APSIPA Transactions on Signal and Information Processing, 2024, 13, e9

Original Paper

Locally-Structured Unitary Network

Yasas Godage[1], Eisuke Kobayashi[1] and Shogo Muramatsu[2*]

[1] *Graduate School of Science & Technology, Niigata University, Japan*
[2] *Faculty of Engineering, Niigata University, Japan*

ABSTRACT

This paper proposes a novel learnable linear transform, locally-structured unitary network (LSUN), that captures tangent spaces of a manifold latent in high-dimensional data, enabling effective, systematic, and highly interpretable data-driven dimensionality reduction. LSUN provides a linear layer that has locally controllable filter kernels with shift-variability under the structural constraint of global unitary property. It is similar to a convolutional layer as the filter kernels share the properties of overlapping and locality, while fixed kernels are not repeated. The kernels can be trained in a self-supervised manner owing to the unitary property. The proposed method can be a candidate for realizing manifold learning. Although local selection of filter kernels, such as sparse modeling, can capture tangent spaces as a set of coordinates, the set of kernels is redundant, and the filters are not very interpretable. To address these problems, this study utilizes a method that locally controls coordinate axes by combining some primitive local linear operations that preserve unitarity, such as Givens rotation, shift, and butterfly operations. This study evaluates the ability to capture the tangent space of the proposed LSUN through low-dimensional approximation and dynamical system modeling experiments.

Keywords: Tangent space sampling, shift-variability, unitarity, linear transforms, self-supervised learning

*Corresponding author: Shogo Muramatsu, shogo@eng.niigata-u.ac.jp. This work was supported by JSPS KAKENHI Grant Number JP22H00512.

Received 04 February 2024; revised 24 April 2024; accepted 24 April 2024.
ISSN 2048-7703; DOI 10.1561/116.00000308

1 Introduction

High-dimensional data is extensively used in many signal processing applications, with examples such as MRIs [22], satellite imagery [49], video sequence data [16], and thermal imaging. Although high-dimensional datasets can capture complex details, they often pose significant challenges. The curse of dimensionality [47] becomes apparent as the data becomes sparser in higher dimensions, leading to increased computational complexity and challenges in visualizing and interpreting the data. The goal of dimensionality reduction is to discover a subspace that captures the most relevant information while reducing computational complexity. The naturally occurring high-dimensional data is assumed to lie on a low-dimensional manifold. Manifold learning offers a non-linear approach to dimensionality reduction commonly used for various data types [8, 11, 36, 21], focusing on learning smooth surfaces in multi-dimensional spaces. Using a linear system to capture the manifold structure can reduce the computational complexity inherent in non-linear processing. Tangent space learning [44, 50, 48, 40] aims to maintain local linear structures by assuming that the underlying manifold can be locally approximated by a linear tangent space. By aligning these tangent spaces across the dataset, linear tangent space learning constructs a lower-dimensional representation that captures the local geometry of the data. Although it may not capture intricate non-linear relationships as effectively as some non-linear methods, the linear approach provides computational benefits, interpretability, and a simpler comprehension of the local structure of the data.

Tangent spaces can be sampled using various methods, including singular value decomposition (SVD), principal component analysis (PCA)[1], independent component analysis (ICA), convolutional auto-encoders (CAE), convolutional dictionary learning (CDL) [5, 26, 29, 25]. Differentiation between capturing tangent spaces between the conventional block-processing/convolutional structure and the proposed structure outlined in this study is illustrated in Figure 1. Among these methods convolutional processing as shown in Figure 1 (a), with a selection mechanism such as sparse approximation, or alternatively, an adaptive mechanism such as block-PCA as shown in Figure 1 (b) can be utilized for capturing the tangent spaces. This study aims to expand the concept of block-PCA to capture overlapping local structures, enabling the learning of tangent spaces in a data-driven manner while maintaining the shift-variance of local coordinates.

Filter banks (FBs) [45] provide a representation of convolutional layers for capturing tangent space. Figure 2 shows the parallel structure of multidimensional (MD) FBs. Analytical FBs examples include block discrete cosine transform (DCT) [33, 39], discrete wavelet transform (DWTs) [42, 46], and

[1]Known as Karhunen-Loève Transform (KLT).

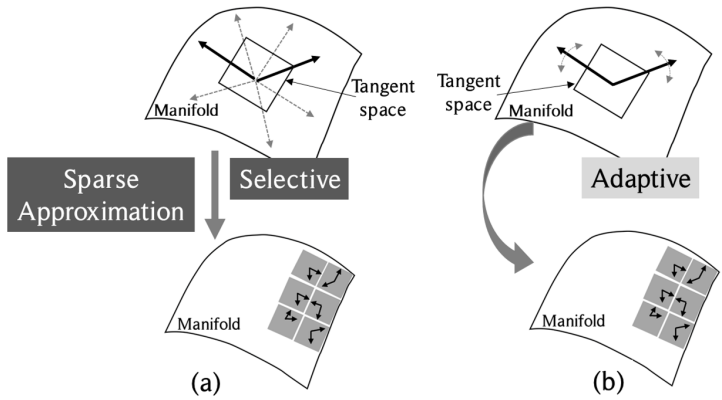


Figure 1: Two different ways of capturing tangent spaces as coordinate systems on the manifold: (a) Sparse approximation (conventional) and (b) Adaptive approximation (proposal).

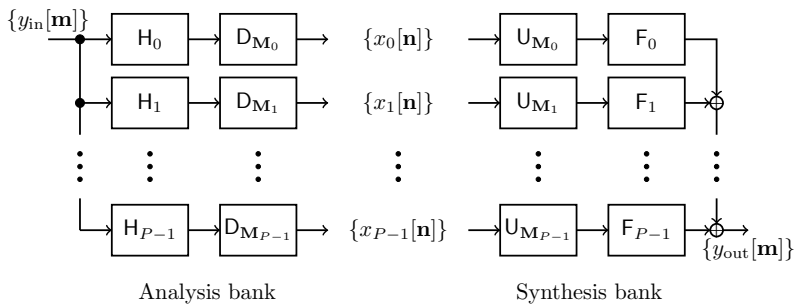


Figure 2: Parallel configuration of P-channel filter banks with downsamplers $\mathsf{D}_{\mathbf{M}p}$ and upsamplers $\mathsf{U}_{\mathbf{M}_p}$. H_p and F_p represent the p^{th} analysis and synthesis filters. Subband signals are denoted by $\{x_p[\mathbf{n}]\}$. All operations are linear.

lapped transforms [28, 10, 30, 27, 38, 37], which can be employed alongside sparsity-aware approximation or other dimensional reduction techniques. Their convolutional kernels correspond to localized coordinate systems embedded in a high-dimensional space. However, it is important to note that such sets of filter kernels, i.e., dictionaries, may not always match a given set of data. This limitation can be overcome by fitting the kernels to the dataset through the use of parametric FBs as learnable dictionaries or convolutional layers [10, 28, 23, 27, 32, 15, 14, 30, 7]. This approach is known as CDL. CDL is a data-driven technique for designing FBs, yielding localized continuous coordinate systems comprised of atomic vector, i.e., filter kernels, which allows for signal representation via a linear combination of a limited number of atoms [34, 4, 9, 12]. This process is streamlined with the convolution process. See Appendix A for reference.

Table 1 summarizes a comparative analysis of the aforementioned dimensional reduction models. These models apply low-dimensional approximation with block processing to capture the tangent spaces, allowing us to locally approximate a low-dimensional embedding for the focused support region and capture tangent vectors. As the intended low-dimensional structure is recognized as a smooth manifold, maintaining the smoothness on the block boundaries is crucial. Lapped transforms and linear convolution processes facilitate smooth connections among blocks, eliminating discontinuities. The shift-invariance nature of convolutional models does not directly correspond to the axes of tangent spaces. In contrast to the above-mentioned models, SVD, PCA, and ICA's localized block processing enables us to achieve shift-variance of coordinate systems. However, the continuous property of tangent spaces is not explicitly considered due to the lack of overlapping in the block processing.

Table 1: Comparison of dimensional reduction models. Block DCT, block PCA, convolutional auto-encoders (CAEs), and dictionary learning with paraunitary filter banks (DL-PUFB) are compared, where all use shift-invariant kernels. ✓ and ✗ represent satisfactory or unsatisfactory, respectively.

	Unitary	Overlap	Learnable
Block DCT [33]	✓	✗	✗
Block PCA [5]	✓	✗	✓
CAE [29]	✗	✓	✓
DL-PUFB [30]	✓	✓	✓

Evidently, the sampling of tangent spaces is insufficient with the conventional linear transforms by themselves due to the lack of discontinuity of block transforms and/or shift-invariant nature of the convolutional structure. The non-linear activation and sparsity-aware selection of coordinate systems have issues with interpretability as the localized linear transforms do not capture bases of tangent spaces by themselves. To solve this problem, this study aims to expand the concept of block-PCA to capture overlapping local structures, enabling the learning of tangent spaces in a data-driven manner while maintaining the shift-variance of local coordinates. We propose a method to locally control coordinate axes by combining some primitive linear transforms, such as Givens rotation, shift, and butterfly operations, under the constraints of the unitarity so that a self-supervised learning approach is applicable. We observe the utilization of lapped transforms with kernel switching in previous works [19, 43], where variable kernels are applied during processing. However, these works propose methods to control the filter shapes and lengths while keeping the filter coefficients constant to enhance boundary processing and the efficiency of coding. In contrast, our proposal emphasizes the capability to dynamically control the filter itself by adaptively varying the filter coefficients. The major contributions of this study are summarized as follows:

Shift-variability
Filter kernels of the network can be varied adaptively according to the input.

Unitarity
Preserves total energy through the network.

Self-supervised learning
Specific dataset is not required for training. Input data can be utilized as training data.

Note that we have already applied one of the specific examples of our proposed locally-structured unitary network (LSUN) to video denoising in [16]. The difference is that this paper focuses on the theoretical background, design method, implementation method, and detailed evaluation of low-dimensional embedding performance of LSUN, which were not described in detail in [16]. The remainder of this paper is organized as follows. Section 2 reviews the linear models for dimensional reduction. Section 3 introduces the proposed network, in Section 4, we share some design examples to verify the significance of LSUN, and Section 5 evaluates the performance of the proposed network, followed by the conclusions in Section 6.

2 Review of Linear Dimensional Reduction

This section reviews some conventional data-driven linear transforms which have the potential to capture tangent spaces.

2.1 Principal Component Analysis (PCA)

PCA provides a data-driven coordinate system to represent the statistical variations of high-dimensional correlated data. The resultant principal components (PCs) can be interpreted as coordinate axes of a space that represent the direction of dominant variances. These coordinate axes are orthogonal to each other. PCA refers to the problem of fitting a low-dimensional affine subspace of dimension $p \ll M$ to a set of points $\{\mathbf{y}_n\}_n \subset \mathbb{R}^M$, where $\sum_{n=1}^{S} \mathbf{y}_n = \mathbf{0}$.

