redundancy in the set of features $\mathcal{S}$, even if not explicitly stated.

Now we show that the Frobenius condition C2 is necessary and sufficient for an invariant mapping to apply. This gives a criterion for the possibility of exact decoupling.

Proposition 2.2. *Given $\mathcal{S}$ as above, there exists an invariant mapping $\mathbf{y}_\mathcal{S}$ w.r.t. $\mathcal{S}$ iff the gradients $\{\nabla f_i, i = 1 \dots M\}$ satisfy the Frobenius condition C2 at each point.*

This proof will be given in two steps. The *only if* part will be postponed to the Appendix (Section B.1) because it is rather technical and not relevant to our study. Here we will concentrate in the *if* statement, which will be treated in Subsection 2.3.3 below. Our proof is explicit, giving a precise construction of the invariant map via normalization, which is the crucial ingredient.

2.3.2 Invariance submanifolds

Given a set of features $\mathcal{S} = \{f_i : \overline{\Omega} \to \mathbb{R}, i = 1 \dots M\}$, the *invariance submanifold* $\mathcal{I}(\mathbf{x}_0, \mathcal{S})$ is an M-dimensional submanifold passing through $\mathbf{x}_0$ whose tangent planes at each point are spanned by the gradients $\{\nabla f_i, i = 1 \dots M\}$.

To ensure that the invariance submanifold exists we need to assume conditions C1 and C2 above, as it is explained in Appendix B. Indeed, Frobenius condition C2 is a compatibility condition on the second derivatives of different f_i which is needed for multi-feature integrability. Moreover, it is is vacuous if $M = 1$, as we only integrate along the gradient of a single feature, see Eq. (4).

In addition, Frobenius theorem states that Ω is foliated by invariance submanifolds.

2.3.3 Normalization of several features

Here we give the construction of a normalization of multiple features generalizing the approach in Section 2.2 for the decoupling of two features, and show that this normalization indeed yields an invariant mapping.

For this, we associate a single vector $\mathbf{y}_\mathcal{S}(\mathbf{x})$ to each invariance submanifold $\mathcal{I}(\mathbf{x}, \mathcal{S})$, this is,

$$\mathbf{y}_\mathcal{S}(\mathbf{z}) = \mathbf{y}_\mathcal{S}(\mathbf{x}_0), \ \forall \mathbf{z} \in \mathcal{I}(\mathbf{x}_0, \mathcal{S}), \quad (10)$$

and ensure that such mapping $\mathbf{y}_\mathcal{S}(\mathbf{x})$ is continuous and differentiable. The remaining question is, then, how to choose a representative $\mathbf{y}_\mathcal{S}(\mathbf{x}_0)$ of each invariance submanifold $\mathcal{I}(\mathbf{x}_0, \mathcal{S})$. In the case of a normalization, we choose the vector belonging to $\mathcal{I}(\mathbf{x}_0, \mathcal{S})$ that attains some reference values $\mathbf{v}_\mathcal{S}^{ref}$ in its features, values that we know $\mathcal{I}(\mathbf{x}_0, \mathcal{S})$ may take, for all $\mathbf{x}_0 \in \Omega$.

The following proposition states that, after removing a lower-dimensional subset Λ from Ω, we can attain a valid set of reference values by moving along the invariance submanifold:

Proposition 2.3. *Let $\mathbf{v}^{ref}$ be any jointly compatible set of values of $\mathbf{f}_\mathcal{S}$. Then for all $\mathbf{x}_0 \in \Omega \setminus \Lambda$, there exists $\mathbf{z} \in \mathcal{I}(\mathbf{x}_0, \mathcal{S})$*

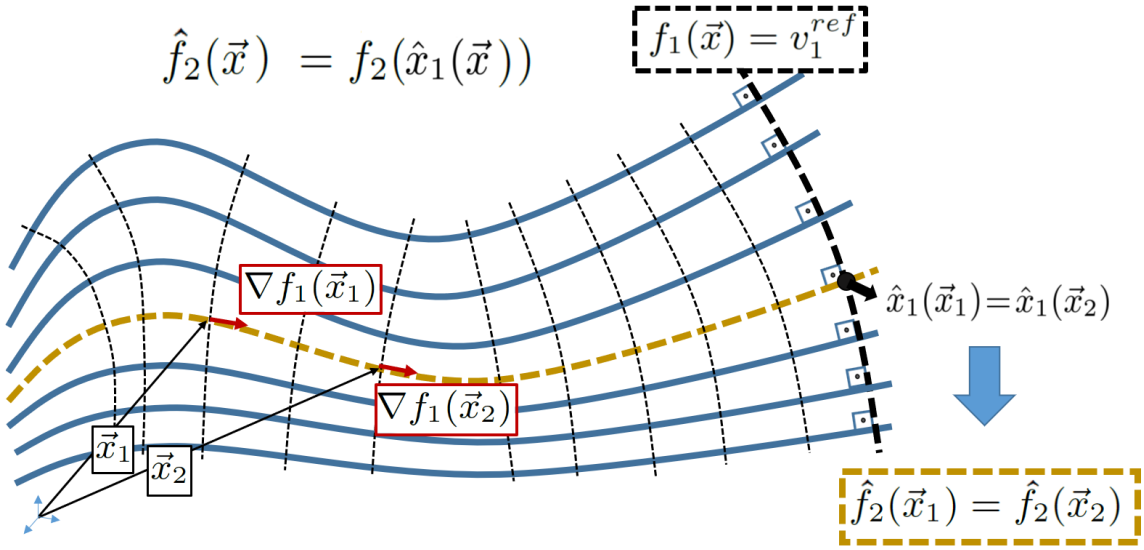


Fig. 1: Decoupling two features through normalization. The proposed normalization consists in finding the intersection of the invariance submanifolds passing by $\mathbf{x}$ (parallel curved thick lines) with the reference manifold $\mathcal{R}_1 = \{\mathbf{x} : f_1(\mathbf{x}) = v_1^{ref}\}$ (black dashed thick line). All vectors belonging to the same invariance submanifold (e.g., the mustard dashed curve), like $\mathbf{x}_1$ and $\mathbf{x}_2$, have the same normalized vector $\hat{x}_1$ and, thus, the same $\hat{f}_2$ value. Therefore, $\nabla \hat{f}_2$ must be locally orthogonal everywhere to the invariance submanifolds (iso-level sets of $\hat{f}_2$), and, as a consequence, also to ∇f_1.

that satisfies $\mathbf{f}_{\mathcal{S}}(\mathbf{z}) = \mathbf{v}^{ref}$*, and it is unique in the connected component of* $\Omega \setminus \Lambda$ *where* $\mathbf{x}$ *belongs to. Thus, the solution set* $\mathcal{I}(\mathbf{x}_0, \mathcal{S}) \bigcap \mathcal{R}_{\mathcal{S}}(\mathbf{v}^{ref})$ *contains exactly one point in this connected component.*

Proof. By our assumptions on the critical points, the basin of attraction of critical points that are not global maxima or minima is lower dimensional. Note that the gradient flow of each f_i starting at $\mathbf{x}_0$ is fully contained in $\mathcal{I}(\mathbf{x}_0, \mathcal{S})$, and thus, the f_i's take all possible values along the flow unless $\mathbf{x}_0$ belongs to the basin of attraction of a saddle.

Next, there cannot be more than one point in $\mathcal{I}(\mathbf{x}_0, \mathcal{S})$ with exactly the same reference values since, under our assumptions on critical points, the flow of each gradient always strictly decreases (resp. increases) the value of the corresponding f_i. □

Thanks to the previous Proposition, for $\mathbf{x} \in \Omega \setminus \Lambda$ we can define the normalization

$$\hat{\mathbf{x}}_{\mathcal{S}}(\mathbf{x}; \mathbf{v}_{\mathcal{S}}^{ref}) = \mathcal{I}(\mathbf{x}, \mathcal{S}) \bigcap \mathcal{R}_{\mathcal{S}}(\mathbf{v}^{ref}). \tag{11}$$

Proposition 2.4. *The normalization* $\hat{\mathbf{x}}_{\mathcal{S}}$ *constructed in* (11) *has the following properties:*

i. $\hat{\mathbf{x}}_{\mathcal{S}}$ *is an invariant mapping.*
ii. *The Jacobian* $\mathbf{J}_{\hat{\mathbf{x}}_{\mathcal{S}}}$*, when evaluated on the reference manifold* $\mathcal{R}_{\mathcal{S}}$*, is an orthogonal projection map on* $\mathcal{R}_{\mathcal{S}}$*. Moreover, it is non-degenerate, i.e.,* $rank(\mathbf{J}_{\hat{\mathbf{x}}_{\mathcal{S}}}(\mathbf{x})) = N - M$ $\forall \mathbf{x} \in \Omega$.
iii. *The pair* $\left(\mathbf{f}_{\mathcal{S}}(\mathbf{x}), \hat{\mathbf{x}}_{\mathcal{S}}(\mathbf{x}; \mathbf{v}^{ref})\right)$ *carries the same information as* $\mathbf{x}$*. In particular,* $\mathbf{x}$ *can always be recovered from it by reversing the normalization, i.e.,* $\mathbf{x} = \hat{\mathbf{x}}_{\mathcal{S}}(\hat{\mathbf{x}}_{\mathcal{S}}(\mathbf{x}; \mathbf{v}^{ref}); \mathbf{f}_{\mathcal{S}}(\mathbf{x}))$, $\mathbf{x} \in \Omega \setminus \Lambda$.

Proof. The fact that $\hat{\mathbf{x}}_{\mathcal{S}}$ is an invariant mapping is obvious from the construction. Indeed, by definition of Jacobian matrix,

$$\mathbf{J}_{\hat{\mathbf{x}}_{\mathcal{S}}}(\mathbf{x}) \nabla f_i(\mathbf{x}) = \frac{d}{dt}\Big|_{t=0} \hat{\mathbf{x}}_{\mathcal{S}}(\alpha_i(t))$$

for a curve $\alpha_i(t)$ satisfying $\alpha_i(0) = \mathbf{x}$ and $\alpha_i'(0) = \nabla f_i(\mathbf{x})$. Since this curve can be taken fully contained inside the invariant submanifold $\mathcal{I}(\mathbf{x}, \mathcal{S})$ (following, for instance, the gradient flow of f_i), then $\hat{\mathbf{x}}_{\mathcal{S}}(\alpha_i(t))$ is a constant function in t and thus, its derivative vanishes.

For the second statement note first that, by applying the chain rule in the condition of Eq. (3) it immediately yields that

$$\mathbf{J}_{\hat{\mathbf{x}}_{\mathcal{S}}}(\hat{\mathbf{x}}_{\mathcal{S}}(\mathbf{x})) \mathbf{J}_{\hat{\mathbf{x}}_{\mathcal{S}}}(\mathbf{x}) = \mathbf{J}_{\hat{\mathbf{x}}_{\mathcal{S}}}(\mathbf{x}), \ \forall \mathbf{x} \in \Omega.$$

Now, recall that for $\mathbf{x}$ in the reference manifold $\mathcal{R}_{\mathcal{S}}$ we have $\hat{\mathbf{x}}_{\mathcal{S}}(\mathbf{x}) = \mathbf{x}$, so the previous equation reduces to

$$(\mathbf{J}_{\hat{\mathbf{x}}_{\mathcal{S}}})^2(\mathbf{x}) = \mathbf{J}_{\hat{\mathbf{x}}_{\mathcal{S}}}(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathcal{R}_{\mathcal{S}}.$$

Moreover, since $\hat{\mathbf{x}}_{\mathcal{S}}$ is an invariant mapping, the rows of $\mathbf{J}_{\hat{\mathbf{x}}_{\mathcal{S}}}$ are orthogonal to gradients $\{\nabla f_j, j = 1 \dots M\}$, and then the projection is on the space orthogonal to the linear span of the previous gradients. That is, on the local tangent space in $\mathcal{R}_{\mathcal{S}}$.

In addition, the non-degeneracy of the Jacobian follows from a classical result in linear algebra that states that the dimension N is the sum of the dimension of the kernel (M in our case) plus the rank of the matrix.

The last statement is a consequence of $\Omega \setminus \Lambda$ being foliated by invariance submanifolds. □

3 The Nested Normalization Method

So far we have proposed a method for decoupling a new feature with respect to a given set of features. Here we apply

the results of the previous analysis in a particular hierarchical fashion, and study how and under which conditions we can obtain a set of mutually decoupled features.

3.1 Analysis

Let us consider a set $\mathcal{S}$ of M ordered non-trivially coupled global features $\{f_i(\mathbf{x}), i = 1..M\}$. We propose here a sequential algorithm that, starting by taking the first original feature $\hat{f}_1 = f_1$ unchanged, aggregates at each step k a new feature $\hat{f}_{k+1}$, as shown in Algorithm 1:

Algorithm 1 NeN: A hierarchical decoupling approach.

Require: Coupled features $\{f_j,\ j = 1, \dots, M\}$
1: **Initialization:** $\hat{f}_1 = f_1$
2: **for** $k = 1$ to $M - 1$ **do**
3: $\quad \hat{f}_{k+1} \leftarrow \text{decouple}(f_{k+1}, \{\hat{f}_i, i = 1 \dots k\})$
4: **end for**
5: **return** Decoupled features $\{\hat{f}_j, j = 1, \dots, M\}$

Our particular strategy involves constructing suitable normalization maps following this sequential aggregation scheme. For this, we use hierarchically nested reference manifolds:

$$\mathcal{R}_{M-1} \subset \cdots \subset \mathcal{R}_1 \subset \mathcal{R}_0 = \Omega \subset \mathbb{R}^N,$$

where, in the notation of Section 2.1.3, $\mathcal{R}_k = \hat{\mathbf{f}}_k^{-1}(\mathbf{v}^{ref})$, being $\hat{\mathbf{f}}_k$ a map made from the ordered set of features $\hat{\mathcal{S}}_k = \{\hat{f}_1, \dots, \hat{f}_k\}$, and a corresponding set of reference values $\mathbf{v}_k^{ref}$. At each step k, we obtain a normalization map $\hat{\mathbf{x}}_k$ with respect to $\hat{\mathcal{S}}_k$ and this is, precisely, what defines and gives its name to the *Nested Normalization* (NeN) method.

The proposed nested structure has some consequences. First, although each normalization onto $\mathcal{R}_k$ imposes a new reference value to the k feature, it respects the previously normalized values for the features $j = 1 \dots k-1$. Second, it implies that

$$\mathcal{R}_k = \hat{\mathbf{f}}_k^{-1}(\mathbf{v}_k^{ref}) = \mathbf{f}_k^{-1}(\mathbf{v}_k^{ref}), \tag{12}$$

because $\hat{\mathbf{f}}_k(\mathbf{x}) = \mathbf{f}_k(\mathbf{x})$ when $\mathbf{x} \in \mathcal{R}_{k-1}$. This property shows that, under these constraints, one can define reference values with respect to $\mathbf{f}_k$ or $\hat{\mathbf{f}}_k$, interchangeably.

We present two variants of the NeN algorithm: *Broad* and *Narrow* paths, presented in Subsections 3.1.1 and 3.1.2, respectively, depending on the choice of the integration path.

3.1.1 A broad path to normalization

This scheme is precisely explained in Algorithm 2. At the k-th step in the Algorithm, we start with a set of features $\hat{\mathcal{S}}_k = \{\hat{f}_1, \dots, \hat{f}_k\}$, constructed inductively. Assume that these satisfy the Frobenius condition C2. Then the broad path scheme yields a normalization $\hat{\mathbf{x}}_k$ with respect to the $\hat{\mathcal{S}}_k$ by integrating along trajectories inside the invariance submanifold of $\hat{S}_k$. Note that all trajectories made of linear combinations of the features' gradients imposing the desired normalization values provide the same normalization result, as they belong to the same integral manifold (which tells us that the order of integration does not change the output normalization). Therefore, this method provides us with valuable degrees of freedom for choosing convenient integration paths.

More formally, we look for suitable combinations of coefficients $\alpha_{j,k}$, such that the initial value problem:

$$\frac{d\mathbf{y}_k(t)}{dt} = \sum_{j=1}^{k} \alpha_{j,k}(t) \nabla \hat{f}_j(\mathbf{y}_k(t)), \tag{13}$$

with $\mathbf{y}_k(0) = \mathbf{x}$, can be integrated in $\mathbf{y}_k(t, \vec{\alpha}_k(t))$.

We can write, taking into account Eq. (12) for the choice of the reference values,

$$\begin{aligned} t_s &= \arg_t \left\{ \mathbf{f}(\mathbf{y}_k(t, \vec{\alpha}_k(t))) = \mathbf{v}_k^{ref} \right\}, \\ \hat{\mathbf{x}}_k(\mathbf{x}; \mathbf{v}_k^{ref}) &= \mathbf{y}_k(t_s, \vec{\alpha}_k(t_s)), \end{aligned} \tag{14}$$

with certainty that such a solution exists and is unique in a connected domain, as it only depends on the reference values of the adjusted features, and not on the choice of the α coefficients. In any case, α_j coefficients need to respect two constraints: (i) having all the sign of $(v_j^{ref} - f_j(\mathbf{x}))$, in order to go coordinately in the direction of imposing the reference values to the features; and (ii) they should not introduce any additional stationary solutions apart from the already discussed admissible critical points of the features. Under these constraints, each feature can be adjusted in its full range.

Algorithm 2 Nested Normalization, Analysis - broad path.

Require: $\{f_j, v_j^{ref},\ j = 1, \dots, M\}$
1: **Initialization:** $\hat{f}_1(\mathbf{x}) = f_1(\mathbf{x})$
2: **for** $k = 1$ to $M - 1$ **do**
3: $\quad \mathbf{f}_k(\mathbf{x}) = [f_j(\mathbf{x})],\ \mathbf{v}_k^{ref} = [v_j^{ref}],\ j = 1, \dots, k$
4: $\quad \mathcal{R}_k = \mathbf{f}_k^{-1}(\mathbf{v}_k^{ref})$
5: $\quad \hat{\mathcal{S}}_k = \{\hat{f}_j, j = 1 \dots k\}\ (*)$
6: $\quad$ Compute (ODEs) $\hat{\mathbf{x}}_k(\mathbf{x}; \mathbf{v}_k^{ref}) = \mathcal{I}(\mathbf{x}, \hat{\mathcal{S}}_k) \cap \mathcal{R}_k\ (*)$
7: $\quad \hat{f}_{k+1}(\mathbf{x}) = f_{k+1}(\hat{\mathbf{x}}_k(\mathbf{x}))$
8: **end for**
9: **return** $\{\hat{f}_j, j = 1, \dots, M\}$
(*) Modifications for the broad path relaxation in Section 3.1.3: Substitute $\mathcal{S}_k$ by $\hat{\mathcal{S}}_k$ and f_j by $\hat{f}_j$ in Steps 5 and 6, for the relaxed version of the NeN broad path.

The proposed Algorithm 1 (and its particular realization Algorithm 2) produces a new set of features $\hat{f}_1, \dots, \hat{f}_M$ such that each $\hat{f}_{k+1}$ is decoupled from the previous ones $\hat{f}_1, \dots, \hat{f}_k$. Unfortunately, it has several drawbacks that often make its implementation difficult in practice.

A first obstacle in the method is the fact that the ODEs involved in computing Step 6 of Algorithm 2 are typically difficult to solve analytically. In fact, lacking an analytical solution for the normalization at the k-th iteration translates into not being able to obtain the expression of the decoupled feature for the $k + 1$ iteration (Step 7), and beyond.

However, a more significant drawback of this scheme is that a new decoupled feature is defined upon the previous ones. As a consequence, the loop stops after a single decoupled feature no longer fulfills the requirements, meaning that the next "decoupled features" in the loop simply do not exist. In particular, Frobenius condition C2 is rather stringent, besides being usually hard to verify since the

Algorithm 3 Nested Normalization - narrow path.

Require: $\mathbf{x} \in \Omega \setminus \Lambda,\ \{f_j, v_j^{ref},\ j = 1, \dots, M\}$
1: **Initialization:** $\hat{f}_1 = f_1$; $\mathcal{R}_0 = \Omega \setminus \Lambda$; $\hat{\mathbf{x}}_0(\mathbf{x}) = \mathbf{x}$
2: **for** $k = 1$ to $k = M - 1$ **do**
3: Compute $\mathbf{g}_k = P_{\mathcal{R}_{k-1}}(\nabla f_k)$
4: $\mathbf{y}_k(0) = \hat{\mathbf{x}}_{k-1}(\mathbf{x})$
5: Follow $\mathbf{g}_k$ until $f_k(\mathbf{y}_k(t)) = v_k^{ref}$
6: $\hat{\mathbf{x}}_k(\mathbf{x}; \mathbf{v}_k^{ref}) = \mathbf{y}_k(t)$
7: $\hat{f}_{k+1}(\mathbf{x}) = f_{k+1}(\hat{\mathbf{x}}_k)$
8: **end for**
9: **return** $\{\hat{f}_j(\mathbf{x}), j = 1, \dots, M\}$, $\hat{x}_{M-1}(\mathbf{x})$

gradients of the new features $\hat{f}_1, \dots, \hat{f}_M$ will tend to have very convoluted mathematical expressions.