The problem setting is represented as

$$\{\hat{\mathbf{\Phi}}, \{\hat{\mathbf{x}}_n\}_n\} = \arg \min_{\{\mathbf{\Phi}, \{\mathbf{x}_n\}_n\}} \frac{1}{2S} \sum_{n=1}^{S} \|\mathbf{y}_n - \mathbf{\Phi}\mathbf{x}_n\|_2^2, \tag{1}$$

subject to

$$\mathbf{\Phi}^\mathsf{T}\mathbf{\Phi} = \mathbf{\Phi}\mathbf{\Phi}^\mathsf{T} = \mathbf{I}_M, \tag{2a}$$

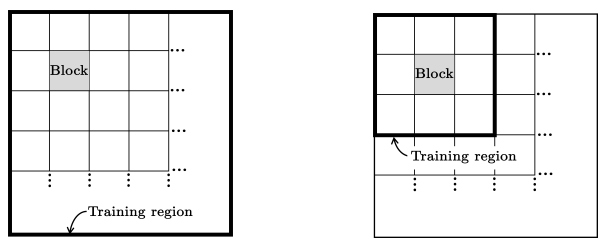
$$\|\mathbf{x}_n\|_0 \leq p, n \in \{1, 2, \ldots, S\}, p \in \{1, 2, \ldots, M-1\} \tag{2b}$$

to find the unknown synthesis dictionary $\mathbf{\Phi} \in \mathbb{R}^{M\times M}$ and features $\{\mathbf{x}_n\}_n \subset \mathbb{R}^M$ for $\{\mathbf{y}_n\}_n$, where $\mathbf{I}_M$ is the $M \times M$ identity matrix and $\|\cdot\|_0$ denotes the number of non-zero entries. (1) is equivalently represented by $\widehat{\mathbf{\Phi}} = \arg\max_{\mathbf{\Phi}\in\mathbb{R}^{M\times M}} \mathrm{tr}(\mathbf{\Phi}^{\intercal}_{:,1:p}\widehat{\mathbf{\Sigma}}_{\mathbf{y}}\mathbf{\Phi}_{:,1:p})$, s.t. $\mathbf{\Phi}^{\intercal}\mathbf{\Phi} = \mathbf{I}_M$, where $\widehat{\mathbf{\Sigma}}_{\mathbf{y}}$ is the sample covariance matrix of the given set $\{\mathbf{y}_n\}_n$. The first p columns of $\mathbf{\Phi}$ is expressed as $\mathbf{\Phi}_{:,1:p}$. The solution is the eigendecomposition as $\widehat{\mathbf{\Phi}}^{\intercal}\widehat{\mathbf{\Sigma}}_{\mathbf{y}}\widehat{\mathbf{\Phi}} = \mathbf{\Lambda}$, where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_M)$, and $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_M$ are the eigenvalues of $\widehat{\mathbf{\Sigma}}_{\mathbf{y}}$, and $\hat{\mathbf{x}}_n = \widehat{\mathbf{\Phi}}^{\intercal}\mathbf{y}_n$.

This study cites two variants of block PCAs as reference methods, namely global block PCA (GBPCA) and local block PCA (LBPCA). Both methods are originated from PCA [5, §1.5],[17, §2.12] where we suggest the latter for showing the advantage of local adaptability inspired by some works regarding local processing of PCA [20].

2.1.1 Global block PCA (GBPCA)

Consider subdividing a high-dimensional signal $\mathbf{y} \in \mathbb{R}^{BM}$, where BM denotes the number of signal or pixel values, such as an image, into B blocks to generate a dataset $\{\mathbf{y}_n\}_n$ as shown in Figure 3 (a). PCA is performed on the vectorized dataset $\{\mathbf{y}_n\}_n$ and PCs are obtained. Since a single dictionary is derived for all blocks, we refer to this approach as global block PCA (GBPCA) in this paper. GBPCA exhibits the properties of linearity and unitarity, and it captures localized constant coordinate axes to the entire input. The resulting dictionary can be regarded as a special case of a convolutional dictionary with a block-size stride. Note that it lacks the ability to capture tangent spaces of a manifold by itself. By changing the way of defining the input dataset, the global representation of GBPCA can be changed to a local representation.



(a) (b)

Figure 3: 2-D illustration of training region in (a) GBPCA and (b) LBPCA, respectively. The training region for GBPCA is extracted from the whole data array, while LBPCA extracts the training region only from surrounding local blocks.

2.1.2 Local block PCA (LBPCA)

The dataset $\{\mathbf{y}_n\}_n$ is defined in the following way. After subdividing a high-dimensional signal $\mathbf{y}$, PCA can be applied for a collection of blocks marked as "Training region" in Figure 3 (b). Since multiple local dictionaries $\{\hat{\mathbf{\Phi}}_b\}_b$ are derived for the original signal $\mathbf{y}$, we refer to this approach as local block PCA (LBPCA) in this paper, where b denotes the block index, and the number of training blocks S for each target block is less than the total number of blocks B. LBPCA exhibits linearity, unitarity, and shift-variability properties, making it possible to capture PCs independently for each of the blocks. Therefore, it has the potential to capture tangent spaces. However, LBPCA does not satisfy the overlapping property, which is essential for continuously capturing tangent spaces on a smooth manifold.

2.2 Review of Filter Banks

A typical FB arrangement features analysis and synthesis banks as demonstrated in Figure 2. It exhibits a parallel structure of a P channel FB described by the filtering operators H_p and F_p respectively for the p^{th} analysis and synthesis filters. These filters are usually realized by linear shift-invariant (LSI) systems via convolutional filter kernels. $\mathsf{D}_{\mathbf{M}_p}$ and $\mathsf{U}_{\mathbf{M}_p}$ are the p^{th} downsampler and upsampler, respectively, with resampling matrix $\mathbf{M}_p \in \mathbb{Z}^{D\times D}$ which exactly denotes the stride. The resampling ratio for each channel is calculated as $M_p = |\det(\mathbf{M}_p)|$, and the redundancy as $R = \sum_{p=0}^{P-1} M_p^{-1}$. A FB with $R > 1$ is referred to as oversampled, whereas critically sampled FBs have $R = 1$. Note that perfect reconstruction (PR) requires that $R \geq 1$, and that the output $\{y_{\text{out}}[\mathbf{m}]\}$ is a scaled and shifted form of the input $\{y_{\text{in}}[\mathbf{m}]\}$.

Oversampled FBs increase the degree of freedom, resulting in a set of redundant filter kernels. Nonetheless, the system's complexity is increased by the realization of an infinite number of analysis and synthesis banks. The FBs can be created by introducing various constraints, including linear phase, orthogonality and so forth, with consideration for potential applications. Examples falling under this category include LPPUFB [32, 31], GenLOT [10], NSOLT [30], modulated lapped transform (MLT) [41], and extended lapped transform (ELT) [27]. These FBs are constructed through cascaded primitive block operations, which inherently maintain locality and unitarity. Owing to the unitarity of both analysis and synthesis FBs, they can be represented by structured matrices.

Let $\mathbf{A}$ and $\mathbf{D}$ represent the operators by analysis and synthesis FBs, respectively. If $\mathbf{DA} = \mathbf{I}$, perfect reconstruction is sufficiently achieved. Furthermore, if $\mathbf{A}^\mathsf{T}\mathbf{A} = \mathbf{D}\mathbf{D}^\mathsf{T} = \mathbf{I}$ holds, they satisfy orthonormality, which ensures that energy is preserved before and after the transform (Perceval's theorem). Fac-

torizations of analysis and synthesis FBs into product forms, i.e., the cascade of primitive block operations, can be expressed as

$$\mathbf{A} = \mathbf{H}_{I-1}\mathbf{H}_{I-2}\cdots\mathbf{H}_1\mathbf{H}_0, \tag{3a}$$
$$\mathbf{D} = \mathbf{F}_0\mathbf{F}_1\cdots\mathbf{F}_{I-2}\mathbf{F}_{I-1}. \tag{3b}$$

The structural constraints for the FBs can be attained by parameterizing the I factors $\{\mathbf{H}_i\}$ or $\{\mathbf{F}_i\}$ while imposing some structural constraints. Examples of primitive block operations include Givens rotations and shift operations, as discussed in Appendix B [31, 13]. The combination of these primitive operations allows for the parametric design of convolutional dictionaries that adhere to preferred structural constraints. However, these FBs consist of a global representation of coordinate axes, similar to GBPCA in Section 2.1.1. Consequently, they lack the ability to define coordinate axes locally.

2.3 Dictionary Learning

GBPCA can be regarded as a dictionary learning method in the form of $\mathbf{D} = \mathbf{A}^\intercal = \text{blkdiag}(\mathbf{\Phi}, \mathbf{\Phi}, \cdots, \mathbf{\Phi})$ with $\mathbf{\Phi} \in \mathbb{R}^{M\times M}$ and LBPCA as $\mathbf{D} = \mathbf{A}^\intercal = \text{blkdiag}(\mathbf{\Phi}_1, \mathbf{\Phi}_2, \cdots, \mathbf{\Phi}_B)$ with $\mathbf{\Phi}_b \in \mathbb{R}^{M\times M}$, where $\text{blkdiag}(\cdots)$ denotes the block diagonal matrix of the entries. We can find other instances of dictionary learning such as K-SVD [1, 12]. Dictionary learning can be regarded as a variant of PCA in which the orthogonal constraint is relaxed and redundant systems emphasizing sparse representation are preferred.

Dictionary learning is capable of capturing a low-dimensional representation of a high-dimensional signal with sparse approximation by (1) subject to

$$\|\mathbf{x}_n\|_0 \leq K,\ n \in \{1, 2, \cdots, S\}. \tag{4}$$

Here $K(\ll M)$ denotes the upper limit of the number of non-zero coefficients. $\hat{\mathbf{\Phi}} \in \mathbb{R}^{M\times N}$ represents the dictionary learned from $\{\mathbf{y}_n\}_n \subset \mathbb{R}^M$, and $\{\mathbf{x}_n\}_n \subset \mathbb{R}^N$ is a set of coefficients, where $M \leq N$. Typical dictionary learning alternatively applies sparse approximation and dictionary update steps [12, 1].

2.3.1 Block K-SVD

Note that K-SVD, a representative dictionary learning technique, can be used in a global block processing manner. After subdividing a high-dimensional signal $\mathbf{y} \in \mathbb{R}^{BM}$ into patches $\{\mathbf{y}_n\}_n$, we can design a single redundant dictionary $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$. Although $\mathbf{\Phi}$ is globally applied to every block, the dictionary works in a localized manner. The axes are locally selected with sparse approximation. While the block K-SVD can be designed like GBPCA, it is also possible to produce local dictionaries $\{\hat{\mathbf{\Phi}}_b\}_b$ as LBPCA.

2.3.2 Convolutional dictionary learning (CDL)

Block K-SVD operates in a block-processing manner regardless of the adjacent blocks. A convolutional dictionary also realizes a local process [34, 4, 9]. Unlike block PCA and K-SVD, it processes a local signal in relation to the neighboring region. CDL estimates local atoms to represent signals sparsely. In contrast to K-SVD, a convolutional dictionary consists of a set of linear shift-invariant filters with some stride. The redundant design allows us to locally select coordinate systems related to adjacent region. A coordinate selection, such as sparse approximation, is used to pick the major coordinate axes which capture low-dimensional structure out of the redundant filters.

Let $\mathbf{D} \in \mathbb{R}^{BM \times BN}$ be a global synthesis convolutional dictionary, $\mathbf{x} \in \mathbb{R}^{BN}$ be a concatenation of coefficient vectors $\{\mathbf{x}_n\}_n$, and $\mathbf{y} \in \mathbb{R}^{BM}$ be a target signal. Then, we can rewrite the problem setting of the dictionary learning as

$$\{\hat{\mathbf{D}}, \hat{\mathbf{x}}\} = \arg \min_{\{\mathbf{D},\mathbf{x}\}} \frac{1}{2}\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 \text{ s.t. } \|\mathbf{x}\|_0 \leq BK, \tag{5}$$

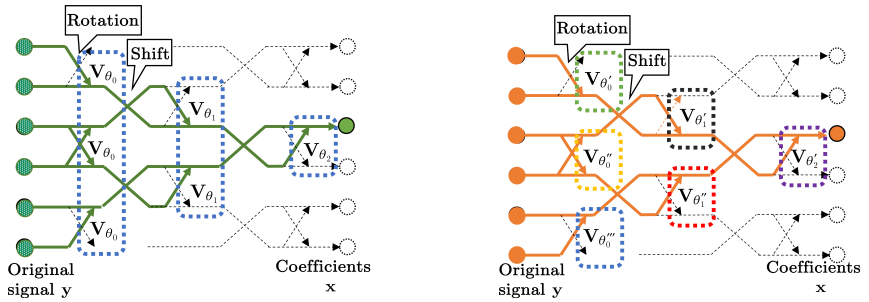
where $K \ll M$, i.e., $K \ll N$. Note that, as special cases, the global distinct block transform matrix can be expressed as $\mathbf{D} = \text{blkdiag}(\mathbf{\Phi}, \mathbf{\Phi}, \cdots, \mathbf{\Phi})$.

A synthesis FB can be regarded as a convolutional dictionary $\mathbf{D}$. A parametric FB enables the design of a dictionary comprising overlapping linear shift-invariant filter kernels with specified strides subject to certain structural constraints [10, 28, 23, 27, 32, 14, 30]. Such FBs can be employed for CDL while incorporating various structural constraints such as unitarity. This, in turn, aligns the discussed CDL models closely with our intended framework which is capable of capturing tangent spaces within a smooth manifold. Nevertheless, the inherent shift-invariance in these FBs limits the direct capture of local coordinate axes. Consequently, the redundant local coordinate system employed by the FBs does not directly correspond to the tangent spaces.

3 Locally-structured Unitary Network

Let us propose a locally-structured unitary network (LSUN). We aim to establish a linear shift-variant network, as shown in Figure 1 (b). Our proposed system is a learnable linear network that satisfies global unitarity $\mathbf{A}^\intercal\mathbf{A} = \mathbf{D}\mathbf{D}^\intercal = \mathbf{I}$.

Figure 4 illustrates the basic idea of our proposal in a simple network, where (a) and (b) denote signal flows of weighted sum operations in a convolutional paraunitary FB (PUFB) and a shift-variant FB with the global unitarity, respectively. Specific examples will be shown in Section 4. The former gives a convolution layer with structural unitary constraints. On the other



(a) (b)

Figure 4: Signal flows of weighted sum operations in (a) convolutional structure and (b) shift-variant structure, where $\mathbf{V}_\theta$ denotes Givens rotation with angle θ.

hand, the latter can realize a local shift-variant system to capture tangent vectors under the unitary constrained structure. Note that both structures can be decomposed into a multilayer network with localized linear layers by block processing with independent design parameters and can be expressed as $\mathbf{D}_{\boldsymbol{\theta}} = \mathbf{F}_0\mathbf{F}_1\cdots\mathbf{F}_{I-2}\mathbf{F}_{I-1}$.

Multi-layer PUFBs will be selected as our baseline, considering its simplicity and ease of understanding. In the following, we summarize the requirements for baseline filter banks:

- Layered structure with independent design parameters,
- Local and unitary processing layers that control overlapping blocks.

In this study, under the assumption that a shift-variable unitary network $\mathbf{D}_{\boldsymbol{\theta}}$ is adopted, we set the following problem:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} -\|\boldsymbol{\Gamma}_{B,K}\mathbf{D}_{\boldsymbol{\theta}}^{\intercal}\mathbf{y}\|_2^2, \text{ s.t. } \mathbf{D}_{\boldsymbol{\theta}}\mathbf{D}_{\boldsymbol{\theta}}^{\intercal} = \mathbf{I}_{BM}, \tag{6}$$

where $\mathbf{D}_{\boldsymbol{\theta}} \in \mathbb{R}^{BM\times BM}$ is a locally-structured synthesis unitary dictionary with shift-variant filter kernels controlled by design parameters in $\boldsymbol{\theta}$, $\boldsymbol{\Gamma}_{B,K} \in \{0,1\}^{BM\times BM}$ is a block diagonal matrix of B diagonal matrices with K diagonal entries of 1 and $(M-K)$ diagonal entries of 0 to reduce dimensionality of the input $\mathbf{y}$, i.e., $K \leq M$. Note that $\|\mathbf{y}\|_2^2 - \|\boldsymbol{\Gamma}_{B,K}\mathbf{D}_{\boldsymbol{\theta}}^{\intercal}\mathbf{y}\|_2^2 \geq 0$ holds from Parseval's identity since $\mathbf{D}_{\boldsymbol{\theta}}$ is unitary. The identity sufficiently holds when the dimension reduction is not applied, i.e., $K = M$.

Figure 5 displays an example configuration of the learning process of LSUN. LSUN can learn local variable filters to preserve the original energy $\|\mathbf{y}\|_2^2$ in low dimensions in a self-supervised manner. Let us denote the cost by

$$\ell_{\mathbf{y}}(\boldsymbol{\theta}) := -\|\boldsymbol{\Gamma}_{B,K}\mathbf{D}_{\boldsymbol{\theta}}^{\intercal}\mathbf{y}\|_2^2 \tag{7}$$

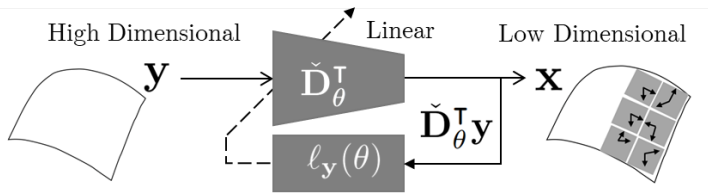


Figure 5: Example configuration of the learning process, where $\mathbf{D}_{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$ are unitary transform and design parameters, respectively. The loss function is given by $\ell_{\mathbf{y}}(\boldsymbol{\theta}) = -\|\boldsymbol{\Gamma}_{B,K}\mathbf{D}_{\boldsymbol{\theta}}^\intercal\mathbf{y}\|_2^2 = -\|\check{\mathbf{D}}_{\boldsymbol{\theta}}^\intercal\mathbf{y}\|_2^2$, where $\check{\mathbf{D}}_{\boldsymbol{\theta}} = \mathbf{D}_{\boldsymbol{\theta}}\mathbf{S}_{B,K}^\intercal$ and $\mathbf{S}_{B,K} \in \{0,1\}^{BK\times BM}$ is the subsampling matrix that holds $\boldsymbol{\Gamma}_{B,K} = \mathbf{S}_{B,K}^\intercal\mathbf{S}_{B,K}$.

If this loss function $\ell_{\mathbf{y}}(\boldsymbol{\theta})$ is differentiable by $\theta_i = [\boldsymbol{\theta}]_i$, $\forall i$, The gradient $\nabla_{\boldsymbol{\theta}}\ell_{\mathbf{y}}(\boldsymbol{\theta})$ is given as

$$[\nabla_{\boldsymbol{\theta}}\ell_{\mathbf{y}}(\boldsymbol{\theta})]_i = \frac{\partial}{\partial\theta_i}\ell_{\mathbf{y}}(\boldsymbol{\theta}) = -2\left\langle \check{\mathbf{D}}_{\boldsymbol{\theta}}^\intercal\mathbf{y}, \frac{\partial}{\partial\theta_i}\check{\mathbf{D}}_{\boldsymbol{\theta}}^\intercal\mathbf{y}\right\rangle \tag{8}$$

where $\langle\cdot,\cdot\rangle$ means the inner product, $\check{\mathbf{D}}_{\boldsymbol{\theta}} := \mathbf{D}_{\boldsymbol{\theta}}\mathbf{S}_{B,K}^\intercal$ and $\mathbf{S}_{B,K} \in \{0,1\}^{BK\times BM}$ is the subsampling matrix that holds $\boldsymbol{\Gamma}_{B,K} = \mathbf{S}_{B,K}^\intercal\mathbf{S}_{B,K}$. We here used the fact that $\boldsymbol{\Gamma}_{B,K}^\intercal\boldsymbol{\Gamma}_{B,K} = \boldsymbol{\Gamma}_{B,K}$.

Using the gradient in (8), we can set the initial value $n \leftarrow 0$, $\boldsymbol{\theta}^{(0)}$ and optimize parameters sequentially, e.g., by gradient descent

$$n \leftarrow n+1, \tag{9a}$$

$$\boldsymbol{\theta}^{(n)} \leftarrow \boldsymbol{\theta}^{(n-1)} - \mu\nabla_{\boldsymbol{\theta}}\ell_{\mathbf{y}}(\boldsymbol{\theta}^{(n-1)}), \tag{9b}$$

where $\mu \in [0,\infty)$ is the step size.

As a result of selecting a simple system, we are able to establish that the parameters $\{\theta_i\}$ are independent of each other and $\mathbf{F}_i$ depends only on θ_i; we obtain,

$$[\nabla_{\boldsymbol{\theta}}\ell_{\mathbf{y}}(\boldsymbol{\theta})]_i = -2\left\langle \mathbf{F}_{i+1}\cdots\mathbf{F}_{I-1}\mathbf{x}(\boldsymbol{\theta}), \frac{\partial}{\partial\theta_i}\mathbf{F}_i^\intercal\mathbf{F}_{i-1}^\intercal\cdots\mathbf{F}_0^\intercal\mathbf{y}\right\rangle \tag{10}$$

from the inner product property $\langle\mathbf{x},\mathbf{A}\mathbf{y}\rangle = \langle\mathbf{A}^\intercal\mathbf{x},\mathbf{y}\rangle$, where $\mathbf{x}(\boldsymbol{\theta}) := \boldsymbol{\Gamma}_{B,K}\mathbf{D}_{\boldsymbol{\theta}}^\intercal\mathbf{y}$. (10) represents a specific instance of the error backpropagation method, and its efficient solution can be achieved by employing a deep learning framework, such as the MATLAB Deep Learning Toolbox or PyTorch in Python.