In next subsection we develop a second version of the algorithm (*narrow path*) that provides: i) an alternative method for feature decoupling that does not require analytical calculations; and, ii) a way to demonstrate that it is possible to relax the normalization at each step k (also in the broad path algorithm), by making it w.r.t. to the original feature set $\mathcal{S}_k$, instead of w.r.t. the decouple features set, $\hat{\mathcal{S}}_k$. In Subsection 3.1.3 below we will discuss when both approaches are equivalent.

3.1.2 The narrow path algorithm

Here we propose Algorithm 3 to construct a normalization $\hat{\mathbf{x}}_k$ with respect to the features in $\mathcal{S}_k = \{f_1, \dots, f_k\}$. Such normalization is obtained, at each step k, by moving along the gradient of each feature, projected over the previous reference manifold. Thus our normalization $\hat{\mathbf{x}}_k$ is made of a sequence of 1-D integral manifolds, whose only requirement is being each free from critical points (conditions B1-B2, see Subsection 2.2.1).

Assuming that Frobenius condition C2 holds for the gradients of $\{f_1, \dots, f_k\}$, then the normalization map $\hat{\mathbf{x}}_k(\mathbf{x})$ we obtain from the Algorithm 3 is an invariant mapping with respect to the features in $\mathcal{S}_k$. This fact follows from Proposition 2.4, since we have just provided an admissible integration path inside the invariance submanifold $\mathcal{I}(\mathbf{x}, \mathcal{S}_k)$ to reach $\hat{\mathbf{x}}_k(\mathbf{x})$ as defined in (11). Another consequence of this Proposition (statement ii.) is that, for $\mathbf{x} \in \mathcal{R}_k$,

$$\nabla \hat{f}_{k+1}(\mathbf{x}) = P_{\mathcal{R}_k}(\nabla f_{k+1}(\mathbf{x})), \tag{15}$$

where we have denoted by $P_{\mathcal{R}_k}$ the (orthogonal) projection map on the reference manifold. This in particular implies that $\nabla \hat{f}_{k+1}$ is orthogonal to the linear span of the gradients $\nabla f_1, \dots, \nabla f_k$.

If, on the contrary, Frobenius condition does not hold for the original gradients, we can still follow the direction of those projected gradients (termed $\mathbf{g}_k$ in Algorithm 3) until we reach the desired feature values. This will not give a new orthogonal gradient $\nabla \hat{f}_{k+1}$. However, the algorithm still yields interesting results since it produces approximate decoupling and improves performance in different applications (an example is shown in Section 6).

Figure 2 illustrates the NeN algorithm in its narrow-path version, for three dimensional vectors, defining two nested normalization levels.

3.1.3 From a narrow path to a broad relaxation

As we have seen in the previous subsection, the narrow path yields a normalization by moving along the gradients of the original features, whereas in the broad path we use the gradients of the modified features. These approaches are equivalent if Frobenius condition C2 holds on the gradients of the modified features. In fact that, under this condition, given $\mathcal{S}_k = \{f_j, j = 1 \dots k\}$ and $\hat{\mathcal{S}}_k = \{\hat{f}_j, j = 1 \dots k\}$ (the latter obtained with the Narrow Path Algorithm 3), then

$$\hat{\mathbf{x}}_{\mathcal{S}_k}(\mathbf{x}) = \hat{\mathbf{x}}_{\hat{\mathcal{S}}_k}(\mathbf{x}). \tag{16}$$

We see that, in this favorable setting, the obtained features in $\hat{\mathcal{S}}_k$ are mutually decoupled. Moreover, since our inductive scheme produces a $\hat{f}_{k+1}$ that is decoupled from the previous ones, then the features in $\hat{S}_{k+1}$ will also be mutually decoupled.

The proof of (16) is a consequence of our construction, since in Algorithms 2 and 3 the reference manifolds are the same thanks to Eq. (12). In addition, we recall Eq. (15) that compares the gradients of the original and the modified features when being on the reference manifolds. Thus, the solution constructed by the narrow path algorithm is both a valid concatenation of 1-D integration paths for the decoupled gradients (as we are following them), and for the original gradients, as projected gradients are linear combinations of original gradients.

As a corollary, given that the normalization result (and, thus, also the resulting set of decoupled features) is unique, if it exists, for a given ordered set $\mathcal{S}$ and their corresponding reference values $\mathbf{v}^{ref}$, then in the broad path Algorithm 2 we can simply substitute $\hat{\mathcal{S}}_k$ by $\mathcal{S}_k$ in its Steps 5 and 6. By doing that we make it totally equivalent to the narrow path algorithm. We call this change a *relaxation* of the broad path algorithm, and refer to this modified version as its *relaxed version*. Besides being much easier to implement, the broad path in its relaxed version (like the narrow path and unlike the broad path in its original version), still provides useful results (but not strictly decoupled) when the Frobenius condition does not hold on the output features, as discussed below.

More generally, our previous discussion yields decoupling in the case when a subset of the output features obtained using Algorithm 3 (or equivalent) have gradients fulfilling the Frobenius condition:

Proposition 3.1. *If the set $\mathcal{S}_k = \{f_j, j = 1 \dots k\}$ and the subset $\hat{\mathcal{S}}_k = \{\hat{f}_j, j = 1 \dots k\}$, $k \leq M$ (the latter obtained from $\hat{\mathcal{S}}$ with Algorithm 3 or equivalent, for a given set of reference values $\mathbf{v}^{ref}$) have gradients fulfilling Frobenius in $\mathcal{S}_k$ and $\hat{\mathcal{S}}_k$, respectively, then all pairs $(\hat{f}_i, \hat{f}_j)$, $i = 1 \dots k$, $j = 1 \dots M$, $i \neq j$, are mutually decoupled in the whole domain Ω.*

Note that Frobenius condition is vacuous if $k = 1$ and consequently, $(\hat{f}_1, \hat{f}_j)$, for $j = 2 \dots M$, are decoupled unconditionally.

If Frobenius condition holds for the gradients of the original features, but not for those of the transformed features, their corresponding gradients will still be orthogonal on their corresponding reference manifolds, i.e., $\nabla \hat{f}_i(\mathbf{x}) \cdot \nabla \hat{f}_j(\mathbf{x}) = 0, i \neq j$, for all $\mathbf{x} \in \mathcal{R}_{k-1}$, where $k = \max\{i, j\}$. The proof of this fact follows from the

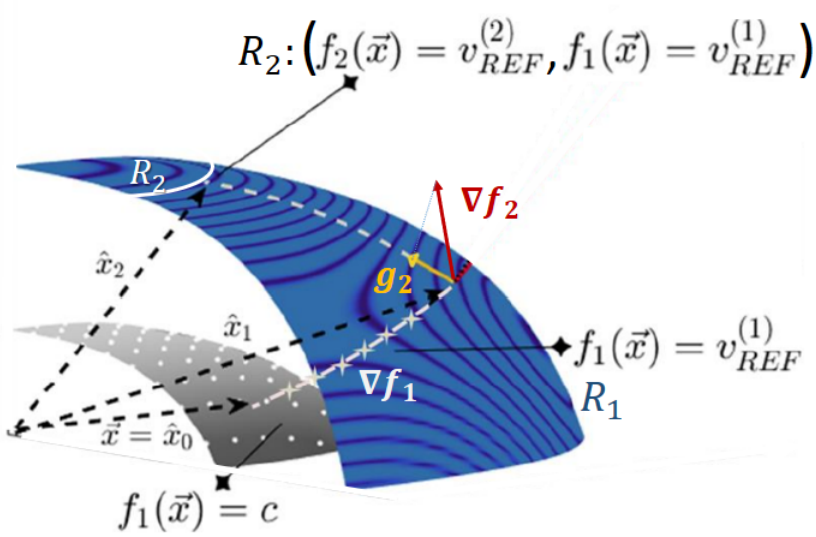


Fig. 2: Illustration of the NeN algorithm, in its narrow path version. The original vector $\mathbf{x}$ is first normalized w.r.t. f_1, that is, it is modified along the gradient of f_1 until reaching $\mathcal{R}_1$, the reference manifold with all its vectors having $f_1(\mathbf{x}) = v^{(1)}_{REF}$. That vector is $\hat{\mathbf{x}}_1(\mathbf{x})$, and there we can evaluate $\hat{f}_2(\mathbf{x}) = f_2(\hat{\mathbf{x}}_1(\mathbf{x}))$. From there we follow the projection of the gradient of f_2 onto the local hyperplane tangent to $\mathcal{R}_1$ until reaching $\mathcal{R}_2$, the set of vectors having $(f_1(\mathbf{x}) = v^{(1)}_{REF}, f_2(\mathbf{x}) = v^{(2)}_{REF})$. That vector is $\hat{\mathbf{x}}_2(\mathbf{x})$, the normalization of $\mathbf{x}$ w.r.t. both f_1 and f_2. There we can evaluate $\hat{f}_3(\mathbf{x}) = f_3(\hat{\mathbf{x}}_2(\mathbf{x}))$ (and so on).

nested structure. Indeed, assume without loss of generality that $i < j$. Then, in $\mathcal{R}_{j-1}$, $\nabla\hat{f}_j$ is the orthogonal projection over $\mathcal{R}_{j-1} \subset \mathcal{R}_{i-1}$ of the original gradient, while $\nabla\hat{f}_i$ is orthogonal to $\mathcal{R}_{i-1}$ by Eq. (12). The interest of this comes from the fact that many times, even if fully decoupling is not possible, one can still have mutual decoupling over high dimensional manifolds in Ω. In such situations, and as a practical consequence, when dealing with probability distributions it is convenient to choose reference values that are the expected values of the density function. This favors obtaining gradients close to mutually orthogonal (see Figs. 8 and 9, panels (a) and (d), in Subsection 6.2), as samples will be close to the reference manifolds, where exact orthogonality holds.

Comparing narrow and broad path versions of the NeN algorithm, the principal advantage of the broad path (especially in its much more convenient relaxed form) is that it provides closed-form solutions in some favorable cases. This usually translates on its solutions being easier to analyze and faster to compute. On the other hand, narrow path is simpler and more systematic at the implementation level, since it only requires explicit functions for the original features' gradients, and it just relies on numerical integration along 1-D trajectories.

3.1.4 Normalization of homogeneous features

A key characteristic of the NeN algorithm as we have presented it so far, is to establish a sequential order among the features to be decoupled. While there are cases, such that of marginal moments, for which it is natural to establish a hierarchical order for normalizing the features, there exist other situations where this does not apply. An example is, given a bank of scaled and rotated filters, to obtain features by applying some functions to the filters' outputs. In this case there is no reason to establish a hierarchy among the features coming from the filters at the same scale, just rotated in different angles. For these situations, a combined graph for the extracted features, having both sequential and parallel nodes seems much more appropriate than a purely sequential scheme.

Fortunately, the tools we have presented so far can be readily applied for (i) "simultaneously"(the sequential order chosen for the integration does not affect the result) changing an input sample by forcing a set of P_k features to have their reference values (i.e., normalizing the vector w.r.t. that set of features), and (ii) obtaining new features that are functions of the normalized vector (applying Equation (5), and Section 2.3.3), which we know that will be decoupled to those P_k parallel features. Therefore, we can easily adapt our algorithms just by changing some of the sequentially adjusted features f_k by P_k-dimensional features $\overline{f}^{(k)}_p$ (not to be confused with the aggregated maps $\mathbf{f}_k$ from $j = 1$ to k), without changing the underlying logic. It must be noted, though, that this procedure does not *mutually* decouple these parallel features [3].

Algorithm 4 shows our strategy to "simultaneously" impose a set of reference values to a set of homogeneous features, i.e., features having all the same functional expression, but different parameters $\{\vec{\lambda}_j, j = 1 \dots P_k\}$. The underlying idea is simple: to express analytically the solution of a single sequential ODE integration, one ODE excursion for each of the homogeneous features, and then obtain an analytical expression concatenating all these excursions. The time values t_j are left as variables, that are computed numerically

3. Mutual decoupling can be obtained by imposing a sequential order to these homogeneous features. Fully parallel (symmetrical) decoupling is a hard problem requiring different techniques from the ones presented here.

in order to fulfill the normalization (or de-normalization). Note the difference with Eqs. (14), where the normalization was obtained by solving a single non-linear equation at a time (scalar t, instead of a vector $\mathbf{t}$, like now). However, note as well that the solution of Algorithm 4 can also be expressed in terms of those equations, by choosing $\{\alpha_j(t)\}$'s that activate sequentially single-gradient combinations, at times $\{t_j\}$. Finally, it must be also noted that a variant of the previous algorithm can be used as well for the case of having homogeneous features not in parallel, but hierarchically ordered (e.g., second-order moment at the output of a bank of filters). In that case we can apply Algorithm 4 to hierarchically normalize nested subsets of features, as a particular procedure for computing Step 6 in Algorithm 2. In Subsection 4.2 we study the different alternatives to approximately decouple the second-order moments at the output of a set of hierarchically ordered filters, among which Algorithm 4 provides the most systematic approach still providing (partially) analytical solutions.

Algorithm 4 Normalization of homogeneous features

Require: $\mathbf{x}_0 \in \Omega \setminus \Lambda$, $\tilde{\mathcal{S}}_k = \{f_{j,k}(\mathbf{x}) = \tilde{f}_k(\mathbf{x}; \vec{\lambda}_j)\}$, $\{v_{j,k}^{ref}\}$, $j = 1, \ldots, P_k$

1: Solve for generic ODE $\mathbf{y}_k(t; \vec{\lambda}, \mathbf{x})$ (analytically)
2: **Initialization:** $\mathbf{x}_{0,1} = \mathbf{x}$
3: **for** $j = 1$ to $P_k - 1$ **do**
4: $\quad \mathbf{x}_{0,j+1} = \mathbf{y}_k(t_j; \vec{\lambda}_j, \mathbf{x}_{0,j})$
5: **end for**
6: $\mathbf{t} = [t_1, \ldots, t_{P_k}]$, $\mathcal{L} = \{\vec{\lambda}_1, \ldots, \vec{\lambda}_{P_k}\}$
7: Solve for $\tilde{\mathbf{y}}_k(\mathbf{t}; \mathcal{L}, \mathbf{x}) = \mathbf{x}_{0,P_k}$ (analytically)
8: Solve for $\hat{\mathbf{t}} = \arg_{\mathbf{t}}\{f_{j,k}(\tilde{\mathbf{y}}_k(\mathbf{t}; \mathcal{L}, \mathbf{x}_0)) = v_{j,k}^{ref}\}_{j=1}^{P_k}$
9: **return** $\hat{\mathbf{x}}_{\tilde{\mathcal{S}}_k}(\mathbf{x}_0) = \tilde{\mathbf{y}}_k(\hat{\mathbf{t}}; \mathcal{L}, \mathbf{x}_0)$

3.2 Feature Transfer

An essential characteristic of the integration along the direction of one or several gradients is its reversibility. First, the adjustment of a set of feature values, under the given constraints and assumptions, is always possible for every $\mathbf{x} \in \Omega \setminus \Lambda$, whenever the set of desired values are algebraically compatible (see Proposition 2.3). It is also true, in particular, that we can *de-normalize* a normalized vector (whenever the reference manifold itself is also contained in $\Omega \setminus \Lambda$), not just for recovering the original vector (as pointed out in the property (iii) from Proposition 2.4), but also for imposing whatever new feature values we may aim for. Thus, the good properties of the NeN analysis methodology presented so far allow us to change the role of vector transformation (by integrating the gradient flows) from instrumental to the main goal, and, as such, changing the focus from analysis to feature transfer or even synthesis.

However, before going into how to do that, it is important to realize that a set of mutually decoupled features have their joint range decoupled, as shown next. Given two features f_i and f_j it is immediate to assess that the decoupling condition of Eq. (2) implies that a local change in $\mathbf{x}$ along the gradient of one of the features does not affect the value of the other feature. More precisely, let us assume $\hat{\mathcal{S}}$ is a set of decoupled features defined in a set $\hat{\Omega} \setminus \hat{\Lambda}$. Since we can modify each feature within its whole range by navigating along its one-dimensional flow, independently of the values of the other features, we obtain as a corollary that

$$\mathrm{Rg}(\hat{\mathbf{f}}) = \mathrm{Rg}(\hat{f}_1) \times \ldots \times \mathrm{Rg}(\hat{f}_M) \tag{17}$$

in the set $\hat{\Omega} \setminus \hat{\Lambda}$.

The problems of finding the largest admissible domain $\hat{\Omega}$ for the decoupled set of features (which implies knowing the location and subsequently excluding all its critical points, C1 condition), and the set $\hat{\Lambda}$ of basins of all their saddles, as well as the range of each decoupled feature, are not trivial, and depend on the particular set of features. Thus, we leave that analysis for Section 4 and the Appendix C (C.1.1 and C.2.1), where the decoupling of two particular sets of features is studied in detail.

3.2.1 Peeling the onion and covering it back with new layers

Now, the decoupling range property (17) allows to modify a signal by enforcing arbitrary desired values (each within its valid range) for the decoupled features without iterative corrections, opening up an unprecedented scenario.

We show below how to achieve this by means of our NeN algorithm. All we need is applying the following 3-step procedure: (i) obtaining a set of desired decoupled features $\mathbf{v}^{des}$ we want to transfer, either by applying any of the NeN analysis algorithms we have presented so far to some *target* vector data $\mathbf{y}$, or simply by choosing any combination of decoupled features' values within their valid range; (ii) normalizing the *source* vector data $\mathbf{x}$ (the one we want to transform) up to the $M-1$-level; and (iii) de-normalizing the previously normalized data, in reverse order, until achieving for each decoupled feature $\hat{f}_k$ the same value v_k^{des} previously measured/chosen in step (i). More formally:

$$\begin{aligned}
&\text{(i)} \quad \mathbf{v}^{des} = \hat{\mathbf{f}}(\mathbf{y}) \\
&\text{(ii)} \quad \hat{\mathbf{w}}_{M-1}(\mathbf{x}) = \hat{\mathbf{x}}_{\mathcal{S}}(\mathbf{x}; \mathbf{v}_{M-1}^{ref}) \\
&\text{(iii)} \quad \mathbf{z}_f(\mathbf{x}; \mathbf{v}^{des}) = \check{\mathbf{x}}_{\mathcal{S}}(\hat{\mathbf{w}}_{M-1}(\mathbf{x}); \mathbf{v}^{des}),
\end{aligned}$$

where we have used the symbol $\check{}$ to indicate de-normalization. Thanks to the reversibility of the whole process (based on the reversibility of each ODE integration), the resulting transformed data vector $\mathbf{z}_f$ shares the same exact decoupled features $\mathbf{v}^{des}$ with the target vector, and the same normalized *kernel* $\hat{\mathbf{w}}_{M-1}$ with the source vector $\mathbf{x}$:

$$\begin{aligned}
\hat{\mathbf{w}}_{M-1}(\mathbf{z}_f; \mathbf{v}_{M-1}^{ref}) &= \hat{\mathbf{w}}_{M-1}(\mathbf{x}; \mathbf{v}_{M-1}^{ref}) \\
\hat{\mathbf{f}}(\mathbf{z}_f) &= \hat{\mathbf{f}}(\mathbf{y}),
\end{aligned}$$

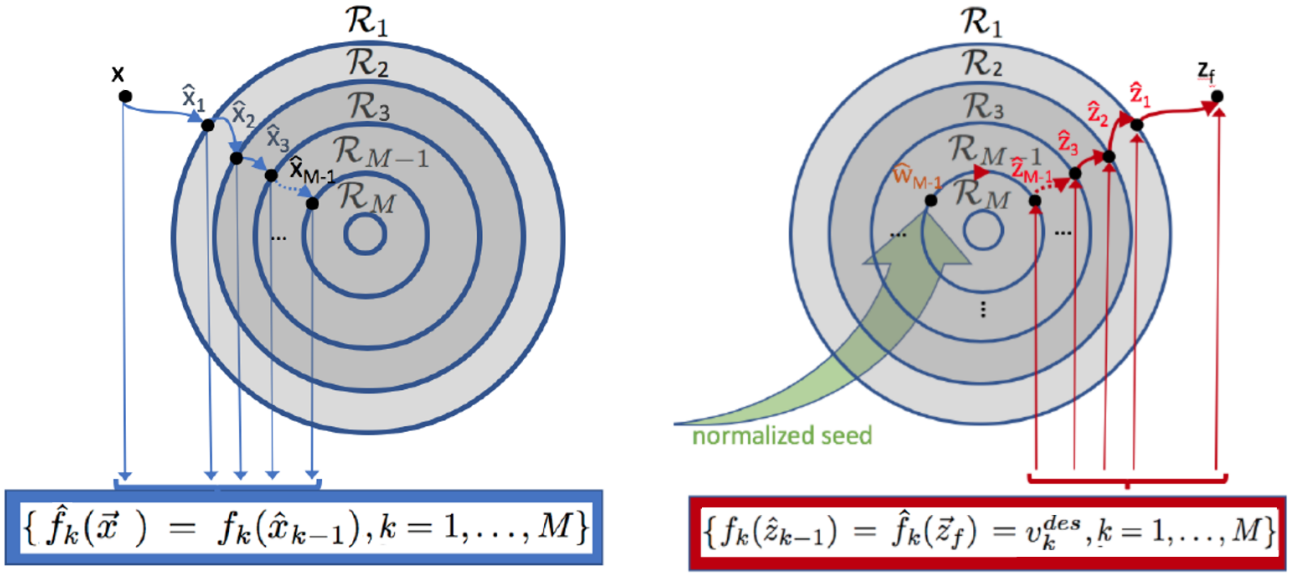
Figures 3 and 4 illustrate this process. In Algorithm 5 we describe the de-normalization steps using the (reversed) narrow-path algorithm.

This approach to feature transfer, termed Controlled Feature Adjustment in [2], and applied to photo-realistic style transfer in [28], opens up a new catalogue of transfer and synthesis possibilities, as explained in those references.

3.3 Critical points and perturbations

3.3.1 Critical points

In this subsection we look at the conditions (assuming B1, B2, C1, C2 hold) for the critical points of the decoupled features obtained using the NeN algorithm. This will be needed



Analysis via Normalization Transfer via De-normalization

Fig. 3: Feature transfer can be done as a sequence of (i) extracting decoupled features in a target vector data, (ii) normalization of the source data (both (i) and (ii) represented on the left), and (iii) de-normalization of the data normalized in (ii), by imposing the features $\mathbf{v}^{des}$ measured in (i), in reverse order (on the right). See Algorithm 5.

Algorithm 5 NeN, De-normalization (narrow path).

Require: $\hat{\mathbf{w}}_{M-1} \in \Omega \setminus \Lambda$, $\mathbf{f}^{ref} \in \mathrm{Rg}(\mathbf{f}), \mathbf{f}^{des} \in \mathrm{Rg}(\hat{\mathbf{f}})$
1: **Initialization:** $\hat{z}_M = \hat{w}_{M-1}$
2: **for** $k = M$ to 1 **do**
3: $\mathbf{y}(0) = \hat{z}_k$
4: Compute $\mathbf{g}_k = P_{\mathcal{R}_{k-1}}(\nabla f_k)$
5: Follow $\mathbf{g}_k$ until $f_k(\mathbf{y}_k(t)) = v_k^{des}$
6: $\hat{z}_{k-1} = \mathbf{y}(t)$
7: **end for**
8: **return** $\mathbf{z}_f = \hat{z}_0$

in order to establish the range of the newly constructed features.

The critical points of a feature $\hat{f}_k$ are those points $\mathbf{x}_k^*$ satisfying $\nabla \hat{f}_k(\mathbf{x}_k^*) = 0$. As, when using the Nested Normalization algorithm, we have that $\nabla \hat{f}_k = P_{\mathcal{R}_k}(\nabla f_k)$ within $\mathcal{R}_{k-1}$, the critical point condition corresponds to the orthogonality of ∇f_k to the reference manifold $\mathcal{R}_{k-1}$. Because, by the definition of reference manifold, the local tangent space at $\mathbf{x} \in \mathcal{R}_{k-1}$ is the orthogonal complement of the linear span $\mathcal{L}(\{\nabla f_j\}_{j=1}^{k-1})$, having a null projection on that local plane implies in this case that $\nabla f_k \in \mathcal{L}(\{\nabla f_j\}_{j=1}^{k-1})$, i.e., that there exist $\{\lambda_{j,k} \in \mathbb{R}\}_{j=1}^{k-1}$ not all zero such that:

$$\nabla f_k(\mathbf{x}_k^*) = \sum_{j=1}^{k-1} \lambda_{j,k} \nabla f_j(\mathbf{x}_k^*). \tag{18}$$

(Note also that a trivial solution of Eq. (18) comes from having common critical points of the original features for orders $j < k$, for which both sides of the equation vanish.) This is precisely the same condition as C1.

By introducing the (known) structure of the gradients, we obtain a general expression for the critical points (which, by the structure of the NeN method, are not isolated, but entire submanifolds). In addition, by using the previous equation plus the constraints derived from $\mathbf{x}_k^* \in \mathcal{R}_k$ we obtain the expression of the critical points of $\hat{f}_k$ on $\mathcal{R}_k$ (see the corresponding calculations in the two study cases, in Section 4).

Finally, it is important to note that decoupled features at level j *are not defined* at critical points of decoupled features at level i, for $i < j$. The reason is that iso-level sets of the feature j all cross orthogonally the iso-level sets of the feature i, until converging all to a critical point (maximum or minimum, a source or a drain of the gradient field of feature i), and therefore the feature j is not defined there. This is a direct consequence of the normalization $\hat{\mathbf{x}}_i(\mathbf{x})$ requiring the existence of a non-null gradient $\nabla f_i(\mathbf{x})$ for initiating the ODE trajectory that adjusts the i-th feature to its reference value, a previous step for computing $\hat{f}_{i+1}(\mathbf{x}) = f_{i+1}(\hat{\mathbf{x}}_i(\mathbf{x}))$. Figure 5 (comparing the orthokurtosis to skewness and kurtosis) may help to understand the involved concepts.

3.3.2 *Introducing perturbations to avoid spurious basins*

We have imposed that the domain in which the features in $\mathcal{S}$ are defined, Ω, is free from critical points. A practical way to choose the common domain Ω for a set of features $\mathcal{S}$ is to find the largest admissible set, consisting of the intersection of the domains on which each feature is defined, and then to remove from it all critical points.

We have also assumed B1-B2 conditions, i.e., that the basins of attraction of critical points other than the absolute maxima and minima are a set Λ of submanifolds with a joint dimension lower than that of Ω. In that case, it is well known that a small perturbation of a point inside these basins will

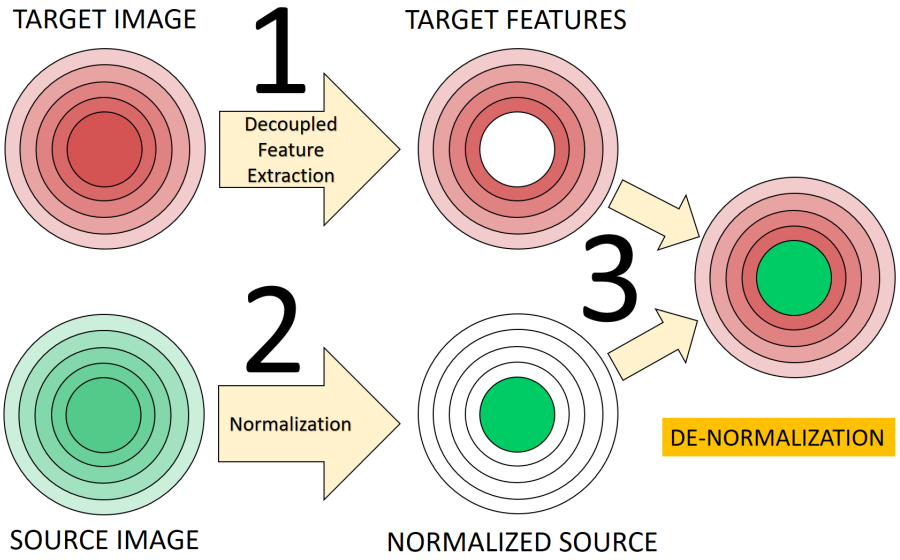


Fig. 4: Three-step feature transfer using de-normalization, within the Nested Normalization framework. The left half of the figure represent the normalization+analysis process (first and second steps, in parallel), and the right half of the figure (converging arrows) is the de-normalization+transfer process (third step).

take it back into $\Omega \setminus \Lambda$ with probability one.[4] We should also verify that for *almost all* $\mathbf{x} \in \Omega \setminus \Lambda$, the perturbed $(\mathbf{x} + \epsilon)$ will remain in $\Omega \setminus \Lambda$[5]. At which stage should we add a perturbation, then? The simplest solution, because it does not even require to check if $\mathbf{x} \in \Lambda$, is to *always* add the perturbation just before performing gradient integration.

In addition, in the context of a real application, there are two more concerns on the practical impact of the above constraints and assumptions: (i) usually, real-world signals are quantized, and, thus, they are not in $\mathbb{R}^N$, but in a finite, numerable subset. In particular, within a quantized representation, it is no longer true that the probability of *falling* on a lower dimension basin of attraction of a spurious critical point is zero. Actually, quite the opposite, signal quantization will reduce the entropy and favor the symmetry. (ii) Whereas being directly on a spurious basin of attraction (of a saddle) will make our method to eventually get stuck, that is not the only problem. Close-distance neighbor points, although outside those basins of attraction, may be affected negatively in terms of the speed of convergence of the differential equations, and should be avoided. Same can be said about the adjustment of features to values too close to their absolute extrema, especially when using numerical integration because of lacking analytical solutions.

3.3.3 Choosing a suitable perturbation

Critical points, as we will see in the study cases (Section 4), are typically points presenting low entropy configurations (high symmetry, a few repeated values/patterns, etc.). The role of the added perturbation is to increase the entropy of the signal in a suitable way, without affecting its relevant (e.g., perceptual) features. Here is a list of desirable characteristics of a perturbation ϵ on a digital signal:

- it should not cause a direct loss of information, i.e., $\mathbf{q}(\mathbf{x} + \epsilon) = \mathbf{q}(\mathbf{x}) = \mathbf{x}$, where here $\mathbf{q}(\cdot)$ represents the quantization already present in $\mathbf{x}$,
- it should be reproducible,
- it should increase the entropy (break the symmetry),
- it should not affect the operational conditions (e.g., not noticeable under human observation).

Among previous characteristics, the third and fourth ones discourage us from naively using noise-like perturbations (i.i.d. pseudo-random coefficients), because they may produce (i) some samples having very close values (just by chance); and (ii) noticeable (e.g., visual, auditory, etc.) artifacts, because of introducing unnecessarily large differences among neighbors. In appendix subsections C.1.2 and C.2.1 we propose ad-hoc perturbations for two different decoupling problems. In both studied cases the perturbation aims at increasing the entropy of the signal, by increasing the number either of the distinct signal values or of the active frequencies in the Fourier domain. As such, these perturbations expand the decoupled feature ranges to their theoretically broadest possible intervals.

4 Two Study Cases

4.1 Marginal Moments

We study here the problem of hierarchically decoupling the first M sample moments:

$$\mathcal{S} = \left\{ f_j(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} x_n^j \right\}_{j=1}^{M}.$$

First of all, we check that in this case the Frobenius condition C2 holds (see Appendix B for a proof). Therefore, the corresponding set of gradients defines a proper invariance submanifold and we can use the broad path from Subsection 3.1.1. This fact will be crucially exploited for computing

4. Although it is very easy to fabricate ad-hoc counter-examples, these counter-examples will never happen by chance using continuous perturbation densities in $\mathbb{R}^N$.

5. Idem previous footnote.

in a quasi-explicit way the fourth-order decoupled moment, that we termed the *orthokurtosis*. In Section 6 we will show how the pair skewness-kurtosis is clearly inferior for several analysis tasks than the couple skewness-orthokurtosis.

To fix notations, we will use for the normalization the moments of a zero-mean uni-variate Gaussian distribution:

$$v_j^{ref} = \begin{cases} (2j)!/(2^j j!) & \text{for } j \text{ even}, \\ 0, & \text{for } j \text{ odd}. \end{cases}$$

4.1.1 *Analytic solutions: a path to the* orthokurtosis

First, the gradient of the sample mean (the sample mean is both f_1 and $\hat{f}_1$) is:

$$\nabla \hat{f}_1(\mathbf{x}) = \frac{1}{N}\mathbf{1},$$

and the normalization comes from solving the ODE

$$\frac{d\mathbf{y}_1(t)}{dt} = \frac{1}{N}\mathbf{1}$$

starting from $\mathbf{y}_1(0) = \mathbf{x}$ and finding $t_1 = \arg_t\{f_1(\mathbf{y}_1(t)) = 0\}$ (recall that $\mathcal{R}_1 = \mathbf{f}_1^{-1}(\mathbf{v}_1^{ref}) = \{\mathbf{z} \in \Omega \,:\, f_1(\mathbf{z}) = 0\}$). The normalization will then be $\hat{\mathbf{x}}_1(\mathbf{x}) = \mathbf{y}_1(t_1)$. In this case the solution is straightforward:

$$\begin{aligned} \mathbf{y}_1(t) &= \mathbf{x} + \frac{1}{N}t \\ f_1(\mathbf{y}_1(t)) &= f_1(\mathbf{x}) + \frac{1}{N}t \\ t_1 &= -N f_1(\mathbf{x}) \\ \hat{\mathbf{x}}_1(\mathbf{x}) &= \mathbf{x} - f_1(\mathbf{x})\mathbf{1}, \end{aligned}$$

which corresponds to the original vector with the sample mean subtracted to every sample. Now we obtain the next decoupled feature, $\hat{f}_2(\mathbf{x}) = f_2(\hat{\mathbf{x}}_1(\mathbf{x}))$:

$$\hat{f}_2(\mathbf{x}) = f_2(\mathbf{x} - f_1(\mathbf{x})\mathbf{1}),$$

which is a (biased) version of the classical sample variance. Now we compute its gradient,

$$\begin{aligned} \nabla \hat{f}_2(\mathbf{x}) &= \frac{2}{N}\hat{\mathbf{x}}_1(\mathbf{x}) \\ &= \frac{2}{N}(\mathbf{x} - f_1(\mathbf{x})\mathbf{1}). \end{aligned} \tag{19}$$

Because of the irrelevance of the factor $2/N$ for the subsequent calculations, we drop it. Now we can modify $\hat{f}_2$ by moving along its gradient without leaving $\mathcal{R}_1$, until reaching $\mathcal{R}_2 = \mathbf{f}_2^{-1}(\mathbf{v}_2^{ref}) = \{\mathbf{z} \in \Omega : f_1(\mathbf{z}) = 0,\, f_2(\mathbf{z}) = 1\}$. Now $\mathbf{y}_2(0) = \hat{\mathbf{x}}_1(\mathbf{x})$,

$$\frac{d\mathbf{y}_2(t)}{dt} = (\mathbf{y}_2(t) - f_1(\mathbf{y}_2(t))\mathbf{1})\,.$$

This ODE simplifies by noting that, when the gradient has zero sample mean, the sample mean can not change when integrating it (the resulting curve belongs to $\mathcal{R}_1$). As the initial value has zero mean, $f_1(\mathbf{y}_2(t)) = 0\ \forall t$, the ODE simplifies to:

$$\frac{d\mathbf{y}_2(t)}{dt} = \mathbf{y}_2(t),$$

whose solution $\log(\mathbf{y}_2(t)) + C = t\mathbf{1}$ results in

$$\mathbf{y}_2(t) = \exp(t)\hat{\mathbf{x}}_1(\mathbf{x}),$$

by enforcing $\mathbf{y}_2(0) = \hat{\mathbf{x}}_1(\mathbf{x})$. We see that $f_2(\mathbf{y}_2(t)) = f_2(\hat{\mathbf{x}}_1(\mathbf{x}))\exp(2t)$ and the t value intersecting with $\mathcal{R}_2$ is $t_2 = -1/2\log(f_2(\mathbf{x}_1(\mathbf{x})))$. Then

$$\hat{\mathbf{x}}_2(\mathbf{x}) = \mathbf{y}_2(t_2) = \hat{\mathbf{x}}_1(\mathbf{x})/\sqrt{\hat{f}_2(\mathbf{x})}.$$

We see that this second normalization is the standardization of $\mathbf{x}$. Now we can compute the next decoupled moment $\hat{f}_3(\mathbf{x}) = f_3(\hat{\mathbf{x}}_2(\mathbf{x}))$:

$$\hat{f}_3(\mathbf{x}) = f_3\left(\frac{\mathbf{x} - f_1(\mathbf{x})\mathbf{1}}{\sqrt{f_2(\mathbf{x} - f_1(\mathbf{x})\mathbf{1})}}\right), \tag{20}$$

which is the sample skewness. Here again, we compute $\nabla \hat{f}_3(\mathbf{x})$ on $\mathcal{R}_2$ by differentiating in Eq. (20). We obtain:

$$\nabla \hat{f}_3(\mathbf{x}) = \frac{3}{N}\left(\hat{\mathbf{x}}_2(\mathbf{x})^{\odot 2} - \hat{f}_3(\mathbf{x})\hat{\mathbf{x}}_2(\mathbf{x}) - \mathbf{1}\right), \tag{21}$$

where we use the symbol "$\odot$" for representing a pointwise scalar operation for the vector coefficients (here a power).

Unlike in previous cases, now the resulting ODE equation $\frac{d\mathbf{y}_3(t)}{dt} = \nabla \hat{f}_3(\mathbf{y}_3(t))$ has no known closed-form solution. However, we note two facts. First one, as mentioned above, the original gradients fulfill the C1-C2 conditions and, thus, define a proper invariance submanifold. Second, in this case we observe that the linear span of $\{\nabla \hat{f}_j(\mathbf{x})\}_{j=1}^3$ is the same as the linear span of $\{\nabla f_j(\mathbf{x})\}_{j=1}^3$, $\forall \mathbf{x} \in \Omega$. As a consequence, both sets of features produce the same invariance submanifolds. Thus to compute the normalization, it is equivalent to use both broad path (Algorithm 2) or its relaxed version. In particular, using the gradients of the original features instead of the decoupled ones allows to find a closed-form solution for the normalization. More precisely, we proceed as follows: first find a solution by following ∇f_3 until achieving zero skew, and then standardizing that result, i.e., imposing also zero-mean and standard deviation one. The latter adjustments are a shift and re-scaling that correspond to moving along ∇f_1 and ∇f_2, admissible operations within the submanifold, that do not affect the zero-skew condition. Thus, we can pose the much easier Ricatti ODE equation obtained moving along the ∇f_3, $\frac{d\mathbf{z}_3(t)}{dt} = \nabla f_3(\mathbf{z}_3(t))$[6]:

$$\frac{d\mathbf{z}_3(t)}{dt} = \frac{3}{N}(\mathbf{z}_3(t))^2,$$

with $\mathbf{z}_3(0) = \hat{\mathbf{x}}_1(\mathbf{x})$, whose solution is (because of its irrelevance for the next calculations, we ignore the $\frac{3}{N}$ factor):

$$\mathbf{z}_3(t) = \hat{\mathbf{x}}_1(\mathbf{x}) \odot /(\mathbf{1} - t\hat{\mathbf{x}}_1(\mathbf{x})). \tag{22}$$

Then we find a numerical solution for

$$t_0(\mathbf{x}) = \arg_t\{\hat{f}_3(\hat{\mathbf{x}}_1(\mathbf{x}) \odot /(\mathbf{1} - t\hat{\mathbf{x}}_1(\mathbf{x}))) = 0\}, \tag{23}$$

and we obtain the third-order normalization by standardizing $\mathbf{z}_3(t_0(\mathbf{x}))$[7]:

$$\hat{\mathbf{x}}_3(\mathbf{x}) = \hat{\mathbf{x}}_2\left(\hat{\mathbf{x}}_1(\mathbf{x}) \odot /(\mathbf{1} - t_0(\mathbf{x})\hat{\mathbf{x}}_1(\mathbf{x}))\right).$$

6. This is not the only possibility for obtaining analytic solutions for the normalization, although it seems the best, in this case. Note that other integrable gradients, such as $\mathbf{z}_3(t)^2 - 1$ (giving rise to an hiperbolic tangent solution) have a reduced range and converge to spurious stationary solutions.

7. It is easy to check that Eq. (22) always has solution within the open interval $(1/\min(\hat{\mathbf{x}}_1(\mathbf{x})), 1/\max(\hat{\mathbf{x}}_1(\mathbf{x})))$ (note that $sign(\min(\hat{\mathbf{x}}_1(\mathbf{x}))) \neq sign(\max(\hat{\mathbf{x}}_1(\mathbf{x})))$ for all $\mathbf{x} \neq \mathbf{0}$).

Finally, this allows us to define the fourth-order decoupled moment as $\hat{f}_4(\mathbf{x}) = f_4(\hat{\mathbf{x}}_3(\mathbf{x}))$. We have termed this new decoupled sample moment $\hat{f}_4$ the *orthokurtosis*, a function that, unlike the classical standardized fourth-order sample moment (the kurtosis) is not just decoupled from the mean and variance, but also from the skewness.

Note that the computation of the orthokurtosis includes a non-explicit function, namely $t_0(\mathbf{x})$. Although we could apply a similar strategy for obtaining closed-form solutions (up to the integration parameter value) for decoupling higher-order moments by integrating separately along integer power gradients, that would not provide us with efficient solutions, because we would still need to numerically find the parameter t_0 for which the reference value for the decoupled moment is reached. For instance, for computing the fifth order decoupled moment we need to normalize the orthokurtosis. This implies evaluating every time in a loop this function, which, in turn, requires the normalization in loop of the skewness. In summary, once there are no fully analytic expressions, computationally expensive nested search loops appear, and a piece-wise 1-dimensional ODE integration strategy (as the one described in Algorithm 3) is preferable.