4 Construction Examples

LSUN can be defined by a distinctive class of convolutional FBs built on cascaded primitive block operations. Some of the candidate systems that

meet the conditions outlined in Section 3 include linear-phase PUFB (LP-PUFB) [32], cosine modulated FBs [23], non-separable oversampled lapped transform (NSOLT) [30], generalized linear-phase lapped orthogonal transform (GenLOT) [10] among others. This section presents two examples of LSUN constructions using existing 1-D and 2-D FBs. Examples of notable differences in 2-D systems are illustrated in Figure 6.

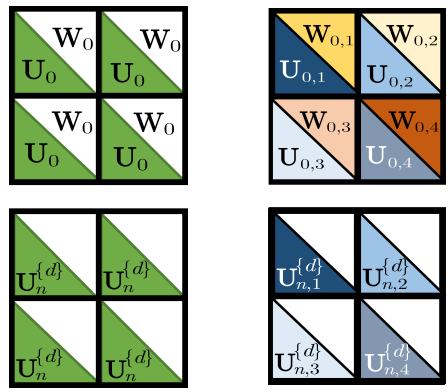


Figure 6: Comparison between (a) shift-invariant unitary parameters $\mathbf{W}_0$, $\mathbf{U}_0$, and $\mathbf{U}_j^{\{d\}}$ of NSCLT, a convolutional FB and (b) the adaptively variable unitary parameter matrices $\mathbf{W}_{0,i}$, $\mathbf{U}_{0,i}$, and $\mathbf{U}_{j,i}^{\{d\}}$ of LSUN, a shift-variant FB.

4.1 1-D LSUN Example

The Gan's PUFBs in [14] can be used as a base system to construct 1-D LSUN. The original PUFB configurations are given by a polyphase representation. For example, the analysis system in even-channel real-coefficient symmetric delay factorization (Real SDF) configuration [14, Fig. 7 (b)] is represented by

$$\mathbf{E}(z) = \mathbf{G}_{J-1}(z)\mathbf{G}_{J-2}(z)\cdots\mathbf{G}_2(z)\mathbf{G}_1(z)\mathbf{E}_0, \tag{11}$$

where the number of channels and resampling ratio are all set to $P = M = 2m$ with a natural number m and

$$\mathbf{G}_j(z) = \mathbf{V}_j\mathbf{\Sigma}_i\mathbf{\Lambda}(z), \tag{12a}$$

$$\mathbf{V}_j = \mathrm{blkdiag}(\mathbf{W}_j, \mathbf{U}_j), \tag{12b}$$

$$\mathbf{\Sigma}_j = \begin{pmatrix} \mathbf{C}_j & -\mathbf{S}_j \\ \mathbf{S}_j & \mathbf{C}_j \end{pmatrix}, \tag{12c}$$

$$\mathbf{C}_j = \mathrm{diag}(\cos\theta_{j,0}, \cos\theta_{j,1}, \cdots, \cos\theta_{j,m-1}), \tag{12d}$$

$$\mathbf{S}_i = \mathrm{diag}(\sin\theta_{j,0}, \sin\theta_{j,1}, \cdots, \sin\theta_{j,m-1}), \tag{12e}$$

$$\mathbf{\Lambda}(z) = \mathrm{blkdiag}(\mathbf{I}_m, z^{-1}\mathbf{I}_m), \tag{12f}$$

$$\mathbf{E}_0 = \mathbf{V}_0\mathbf{\Sigma}_0. \tag{12g}$$

In (11), $J > 1$ controls the number of overlapping local blocks. In (12), $\mathbf{W}_j, \mathbf{U}_j \in \mathbb{R}^{m\times m}$ are parameter orthonormal matrices and $\theta_{j,i}$ is an angular parameter. In the original construction, these parameters are shift-invariant, but in LSUN, they are made shift-variant. Since the polyphase matrix representation consisting of transfer functions is no longer valid, it is represented by a global matrix as in (3a).

4.2 2-D LSUN Example

M-D PUFBs proposed in [32, 15, 30] can also be used as a base system to construct M-D LSUN. For example, the analysis system in 2-D even-channel non-separable oversampled lapped transform (NSOLT) configuration [30, Figs. 3(a) and 4(a)] is expressed in terms of the polyphase matrix by

$$\mathbf{E}(\mathbf{z}) = \left(\prod_{j=1}^{J_\mathrm{h}-1} \mathbf{G}_j^{\{\mathrm{h}\}}(z_\mathrm{h})\right)\left(\prod_{j=1}^{J_\mathrm{v}-1} \mathbf{G}_j^{\{\mathrm{v}\}}(z_\mathrm{v})\right)\mathbf{E}_0, \tag{13}$$

where $\prod_{j=1}^{J-1}\mathbf{G}_j \coloneqq \mathbf{G}_{J-1}\mathbf{G}_{J-2}\cdots\mathbf{G}_2\mathbf{G}_1$, $\mathbf{z} = (z_\mathrm{v}, z_\mathrm{h})^\intercal$,

$$\mathbf{G}_j^{\{d\}}(z_d) = \mathbf{V}_j^{\{d\}}\mathbf{Q}(z_d), \tag{14a}$$

$$\mathbf{V}_j^{\{d\}} = \mathrm{blkdiag}(\mathbf{I}_m, \mathbf{U}_j^{\{d\}}), \tag{14b}$$

$$\mathbf{Q}(z_d) = \mathbf{B}_M\mathbf{\Lambda}(z_d)\mathbf{B}_M, \tag{14c}$$

$$\mathbf{B}_M = \frac{1}{\sqrt{2}}\begin{pmatrix}\mathbf{I}_m & \mathbf{I}_m\\ \mathbf{I}_m & -\mathbf{I}_m.\end{pmatrix}, \tag{14d}$$

$$\mathbf{\Lambda}(z) = \mathrm{blkdiag}(\mathbf{I}_m, z^{-1}\mathbf{I}_m), \tag{14e}$$

$$\mathbf{E}_0 = \mathbf{V}_0\mathbf{C}_\mathbf{M}\mathbf{J}_M, \tag{14f}$$

$$\mathbf{V}_0 = \mathrm{blkdiag}(\mathbf{W}_0, \mathbf{U}_0) \tag{14g}$$

for $d \in \{\mathrm{h}, \mathrm{v}\}$. J_h and J_v control the numbers of overlapping blocks in the horizontal and vertical directions, respectively. It is possible to construct a 2-D LSUN based on this system. Since redundancy is not needed in the LSUN configuration, the number of channels and resampling ratio are all set to $P = M = 2m$ with a natural number m. In other words, since we are based only on the case of critical sampling, we will refer to this base system as the non-separable critically sampled lapped transform (NSCLT) instead of NSOLT.

In (14), $\mathbf{W}_0, \mathbf{U}_0, \mathbf{U}_j^{\{d\}} \in \mathbb{R}^{m\times m}$ are parameter orthonormal matrices, where d denotes either of the horizontal or vertical direction, expressed as 'h' or 'v', respectively. $\mathbf{J}_M \in \mathbb{R}^{M\times M}$ is the counter identity matrix. $\mathbf{C_M} \in \mathbb{R}^{M\times M}$ is an orthonormal matrix that consists of basis vectors satisfying even or odd symmetry. We adopt the 2-D separable DCT for $\mathbf{C_M}$. See Appendix B for further discussion on NSCLT.

In the original construction of NSCLT, the design parameters are shift-invariant, but LSUN adopts a shift-variant construction. Therefore, the polyphase matrix representation is no longer valid. Figure 6 compares primitive block operations with learnable parameters, where (a) and (b) illustrate the shift-invariant construction in NSCLT and the shift-variant construction in LSUN, respectively. Note that both implementations structurally maintain the unitary property, while the operations in Figure 6 (b) realize variability of local block transform. In combination with block shift and butterfly operations, as with convolution, surrounding blocks can be related to each other.

The overlapped block process maintains the continuity of tangent spaces. The adaptive filter kernels can directly capture tangent spaces on a manifold embedded in the high-dimensional space. The unitary property of LSUN enables efficient network training using only the analyzer, eliminating the need for an encoder-decoder structure. Furthermore, specialized training datasets are not necessary, but can be used if desired. LSUN provides a novel linear layer which is an alternative to the conventional convolutional layers.

5 Performance Evaluation of LSUN

Let us evaluate the performance of LSUN through approximation experiments. We aim to investigate the ability of LSUN to capture the underlying low-dimensional structure. The approximation process

$$\tilde{\mathbf{y}} = \mathbf{D}_{\boldsymbol{\theta}} \boldsymbol{\Gamma}_{B,K} \mathbf{D}_{\boldsymbol{\theta}}^{\intercal} \mathbf{y} \tag{15}$$

is applied as shown in Figure 7, where $\mathbf{D}_{\boldsymbol{\theta}} \in \mathbb{R}^{BM\times BM}$ is a synthesis dictionary, and $\mathbf{y} \in \mathbb{R}^{BM}$ is a target signal. The experiments presented in Sections 5.1, 5.2, and 5.3 are conducted to evaluate the performance of 1-D and 2-D LSUN.

5.1 1-D LSUN Experiment

We use the 1-D LSUN built on the Gan's PUFBs [14] in Section 4.1 for the 1-D experimental evaluation.[2]

The task involves approximating the Rosenbrock function

$$f(w_1, w_2) = (a - w_1)^2 + b(w_2 - w_1^2)^2, \tag{16}$$

[2]The code can be viewed at https://codeocean.com/capsule/3430476/tree/v2.

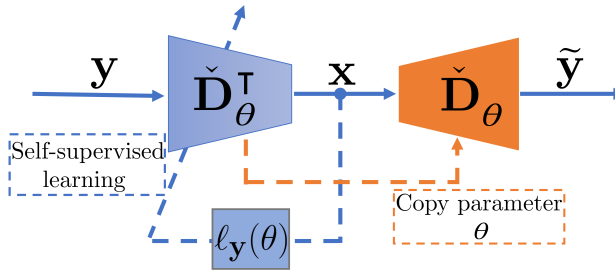


Figure 7: Layout for the approximation experiment.

where $a, b \in \mathbb{R}$ and $w_1, w_2 \in \mathbb{R}$. The function $f(w_1, w_2)$ is discretized on a 2-D grid to generate a dataset as

$$[\mathbf{y}^{(t)}]_\ell := f(\ell \Delta w_1 - 2, t\Delta w_2 - 1) \tag{17}$$

where $\ell, t \in \{0, 1, \cdots, 99\}$, $\Delta w_1 = \Delta w_2 = 0.04$, and $[\cdot]_\ell$ denotes the ℓ-th element of the vector. This 1-D vector sequence is treated as a dataset $\{\mathbf{y}^{(t)}\} \subset \mathbb{R}^{BM}$. The parameters were set to $a = 1$ and $b = 100$.