Finally, it is worth emphasizing how the first three decoupled moments obtained using the NeN algorithm are precisely the classical standardized moments: mean, variance and skewness. This clearly reflects how our method has captured the pre-existing natural intuitions about decoupling features through normalization. However, the next standardized moment, the kurtosis, breaks the pattern of being decoupled from all previous standardized moments, as it is algebraically coupled to skewness. This algebraic coupling has been previously noted, and some solutions have been proposed (see, e.g., [16]). Previous efforts have not aimed at orthogonalizing the involved gradients, and the few proposed modifications of the kurtosis lack a theoretical foundation and have proven inferior in their practical application to the solutions presented here (see Fig.12 in Section 6).

Figure 5 illustrates the orthogonalization of the kurtosis with respect to the skewness, giving rise to the orthokurtosis. It shows the actual iso-level curves, for the case of $N = 4$ (for visualization purposes, a dimension has been removed, namely forcing that the solutions belong to the hyperplane $\mu(\mathbf{x}) = 0$). Each trajectory shown in the orthokurtosis representation has been actually computed by integrating the projected gradients, starting from a randomly perturbed maximum of the skewness (a perturbation is necessary, because the new function is not defined at the skewness' critical points) and finishing in one minimum.

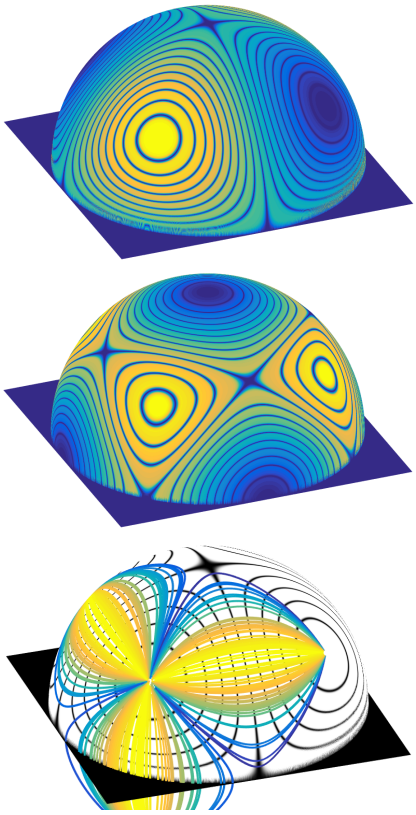


Fig. 5: Visual represention of sample skewness (up), kurtosis (middle) and *orthokurtosis*, the new fourth-order normalized moment (bottom). Dark curves (at all three panels), and coloured curves (at the bottom), are actual iso-level curves. Yellow/bright represents high values, and blue/dark low values. At the bottom panel, the iso-orthokurtosis colored curves are drawn over the iso-skewness curves (black) to show mutual orthogonality. Note how the orthokurtosis is not defined at the extrema of the skewness.

4.1.2 Beyond orthokurtosis: higher-order approximately decoupled moments

As explained in Section 3, universal and exact feature decoupling is only possible when the gradients obtained by any of the proposed algorithms fulfill the Frobenius condition C2. The gradient of the orthokurtosis, together with the gradients of its preceding decoupled moments (sample mean, variance and skewness), no longer fulfill C2 condition. Therefore, an exact unconstrained hierarchical decoupling solution is not possible beyond four-order moments. Nevertheless, as shown in Section 3.1.3, gradients orthogonality can still hold exactly within the reference manifolds and, as shown in Section 6.2, approximately outside of them.

Thus, in a looser sense of "decoupling", the lack of analytic solutions fulfilling Frobenius beyond the fourth order is not an insurmountable obstacle for computing higher-order approximately decoupled moments. In fact, we have seen (Proposition 2.4 and posterior discussion, in Subsection 2.3.3) how easy is to compute the gradient of a decoupled feature $\nabla\hat{f}_k(\mathbf{x})$ if we constrain $\mathbf{x}$ to belong to the reference manifold $\mathcal{R}_{k-1}$. In that case it holds $\nabla\hat{f}_k(\mathbf{x}) = \mathbf{g}_k(\mathbf{x}) = P_{\mathcal{R}_{k-1}}(\nabla f_k(\mathbf{x}))$, where $P_{\mathcal{R}_{k-1}}$ is the projection operator on the local tangent plane to $\mathcal{R}_{k-1}$ in $\mathbf{x}$, $\mathbf{x} \in \mathcal{R}_{k-1}$. As the orthogonal space to that tangent plane is the linear span of the previous gradients $\nabla f_j, j = 1 \dots k-1$, the projection can be computed by finding a linear combination of all gradients (including ∇f_k) that is orthogonal to the previous gradients. Because in our case the original gradients are made of "monomial vector" of increasing orders, it is easy to solve the triangular system of equations resulting from equating their inner products to

zero [6], yielding the projected gradients (a set of orthogonal vectors):

$$\begin{aligned}\mathbf{g}_1(\mathbf{x}) &= \frac{1}{N}(\mathbf{1})\\ \mathbf{g}_2(\mathbf{x}) &= \frac{2}{N}(\mathbf{x}-f_1\mathbf{1})\\ \mathbf{g}_3(\mathbf{x}) &= \frac{3}{N}\left(\mathbf{x}^{\odot 2}-f_2\mathbf{1}-a_{2,3}\mathbf{g}_2(\mathbf{x})\right)\\ &\vdots\\ \mathbf{g}_k(\mathbf{x}) &= \frac{k}{N}\left(\mathbf{x}^{\odot k-1}-\sum_{j=1}^{k-1}a_{j,k}\mathbf{g}_j(\mathbf{x})\right), \end{aligned} \quad (24)$$

with

$$\begin{aligned} c_{i,j}^{(\ell)} &= \begin{cases} f_{i+j}-f_if_j & \text{if } \ell=1,\\ c_{i,j}^{(\ell-1)}c_{\ell-1,\ell-1}^{(\ell-1)}-c_{\ell-1,i}^{(\ell-1)}c_{\ell-1,j}^{(\ell-1)} & \text{if } \ell>1,\end{cases}\\ a_{j,k} &= \begin{cases} f_{k-1} & \text{if } j=1,\\ c_{j-1,k-1}^{(j-1)}/c_{j-1,j-1}^{(j-1)} & \text{if } j>1.\end{cases}\end{aligned}$$

We remind the reader that, in general, $\mathbf{g}_k = \nabla\hat{f}_k$ only for points belonging to their corresponding reference manifolds (the k-th gradient is computed and applied in the $\mathcal{R}_{k-1}$ manifold). Some of them may get simpler expressions when imposing the corresponding reference values (particularly, lower than order k odd moments vanish if we take as reference the moments of even symmetric pdf's, e.g., Gaussian). They have the cross-invariance property, i.e., by integrating a curve along the k-th decoupled gradient we do not change the previous featuress $f_j, \hat{f}_j\ j=1\dots k-1$. Although in this case we obtained a closed-form (recursive) solution for the gradient projection, in case of lacking close-form expressions we can always apply a purely numerical orthogonalization method to the gradient vectors of the original features (like Gram-Schmidt).

In our practical examples of applying decoupled moments to signal analysis in Section 6, we demonstrate the usefulness of higher than four order decoupled moments. This is, we believe, a relevant result, as the original moments of so high order are very rarely used in the literature due to their instability and high redundancy.

4.2 second-order moments at the output of a set of filters

We study here the problem of hierarchically decoupling second-order moments measured at the output of a set of M linearly independent band-pass (zero DC-response) filters:

$$\mathcal{S}=\left\{f_j(\mathbf{x})=\frac{1}{N}\sum_{n=1}^{N}[\mathbf{x}*\mathbf{h}_j]_n^2\right\}_{j=1}^{M}.$$

Such a set of features provides an economical description of the auto-correlation of $\mathbf{x}$. We will use for the normalization $\{v_j^{ref}=f_j(\mathbf{w})\}_{j=1}^M$, being $\mathbf{w}=N\delta$, a scaled Kronecker delta. This is equivalent to taking for reference values the expected value of $f_j(\mathbf{y})$ for $\mathbf{y}$ a vector made of i.i.d. zero-mean and unit variance coefficients. This choice for reference values, being the values of the functions applied to a given vector $\mathbf{w}$, guarantees the algebraic compatibility of $\mathbf{v}^{ref}$, regardless of the chosen set of filters $\{\mathbf{h}_j\}$. In addition, it makes the normalization to *whiten* the input.

Same as in previous Subsection 4.1, we first check that in this case the Frobenius condition C2 holds for the original gradients (see Appendix B for a proof), and, therefore, the corresponding set of gradients defines a proper invariance submanifold. This allows us to apply Algorithms 2 (in its relaxed form), 3 or 4 for trying to decouple these features, and see *a posteriori* if Frobenius condition also holds on the gradients of the obtained features.

4.2.1 General analytical approach

We first obtain the feature gradients and study their integrability. We have:

$$\nabla f_j(\mathbf{x})=\frac{2}{N}\mathbf{x}*\mathbf{h}_j*\tilde{\mathbf{h}}_j,$$

where $\tilde{\mathbf{h}}_j(\mathbf{n})=\mathbf{h}_j(-\mathbf{n})$. To simplify this expression and its subsequent integration, we express it in the Fourier domain, by doing the DFT of $\mathbf{x}$ and $\mathbf{h}$:

$$G_j(X(\xi))=\mathcal{F}\{\nabla f_j(\mathbf{x})\}=\frac{2}{N}|H_j(\xi)|^2X(\xi), \quad (25)$$

where ξ represents the (possibly vectorial, for n-D signals, $n>1$) discrete frequencies, and, as usual, upper case letters represent the Fourier transforms of their original lower-case counterparts (except for G, that corresponds to the Fourier transform of the gradient of the features).

In order to normalize the first k features, following the broad path algorithm, we can write the relaxed version of Eq. (13) in the Fourier domain as:

$$\frac{dY_k(\xi,t)}{dt}=\frac{2}{N}\left(\sum_{j=1}^{k}\alpha_{j,k}|H_j(\xi)|^2\right)Y_k(\xi,t),$$

for some convenient choice of the integration path, encoded by the coefficients $\vec{\alpha}_k$. Setting $Y_k(\xi,0)=X(\xi)$ and calling $L(\xi;\vec{\alpha}_k)$ the filter of the sum in brackets in previous equation, the integration of the initial value problem for computing the signal normalization is straightforward:

$$Y_k(\xi,t;\vec{\alpha}_k)=X(\xi)\exp\Big(\frac{2}{N}L(\xi;\vec{\alpha}_k)t\Big). \quad (26)$$

To obtain $\hat{\mathbf{x}}_k(\mathbf{x},\mathbf{v}^{ref})$, we first normalize $\alpha_{1,k}=1$ and then solve for

$$(\vec{\alpha}_k^{ref},t_k^{ref})=\underset{\vec{\alpha},t}{\arg}\Big\{\sum_{\xi}|H_j(\xi)|^2\,|Y_k(\xi,t;\vec{\alpha}_k)|^2=v_j^{ref}\Big\}_{j=1}^{k} \quad (27)$$

Finally, $\hat{X}_k(\xi)=Y_k(\xi,t_k^{ref};\vec{\alpha}_k^{ref})\}$, and $\hat{\mathbf{x}}_k(\mathbf{x},\mathbf{v}^{ref})=\mathcal{F}^{-1}\{\hat{X}_k(\xi)\}$. Solving the non-linear system of k equations and k unknowns of Eq. (27) will require a numerical computation in a general case. However, it may also have simplified solutions in some special cases, as we will see in next subsection.

An equivalent possibility for computing $\hat{\mathbf{x}}_k(\mathbf{x},\mathbf{v}^{ref})$ consists in following the Algorithm 4: calculating analytically the result of a generic ODE excursion using a single gradient ∇f_j from a given point, with $\alpha_{j,k}=1$, as a function of $t_{j,k}$ (initially left as unknown), Step 1. Then we concatenate these 1-D trajectories (Steps 2-5) into a final analytical solution depending on $\mathbf{t}_k=[t_{j,k}], j=1\dots k$ (Step 7). It is

easy to see that such analytical solution has the same form of Eq. (26) (but depending on a set of consecutive times $t_{j,k}$, instead of $\alpha_{j,k}$), and that the Step 8 of that algorithm, which enforces the solution achieving all k reference values in $\mathbf{f}_k(\mathbf{x})$, is equivalent to Eq. (27).

4.2.2 A case including two complementary filters

Let us consider a Parseval frame representation [30] using two filters, $H_1(\xi), H_2(\xi)$ fulfilling $|H_1(\xi)|^2 + |H_2(\xi)|^2 = 1$, e.g., a low and high-pass kernels of a redundant wavelet transform. Here the features are simply $f_j(\mathbf{x}) = \sigma_j^2(\mathbf{x})$, $j = 1, 2$. Since the Euclidean metric is preserved, we have $\sigma_1^2 + \sigma_2^2 = \sigma^2$, the total signal variance (for simplicity sake we assume here zero mean).

For normalizing the first feature, we integrate its corresponding gradient, obtaining:

$$Y_1(\xi, t) = X(\xi) \exp\Big(\frac{2}{N}|H_1(\xi)|^2 t\Big).$$

As usual, we then find the t parameter providing us the desired reference value:

$$t_1^{ref}(\mathbf{x}) = \arg_t \Big\{ \sum_\xi |H_1(\xi)|^2 \, |Y_1(\xi, t)|^2 = v_1^{ref} \Big\}$$

and we obtain $\hat{\mathbf{x}}_1(\mathbf{x}) = \mathcal{F}^{-1}\{Y_1(\xi, t_1^{ref}(\mathbf{x}))\}$, from which we obtain the decoupled feature $\hat{f}_2(\mathbf{x}) = f_2(\hat{\mathbf{x}}_1(\mathbf{x}))$.

If we wanted to normalize also the second feature (e.g., in order to add more decoupled features), we could move along the gradient of the second feature, but, at the same time, control that the first feature does not change its value (this is equivalent to project ∇f_2 onto $\mathcal{R}_1$). We can achieve that by dividing by the square root of the first feature evaluated for each t. Such an adjustment is valid in this case because it corresponds to moving along the sum of the two gradients (which in this case corresponds to simply applying a scale factor to the vector):

$$Y_2(\xi, t) = \frac{\hat{X}_1(\xi) \exp(\frac{2}{N}|H_2(\xi)|^2 t)}{\sqrt{(v_1^{ref})^{-1} \sum_\xi |H_1(\xi)|^2 \, |\hat{X}_1(\xi) \exp(\frac{2}{N}|H_2(\xi)|^2 t)|^2}}, \tag{28}$$

thus enforcing that $\sum_\xi |H_1(\xi)|^2 |Y_2(\xi, t)|^2 = v_1^{ref}$. Same as before, we solve for the t value that achieves the desired normalization:

$$t_2^{ref}(\mathbf{x}) = \arg_t \Big\{ \sum_\xi |H_2(\xi)|^2 |Y_2(\xi, t)|^2 = v_2^{ref} \Big\}$$

and we obtain $\hat{\mathbf{x}}_2(\mathbf{x}) = \mathcal{F}^{-1}\{Y_2(\xi, t_2^{ref}(\mathbf{x}))\}$, from which we could obtain another decoupled feature from any arbitrary (non-trivially redundant) feature $g(\mathbf{x})$; indeed, $\hat{g}(\mathbf{x}) = g(\hat{\mathbf{x}}_2(\mathbf{x}))$.

Note also that Eq. (28), although looking quite different from Eq. (26), is a particular case of the latter with $\alpha_{1,2} = t_1^{ref} - \nu$, $\alpha_{2,2} = t_2^{ref} - \nu$, ν being the logarithm of the square root in the denominator of Eq. (28). No matter the adjustment method applied here may seem arbitrary, the fulfillment of Frobenius conditions on the gradients of the original features guarantees, jointly with the additional constraints explained in Section 2, that the result of such normalization exists and is unique, as it only depends on the reference values of the adjusted features, and not on the choice for the coefficients α's in the linear combinations of the feature gradients, in the ODEs.

Finally, although not mathematically proven here[8], it turns out that the resulting gradients of the new features obtained in this case do not fulfill the Frobenius condition. This implies that strict gradients' orthogonality only holds for all pairs within their reference manifolds, and in the whole domain for the pairs $(\hat{f}_1, \hat{f}_j), j = 1 \dots M$, as explained in Subsection 3.1.3. Nevertheless, we have obtained gradients that are very close to being mutually orthogonal also in the other cases when applying a set of bar and edge detectors both to white noise and to textured patches of photographic images (see Fig. 9 in Section 6.2).

4.3 A summary of feature-decoupling scenarios

Table 1 summarizes the different situations one may encounter when trying to apply the decoupling framework proposed here to decouple a given set of ordered features. From less favorable to more favorable, the first scenario is when we do not have an explicit expression of the original features gradients. The most extreme case would be that each of our features is a black box function. In that case, a very computationally costly numerical procedure (such as described in Section 6.2, see Eq. (33)) is the only option for approximately computing the gradients. A better situation is when using artificial neural networks (ANNs) with automatic differentiation for computing the gradients of the cost functions. In that case we can evaluate the gradients with a reasonable cost (and apply numerical integration, using the narrow-path algorithm), but we may not be able to assess the fulfillment of the Frobenius condition for the original features beyond the first layer. As such, we should not expect a strict and universal decoupling beyond the second feature (the second layer, if we decouple in a layer-wise fashion). For those features, it becomes an empirical matter to test how far new gradients typically are from being mutually orthogonal.

Second scenario corresponds to knowing the explicit expressions of our features and their gradients, and knowing that they do not jointly fulfill the Frobenius condition. In that case we can still apply the narrow-path version of NeN and, again, test empirically how far the resulting gradients of the (approximately) decoupled features are from being orthogonal. This corresponds, for instance, to the case of higher-than-two order moments at the output of a set of filters.

Third scenario is when we know the analytic expressions of the gradients of the original features and they fulfill Frobenius. In that case the definitions of Section 2 for the decoupled features apply for finding decoupled features *to the original ones*, and we may end up finding explicit expressions for their gradients. However, in this scenario these gradients turn out not fulfilling Frobenius condition. First, we recall that the first decoupled feature is just the first feature $\hat{f}_1(\mathbf{x}) = f_1(\mathbf{x})$, as always, and, for that reason $\mathcal{S}_1 = \hat{\mathcal{S}}_1$, so the broad-path algorithm applied to obtaining the second decoupled feature is the same as its relaxed version. Furthermore, the second feature is exactly and univer-

8. Explicit expressions of the second decoupled feature's gradient can be obtained through implicit derivation.

sally decoupled, as Frobenius condition becomes vacuous in 1-D. However, if we are able to compute both gradients and they do not fulfill the Frobenius condition, we then know that no added third or following features will be universally and exactly decoupled from the two previous ones. Still, we may find exact constrained decoupling solutions: the output features will be exactly decoupled within the corresponding reference manifolds, but only approximate outside them. Again, finding how close to be orthogonal are the gradients outside those manifolds requires an empirical measurement. An example of this situation is when decoupling the second-order moments at the output of a set of filters overlapping in the Fourier domain.

Finally, the most favorable scenario corresponds to being able to obtain explicit expressions for the input (coupled) and output features (by using the broad-path NeN algorithm, Algorithm 2, either in its original form or in its relaxed version) and their corresponding gradients, and that the two sets of gradients fulfill Frobenius condition. In this case we obtain the full decoupling solution, which we know is unique, universal and exact. In addition, in this case there is a joint equivalence relationship between the set of original features and the obtained decoupled set. E.g., when decoupling the first three marginal moments, the decoupled result (sample mean, variance and skewness) jointly carries exactly the same information as the original set (first, second and third-order moments). When adding the fourth-order moment we obtain another exact and universally decoupled feature (the orthokurtosis). However, because of the gradient of the orthokurtosis no longer fulfills Frobenius condition jointly with the lower order gradients, then: (1) the joint equivalence relationship between coupled and decoupled moments up to order four does not hold anymore, and (2) we can not further obtain exact universal decoupling for any added feature to this set (and, in particular, for any higher-order moments).

5 Deterministic Decoupling and Local Decorrelation

Here we show how features' decoupling removes local covariance in the feature space, and how this improves discrimination.

5.1 Covariance-free "balls" in the feature space

Let $\mathbf{x} \in \mathbb{R}^N$ be a random vector made of N i.i.d. samples obeying a probability distribution $p(x)$. Let us assume a feature set $\mathcal{S}$ of M global features $\{f_j\}$. Define a vector $\mathbf{c} \in \mathbb{R}^M$ containing the expected value of the features $f_j(\mathbf{x})$ for different realizations of $\mathbf{x}$, i.e., $c_j = \mathbb{E}\{f_j(\mathbf{x})\}, j = 1 \dots M$. Define a manifold $\mathcal{A}(\mathbf{c}; \mathcal{S}) = \mathbf{f}^{-1}(\mathbf{c})$, i.e., the manifold containing the set of all vectors $\mathbf{x}$ having the same $\mathbf{c}$ (note that it is non-degenerate by our initial assumption). Describe vector samples as $\mathbf{x}_i = \mathbf{x}_{0i} + \mathbf{d}_i$, where $\mathbf{x}_{0i} \in \mathcal{A}$, and $\mathbf{d}_i$ is a (relatively small) sampling fluctuation, with $\mathbb{E}\{d_{i,k} d_{i,l}\} = 0$, $k, l \in [1, \dots, N]$, for all k-th and l-th components of the vector $\mathbf{d}_i$.