In the context of our 1-D LSUN experiment, the problem setup described in (6) is modified as follows:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} - \sum_{t=0}^{T-1} \|\boldsymbol{\Gamma}_{B,K} \mathbf{D}_{\boldsymbol{\theta}}^\intercal \mathbf{y}^{(t)}\|_2^2, \text{ s.t. } \mathbf{D}_{\boldsymbol{\theta}} \mathbf{D}_{\boldsymbol{\theta}}^\intercal = \mathbf{I}_{BM}, \tag{18}$$

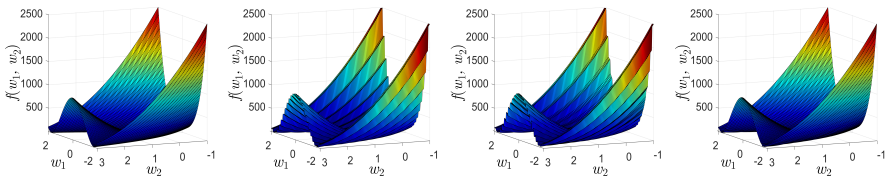
for $\{\mathbf{y}^{(t)}\}_{t=0}^{T-1}$, where $T = 100$. Note that the error back propagation discussed in Section 3 is still valid. As per the boundary operations, circular extensions were applied to signal boundaries within the LSUN process.

The comparison of 1-D LSUN approximation results is presented in Table 2, with visual illustrations provided in Figure 8 and Figure 9, where (a), (b), (c) and (d) illustrate the original representation and the results for GBPCA, LBPCA, and LSUN, respectively. In the 1-D LSUN approximation test, the parameters for all models were configured as follows: the block size (stride) was set to $M = 4$, the number of coefficients per block was chosen as $K \in \{1, 3\}$ for different instances, and the overlapping factor (polyphase order plus one) for LSUN was set to $J = 9$, the number of blocks per training sample for LBPCA was set to $S = 9$, and the number of blocks is $B = 25$.

The experimental results show that the 1-D LSUN performs better than GBPCA and LBPCA, offering insights into their respective abilities to capture the underlying structure of the Rosenbrock function.

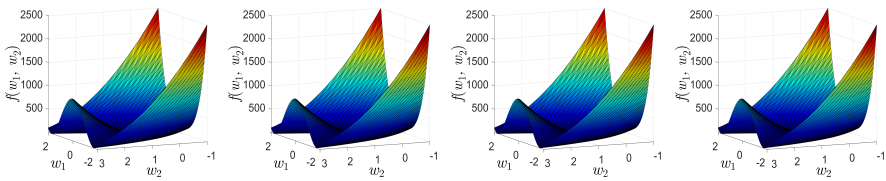
Table 2: Approximation results for the Rosenbrock function in terms of mean squared error (MSE).

K	GBPCA	LBPCA	LSUN
1	2.0301×10^3	1.7213×10^3	4.3499×10^0
3	2.3049×10^{-4}	4.6973×10^{-5}	3.7445×10^{-6}



(a) (b) (c) (d)

Figure 8: Approximation of Rosenbrock function by (a) Original, (b) GBPCA (MSE:2.0301×10^3), (c) LBPCA (MSE:1.7213×10^3), and (d) LSUN (MSE:4.3499×10^0), where $M = 4$, $J = 9$ and the number of coefficients per block is set to $K = 1$.



(a) (b) (c) (d)

Figure 9: Approximation of Rosenbrock function by (a) Original, (b) GBPCA (MSE:2.3049×10^{-4}), (c) LBPCA (MSE:4.6973×10^{-5}), and (d) LSUN (MSE:3.7445×10^{-6}), where $M = 4$, $J = 9$ and the number of coefficients per block is set to $K = 3$.

5.2 2-D LSUN Experiment

We use 2-D LSUN built on NSCLT from the framework of the LPPUFBs [31] for the evaluation. The 2-D LSUN experiment was conducted using image approximation, comparing the performance of GBPCA, LBPCA, NSCLT, NSOLT, and K-SVD, where K-SVD and NSOLT are processed with the normalized iterative hard thresholding (IHT) [3]. The selection of models for evaluation was based on the variation of properties outlined in Table 3. The parameters for each model were configured as follows. For NSCLT, the number of channels $p_\mathrm{s} + p_\mathrm{a} = 8 + 8$, block size $[M_\mathrm{v}, M_\mathrm{h}] = [4, 4]$, the number of overlapping blocks $[J_\mathrm{v}, J_\mathrm{h}] = [9, 9]$ (polyphase order $[N_\mathrm{v}, N_\mathrm{h}] = [8, 8]$ plus

Table 3: Comparison of the properties of the models used in performance evaluation, ✓ and ✗ represent satisfactory or unsatisfactory, respectively.

	Overlap	Shift-variability
GBPCA	✗	✗
LBPCA	✗	✓
NSCLT	✓	✗
K-SVD	✗	✗
NSOLT	✓	✗
LSUN	✓	✓

one for each direction) and, tree levels $\tau = 1$ were used.[3] NSOLT, a redundant extension of M-D LPPUFB, was designed and used as a convolutional dictionary. It was paired with IHT as a method for finding the solution in a greedy manner, as denoted in (5). Type-I NSOLT is used with the following configuration, where the number of channels $p_{\mathrm{s}} + p_{\mathrm{a}} = 34 + 34$, block size $[M_{\mathrm{v}}, M_{\mathrm{h}}] = [4, 4]$, the number of overlapping blocks $[J_{\mathrm{v}}, J_{\mathrm{h}}] = [9, 9]$, and tree levels $\tau = 1$. K-SVD also uses a similar configuration with block size $[M_{\mathrm{v}}, M_{\mathrm{h}}] = [4, 4]$, the number of coefficients per block K is 1 and the number of atoms per dictionary is 38, where there is no overlapping among blocks, i.e. $[J_{\mathrm{v}}, J_{\mathrm{h}}] = [1, 1]$.

The LSUN architecture for this evaluation was based on the reference NSCLT, where the number of channels $p_{\mathrm{s}} + p_{\mathrm{a}} = 8 + 8$. The number of coefficients per block K is 1, block size (stride) is $[M_{\mathrm{v}}, M_{\mathrm{h}}] = [4, 4]$, the number of overlapping blocks for LSUN was set to $[J_{\mathrm{v}}, J_{\mathrm{h}}] = [9, 9]$. Similar to the 1-D experiment, circular extensions are utilized to signal boundaries within the 2-D LSUN process. LBPCA and GBPCA also employed with similar parameters to LSUN, where the block size (stride) is $[M_{\mathrm{v}}, M_{\mathrm{h}}] = [4, 4]$, $K = 1$ and the number of blocks is $B = 128 \times 192 = 24576$. The number of sample blocks per training region for LBPCA was set to $S = 9 \times 9 = 81$.

For the 2-D LSUN approximation, the Kodak dataset consisting of 24 images was used. Each image has a size of 512×768 pixels and a bit depth of 8 bpp. The results are presented in Table 4, while the average for the total dataset is given in Table 5. Figure 10 and Figure 11 visualize the results of the approximation test for the Kodak images kodim01 and kodim21, respectively. It is evident that LSUN outperforms in terms of approximation performance for most of the instances, showing better average performance for the images in the Kodak dataset.

5.3 Dimensional Reduction for Dynamic System Modeling

LSUN can reduce dimensionality and decrease computational complexity in applications involving high-dimensional signals, such as the modeling of

[3] The symbols are inherited from [30, 13].

Table 4: Approximation results for each image in PSNR [dB], where K-SVD and NSOLT are paired with IHT. The number of channels K is set to 1. Best results are highlighted in bold.

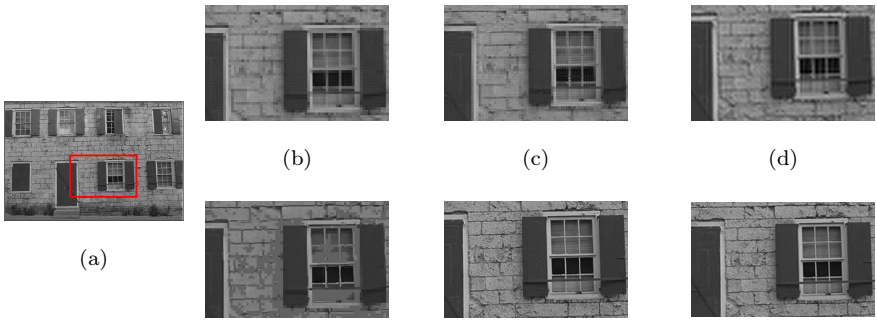
Im	GBPCA	LBPCA	NSCLT	K-SVD	NSOLT	LSUN
01	21.5746	22.2596	22.2499	23.3883	45.5939	**55.5688**
02	28.7108	29.6735	29.2189	34.3887	**51.8891**	49.2904
03	28.4433	29.0207	29.5146	31.1289	**54.1636**	49.5122
04	27.6568	28.3572	28.8698	27.7029	51.0219	**51.0437**
05	20.9029	21.1124	22.2294	20.7751	46.7815	**57.5899**
06	23.4314	23.8015	24.0569	22.0179	48.5984	**52.2242**
07	25.6959	26.0441	27.9085	27.8332	**53.3019**	51.6631
08	18.9501	19.3258	19.9320	18.9712	44.2331	**55.2865**
09	25.7672	26.6639	27.0177	28.1891	**50.6422**	47.1693
10	26.0671	26.6249	27.3315	24.4945	**50.7356**	49.0067
11	24.4906	25.0829	25.3446	25.5364	49.0279	**51.1946**
12	26.6003	27.1958	28.1615	27.2001	**51.7021**	49.4966
13	20.1337	20.3206	20.7878	20.8253	40.6788	**64.0321**
14	23.9734	24.1802	24.9775	24.1972	47.0001	**54.4873**
15	26.6497	27.2781	27.8306	21.1087	**51.6795**	50.3687
16	27.3089	27.7478	27.7075	26.8729	51.0189	**51.9633**
17	26.5895	26.8366	28.1092	25.0858	50.2599	**51.4081**
18	23.4821	23.6324	24.2766	24.5879	45.2982	**51.1779**
19	23.2453	24.1077	23.8718	23.8947	48.3095	**50.0852**
20	25.1995	25.5980	26.4054	26.5395	**52.3848**	51.9855
21	23.6225	23.9749	24.5378	25.0235	48.6529	**48.6653**
22	25.8677	26.4694	26.3343	26.7786	49.1581	**50.1351**
23	28.0388	28.6753	29.2068	26.9834	**52.6607**	50.0164
24	22.7534	22.9423	23.3451	21.8009	46.7303	**52.2141**

Table 5: The average of approximation results for Kodak dataset in PSNR [dB], where K-SVD and NSOLT are paired with IHT. The number of channels K is set to 1. Best result is highlighted in bold.