Proposition 5.1. (Decoupled features are locally uncorrelated). *Under previous assumptions, for N large, decoupled features will have uncorrelated deviations from their expected values, i.e., $\mathbb{E}\{(\hat{f}_n(\mathbf{x}) - c_n)(\hat{f}_m(\mathbf{x}) - c_m)\} \approx 0$, $n, m \in [1, \dots, M]$.*

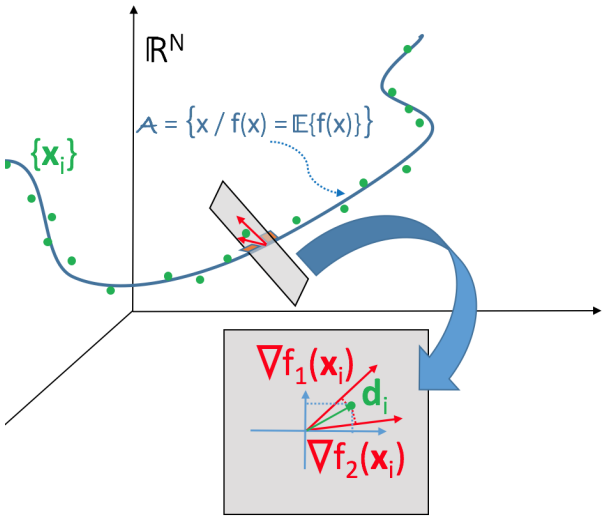


Fig. 6: A representation of the expected feature vector manifold in the input space domain, showing some input samples, and the influence of feature gradients' correlation on the correlation in the feature domain.

Proof. For N large, features' values, as they behave like sample statistics (see Eq. (1)), will not deviate much from their expected values and thus vector samples $\{\mathbf{x}_i\}$ will be located in the vicinity of $\mathcal{A}$. Thus, a first-order local approximation can be applied, which gives

$$f_j(\mathbf{x}_i) \approx f_j(\mathbf{x}_{0i}) + \nabla f_j(\mathbf{x}_{0i}) \cdot \mathbf{d}_i = c_j + \nabla f_j(\mathbf{x}_{0i}) \cdot \mathbf{d}_i. \quad (29)$$

Therefore, $\mathbb{E}\{(f_n(\mathbf{x}) - c_n)(f_m(\mathbf{x}) - c_m)\} \approx \mathbb{E}\{(\nabla f_n(\mathbf{x}_0) \cdot \mathbf{d}(\mathbf{x}_0))(\nabla f_m(\mathbf{x}_0) \cdot \mathbf{d}(\mathbf{x}_0))\}$, yielding the covariance:

$$Cov(f_n, f_m)(\mathbf{x}) \approx \sigma_d^2 \nabla f_n(\mathbf{x}_0) \cdot \nabla f_m(\mathbf{x}_0), \quad (30)$$

where σ_d^2 is the expected quadratic dispersion of the features fluctuations. □

In the decoupled features case gradients are mutually orthogonal, and thus vector differences for the different features will be uncorrelated. In contrast, when using coupled features, $\mathbf{d}_i$ is projected onto non-orthogonal directions, leading to correlated sampling fluctuations in the feature space, as illustrated in Figure 6.

5.2 Features' covariance and discriminability

Let us assume now that our pdf depends on a parameter θ, $p(x; \theta)$. Consider also a global feature transformation $\mathbf{f}(\mathbf{x}) : \mathbb{R}^N \to \mathbb{R}^M$ meant to be applied to vectors $\mathbf{x}(\theta) \in \mathbb{R}^N$ made of samples from $p(x; \theta)$. How well can we discriminate samples coming from similar values of θ, based on $\mathbf{f}(\mathbf{x}(\theta))$? For studying this problem it is convenient to represent the samples $x(\theta)$ using an intermediate stochastic sample $x_0 \sim p(x; \theta_0)$ that does not depend on θ; then we obtain the final sample by applying a deterministic invertible mapping $s_\theta : \mathbb{R} \to \mathbb{R}$ of x_0 depending on θ: $x(\theta) = s_\theta(x_0)$, such that $x(\theta) \sim p(x; \theta)$ (re-parametrization trick [31], [32].

TABLE 1: Four feature decoupling scenarios.

		#SCENARIO 1	2	3	4
Original Feat. Gradients:	Analytic Expression	Non-Explicit	Yes	Yes	Yes
	Frobenius	?	No	Yes	Yes
Output Feat. Gradients:	Analytic Expression	?	No	Yes (2)	Yes
	Frobenius	?	No	No	Yes
Joint Equivalence Coupled/Decoupled Set		No	No	No	Yes
Does accept one additional decoupled feature?		No	No	No	Yes
NeN Applicability:	Computation	Heavy/?	Medium	Medium	Light
	Algorithm	3	3	3,4	2,3,4
	Decoupling	1 layer	Approx/Constr	Approx/Constr	Exact&Univ.
Example (See Table 2 for acronyms' description)		ANN	MF ($p > 2$)	MF ($p = 2$)	MM ($p \leq 3$)
See Secs.& Refs.		Future Work	6.4, 6.4.2 & [6], [7]	4.2, 6.2.2 & [2]	4.1, 6.4.1, 6.2.1, 6.3

This allows us to study the dependency of the expected feature vector $\bar{\mathbf{f}}$ on θ, by expressing:

$$\begin{aligned} \frac{d\bar{\mathbf{f}}(\theta)}{d\theta} &= \frac{d\mathbb{E}\{\mathbf{f}(\mathbf{x}(\theta))\}}{d\theta} \\ &= \mathbb{E}\left\{\mathbf{J_f}\frac{d\mathbf{x}(\theta)}{d\theta}\right\} \\ &= \mathbb{E}\left\{\mathbf{U_f}\mathbf{S_f}\mathbf{V_f}^T\frac{d\mathbf{x}(\theta)}{d\theta}\right\}, \end{aligned} \tag{31}$$

where $\mathbf{J_f}$ is the Jacobian matrix of $\mathbf{f}$ and $\mathbf{U_f}\mathbf{S_f}\mathbf{V_f}^T$ is its singular value decomposition, SVD (we have omitted here their dependency on $\mathbf{x}(\theta)$ for brevity). On the other hand, from Eq. (30) we can write the expected local covariance matrix $\mathbf{C}(\theta)$ of the features fluctuations, as:

$$\begin{aligned} \mathbf{C}(\theta) &\approx \sigma_d^2\,\mathbb{E}\{\mathbf{J_f}\mathbf{J_f}^T\} \\ &= \sigma_d^2\mathbb{E}\{\mathbf{U_f}\mathbf{S_f}^2\mathbf{U_f}^T\}. \end{aligned} \tag{32}$$

Here it is crucial to notice that, under the assumptions made in previous and current subsections, whereas $\mathbf{J_f}(\mathbf{x}(\theta))$ will heavily depend on $\mathbf{x}$, $\mathbf{J_f}(\mathbf{x}(\theta))\mathbf{J_f}^T(\mathbf{x}(\theta))$ will be much less sensitive to $\mathbf{x}$, as it only depends on the inner products of the different features' gradients (see Eq. (30)). Furthermore, in Subsection 6.2 we show how these inner products (at least their correlation factor, which depends only on their relative angle) are fairly stable, especially when inputs are samples from pdf's. Therefore, the $\mathbf{U_f}(\mathbf{x}(\theta))$ and $\mathbf{S_f}(\mathbf{x}(\theta))$ matrices, on their average behavior, will determine both the direction of change of $\bar{\mathbf{f}}$ when changing θ (Eq. (31)) and the dominant direction of $\mathbf{C}(\theta)$ (Eq. (32)), which, thus, will tend to coincide. Our observations indicate that the eigenvalues of $\mathbf{C}(\theta)$ (the diagonal terms of $\mathbf{S_f^2}$) are fairly concentrated in the studied cases. This implies that the features' coupling actually causes a *worst case scenario* for discriminating between similar θ's: the features' pdfs become (i) elongated (due to eigenvalues' concentration), and (ii) locally aligned with the $\bar{\mathbf{f}}(\theta)$ curve. This causes strong overlapping of the pdf's having close θ values, and, as a result, poor discrimination. Fig. 7(left) illustrates this phenomenon in a real experiment with real data. Fig. 7(right) shows the effect of decoupling the kurtosis from the skewness (*orthokurtosis*). We used 128 random vectors of 1024 i.i.d. samples each, from $x(\theta) = x_0^\theta$, being $x_0 \sim U(0,1)$. Ellipses correspond to a Mahalanobis radius of 2, and $\theta = 5$ (black), 6 (blue), and 7 (red). Expected error probabilities, using a bi-variate Gaussian model, are 12.4% (coupled) vs. 4.5% (decoupled).

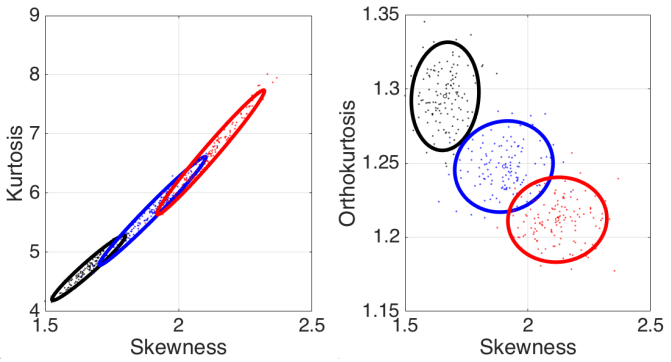


Fig. 7: Comparing Gaussian classes in the original feature space (left) and in the decoupled feature space (right). See text for details.

To conclude this section, the techniques presented here attack the core of the poor discrimination problem due to using coupled features, by orthogonalizing the feature gradients, which has the effect of approximately diagonalizing the local covariance matrix $\mathbf{C}(\theta)$. It is crucial to note that this diagonalization is effective because it is *local*. A *global* diagonalization (such as the classical Principal Component Analysis, PCA) would be useless for reducing the pdfs' overlapping corresponding to close θ values in the feature space, as such overlapping is insensitive to global affine transformations of that space. In contrast, a global linear correlation (as the one shown in Figure 7(b)) can be trivially removed, if needed, by applying PCA *after* feature decoupling.

6 Experiments and Applications

6.1 Using two families of global features

In this section we define the two features' families (sets) that will be studied in the experiments, namely: (i) marginal moments of arbitrary order p (MM); (ii) p-th order moment at the output of a 2-D filter bank defined from a filter h (MF). Specifically, to extract the MF features, we first applied the Translation Invariant Laplacian separable (TILs) representation [33], a tight frame acting as bar and edge detector that provides nine subbands, each with the same number of coefficients as pixels in the image. Then, the p-th order moment was obtained for every subband. To

obtain the corresponding decoupled sets of the MM and MF families (DF_{MM} and DF_{MF} respectively), we used the Nested Normalization-narrow path (Algorithm 3). For the MM set of features, we used as reference values the p-th moment of a standardized Gaussian distribution (i.e., $(p-1)!!$ for even p, 0 for odd [34]). For the MF set of features, reference values corresponded to moments obtained by convolving zero-mean univariate white Gaussian noise with a kernel h, which, in our case, using use the set of kernels $\{h_j, j = 1 \dots 9\}$ from the TILs representation, are the same function of p as for the MM family in all subbands. Table 2 shows the original features for MM and MF. It also shows their corresponding gradients expressions (ignoring scaling factors which do not influence the result) and indicates in which cases the set of original gradients fulfills the Frobenius condition.

TABLE 2: Families of features (marginal moments, MM, and moments at the output of filters, MF), and their gradients.

Family	$f(\mathbf{x})$	$\partial f(\mathbf{x})/\partial x_i$	Frobenius
MM	$1/N \sum_{n=1}^{N} x_n^p$	$x_i^{(p-1)}$	For all p
MF	$1/N \sum_{n=1}^{N} (x * h)_n^p$	$(x * h)^{(p-1)} * \tilde{h}$	Only for $p = 2$

6.2 Measuring the amount of mutual coupling

In this section we evaluate how close to being mutually orthogonal are the feature's gradients, for two sets of coupled features and their corresponding decoupled sets, namely: (i) a set composed of the first six orders of the classical MM, in its standardized version: mean, variance, and the rest the moments of the standardized sample to zero mean and unit variance (that is, skewness, kurtosis, etc.). We will refer to this classical set of statistical features by "MSM" (from Marginal Standardized Moments), and DF_{MSM} its corresponding decoupled set; (ii) a set composed of the second-order moments at the output of a filter bank ("VF", a particular case of MF with $p = 2$ and assuming zero mean) and its corresponding decoupled set (DF_{VF}).

Let $\{f_j(\mathbf{x})\}$ represent the original features and $\{\hat{f}_j(\mathbf{x})\}$ its corresponding decoupled set. Note that $j = 1, \dots, 6$ for MSM and $j = 1, \dots, 9$ for VF (for the 9 subbands of the TILs representation). Let $\mathbf{x_0}$ represent an N-D vector of i.i.d. samples drawn from a Gaussian or an uniform distribution; or an N-D vector that represents the pixel values of a texture patch extracted from an image of the Broadtz database [35]. To obtain the gradient of a feature at $\mathbf{x_0}$, we numerically calculated the partial derivatives with respect to the i-th variable $x_i \in \mathbf{x}$ by adding a differential perturbation ϵ to the i-th element of vector $\mathbf{x}$:

$$\frac{\partial f(\mathbf{x_0})}{\partial x_i} = \lim_{\epsilon \to 0} \frac{f(\mathbf{x_0} + \epsilon \mathbf{e_i}) - f(\mathbf{x_0} - \epsilon \mathbf{e_i})}{2\epsilon}. \quad (33)$$

This expression yields the gradients for each feature of the MSM, the VF, and their corresponding decoupled sets (DF_{MSM} and DF_{VF} respectively). To evaluate the function $\hat{f}_j(\mathbf{x})$ in the DFs cases we used the Nested Normalization-narrow path (Algorithm 3) using equation (24) for a fast analytical computation of the gradient's orthogonal projections.

Then we measured the angle α between pairs of gradient vectors of the different features that belong to the MSM and VF sets $\{f_j(\mathbf{x})\}$ and that belong to the DF_{MSM} and DF_{VF} sets $\{\hat{f}_j(\mathbf{x})\}$. The deviation from orthogonality (DO) was obtained as the difference between 90 degrees (perfect orthogonality) and the actual calculated angles ($DO = 90 - \alpha$). As such, $DO > 0$ indicates acute angles and $DO < 0$ obtuse angles. The number of samples were N=512 for the Gaussian and uniform distributions and N=529 (23×23 pixels) for textures. The experiment was repeated M=256 times for the Gaussian and uniform distributions. In the case where $\mathbf{x_0}$ came from textures, we used a single patch for each of the M=112 different textures in the Brodatz database. Table 3 shows the average of the absolute value of the DO across the different pairs of feature's gradients for the different distributions tested, for the MSMs, VFs, DF_{MSM} and DF_{VF}. We excluded from the average calculation the mean and the variance in the MSM and DF_{MSM} cases, as they are orthogonal by definition. Figures 8 and 9 show the DO results. Panels (a) and (b) show the DO between different pairs of gradient's features for the original set and its decoupled counterpart (Gaussian and Textures cases respectively). Panels (c-f) show the DO, in absolute value, between different pairs of gradients. Panels (c) and (d) show results for the Gaussian case; panels (e) and (f) show the results obtained for the Textures case. Blue color indicates |DO| close to 0 degrees (orthogonality), while yellow color indicates a deviation from orthogonality close to 90 degrees (angle of 0 degrees). Note that the main diagonal only acts as a reference (0 degrees) here.

6.2.1 *Marginal moments*

Let us analyze first the case of the marginal moments. In agreement with the theory (the first three decoupled moments have gradients fulfilling Frobenius condition, see Proposition 3.1) we see that the DO is exactly zero (perfect orthogonality) for all standardized moments combined with orders 1 and 2 (note that the MSM set is already a partially decoupled version of the original MM family set), and for all decoupled moments combined with orders 1, 2 and 3. In addition, it is close to zero on average in the rest of odd-even combinations, in which case it presents a wide variance for MSM, and a much narrower one for DF_{MSM}. For the odd-odd and even-even rest of the cases, MSM presents highly acute angles (strong average coupling) with an extremely low variance, whereas DF provides either perfect (3-5 case) or approximate (4-6 case) orthogonality, also with low variance. In summary, the decoupled moments DF_{MSM} provide close to orthogonal gradients also for orders greater than 3, especially for the random sampling experiments (samples are close to their expected values, in the reference manifolds, where exact orthogonality holds), but also, to a lesser extent and with higher variability, for photographic images (textures). In contrast, MSM presented in those cases either very high deviations (with low variability) or low-to-moderate average deviations with high variability (especially in real images). In Table 3 we can see that, for real photographic images, the average absolute DO has been reduced in a

factor 6, approx., whereas for Gaussian samples it has been reduced 16 times.

6.2.2 Second-order moments at the output of a filter bank

Figure 8 shows the empirical results obtained using the VF features. First, we note that original features are fairly coupled, although not as much as in the marginal moments' case. Now the new DF_{VF} features are exactly and universally decoupled only for the eight pairs $(\hat{f}_1, \hat{f}_j), j = 2 \dots 9$, for which the theory tells us that the gradients of the new features achieve perfect orthogonality when Frobenius condition holds for the gradients of the original features (as it happens in this case; see Proposition 3.1 and the note about the special case $\hat{\mathcal{S}}_1 = \mathcal{S}_1$). Although this particular condition on the first eight pairs of gradients is difficult to appreciate in Fig. 8(a) and (d), where deviation from orthogonality seems approximately zero for all pairs of features using white Gaussian samples, we measured a RMS value of the deviation from orthogonality for the first eight couples of 1.6×10^{-5} degrees. This is a negligible numerical error exclusively due to the numerical computation of the gradient (see Eq. (33)). For the rest of feature couples we obtained an RMS of 0.42 degrees, still small, but four orders of magnitude larger. Aside from the first eight pairs of features, the excellent practical decoupling of the other ones has been favored in this case by using a random distribution (zero-mean uni-variate white Gaussian noise) for the samples, with features whose expected values are precisely the reference values used in the NeN decoupling algorithm. As explained in the theory (Subsection 3.1.3), exact decoupling is also obtained for samples on the reference manifolds, even if Frobenius condition does not hold for the output features. For large enough samples, sample feature values do not deviate much from their expected values, and, as a consequence of the features' smoothness, the gradients of these samples will also be close to orthogonal. To test how the decoupling quality degrades when using real samples instead of pseudo-random ones, we have tested the method, again, with the referred collection of 112 textured 23×23 pixel patches. We first note that the amount of mutual coupling between features is almost exactly the same as for the white noise case, a relevant fact that adds support to the assumptions made in Section 5: the angle between features is fairly constant for each couple of features, both within a given distribution/collection (see the relatively small amplitude of error bars) and also across different distributions/collections (compare panel (a) with (b) and (c) with (e)). We can say that, in this case, it is especially accurate to say that the local covariance matrix is fairly independent of $\mathbf{x}$, behaving the Jacobian pretty close to a moving frame. As for the decoupling quality for this collection of real photographic patches, we now observe than in about half of the pairs the decoupling is either perfect (first eight couples) or almost perfect. For the other half, DO is still very moderate and much smaller than for the original pairs, in the great majority of cases. In Table 3 we can see that, for real photographic images, the average absolute DO has been reduced in a factor 6, approx., whereas for Gaussian samples it has been reduced 65 times.

TABLE 3: Average ($\pm\sigma$) absolute deviation from orthogonality (|DO|), in degrees, between feature's gradients for the MSM, VF, DF_{MSM} and DF_{VF} sets of features.

	Gaussian	Textures
MSM	32±26	59±21
DF_{MSM}	2±3	9±17
VF	13±10	12±11
DF_{VF}	0.2±0.3	2±3

6.3 Statistical regression

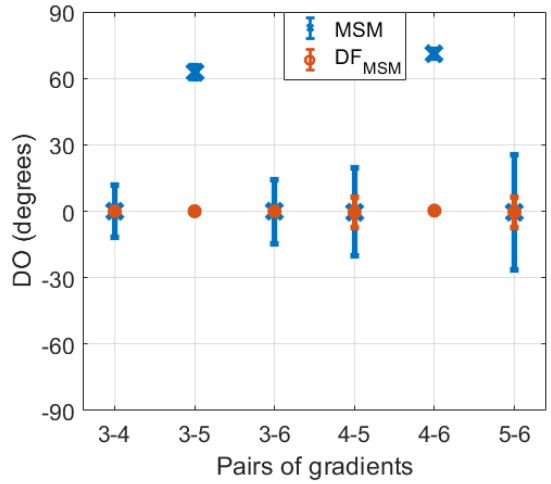
In this section we focus on the estimation of the parameters of a distribution that best describe a dataset[9]. Different approaches have been proposed in the literature for different families of parametric distributions, such as the classical maximum likelihood estimation or the method of moments. We approached the estimation task as a regression problem, where the parameters of the distribution are estimated from a set of global features obtained from the observed data. Specifically, we tested, as global features: (i) a set of classical MSM, and (ii) the corresponding decoupled set (DF_{MSM}). In order to thoroughly compare the descriptive capabilities of both sets of features, we used several regression methods, namely: linear regression models (LRM), regression trees (RT), support vector regression (SVR), Gaussian process regression (GPR), ensembles of trees (ET) and neural networks (NNR). All of these methods are implemented in the Regression Learner App, ®Matlab. For reproducibility purposes, we used the hyper-parameters set by default in the referred app. Specific information about implementation, hyper-parameters selection and methodological details can be found in [36].