GBPCA	LBPCA	NSCLT	K-SVD	NSOLT	LSUN
24.7981	25.2886	25.8011	25.2218	49.2301	**51.8994**

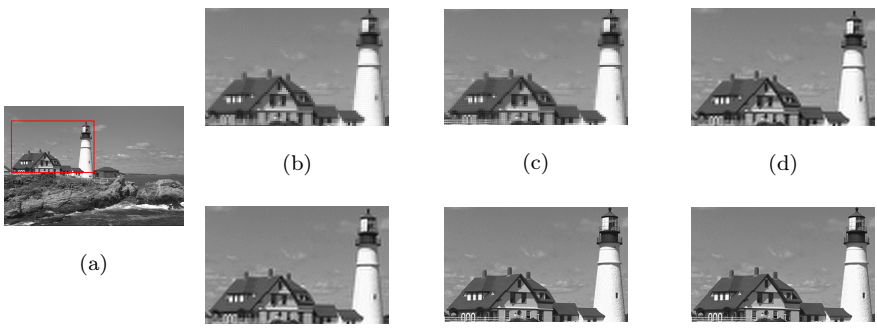
dynamical systems. As well, LSUN is useful for acquiring essential features from contaminated data.

Recent advancements in dynamical system modeling, such as physics-informed dynamic mode decomposition (piDMD) [2], have shown promising results under some physical constraints. We utilize LSUN to assist the proper orthogonal decomposition (POD) [6] before piDMD and to improve the performance of piDMD for modeling conservative systems.



(a) (b) (c) (d) (e) (f) (g)

Figure 10: Results of the approximation experiments for the image kodim01, where (a) Original, (b) GBPCA (PSNR:21.574 dB), (c) LBPCA (PSNR:22.259 dB), (d) NSCLT (PSNR:22.249 dB), (e) K-SVD (PSNR:23.388 dB), (f) NSOLT (PSNR:45.593 dB), (g) LSUN (PSNR:55.568 dB).



(a) (b) (c) (d) (e) (f) (g)

Figure 11: Results of the approximation experiments for the image kodim21, where (a) Original, (b) GBPCA (PSNR:23.622 dB), (c) LBPCA (PSNR:23.974 dB), (d) NSCLT (PSNR:24.537 dB), (e) K-SVD (PSNR:25.023 dB), (f) NSOLT (PSNR:48.652 dB), (g) LSUN (PSNR:48.665 dB).

We conducted a conservative dynamical system modeling with orthogonal DMD, which is a type of piDMD, and 2-D LSUN from contaminated noisy simulation data. The following steps show our evaluation procedure.

Step 1: Generate noisy observation data $\{\mathbf{v}_k\}$ from clean data $\{\mathbf{u}_k\}$

Step 2: Train LSUN $\check{\mathbf{D}}_{\boldsymbol{\theta}}^{\mathsf{T}}$ with $\{\mathbf{v}_k\}$

Step 3: Project coefficients $\{\check{\mathbf{D}}_{\boldsymbol{\theta}}^{\mathsf{T}}\mathbf{v}_k\}$ to the leading POD modes $\boldsymbol{\Psi}$

Step 4: Train piDMD $\mathbf{K}$ from $\{\boldsymbol{\Psi}^{\mathsf{T}}\check{\mathbf{D}}_{\boldsymbol{\theta}}\mathbf{v}_k\}$ under unitary constraint

Step 5: Develop time-evolution equation $\mathbf{u}_{k+1} \simeq \check{\mathbf{D}}_{\boldsymbol{\theta}}^{\mathsf{T}}\boldsymbol{\Psi}\mathbf{K}\boldsymbol{\Psi}^{\mathsf{T}}\check{\mathbf{D}}_{\boldsymbol{\theta}}\mathbf{u}_k$

Step 6: Estimate $\mathbf{u}_k$ for $k > 0$ from the initial noisy observation $\mathbf{v}_0$

Step 7: Evaluate the estimation $\{\hat{\mathbf{u}}_k\}$ in terms of MSE with $\{\mathbf{u}_k\}$

In Step 1, to investigate the incompressible flow around a cylinder, a standard problem for modal decomposition is adopted [24]. The analysis focuses on a system comprising 151 vorticity measurement samples,[4] representing five periods of vortex shedding, collected on a grid with dimensions of 199×449 points. The observation $\mathbf{v}_k$ is assumed to be contaminated by additive white Gaussian noise (AWGN) $\mathbf{w}_k$ as

$$\mathbf{v}_k = \mathbf{u}_k + \mathbf{w}_k, \tag{19}$$

where $\mathbf{u}_k$ denotes the original data at the k-th frame. Throughout the experiment, the signal-to-noise ratio (SNR) varies from 10% to 50%.

In this evaluation, we try to obtain the following time-evolution equation in a data-driven manner:

$$\hat{\mathbf{u}}_k = \check{\mathbf{D}}_{\boldsymbol{\theta}}(\boldsymbol{\Psi}^\intercal \hat{\mathbf{K}} \boldsymbol{\Psi})^k \check{\mathbf{D}}_{\boldsymbol{\theta}}^\intercal \mathbf{v}_0, \tag{20}$$

where $\boldsymbol{\Psi}$ denotes the leading POD modes obtained by SVD for LSUN coefficient data matrix

$$\mathbf{X}_0 \coloneqq \check{\mathbf{D}}_{\boldsymbol{\theta}}^\intercal \begin{pmatrix} \mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_{149} \end{pmatrix}, \tag{21}$$

and $\hat{\mathbf{K}}$ is a matrix given by solving the problem

$$\hat{\mathbf{K}} = \arg \min_{\mathbf{K} \text{ s.t. } \mathbf{K}^\intercal \mathbf{K} = \mathbf{K}\mathbf{K}^\intercal = \mathbf{I}} \|\mathbf{X}_1 - \boldsymbol{\Psi}^\intercal \mathbf{K} \boldsymbol{\Psi} \mathbf{X}_0\|_F^2, \tag{22}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and

$$\mathbf{X}_1 \coloneqq \check{\mathbf{D}}_{\boldsymbol{\theta}}^\intercal \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_{150} \end{pmatrix}. \tag{23}$$

In Step 2, the block size (stride) is fixed at $[M_\mathrm{v}, M_\mathrm{h}] = [4, 4]$. The number of channels K is varied from 1 to 4, and overlapping blocks $[J_\mathrm{v}, J_\mathrm{h}]$ is varied for $J_\mathrm{v} = J_\mathrm{h} \in \{1, 3, 5\}$. By varying the number of channels used for LSUN reconstruction, the spatial compression can be controlled by a factor of $R = K/(M_\mathrm{v} \times M_\mathrm{h})$, thereby relaxing the computational complexity of the SVD process in piDMD. In Step 3, the number of the leading POD modes is fixed to 15 for all the cases following the original piDMD construction [2] i.e., $\hat{\mathbf{K}} \in \mathbb{R}^{15 \times 15}$. In Step 4, the orthogonal piDMD framework is used for obtaining $\hat{\mathbf{K}}$, then Step 5 follows.

The estimated results in Step 6 are presented in Figure 12, which compares the performance of piDMD with LSUN and the original piDMD, i.e., the case that $\check{\mathbf{D}}_{\boldsymbol{\theta}} = \mathbf{I}$. In Step 7, the detailed results are provided in Tables 6 and 7.

[4]The original data is obtained from: http://databookuw.com/DATA.zip.

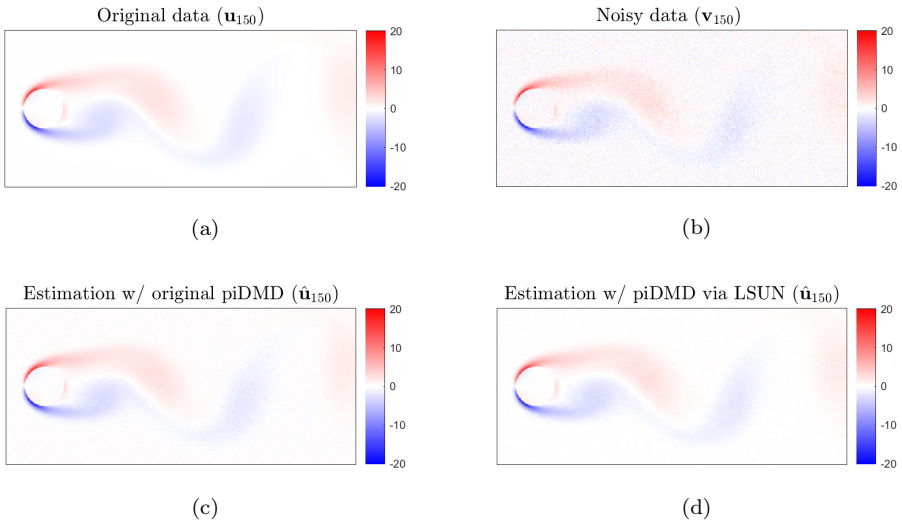


Figure 12: Estimation with noisy data using piDMD with and without LSUN: (a) Source data $\mathbf{u}_{150}$, (b) Noisy data $\mathbf{v}_{150}$ with noise energy set to 50% of signal energy (MSE: 0.6283), (c) Estimation using piDMD $\hat{\mathbf{u}}_{150}$ (MSE: 0.0649), (d) Estimation using piDMD with LSUN $\hat{\mathbf{u}}_{150}$ (MSE: 0.0110). LSUN is configured as the number of channels $K = 2$ and the number of overlapping blocks $[J_\mathrm{v}, J_\mathrm{h}] = [3,3]$.

Table 6 demonstrates the performance of piDMD with LSUN for varying the numbers of channels K and overlapping factors $[J_\mathrm{v}, J_\mathrm{h}]$. Table 7 presents a comparison of piDMD with and without LSUN under different noise energy levels. All the values are presented in terms of MSE at the 150^{th} estimation $\hat{\mathbf{u}}_{150}$ with the source $\mathbf{u}_{150}$. Note that the MSE values are taken from the average of five independent trials since the noise $\mathbf{w}_k$ is random. The results demonstrate that LSUN introduces spatial compression to the data while preserving the essential data structure of the system.

Figure 13 presents a comparison of the learned spectra between the original piDMD and piDMD via LSUN. It is apparent that the eigenvalue distributions are on the unit circle in both cases, that the orthogonal piDMD requests. Figure 14 shows the comparison of estimation averaged MSEs with original piDMD and piDMD via LSUN. Note that there is no estimation for the initial frame $\mathbf{u}_0$. The plot starts from the first frame $\hat{\mathbf{u}}_1$. In addition to POD, LSUN can improve the low-dimensional representation by providing spatial compression to the data. Incorporating LSUN with piDMD is expected to increase the efficiency of conservative dynamical system modeling.