In our experiments we used different statistical distributions, specifically: generalized Gaussian distribution (GGD), Gamma distribution (GMD), and absolute value of a Normal distribution raised to a positive number (GND). The shape of these distributions depends on a shape parameter (β), and the regression problem consists in estimating β from an observed data set. See Appendix D for the expressions of the probability density functions of these distributions and their dependence with β.

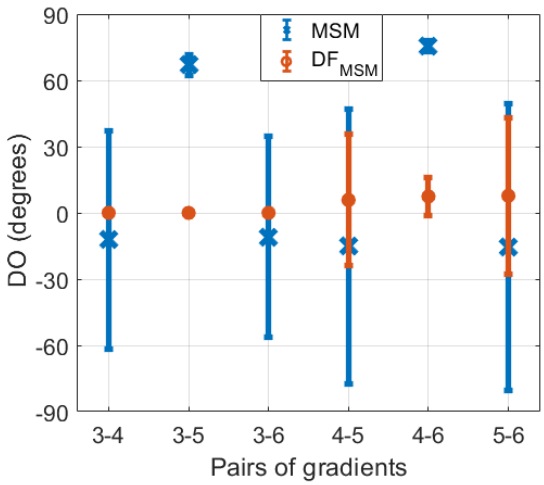
Let $\mathbf{x}$ represent an N-D vector of i.i.d. samples drawn from a GGD, GMD or GND distributions (we generated the samples following [37], [38] for the GGD, and [39] for the GMD, using the ®Matlab function *gamrnd.m*), normalized to have zero mean and unitary variance. For the GGD and GMD cases, given that the kurtosis of these distributions changes very fast for small values of the β parameter, we defined $\beta = 2^A$ and sampled uniformly the exponent A in $[-3, 3]$, resulting in β ranging in $[1/8, 8]$. In this way, we obtained a quite uniform distribution of kurtosis values. In the GND case, β was sampled uniformly in the range $[1, 6]$.

MSM features and their corresponding decoupled features, DF_{MSM}, were obtained from the third (skewness) to the sixth order. This led to two sets of 4 dimensional predictors $\{f_j(\mathbf{x}), j = 3, \dots, 6\}$ and $\{\hat{f}_j(\mathbf{x}), j = 3, \dots, 6\}$ for each parameter β. We compared the β prediction accuracy of these two sets. We generated $d = 2048$ vectors $\mathbf{x}$

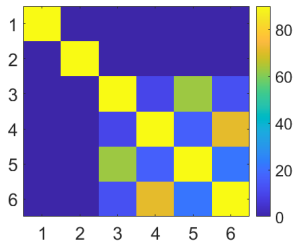
9. Some results of this subsection have been presented in [7].



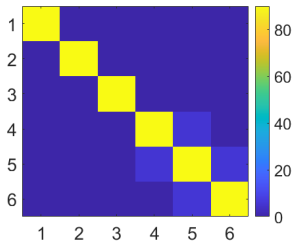
(a) DO between different pairs of gradients. Gaussian case.



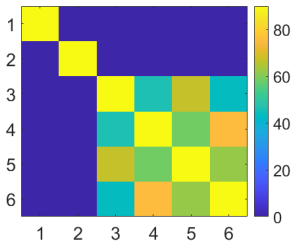
(b) DO between different pairs of gradients. Textures case.



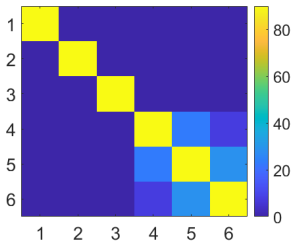
(c) MSM set. Gaussian case.



(d) DF_{MSM} set. Gaussian case.



(e) MSM set. Textures case.



(f) DF_{MSM}. Textures case.

Fig. 8: Deviation from orthogonality (DO), in degrees, for the different pairs of gradient's features for the standardized marginal moments (MSM) and the corresponding decoupled sets (DF_{MSM}). (a), (c) and (d) show results for the Gaussian case, $N = 512$. (b), (e) and (f) show results for the Textures case, $N = 529$ (23×23 pixels). In (a)-(b) positive values indicate acute angle and negative obtuse. In (c-f) the color level represents the average of the absolute value of DO.

of different lengths N of i.i.d. samples, thus having 2048 pairs of predictors ($\{f_j\}$ or $\{\hat{f}_j(\mathbf{x})\}$) and targets (known β values). We averaged 100 5-fold cross-validation runs to measure the accuracy of the methods in terms of the RMSE in the estimation of the exponent $A = \log_2 \beta$ in the GGD and GMD cases, and the estimation of β, in the GND case.

Figure 10 shows the RMSE of the results as a function of N for the MSM and DF_{MSM} sets of descriptors, by using the NNR regression method, the method providing the best results for almost all the tested cases (see Appendix D for more detailed results). Figure 10(a) shows the results for GGD, Figure 10(b) for GMD and Figure 10(c) for GND. The shadow area represents the standard deviation across 100 repetitions of the experiment. The proposed DF_{MSM} clearly outperformed MSM for all the tested distributions and sample sizes. For instance, in the GGD case, using the proposed DF_{MSM} the RMSE was reduced by factors of 0.50, 0.54, 0.63, 0.68, 0.76, and 0.86, for $N = 2048, 1024, 512, 256, 128$, and 64, respectively.

Tables 6, 7 and 8 in Appendix D show the RMSE obtained using MSM and DF descriptors for the different sample sizes N and regression methods, for the GGD, GMD and GND distributions, respectively. The best regression method for each N is highlighted in bold. Our decoupled descriptors DF_{MSM} outperformed the MSM across all the compared regression methods and sample sizes in 105 out of 108 cases (97.2%), which shows the robustness and generalization ability of our approach.

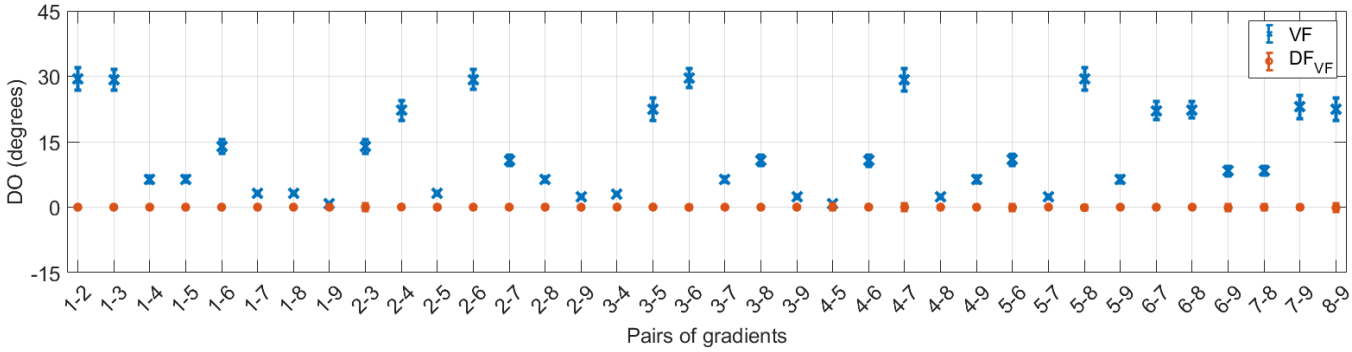
6.4 Texture classification

In this section we apply the proposed method for texture classification in two different settings: (i) comparing standardized moments (MSM) with their corresponding decoupled features DF_{MSM}, both applied to a set of subbands, the output of a filter bank (Section 6.4.1); (ii) using features that directly are defined as marginal moments at the output of a filter bank (MF), and their corresponding fully decoupled features, DF_{MF} (Section 6.4.2).

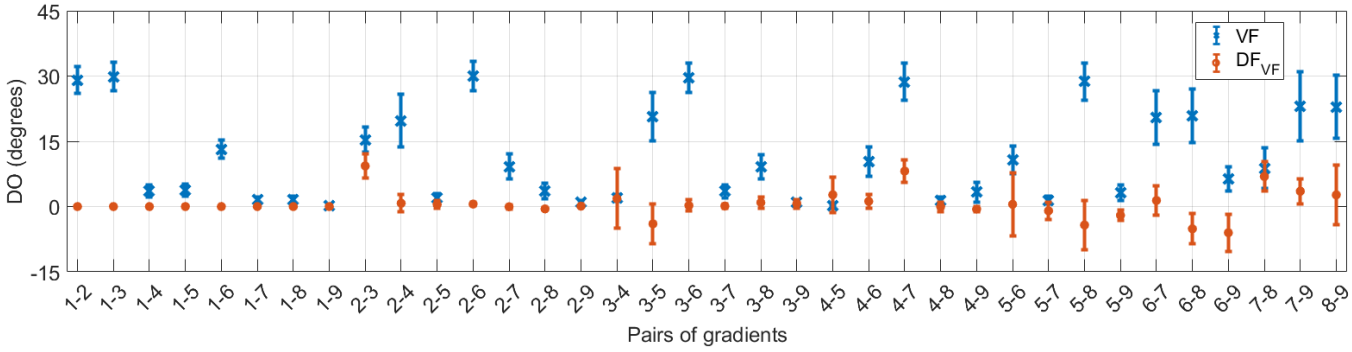
6.4.1 Texture classification based on marginally decoupled moments in a filter bank

Here[10] we compare the performance of two classifiers using features derived (i) from MSM of order 2nd to 10th, (ii) from a generalized form of Ref. [16], and (iii) from the corresponding decoupled features DF_{MSM}, all of them at the output of subbands of the TILs representation [33]. We selected 54 textures of 640×640 pixels from the Broadtz [35] database under the criterion of looking homogeneous in 64×64 pixel

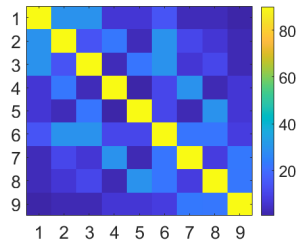
10. This subsection is a summary of the results published in [6].



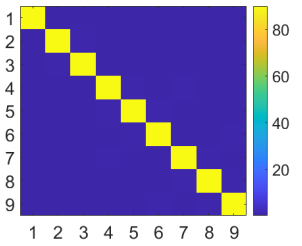
(a) DO between different pairs of gradients. White Gaussian noise case.



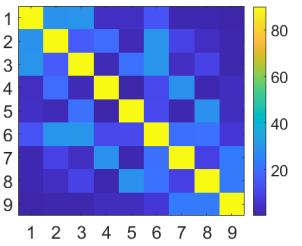
(b) DO between different pairs of gradients. Textures case.



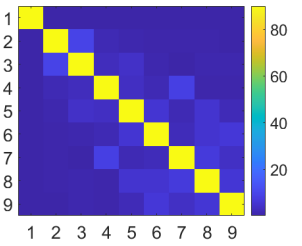
(c) VF, White Gaussian case.



(d) DF_{VF}, White Gaussian case.



(e) VF, Textures case.



(f) DF_{VF}, Textures case.

Fig. 9: Deviation from orthogonality (DO), in degrees, for the different pairs of gradient's features for the second-order moment at the output of a filter bank (VF), and the corresponding decoupled set (DF_{VF}). We used $N = 529$ (23×23 pixels). (a), (c) and (d) show results for the Gaussian case. (b), (e) and (f) show results for the Textures case. In (a)-(b) positive values indicate acute angle and negative obtuse. In (c-f) the color level represents the average of the absolute value of DO.

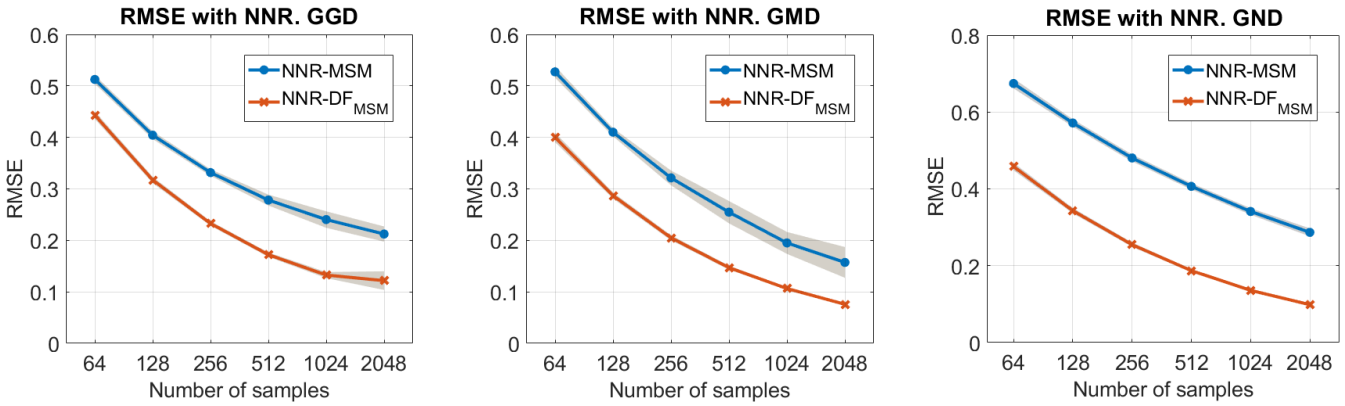
patches[11]. Every texture was divided into 10×10 disjoint 64×64 patches. The problem consisted in classifying patches in their corresponding textures. To extract the features we first applied the TILs representation [33] and discarded the low-pass band (having 8 subbands). Then, for every subband of every patch, three sets of features were obtained: (i) classical MSM features $\{f_j(\mathbf{x}), j = 2, \ldots, 10\}$ (something commonly used to characterize textures, but to a lower order, see e.g. [40]); (ii) modified moment set: same as MSM, except that now even order moments are shift-minimized ($\tilde{\mu}_n = \min_{\alpha_n} \mathbb{E}\{(\hat{x} - \alpha_n)^n\}$, $\hat{x}$ being the standardized observation) [16]; and (iii) proposed marginally decoupled moments DF_{MSM} $\{\hat{f}_j(\mathbf{x}), j = 2, \ldots, 10\}$.

In order to quantify the redundancy between features, we estimated the mutual information (MI) between pairs of features [41]. Some mean values across subbands are shown in Table 4. We see that, except for the (3,4)-th case, the proposed DF_{MSM} features present a drastic reduction of redundancy compared to the classical MSM and modified [16] ones. We recall the reader that the aim of using decoupled features is not to compensate for the statistics of the data (to which the decoupling method is totally transparent), like non-linear ICA aimes for, but rather to avoid adding spurious coupling in the processed data, leaving only the dependencies that are effectively caused by the input data statistics. So, perfectly decoupled features will generally reduce the mutual information among features,

TABLE 4: Averaged mutual information.

Features	(3,4)	(3,5)	(4,5)	(3,6)	(4,6)	(5,6)
MSM	0.26	0.61	1.39	0.55	3.10	3.11
Mod. [16]	0.23	0.61	1.39	0.55	3.07	3.03
DF_{MSM}	0.22	0.04	0.08	0.09	0.54	0.09

11. The list of selected textures and the ®Matlab code of the experiments is available in https://www.researchgate.net/project/Nested-Normalizations-for-Decoupling-Global-Features.



(a) GGD distribution (b) GMD distribution (c) GND distribution

Fig. 10: RMSE of the estimated parameters as a function of the number of samples N for the NNR regression method applied to three density functions (see text for details).

but will not necessarily remove it, as that depends on the data statistics (see, e.g., what happened in the example of Fig. 7). Figure 11 shows a 3-D subset of the MSM (left) and proposed DF_{MSM} (right) features (i.e., before and after our decoupling method) for two texture classes, to illustrate the decoupling effect on the data distribution in the feature space. Shown data correspond to a single subband (number 5 in the representation), of textures D103 and D111, selecting just 3 features for the MSM ($f_3(\mathbf{x})$, $f_4(\mathbf{x})$ and $f_5(\mathbf{x})$[12]) and the proposed DF_{MSM} ($\hat{f}_3(\mathbf{x})$, $\hat{f}_4(\mathbf{x})$ and $\hat{f}_5(\mathbf{x})$) sets. We include the projections onto the three orthogonal planes. The decoupling between every pair of features is apparent, specially for the (3,5)-th and (4-5)-th order cases.

We used two classifiers: a naïve (univariate) Gaussian and a parameter-optimized Support Vector Machine (SVM) using Radial Basis Functions[13]. We applied cross validation with 4 folds, and averaged 8 runs for each result. Figure 12 shows the test classification results as a function of the order of the moments included in the feature's set for the three compared sets and the two classifiers (see legend). We observe a totally different behavior between using classical MSM and modified [16] moments, vs. the marginally decoupled ones DF_{MSM}: whereas the former achieve their optima when using only variance ($f_2(\mathbf{x})$), skewness ($f_3(\mathbf{x})$) and kurtosis ($f_4(\mathbf{x})$), roughly achieving 3% and 2% error ratios for naïve and SVM, respectively, the latter keep on decreasing the error when adding higher-order features, reaching 1.34% and 0.86%, respectively, for $n = 10$. We see how [16] produces just a marginal improvement. It is also very significant how, for MSM, the naïve method behaves very differently from SVM, whereas, for DF_{MSM}, results of SVM and naïve classifiers run in close parallel. We believe this is due to the strong assumption made by the naïve method (namely, that features are mutually independent), which holds approximately true *after* the feature decoupling (as shown in Table 3), but not before.

12. Note that $f_3(\mathbf{x})$ and $f_4(\mathbf{x})$ are equal to the classical definition of sample skewness and kurtosis, respectively.

13. We thank the authors of the Pattern Recognition Toolbox (http://covartech.github.io), which we used in preliminary experiments.

TABLE 5: Texture patch classification, 2nd experiment. Test classification error (%).

	MSM	DF_{MSM} [6]	DF_{MF}
Naïve Bayes	9.1 ± 0.4	7.4 ± 0.3	4.0 ± 0.3
SVM	5.2 ± 0.5	4.5 ± 0.4	3.1 ± 0.4

6.4.2 *Texture classification based on jointly decoupled moments in a filter bank*

In these experiments we compared the performance of a classifier trained using three different sets of features, namely: (i) MSM measured at the output of a filter bank; (ii) the decoupled standardized moments DF_{MSM} (same as in the previous experiment), without considering the coupling caused by the filter bank; and (iii) fully decoupled features, DF_{MF}, using as original features marginal moments at the outputs of the filter bank. The three sets were obtained from order 2 to 6. The first 20 textures from the Broadtz database [35] were taken. The upper left quarter of every texture was normalized and divided into 25 disjoint, 64×64 pixel, patches. The problem consisted of classifying these patches into one of the 20 texture classes. To extract the features that characterized every patch we first applied the TILs representation, using the 9 subbands (thus including the low-pass residual, not used in previous section). Then, for every subband, the above described sets of 45 features (5 moments $\times$ 9 subbands) were calculated. We used two classifiers: a Naïve Bayes and a parameter-optimized Support Vector Machine (SVM) using Gaussian kernels [42]. We applied cross validation with 10 folds, and averaged 200 runs for each result. Table 5 shows the mean $\pm$ standard deviation test classification error across runs when using the MSM (1st column), the partially decoupled DF_{MSM} [6] (2nd column), and the fully decoupled features DF_{MF} (3rd column). First row shows results using Naïve Bayes, and second using SVM. We observed that the probability of error is greatly reduced when using DF_{MSM} in comparison with MSM. In addition, fully decoupled features DF_{MF} also significantly improved the results of the partially decoupled set [6]. Using the Naïve Bayes classifier, we see that the error obtained by the fully decoupled set DF_{MF} decreased the error in a factor of 0.54 and 0.44 w.r.t. DF_{MSM} and original MSM sets, respectively. We also did the experiment of incrementally

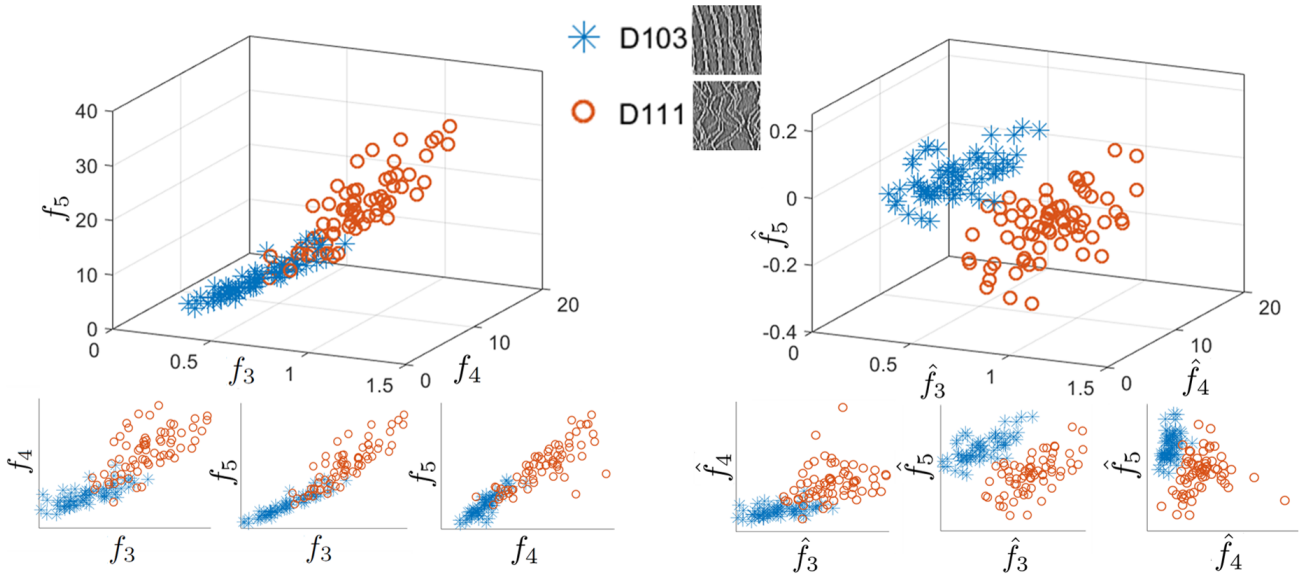


Fig. 11: Illustrating the decoupling effect in the feature space on two classes of texture patches. Left: Using classical MSM features; Right: Using proposed DF_{MSM} features.

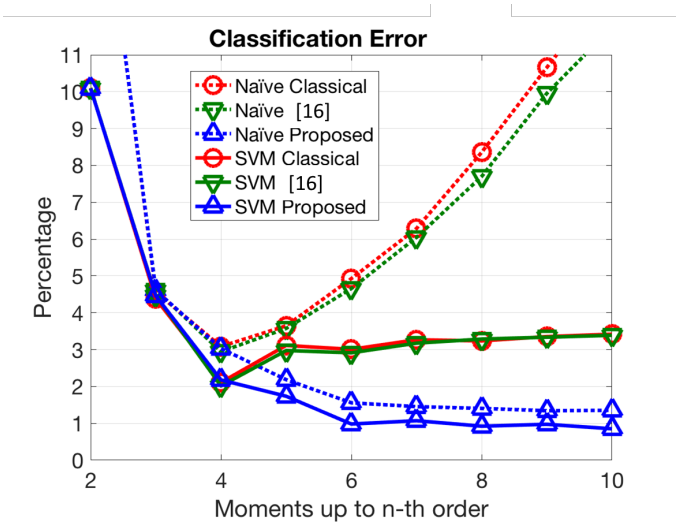


Fig. 12: Texture patch classification, 1st experiment. Test classification error (%) as a function of the order of the moments included in the feature's set.

increasing the highest order in the set of features, from just the variance to including all orders from 2 to 6. Significantly, the error was minimized when including high order DF_{MF}, up to 6, while for the MSM case the minimum error was obtained when including features up to the fourth order (kurtosis). This result is consistent with the one reported in Fig.12.

It is especially relevant, at this point, to recall that the MF set (for orders above 2) does not fulfill Frobenius condition (as mentioned in Section 4). This implies that decoupling solutions do not even exist within the theoretical framework presented here. Nevertheless, Algorithm 3 (NeN, narrow path) provides a well-defined transformation that, as empirically shown in Section 6.2.2 for the case of the second-order moments at the output of a filter bank, greatly reduces the amount of coupling between the features' gradients. This translates here into a substantial performance boost when applied to texture classification.

7 Conclusions and Future Work

We have presented a new mathematical and algorithmic framework for, given a set of differentiable functions acting as global data descriptors, obtaining a closely related set such that its gradients are mutually orthogonal. We have set the conditions under which this can be done hierarchically and progressively, adding a new feature at a time, and devised a new family of algorithms based on nested normalization operations. We have also studied the need of adding small perturbations in some cases, and devised some specific methods for that.

We have shown, first, that the proposed method allows for locally decorrelating features of statistical distributions, and why this has a positive impact in discriminating close values of statistical parameters. We have also tested empirically (with both real textured image patches and pseudo-random numbers, for some distributions) the degree of accomplishment of mutual decoupling (gradient orthogonality) for different global features, obtaining results that are both practically interesting and consistent with the theory.

We have applied our decoupling methods to marginal moments, both in the pixel domain and at the output of a filter bank. Using the new decoupled features as descriptors with state-of-the-art machine learning methods we have obtained a dramatic decrease in error in regression problems (over simulated random variables, under three different distributions) and classification (over real textured image patches), as compared to using classical standardized moments. It is worth noting that we obtained substantial improvements in classification accuracy even when theoretical conditions for perfect decoupling were not met.

It is noteworthy that, applying our decoupling method to the first three raw moments results into their standardized counterparts. In addition, we obtained for the first time, a (quasi-analytic) normalized fourth-order moment that is decoupled from mean, variance, and skewness, which we have termed *orthokurtosis*. For higher orders, we obtained other new well-defined features, numerically computable, which present a much-decreased amount of coupling. Finally, although not fully developed here, natural application fields for our method go beyond analysis and include promising style transfer and synthesis techniques [2], [28].

In future work we want to explore the extension of the deterministic decoupling methods for non-hierarchical sets, as well as its applicability for improving economy and robustness in ANNs, by decoupling their features, either at training phase or *a posteriori*. To conclude, a promising research area is that of exploring actual biological mechanisms of adaptation involving decoupling/cross-invariance, in perceptual neural science.

APPENDIX A
EQUILIBRIUM POINTS OF GRADIENT SYSTEMS

In this section we study equilibrium points for the gradient system (4).

If f is continuous we must have $f(\overline{\Omega}) = [a, b]$ (here a, b maybe $\pm\infty$ if the set is not bounded, but the analysis can be performed similarly). We would like to see if, given $x_0 \in \Omega$, we can reach all values in the range of f by moving in the direction of the gradient. We look at stationary points in $\overline{\Omega}$ that are not a global maximum or minimum points. The basin (or domain) of attraction of an equilibrium point $\bar{\mathbf{x}}$ is the set of all initial conditions with solutions that tend to it. Let us consider all possible cases according to the behavior of the linearized equation $\dot{\mathbf{x}} = -D^2 f(\bar{\mathbf{x}})\mathbf{x}$ (see, for instance, the introduction in [43]). Assume that $D^2 f$ has no zero eigenvalues:

- If the eigenvalues of the Hessian $D^2 f$ are all strictly positive then $\bar{\mathbf{x}}$ is a sink.
- Similarly for max.
- If the Hessian has a negative real eigenvalue then the equilibrium is unstable. If all eigenvalues are nonzero, then the dimension of the unstable manifold is equal to the number of negative eigenvalues counting multiplicity. The dimension of the unstable manifold is the number of negative eigenvalues counting multiplicity. The tangent spaces of these manifolds are the spans of the corresponding eigenspaces so are orthogonal at the equilibrium point.

If some eigenvalue is zero, then the picture is much more complicated and we will make additional assumptions in order to avoid technicalities:

- If $\bar{x}$ is a local minimum of f (not necessarily strict), then $\bar{x}$ is a Lyapunov-stable equilibrium point of the gradient flow of a real-analytic function f [44, Section 3]. Thus we pose the additional constraint that all local maxima and minima are also global.
- If $\mathbf{x}$ is a degenerate saddle, we assume that its basin of attraction has a lower dimension and thus, saddles can be avoided by introducing a suitable perturbation. We will actually assume that the union of all basins of attraction, denoted by Λ, is also lower dimensional.

APPENDIX B
FROBENIUS THEOREM

Let $\mathcal{S} = \{f_i : \Omega \to \mathbb{R}, i = 1 \dots M\}$ be a set of features. Fix p a positive integer, $p \leq M$. A p-dimensional *distribution* $\mathcal{D}$ in Ω is a (smooth) choice of a p-dimensional subspace of the linear span of the $\{\nabla f_i(\mathbf{x})\}_{i=1}^M$ for every point in $\mathbf{x} \in \Omega$. We denote the plane at $\mathbf{x}$ by $\mathcal{D}(\mathbf{x})$. We say that the set of features $\mathcal{S}$ is of *maximal rank* in Ω if the gradients are linearly independent at every point $\mathbf{x}$ so that the distribution spanned by the gradients has $p = M$. This implies, in particular, that we must choose Ω not containing critical points of the f_i.

We say that $\mathcal{D}$ satisfies the *Frobenius condition* if for every $f, g \in \mathcal{S}$, writing the gradient vectors as

$$X = \sum_\ell \partial_{x_\ell} f \frac{\partial}{\partial x_\ell}, \quad Y = \sum_j \partial_{x_j} g \frac{\partial}{\partial x_j}, \tag{34}$$

we have that the combination

$$XY - YX = \sum_{j,\ell} \left(\partial_{x_\ell} f \partial_{x_\ell x_j} g - \partial_{x_\ell} g \partial_{x_j x_\ell} f\right) \frac{\partial}{\partial x_j} \tag{35}$$

also belongs to $\mathcal{D}$.

The classical Frobenius theorem [45, Chapter 6] states that a distribution $\mathcal{D}$ that satisfies the Frobenius condition in Ω is integrable. That is, one can find a submanifold $\mathcal{I}(\mathbf{x}_0, \mathcal{S})$ passing by $\mathbf{x}_0$ whose tangent hyperplane at each location $\mathbf{x}$ is the linear span of $\{\nabla f_i(\mathbf{x}), i = 1 \dots M\}$. In addition, this submanifold is p-dimensional, and Ω is foliated by these submanifolds. A different presentation of Frobenius theorem from the dynamical systems point of view can be found in [46, Chapter VI].

B.1 Back to Proposition 2.2

We give now the proof of the *only if* statement in Proposition 2.2. Assume that, given set of features $\mathcal{S}$, there exists an invariant mapping $\hat{\mathbf{x}}_\mathcal{S}$ exists, and denote by $J := J_{\hat{\mathbf{x}}_\mathcal{S}}$ its Jacobian matrix. We will show that the gradients of those features in $\mathcal{S}$ satisfy the Frobenius condition C2.

For this, it is enough to pick two any two features in $\mathcal{S}$, say f and g. By the definition of invariant mapping, we must have

$$J \cdot \nabla f = \mathbf{0},$$
$$J \cdot \nabla g = \mathbf{0},$$

at each point in the domain (which is not written in to simplify the notation). Differentiating both equations w.r.t. the ℓ-th coordinate, component by component, we obtain

$$\partial_{x_\ell} J \cdot \nabla f + J \cdot \partial_{x_\ell} \nabla f = \mathbf{0},$$
$$\partial_{x_\ell} J \cdot \nabla g + J \cdot \partial_{x_\ell} \nabla g = \mathbf{0},$$

from where we can isolate the second derivatives of each feature

$$\partial_{x_\ell} \nabla f = -J^{-1} \cdot \partial_{x_\ell} J \cdot \nabla f,$$
$$\partial_{x_\ell} \nabla g = -J^{-1} \cdot \partial_{x_\ell} J \cdot \nabla g.$$

Now we extract the j-th component of both vectors above, to have a formula for $\partial_{x_\ell x_j} f$ and $\partial_{x_\ell x_j} g$. This is,

$$\partial_{x_\ell x_j} f = -\sum_{k,s} (J^{-1})_{jk} \cdot (\partial_{x_\ell} J)_{ks} \cdot \partial_{x_s} f,$$
$$\partial_{x_\ell x_j} g = -\sum_{k,s} (J^{-1})_{jk} \cdot (\partial_{x_\ell} J)_{ks} \cdot \partial_{x_s} g.$$

Next, in order to check Frobenius condition (35) we need to calculate

$$\begin{aligned}\sum_\ell &(\partial_{x_\ell} f \partial_{x_\ell x_j} g - \partial_{x_\ell} g \partial_{x_j x_\ell} f) \\ &= -\sum_{\ell,k,s} \Big[\partial_{x_\ell} f \cdot (J^{-1})_{jk} \cdot (\partial_{x_\ell} J)_{ks} \cdot \partial_{x_s} g \\ &\qquad - \partial_{x_\ell} g \cdot (J^{-1})_{jk} \cdot (\partial_{x_\ell} J)_{ks} \cdot \partial_{x_s} f\Big].\end{aligned}$$

If one is able to interchange the indexes s and ℓ, the above quantity is identically zero. And this is possible since, for a Jacobian matrix,

$$\partial_{x_\ell} J_{ks} = \partial_{x_\ell} \partial_{x_s} (\hat{\mathbf{x}}_{\mathcal{S}})_k = \partial_{x_s} \partial_{x_\ell} (\hat{\mathbf{x}}_{\mathcal{S}})_k = \partial_{x_s} J_{k\ell}.$$

We have shown that Frobenius condition (35) holds for any two features in the set $\mathcal{S}$, which completes the proof.

B.2 Examples

B.2.1 Average of scalar functions

If the set $\mathcal{S}$ consists on features of the form

$$f_j(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} m_j(x_n), \quad j = 1..M,$$

for scalar functions $m_j : \mathbb{R} \to \mathbb{R}$, then it clearly satisfies the Frobenius condition. This follows by simple inspection of (35), because the Hessians $\partial_{x_\ell x_j} f_i$ are multiples of the identity.

B.2.2 Second-order moments at the output of a set of filters

Another example of features that satisfy Frobenius is a set of second-order moments measured at the output of a set of filters. To fix notation, we take two functions

$$f(\mathbf{x}) = \frac{1}{N} (\mathbf{x} * h)^{\odot 2}, \quad g(\mathbf{x}) = \frac{1}{N} (\mathbf{x} * h')^{\odot 2}.$$

Calculate

$$\partial_{x_i} f = \sum_k (\mathbf{x} * h)_k h_{k-i} = \frac{2}{N} (\mathbf{x} * (h * \tilde{h}))_i = \frac{2}{N} (\mathbf{x} * \hat{h})_i,$$
$$\partial_{x_i x_j} f = \hat{h}_{i-j},$$

where we have denoted $(\tilde{h})_s = h_{i-s}$, $\hat{h} = h * \tilde{h}$, and similarly for h'. Then, in the notation of (34), dropping the multiplicative constants,

$$\begin{aligned}XY - YX &= \sum_{i,j,s} x_s \hat{h}_{i-s} \hat{h}'_{i-j} \frac{\partial}{\partial x_j} - \sum_{i,j,s} x_s \hat{h}'_{i-s} \hat{h}_{i-j} \frac{\partial}{\partial x_j} \\ &= \sum_j \Big\{[(\mathbf{x} * \hat{h}) * \hat{h}']_j - [(\mathbf{x} * \hat{h}') * \hat{h}]_j\Big\} \frac{\partial}{\partial x_j} = 0,\end{aligned}$$

as desired.

APPENDIX C
ADDING PERTURBATIONS

C.1 Marginal moments

C.1.1 Critical points of the decoupled moments and their basins of attraction

In order to apply the methods proposed in Section 3, conditions B1 (no local non-global extrema) and B2 (lower dimensional basins for saddles) must hold. Here we test this, and search for explicit formulas for all critical points of the new decoupled features. In particular, this is essential to understand the structure of basins of attraction and discuss the effectiveness of perturbations.

From the method in Subsection 3.3.1, obtaining the critical points of the decoupled features is sort of dual to finding the decoupled gradients within their corresponding reference manifolds. Whereas for the latter we imposed orthogonality on each new gradient with respect to the previous ones (see Subsection 4.1.2), for finding the critical points we impose co-linearity on each gradient with respect to the previous ones, see Eq. (18). A particular solution corresponds to finding common critical points (through order k) of the original features, where all gradients for $j \leq k$ vanish. In this case there are no local (non-global) extrema, as gradients are made of monomials, which either have a single minimum at zero, for even orders, or a saddle point at zero and no extrema in $\mathbb{R}$ (odd orders). Therefore, the only solution coming from the original features' gradients vanishing corresponds to $\mathbf{x}_0^* = \mathbf{0}$, i.e., at zero all decoupled moments have a critical point, same as the original moments.

For the rest of solutions, we start by solving for the critical points of $\hat{f}_2$, denoted by $\mathbf{x}_1^*$ [14], in the equation $\nabla f_2(\mathbf{x}_1^*) = \lambda_{1,2} f_1(\mathbf{x}_1^*)$, which, substituting its corresponding expressions, gives us $\mathbf{x}_1^* = c\mathbf{1}$, $c \in \mathbb{R}$. No surprisingly, constant signals provide the minimal (zero) variance, the only extreme value of this feature. Moreover, as explained in Subsection 3.3.1, decoupled features are not defined at $\mathbf{x}_1^*$ (being the sample variance null it can not be normalized to one). This manifold only intersects $\mathcal{R}_1$ in $\mathbf{x} = \mathbf{0}$, and has no intersections with $\mathcal{R}_2$ and subsequent. Next, the calculation of the critical points of $\hat{f}_3$ comes from solving for $\mathbf{x}_2^*$ in $f_3(\mathbf{x}_2^*) = \lambda_{1,3} f_1(\mathbf{x}_2^*) + \lambda_{2,3} f_2(\mathbf{x}_2^*)$. This is a quadratic equation, whose solution is a vector $\mathbf{x}_2^*$ made of only two arbitrary values, repeated in arbitrary proportions (p and $1-p$), for all the coefficients. If we impose, in addition, $f_1(\mathbf{x}_2^*) = 0$ and $f_2(\mathbf{x}_2^*) = 1$ (conditions of $\mathbf{x}_2^* \in \mathcal{R}_2$), we obtain, after some operations, that the only possible values of the coefficients, for $\mathbf{x}_2^*$ in $\mathcal{R}_2$, are:

$$\begin{aligned}x_{2,1}^* &= \sqrt{p}/(1-p) \\ x_{2,2}^* &= -\sqrt{1-p}/p,\end{aligned} \tag{36}$$

where $0 < p < 1$ denotes the proportion of the first value. From this we can easily compute the skewness of $\mathbf{x}_2^*$, that only depends on p, not on the particular values: $\hat{f}_3(\mathbf{x}_2^*) = (2p-1)/\sqrt{p(1-p)}$. Although the previous expression has no extrema for a continuous p, given the discrete nature of

14. Although for notation simplicity critical points here are denoted as a vector $\mathbf{x}_j^*$, they actually represent sets of vectors.

p for discrete signals (from $1/N$ to $(N-1)/N$) we obtain that the maximal and minimal skewness are produced when p is either $1/N$ or $(N-1)/N$, that is, when the vector is constant except for one coefficient. The rest of the values of p produce the vast majority of the critical points of $\hat{f}_3$, which are saddles. There are no local (non-global) extrema (condition B1). These critical points have subsequent undefined decoupled moments (see Subsection 3.3.1), with the only exception of the case $p = 1/2$ (possible just when N is even), the only intersection of $\mathbf{x}_2^*$ with $\mathcal{R}_3$, which, having already zero skewness, its (minimal) kurtosis is also its orthokurtosis. One can see that the skewness of a bivaluated vector can not be adjusted by applying a coefficient-wise reversible (monotonous) non-linearity, because the skewness only depends on p.

For $k > 3$ we apply the same procedure for obtaining the critical points of $\hat{f}_k$ in $\mathcal{R}_{k-1}$, namely, solving an algebraic equation of degree $k-1$, and obtaining any combination of distinct $k-1$ solutions for the values in the vector. Same as for the previous case only the $k-2$ distinct proportions ($k-1$ in total, adding up to 1) of each of these $k-1$ different values matter for the computation of $\hat{f}_k$. And, again, the vast majority of these critical points are saddles, presenting no local (non-global) extrema (condition B1). For instancce, in the case of $k = 4$, critical points have three different distinct values or less, the minimal orthokurtosis point being a degenerate case (having two single values with $p = 1/2$, as mentioned above), whereas the maximal orthokurtosis is produced for $p_1 = (N-2)/N$, $p_2 = 1/N$, corresponding to all pixels having the same value, except for two pixels, now having each of these two a different value from the dominant and from each other. These vectors can be adjusted to have the desired mean, variance and skewness, but, having only three distinct values, we run out of degrees of freedom to adjust their orthokurtosis too. Therefore, except for the case when the orthokurthosis is already three (its reference value), $\hat{\mathbf{x}}_4(\mathbf{x}_3^*)$ does not exist, and, as a consequence, the fifth order and subsequent decoupled moment are not defined at these points. Note also that the range of the decoupled features generally change with respect to their original counterparts. Whereas the second-order moment range does not change when decoupled (it is still $\mathbb{R}^+$), the fourth-order (skewness) is constrained from $\mathbb{R}$ to $[-\frac{N-2}{\sqrt{N-1}}, \frac{N-2}{\sqrt{N-1}}]$, and the fourth order (orthokurtosis) from $\mathbb{R}^+$ to $[1, N/2]$. Finally, we note that the existence of higher-order decoupled moments also depends on having a large enough number of samples, N, In particular, the last result implies that, for $N < 6$, there are no decoupled moment of higher-than-four orders, as the reference value for normalizing the orthokurtosis, 3, is not reachable within its valid range. This limitation also comes from the already explained requirement of having enough distinct values for the samples.