The LSUN demonstrates superior performance in both 1-D and 2-D approximation experiments and exhibits enhanced performance in dynamical system modeling when integrated with piDMD. The efficacy of its shift-variant, unitary

Table 6: Estimation performance of piDMD with LSUN for varying numbers of coefficients K and overlapping blocks $[J_{\mathrm{v}}, J_{\mathrm{h}}]$. Noise energy is set to 50% of signal energy. R indicates the factor of spatial compression. All the values are presented in terms of averaged MSE processed for five trials. Indicated MSE values represent the last 150^{th} frame of estimated data.

	K	R	$[J_{\mathrm{v}}, J_{\mathrm{h}}]$	MSE
Original piDMD	-	-	-	0.0637
piDMD via LSUN	1	1/16	[1,1]	0.0894
	2	2/16	[1,1]	0.0132
	3	3/16	[1,1]	0.0180
	4	4/16	[1,1]	0.0218
	2	2/16	[3,3]	0.0121
	2	2/16	[5,5]	0.0120

Table 7: Comparison of estimation of a dynamical system with piDMD w/o LSUN and piDMD w/ LSUN against different noise levels, where LSUN is configured as the number of channels $K = 2$ and the number of overlapping blocks $[J_{\mathrm{v}}, J_{\mathrm{h}}] = [3,3]$. All the values are presented in terms of averaged MSE processed for five trials. Indicated MSE values represent the last 150^{th} frame estimated data.

SNR	Original piDMD	piDMD via LSUN
10%	0.0136	0.0039
20%	0.0259	0.0059
30%	0.0339	0.0079
40%	0.0591	0.0112
50%	0.0637	0.0121

structure in capturing low-dimensional structures from high-dimensional data is confirmed by the provided examples. Additionally, due to its local structure and linearity, LSUN is well-suited for parallel processing of large-scale problems, similar to convolutional structures, utilizing overlap-add or overlap-save methods.

6 Conclusion

This paper proposed a locally-structured linear dictionary for tangent space sampling. The proposed model was validated through approximation and dynamical system modeling experiments. The shift-invariant kernels of the convolutional dictionary were replaced with shift-variant ones while preserving the locality, unitarity, and overlapping properties. The introduction of adaptively variable filters throughout the dataset may lead to more learnable parameters than the CDL counterpart, depending on the configuration. However, the

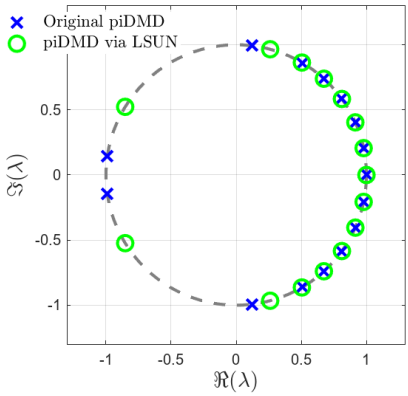


Figure 13: Comparison of learned spectra for original piDMD and piDMD via LSUN, where SNR is set at 50%, and the number of channels $K = 2$, the number of overlapping blocks $[J_\mathrm{v}, J_\mathrm{h}] = [3,3]$, and λ denotes the eigenvalues of $\mathbf{K}$.

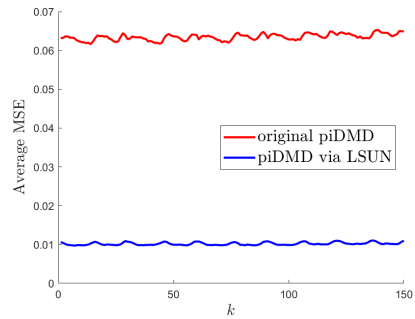


Figure 14: Comparison of average MSE for estimation with original piDMD and piDMD via LSUN, where SNR is set at 50%, the number of channels $K = 2$, and the number of overlapping blocks $[J_\mathrm{v}, J_\mathrm{h}] = [3,3]$.

encoder-based self-supervised learning approach avoids coordinate selection mechanisms, such as sparse approximation. The proposed architecture was inspired by M-D LPPUFB, and the design parameters were adaptively varied block by block. We achieved a better low-dimensional representation than K-SVD, NSCLT, and NSOLT. The LSUN introduces a novel, data-driven, self-supervised learnable network designed for tangent space sampling. While this study is evaluated with 1-D and 2-D data approximation and dynamic system modeling examples, the underlying idea exhibits the potential for broader applications, extending to other signal types.

For future research directions, we propose investigating the application of LSUN in modeling nonlinear conservative dynamic systems. Another potential application of LSUN is realizing invertible networks such as normalizing flows [18]. We can replace the convolutional layer with LSUN to improve the performance of the network.

Appendices

A Convolutional Dictionary Learning

A convolutional dictionary offers efficient local feature extraction and signal representation similar to block processing. By constructing a composite dictionary with a synthesis FB and parameterizing it, we can design a learnable

convolutional dictionary that encourages sparsity for specific data instances, similar to methods such as PCA and K-SVD. In this paper, the CDL problem is formulated as (5). Typical dictionary learning alternates between sparse approximation and dictionary update as in K-SVD [12, 1, 30].

- Sparse approximation: Solve the following problem for the convolutional dictionary $\hat{\mathbf{D}}$:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \hat{\mathbf{D}}\mathbf{x}\|_2^2 \text{ s.t. } \|\mathbf{x}\|_0 \leq BK. \tag{24}$$

  (24) is NP-hard; therefore, heuristic methods such as orthogonal matching pursuit (OMP) [35] and iterative hard thresholding (IHT) [3] are used.

- Dictionary update: Solve the following problem for the sparse coefficients $\hat{\mathbf{x}}$:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\|\mathbf{y} - \mathbf{D}_{\boldsymbol{\theta}}\hat{\mathbf{x}}\|_2^2, \tag{25}$$

$$\hat{\mathbf{D}} = \mathbf{D}_{\hat{\boldsymbol{\theta}}}, \tag{26}$$

  where $\boldsymbol{\theta}$ is a vector representation of the design parameter set.

A.1 Sparse Approximation Step

The objective of this step is to find a sparse representation of the high-dimensional signal using the given convolution dictionary. Because the atoms of $\mathbf{D}$ are utilized in a manner that allows for local overlap while contributing to the overall representation of $\mathbf{y}$, it is desirable to employ a feasible sparse approximation technique available for high-dimensional signals. IHT is a good candidate [3].

A.2 Dictionary Update Step

The purpose of this step is to find a convolutional dictionary $\mathbf{D}$ that provides a better approximation of the high-dimensional signal $\mathbf{y} \in \mathbb{R}^{BM}$ using the given sparse coefficients $\mathbf{x} \in \mathbb{R}^{BN}$, or their support $\text{supp}(\mathbf{x})$.

If the convolutional dictionary can be controlled by the design parameters $\boldsymbol{\theta}$, we can define the loss function as

$$\ell_{\mathbf{y}}(\boldsymbol{\theta}) \coloneqq \frac{1}{2}\|\mathbf{r}_{\mathbf{y}}(\boldsymbol{\theta})\|_2^2 \tag{27}$$

with the approximation error

$$\mathbf{r}_{\mathbf{y}}(\boldsymbol{\theta}) \coloneqq \mathbf{y} - \mathbf{D}_{\boldsymbol{\theta}}\hat{\mathbf{x}}. \tag{28}$$

When the dictionary $\mathbf{D}_{\boldsymbol{\theta}}$ is factorable into matrix products and the design parameters of each building block matrix are independent, then some deep learning framework can be used for optimization design.

A.3 Extension to Training Data Set

The above discussion on CDL can be extended to the problem

$$\{\hat{\mathbf{D}}, \{\hat{\mathbf{x}}^{(t)}\}\} = \arg \min_{\{\mathbf{D}, \{\mathbf{x}^{(t)}\}\}} \frac{1}{2} \sum_{t=0}^{T-1} \|\mathbf{y}^{(t)} - \mathbf{D}\mathbf{x}^{(t)}\|_2^2 \text{ s.t. } \|\mathbf{x}^{(t)}\|_0 \leq BK \quad (29)$$

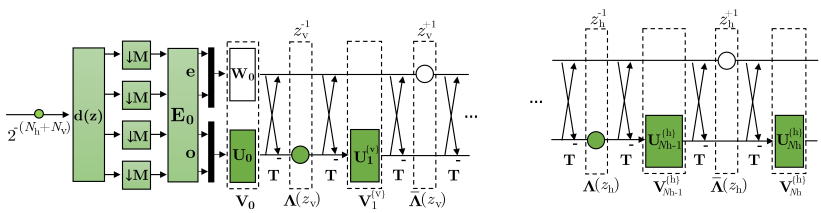
that assumes a training data set $\{(\mathbf{y}^{(t)}, \mathbf{x}^{(t)})\}_{t=0}^{T-1}$ with the loss function $\ell(\mathbf{D}) := \frac{1}{2} \sum_{t=0}^{T-1} \|\mathbf{y}^{(t)} - \mathbf{D}\hat{\mathbf{x}}^{(t)}\|_2^2$ in the dictionary update step.

B Nonseparable Critically Sampled Lapped Transform (NSCLT)

The advantage of data-driven dictionary design lies in the ability to tailor atoms to specific data and applications. The following outlines the design method of NSCLT as an illustrative example of a convolutional dictionary learning procedure. NSCLT is the critically sampled implementation of NSOLT [30].

B.1 Lattice Structure

NSCLT is a convolutional dictionary composed of a lattice structure. The filter kernels satisfy the properties of non-separable, symmetric, real coefficients and overlapping [30]. In addition, structural constraints of Parseval tight and no DC-leakage property can be realized. Figure 15 illustrates the lattice structure of the NSCLT analyzer. In the following, for convenience, we assume that the number of overlapping blocks in each dimension d is even. Figure 15 is given by the polyphase matrix representation in (11).



(a) Initial building block and vertical extension

(b) Horizontal extension

Figure 15: Lattice structure of NSCLT analyzer, where $\mathbf{d}(\mathbf{z})$ is the delay chain determined by the downsampling factor $\mathbf{M} = \mathrm{diag}(M_\mathrm{v}, M_\mathrm{h})$. $\mathbf{E_0}$ denotes the orthonormal transformation matrix which is directly given by 2-D DCT and $\mathbf{W}_0$, $\mathbf{U}_0$, $\mathbf{U}_i^{\{d\}}$ are the parameter matrices, where d denotes the direction of extension $d \in \{\mathrm{h}, \mathrm{v}\}$. $\mathbf{V}_0$, $\mathbf{V}_i^{\{d\}}$, $\mathbf{\Lambda}(z_d)$, $\overline{\mathbf{\Lambda}}(z_d)$ and $\mathbf{T}$ denotes the block butterfly operations illustrated in Figure 16.