Whereas a typical vector will have more than a few distinct quantization values, and, thus, it will not produce critical points for the first few decoupled moments, a different situation is created by the basins of attraction of the saddles, which, as pointed out above, constitute the vast majority of the critical points. It is easy to informally check that any vector having coefficients with repeated maximal (or minimal, for odd orders) values lies within the basin of attraction of a saddle. To illustrate this, let us consider a gray-level image in the range $[0, 255]$, with two pixels having the 255 value. As we increase its fourth-order decoupled moment (the skewness) staying in $\mathcal{R}_2$, these two values are going to grow at exactly the same pace, relatively to the rest of the coefficients, that, due to the normalization, will get relatively lower and progressively closer to each other. If we keep on increasing the skewness we would approach, in the limit, to an image made of all pixels sharing the same value, except for two pixels, both sharing another value. This mental experiment shows how our original image, having more than one pixel with the maximal value, lies in Λ, i.e., in the basin of attraction of a saddle. This situation, unless avoided by adding a proper perturbation, provokes the algorithm to eventually get stuck in the saddle, thus not letting the vector to be adjusted along its full range (as shown before, the maximal/minimal skewness is achieved when all samples except for one have the same value). The desired effect of a perturbation is to "break the tie", thus allowing the gradient to further advance towards the absolute extrema of the feature. An analogous reasoning can be applied for higher-order moments.

Then, how likely is that a vector belongs to Λ? The answer, dealing with vectors having regular density distributions in $\mathbb{R}^N$ is: zero probability (as the basins of attraction of these saddles are lower-than-N dimensional, condition B1). However, the answer for digital (discrete quantized samples) is totally different: potentially *very* likely. Let us assume a vector having N samples quantized in Q levels. Then, the probability of that a particular value v (e.g., the highest) is repeated, assuming a uniform and independent distribution for each of the vector coefficients is given by a binomial distribution:[15] $P(n(v) > 1) = 1 - (1 - 1/Q)^N - N/Q(1-1/Q)^{N-1}$, $n(v)$ being the number of occurrences of v. For instance, for a very small size image of 64×64 pixels ($N = 4096$ - for larger images it gets worse) with pixels ranging from 0 to 255 ($Q = 256$), the probability of a given quantization level appearing more than once is extremely high: $1 - 1.86 \times 10^{-6}$. This example shows the importance of adding a proper perturbation to our digital signal $\mathbf{x}$, as the one proposed in Subsection 3.3.3.

C.1.2 Adding a perturbation

In this subsection we propose a perturbation method that not only ensures that all perturbed coefficient values $x'_i = x_i + \epsilon_i$, $i = 1 \dots N$ are different (high entropy) but it also maximizes the minimal possible difference between them. The latter feature is achieved by using for the perturbation N uniform extra levels within each single quantizing step and assigning a different level to each ϵ_i. On the other hand, noticeable local oscillations are avoided by choosing a very low frequency pattern for the perturbation, which minimizes its perceptual impact.

Algorithm 6 explains the process. First, a random angle tangent is chosen for generating a ramp for the image grid (same method can be straightforwardly generalized to n-dimensional grids). The tangent value must not be a rational

15. This does not pretend to be an accurate estimation of the probability of the repetition of the maximal/minimal value v in a typical signal (e.g., an image). However, it does provide a useful reference value.

p/q number with small $p, q \in \mathbb{N}$, because that would create repeated values on the ramp. Then, after checking that there are no repeated values on the generated ramp, its cells are sorted in a ranking according to their value, corresponding the number 1 to the lowest value and N the highest. Finally, these integers are normalized to the interval $[-1/2, 1/2)$ and returned as the perturbation. The result is a (slightly curved, sigmoid) ramp made of all different values, having an exactly uniform distribution.

Algorithm 6 High-entropy, low-impact perturbation

Require: An empty array of $N_x \times N_y = N$ pixels
1: **repeat**
2: Generate a pseudo-random number $r \in [0, 1]$
3: Compute a ramp $v(n_x, n_y) = n_x + r * n_y$
4: Check for repeated values in v
5: **until** there are no repeated values in v
6: Compute $o(n_x, n_y) = rank(v(n_x, n_y)) \in \{1 \dots N\}$
7: $\epsilon(n_x, n_y) = (o(n_x, n_y) - 1 - N/2)/N \in [-1/2, 1/2)$
8: **return** ϵ ($N_x \times N_y$ array)

C.2 Second-order moments at the output of a filter bank

C.2.1 Critical points, active frequencies and perturbations

Being the sample second-order moment at the output of a filter a positive definite quadratic function, it has no saddle points. Therefore, in this case, in contrast with Section C.1.1, we do not face the problem of their basins of attraction.

The critical points of the decoupled features are those vectors where the gradients are either zero (critical points inherited from the original features) or co-linear. Substituting Eq. (25) into Eq. (18) it yields:

$$|H_k(\xi)|^2 X_k^*(\xi) = \sum_{j=1}^{k-1} \lambda_{j,k} |H_j(\xi)|^2 X_k^*(\xi), \quad \{\lambda_{j,k} \neq 0\}.$$

This equation always admits the solution $\mathbf{x}_0^* = \mathbf{0}$. In addition, in the case there are regions of the signal spectrum that are not covered by any filter (e.g., signals made of a constant value, for band-pass or high-pass filters), the corresponding vectors having only those frequencies will also be critical points. Apart from the previous solutions, all corresponding to the "null space" of the filters' output (which provide the absolute minima of the original features), the equation may only hold if the squared filters are themselves co-linear. Thus, the latter possibility must be prevented - otherwise *all* points in $\bar{\Omega}$ would be critical!

We see that, although in this case the decoupled features do not introduce additional critical points, zeros in the frequency domain, both of the signal and of the filters, act as partial "gradient killers": signal will not change at those frequencies where kernels are zero, and, similarly, the signal will neither change at those frequencies where the signal itself vanishes.

Whereas in some applications it is normal to ignore some regions of the spectrum that are not useful for a given task, and thus they are left uncovered by the filters bank, it seems advisable, nevertheless: i) to filter out those frequencies (having a support D) also in $\mathbf{x}$ (otherwise those spectral components will remain unchanged in $\mathbf{x}$), and ii) to introduce a small perturbation ϵ in $\mathbf{x}$, such that $\mathbf{x}+\epsilon$ will not be zero or too small at any frequency in the above defined support D. Then, a first reasonable concrete choice for the perturbation may be

$$\epsilon = \arg\min_{\mathbf{z}} \|\mathbf{z}\| \; s.t. \; |X(\xi) + Z(\xi)| \geq \theta, \forall \xi \in D$$

which yields, in the Fourier domain, for $\xi \in D$:

$$E(\xi) = \begin{cases} 0, & \text{if } |X(\xi)| > \theta \\ (\theta - |X(\xi)|) e^{i2\pi \arg(X(\xi))}, & \text{if } \theta > |X(\xi)| > 0 \\ \theta e^{i2\pi r}, & \text{if } |X(\xi)| = 0, \end{cases} \tag{37}$$

and 0 for $\xi \notin D$, where $E(\xi) = \mathcal{F}(\epsilon)$. Here r is a uniform random value in $[0, 1]$. The so defined ϵ is a perturbation of maximal entropy amongst all minimal Euclidean norm ensuring a spectral content above a threshold θ in D.[16] The threshold θ can be chosen as the supreme of the set $\{\theta : q(\mathbf{x} + \epsilon(\mathbf{x}, \theta)) = q(\mathbf{x})\}$, or a similar criterion. Furthermore, depending on the complete set of features, different types of perturbations (see Subsection 3.3.3) can be fused into a single one fulfilling all the requirements.

APPENDIX D
REGRESSION EXPERIMENTS

In this section we give some specific details about the regression experiments. The probability density function of the Generalized Gaussian Distribution (GGD) is given by:

$$f(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|x-\mu|/\alpha)^\beta},$$

where Γ denotes the gamma function, β is the shape parameter, α the scale parameter and μ the location parameter. The probability density function of the Gamma Distribution (GMD) is given by:

$$f(x) = \frac{1}{\Gamma(\beta)\theta^\beta} x^{\beta-1} e^{-x/\theta},$$

where β is the shape parameter and θ the scale parameter.

The probability density function of the absolute value of a Normal distribution raised to β (GND), $X = |T|^\beta$, $T \sim \mathcal{N}(0,\, 1)$, is given by, for every positive β:

$$f(x) = \frac{2}{\beta\sqrt{2\pi}} x^{(1/\beta)-1} e^{-\frac{1}{2}x^{2/\beta}},$$

where β is the shape parameter. Note that, for $\beta = 2$, this distribution leads to the chi-squared distribution of one degree of freedom, $\chi^2(1)$.

The RMSE results for the different regression methods and sample sizes N are shown in Tables 6 (Generalized Gaussian Distribution distribution, GGD), 7 (Gamma distribution, GMD) and 8 (absolute value of a Normal distribution raised to β, GND). The regression methods compared are: linear regression models (LRM), regression trees (RT), support vector regression (SVR), Gaussian process regression (GPR), ensembles of trees (ET) and neural networks (NNR). MSM stands for the set of classical marginal

16. A perceptually-based perturbation may be easily obtained from here by considering perceptual metrics/threshold instead.

TABLE 6: RMSE Results for the different regression methods, Generalized Gaussian distribution (GGD).

	N	LRM	RT	SVR	ET	GPR	NNR
MSM	64	0.71	0.59	0.54	0.55	0.52	**0.51**
	128	0.75	0.46	0.46	0.44	0.41	**0.41**
	256	0.85	0.36	0.44	0.35	0.34	**0.32**
	512	0.95	0.30	0.47	0.29	0.29	**0.27**
	1024	1.02	0.26	0.55	0.25	0.25	**0.24**
	2048	1.13	0.22	0.63	0.22	0.22	**0.21**
DF_{MSM}	64	0.79	0.48	0.51	0.45	0.47	**0.44**
	128	0.67	0.34	0.44	0.32	0.38	**0.31**
	256	0.85	0.25	0.38	0.23	0.31	**0.22**
	512	0.51	0.18	0.31	0.17	0.26	**0.17**
	1024	0.45	0.13	0.29	0.13	0.26	**0.13**
	2048	0.41	**0.10**	0.32	0.10	0.31	0.10

TABLE 7: RMSE Results for the different regression methods, Gamma distribution (GMD).

	N	LRM	RT	SVR	ET	GPR	NNR
MSM	64	0.58	0.76	0.59	0.71	**0.50**	0.52
	128	0.45	0.59	0.46	0.56	**0.39**	0.40
	256	0.37	0.48	0.38	0.46	**0.31**	0.31
	512	0.27	0.39	0.30	0.36	**0.23**	0.24
	1024	0.21	0.29	0.22	0.28	0.21	**0.19**
	2048	0.18	0.22	0.18	0.21	0.15	**0.13**
DF_{MSM}	64	0.45	0.45	0.39	0.40	0.39	**0.38**
	128	0.32	0.32	0.28	0.29	**0.27**	0.28
	256	0.25	0.24	0.21	0.22	0.20	**0.20**
	512	0.19	0.18	0.16	0.15	**0.14**	0.14
	1024	0.15	0.14	0.12	0.12	**0.10**	0.10
	2048	0.12	0.10	0.11	0.08	0.07	**0.07**

standardized moments, and DF_{MSM} for its corresponding decoupled set. Highlighted in bold, the regression method that minimizes the RMSE for each N value.

ACKNOWLEDGMENTS

The authors would like to thank Matteo Bonforte, Rafael Molina, Ivan Selesnick, Gustau Camps-Valls and Eero Simoncelli for fruitful discussions.

TABLE 8: RMSE Results for the different regression methods, absolute value of a Normal raised to β (GND).

	N	LRM	RT	SVR	ET	GPR	NNR
MSM	64	0.64	0.77	0.68	0.73	**0.55**	0.59
	128	0.62	0.74	0.65	0.71	**0.53**	0.56
	256	0.53	0.65	0.56	0.61	**0.43**	0.47
	512	0.46	0.56	0.47	0.52	**0.36**	0.40
	1024	0.40	0.49	0.41	0.45	**0.31**	0.34
	2048	0.36	0.41	0.35	0.38	0.34	**0.29**
DF_{MSM}	64	0.41	0.45	0.38	0.39	**0.37**	0.38
	128	0.37	0.40	0.34	0.35	**0.34**	0.34
	256	0.27	0.31	0.25	0.26	**0.25**	0.25
	512	0.20	0.21	0.18	0.18	**0.17**	0.18
	1024	0.16	0.16	0.14	0.14	**0.13**	0.13
	2048	0.13	0.12	0.11	0.10	**0.10**	0.10

REFERENCES

[1] J. Portilla and E. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *International Journal of Computer Vision*, vol. 40(1), pp. 49–71, 2000, ®Matlab code publicly available at http://www.cns.nyu.edu/ lcv/texture/.

[2] E. Martínez-Enríquez and J. Portilla, "Controlled feature adjustment for image processing and synthesis," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, 2020, pp. 1–6.

[3] J. Balas, "Texture synthesis and perception: Using computational models to study texture representations in the human visual system," *Vision Research*, vol. 46, no. 3, pp. 299 – 309, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0042698905002336

[4] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*, 1st ed. Springer Publishing Company, Incorporated, 2007.

[5] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[6] J. Portilla and E. Martinez-Enriquez, "Nested normalizations for decoupling global features," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct 2018, pp. 2112–2116.

[7] E. Martinez-Enriquez and J. Portilla, "Deterministic feature decoupling by surfing invariance manifolds," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6049–6053.

[8] J. R. M. Hosking and J. R. Wallis, "Parameter and quantile estimation for the generalized pareto distribution," *Technometrics*, vol. 29, no. 3, pp. 339–349, 1987. [Online]. Available: http://www.jstor.org/stable/1269343

[9] A. Azzalini, "A class of distributions which includes the normal ones," *Scandinavian Journal of Statistics*, vol. 12, no. 2, pp. 171–178, 1985. [Online]. Available: http://www.jstor.org/stable/4615982

[10] C. Flecher, P. Naveau, and D. Allard, "Estimating the closed skew-normal distribution parameters using weighted moments," *Statistics & Probability Letters*, vol. 79, no. 19, pp. 1977–1984, 2009.

[11] C. G. Justus, W. R. Hargraves, A. Mikhail, and D. Graber, "Methods for estimating wind speed frequency distributions," *Journal of Applied Meteorology and Climatology*, vol. 17, no. 3, pp. 350 – 353, 1978.

[12] S. A. Akdağ and A. Dinler, "A new method to estimate Weibull parameters for wind energy applications," *Energy Conversion and Management*, vol. 50, no. 7, pp. 1761–1766, 2009.

[13] A. Bovik, M. Clark, and W. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 55–73, 1990.

[14] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," in *Proceedings., International Conference on Image Processing*, vol. 3, Oct 1995, pp. 648–651 vol.3.

[15] J. Portilla, R. Navarro, O. Nestares, and A. Tabernero, "Texture synthesis-by-analysis based on a multiscale early-vision model," *Optical Engineering*, vol. 35, no. 8, pp. 2403–2417, 1996.

[16] D. C. Blest, "A new measure of kurtosis adjusted for skewness," *Australian and New Zealand Journal of Statistics*, vol. 45, no. 2, pp. 175–179, 2003. [Online]. Available: http://dx.doi.org/10.1111/1467-842X.00273

[17] K. Pearson, "IX. Mathematical contributions to the theory of evolution.—XIX. Second supplement to a memoir on skew variation," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 216, no. 538-548, pp. 429–457, 1916. [Online]. Available: http://rsta.royalsocietypublishing.org/content/216/538-548/429

[18] R. Sharma and R. Bhandari, "Skewness, kurtosis and Newton's inequality," *ArXiv e-prints*, Sep. 2013.

[19] K. Pearson, "Contributions to the mathematical theory of evolution," *Philosophical Transactions of the Royal Society of London. A*, vol. 185, pp. 71–110, 1894. [Online]. Available: http://www.jstor.org/stable/90667

[20] A. T. Kalai, A. Moitra, and G. Valiant, "Efficiently learning mixtures of two Gaussians," in *Proceedings of the forty-first annual ACM symposium on the Theory of Computing (STOC), 2010*. ACM Press, June 2010, pp. 553–562. [Online]. Available: https://www.microsoft.com/en-us/research/publication/efficiently-learning-mixtures-two-Gaussians/

[21] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193–202, 1980.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the Institute of Radio Engineers*, vol. 86, no. 11, pp. 2278–2323, 1998.

[23] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of Physiology*, vol. 148, no. 3, pp. 574–591, 1959. [Online]. Available:

https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/ jphysiol.1959.sp006308
[24] J. G. Daugman, "Complete discrete 2-d Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 36, pp. 1169–1179, 1988.
[25] R. F. Navarro, A. Tabernero, and G. Cristóbal, "Image representation with Gabor wavelets and its applications," *Advances in Imaging and Electron Physics*, vol. 99, p. 329, 1997.
[26] T. Dau, B. Kollmeier, and A. A. Kohlrausch, "Modeling auditory processing of amplitude modulation. i. detection and masking with narrow-band carriers." *The Journal of the Acoustical Society of America*, vol. 102 5 Pt 1, pp. 2892–905, 1997.
[27] I. Daubechies, "Ten lectures on wavelets," *Computers in Physics*, vol. 6, pp. 697–697, 1992.
[28] T. Canham, A. Martín, M. Bertalmío, and J. Portilla, "Using decoupled features for photo-realistic style transfer," *SIAM Journal on Imaging Sciences, Submitted*, 2022.
[29] S. Robert L. Devaney, M. Hirsch, S. Smale, and R. Devaney, *Differential Equations, Dynamical Systems, and an Introduction to Chaos*, ser. Pure and Applied Mathematics - Academic Press. Elsevier Science, 2004. [Online]. Available: https://books.google.es/books?id=GNOmchErrMgC
[30] J. Kovacević and A. Chebira, "An introduction to frames," *Foundations and Trends in Signal Processing*, vol. 2, no. 1, pp. 1–94, 2008.
[31] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, no. 3–4, p. 229–256, may 1992. [Online]. Available: https://doi.org/10.1007/BF00992696
[32] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013. [Online]. Available: https://arxiv.org/abs/1312.6114
[33] J. Portilla, A. Tristan-Vega, and I. Selesnick, "Efficient and robust image restoration using multiple-feature L2-relaxed sparse analysis priors," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5046–5059, Dec 2015, ®Matlab code publicly available at DOI: 10.13140/RG.2.1.3531.5280.
[34] A. Winkelbauer, "Moments and Absolute Moments of the Normal Distribution," *arXiv e-prints*, p. arXiv:1209.4340, Sep. 2012.
[35] P. Brodatz, *Textures: a photographic album for artists and designers*, ser. Dover pictorial archives. Dover Publications, 1966, available at https://multibandtexture.recherche.usherbrooke.ca/original_brodatz.html. [Online]. Available: http://books.google.es/books?id=4rBk_SemB2QC
[36] "https://in.mathworks.com/help/stats/regression-learner-app.html."
[37] M. Nardon and P. Pianca, "Simulation techniques for generalized Gaussian densities," *Journal of Statistical Computation and Simulation*, vol. 79, no. 11, pp. 1317–1329, 2009.
[38] K. Kokkinakis and A. K. Nandi, "Exponent parameter estimation for generalized Gaussian probability density functions with application to speech modeling," *Signal Processing*, vol. 85, no. 9, pp. 1852–1858, 2005.
[39] G. Marsaglia and W. W. Tsang, "A simple method for generating gamma variables," *ACM Trans. Math. Softw.*, vol. 26, no. 3, p. 363–372, sep 2000. [Online]. Available: https://doi.org/10.1145/358407.358414
[40] M. Mandal, T. Aboulnasr, and S. Panchanathan, "Image indexing using moments and wavelets," *IEEE Transactions on Consumer Electronics*, vol. 42, pp. 557–565, 1996.
[41] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, Aug 2005.
[42] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, Jun 1998. [Online]. Available: https://doi.org/10.1023/A:1009715923555
[43] G. Schneider and H. Uecker, *Nonlinear PDEs*, ser. Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 2017, vol. 182, a dynamical systems approach. [Online]. Available: https://doi.org/10.1090/gsm/182
[44] L. Simon, "Asymptotics for a class of non-linear evolution equations, with applications to geometric problems," *Annals of Mathematics*, vol. 118, no. 3, pp. 525–571, 1983. [Online]. Available: http://www.jstor.org/stable/2006981
[45] M. Spivak, *A Comprehensive Introduction to Differential Geometry*, ser. A Comprehensive Introduction to Differential Geometry. Publish or Perish, Inc., Wilmington, Del., 1979, no. v. 1.
[46] P. Hartman, *Ordinary Differential Equations*, ser. Classics in Applied Mathematics, 38. Society for Industrial and Applied Mathematics, 2002. [Online]. Available: https://books.google.es/books?id=CENAPMUEpfoC