B.2 Block Processing

Each stage of Figure 15 can be realized by primitive block processing as shown in Figure 16. Blocks (a) to (h) represent the primitive block operations for the synthesis process of the 2-D NSCLT [31, 13]. Combining these primitive operations makes it possible to parametrically design convolutional dictionaries with structural constraints such as unitary. The upper and lower triangles correspond to the upper and lower half channels of Figure 15, respectively. The processing in Figure 16 (a)–(d) are non-overlapping blocks that are processed independently of other blocks. On the other hand, Figure 16 (e)-(h) illustrates the coefficient shift between adjacent blocks. It serves as a memory of the polyphase matrix and provides dependencies between blocks.

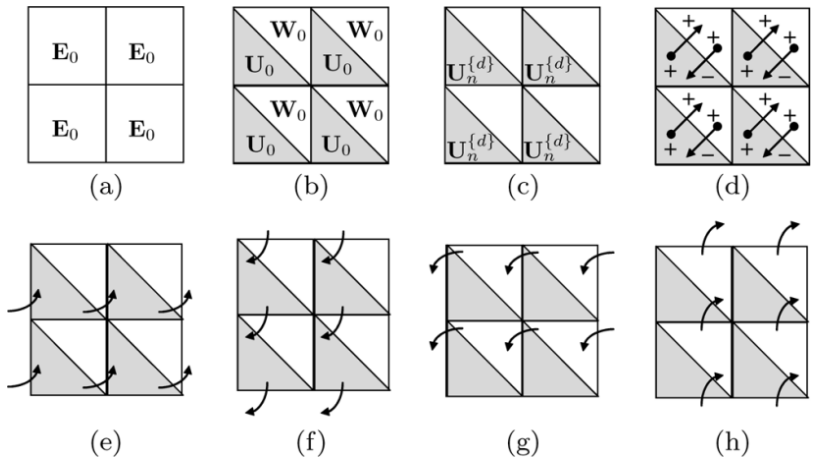


Figure 16: Primitive block operations of 2-D LPPUFBs. $\mathbf{W}_0$, $\mathbf{U}_0$ and $\{\mathbf{U}_j^{\{d\}}\}_{j,d}$ denote parameter orthonormal matrices, where j and d are indices of stages and directions, respectively. (a) a symmetric orthonormal transform, (b), (c) rotation operations, (d) butterfly operation, and (e)-(h) shift operations. ©IEEE Reprinted, with permission, from S. Muramatsu *et al.*, "Boundary Operation of 2-D Nonseparable Linear-Phase Paraunitary Filter Banks," in IEEE Transactions on Image Processing, vol. 21, no. 4, pp. 2314-2318, April 2012, doi:10.1109/TIP.2011.2181527 [31].

B.3 Parameterization

When the parameter matrix $\{\mathbf{U}_j^{\{d\}}\}$ is invertible, $\mathbf{W}_0$ and $\mathbf{U}_0$ are right invertible. Hence, the NSOLT synthesis dictionary $\mathbf{D}_{\boldsymbol{\theta}}$ is right invertible. Furthermore, $\mathbf{R}(\mathbf{z})$ is PU, and NSCLT synthesis dictionary $\mathbf{D}_{\boldsymbol{\theta}}$ is Parseval-tight if these parameter matrices satisfy unitary, i.e., $\mathbf{U}_j^{\{d\}\mathsf{T}}\mathbf{U}_j^{\{d\}} = \mathbf{I}_{P/2}$, $\mathbf{W}_0^{\mathsf{T}}\mathbf{W}_0 = \mathbf{I}_{\lceil M/2 \rceil}$,

$\mathbf{U}_0^\intercal \mathbf{U}_0 = \mathbf{I}_{\lfloor M/2 \rfloor}$. The $p \times p$ orthogonal matrix $\mathbf{U}$ can be decomposed into $p(p-1)/2$ Givens rotations and p signs.

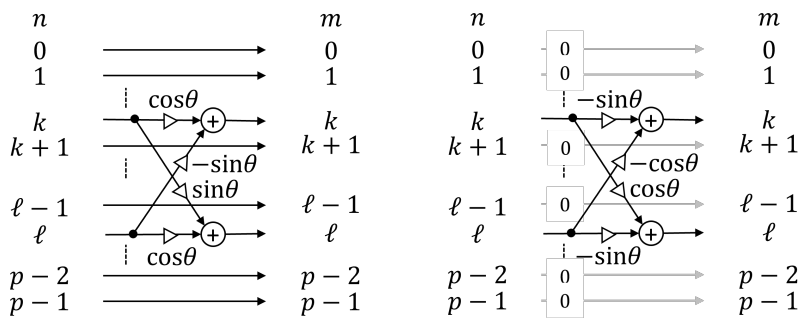
$$\mathbf{U} = \mathrm{diag}\,(s_0, s_1, \cdots, s_{p-1}) \prod_{k=0}^{p-2} \prod_{\ell=k+1}^{p-1} \mathbf{T}_{k,\ell}(\theta_{i(k,\ell)}) \tag{30}$$

where $s_i \in \{-1, 1\}$,

$$[\mathbf{T}_{k,\ell}(\theta)]_{m,n} = \begin{cases} \cos\theta, & m = n \cap (m = k \cup m = \ell) \\ -\sin\theta, & m = k \cap n = \ell \\ \sin\theta, & m = \ell \cap n = k \\ 1, & m = n \cap (m \neq k \cup m \neq \ell) \\ 0, & \text{otherwise} \end{cases} \tag{31}$$

Let $i(k,\ell) \neq i(k',\ell')$ when $k \neq k' \cup \ell \neq \ell'$.

Figure 17 (a) shows an example of Givens rotation configuration. $\theta \in \mathbb{R}$ is the angle of rotation. This structure allows for unitary constraints on the parameter matrix. The right invertible unitary matrix is given by $\mathbf{U}^\intercal (\mathbf{I} \quad \mathbf{O})$. The $\mathbf{T}_{k,\ell}(\theta)$ in (31) is differentiable with θ, and has the structure of Figure 17 (b). This can be used for the error backpropagation method in (10).



(a) $\mathbf{T}_{k,\ell}(\theta)$ (b) $\frac{\partial}{\partial\theta}\mathbf{T}_{k,\ell}(\theta)$

Figure 17: Example of Givens rotation configuration.

Acknowledgements

The authors would like to thank the associate editor and anonymous reviewers for their valuable comments and suggestions on improving the quality of the paper.

Biographies

Yasas GODAGE received the B.S. degree in Mechatronics Engineering from General Sir John Kotelawala Defence University, Sri-Lanka in 2017 and the M.E. degree in Electrical and Electronic Engineering from Niigata University, Japan in 2021. He is currently a Ph.D. candidate at Niigata University. His research interests include image/video restoration, structured networks, and deep learning.

Eisuke KOBAYASHI received the B.S. degree in Electrical and Electronic Engineering from Niigata University, Japan in 2022 and the M.E. degree in Electrical and Electronic Engineering from Niigata University, Japan in 2024. His research interests include image/video restoration, structured networks, deep learning, time series prediction.

Shogo MURAMATSU (Member, APSIPA) received the B.E., M.E., and Ph.D. degrees from Tokyo Metropolitan University, Tokyo, Japan, in 1993, 1995, and 1998, respectively. From 1997 to 1999, he was with Tokyo Metropolitan University. In 1999, he joined Niigata University, Niigata, Japan, where he is currently a Professor with the Faculty of Engineering. From 2003 to 2004, he was a Visiting Researcher with the University of Florence, Firenze, Italy. His research interests include cyber-physical systems, multidimensional signal processing, image restoration, video analysis, and embedded vision systems. Prof. Muramatsu is a senior member of the Institute of Electrical and Electronics Engineers, a senior membor of the Institute of Electronics, Information, and Communication Engineers of Japan, and a member of the Institute of Image Information and Television Engineers of Japan. He was an Associate Editor for IEEE Transactions on Signal Processing from 2019 to 2023 and was an APSIPA distinguished Lecturer from 2020 to 2021.

References

[1] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation", *IEEE Transactions on Signal Processing*, 54(11), November 2006, 4311–22, ISSN: 1053587X, DOI: 10.1109/TSP.2006.881199.

[2] P. J. Baddoo, B. Herrmann, B. J. McKeon, J. Nathan Kutz, and S. L. Brunton, "Physics-informed dynamic mode decomposition", *Proceedings of the Royal Society A*, 479(2271), March 2023, 20220576, ISSN: 14712946, DOI: 10.1098/RSPA.2022.0576, https://royalsocietypublishing.org/doi/10.1098/rspa.2022.0576.

[3] T. Blumensath and M. E. Davies, "Normalized iterative hard thresholding: Guaranteed stability and performance", *IEEE Journal on Selected Topics in Signal Processing*, 4(2), April 2010, 298–309, ISSN: 19324553, DOI: 10.1109/JSTSP.2010.2042411.

[4] H. Bristow, A. Eriksson, and S. Lucey, "Fast convolutional sparse coding", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, 391–8, DOI: 10.1109/CVPR.2013.57.

[5] S. L. Brunton, J. N. Kutz, and, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, Cambridge University Press, 2022.

[6] A. Chatterjee, "An introduction to the proper orthogonal decomposition", *Current science*, 2000, 808–17.

[7] H. Chen, M. D. Sacchi, and J. Gao, "Parametric Convolutional Dictionary Learning and Its Applications to Seismic Data Processing", *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[8] J. Chen and Y. Liu, "Locally linear embedding: a survey", *Artificial Intelligence Review*, 36, 2011, 29–48.

[9] I. Y. Chun and J. A. Fessler, "Convolutional Dictionary Learning: Acceleration and Convergence", *IEEE Transactions on Image Processing*, 27(4), April 2018, 1697–712, ISSN: 10577149, DOI: 10.1109/TIP.2017.2761545.

[10] R. L. De Queiroz, T. Q. Nguyen, and K. R. Rao, "The GenLOT: Generalized Linear-Phase Lapped Orthogonal Transform", *IEEE Transactions on Signal Processing*, 44(3), 1996, 497–507.

[11] M. Du, S. Zhang, and H. Wang, "Supervised isomap for plant leaf image classification", in *Emerging Intelligent Computing Technology and Applications. With Aspects of Artificial Intelligence: 5th International Conference on Intelligent Computing, ICIC 2009 Ulsan, South Korea, September 16-19, 2009 Proceedings 5*, Springer, 2009, 627–34.

[12] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Vol. 2, Springer, 2010.

[13] K. Furuya, S. Hara, K. Seino, and S. Muramatsu, "Boundary operation of 2D non-separable oversampled lapped transforms", *APSIPA Transactions on Signal and Information Processing*, 5 (April), April 2016, ISSN: 20487703, DOI: 10.1017/ATSIP.2016.3.

[14] L. Gan and K. K. Ma, "On Simplified Order-One Factorizations of Paraunitary Filterbanks", *IEEE Transactions on Signal Processing*, 52(3), March 2004, 674–86, ISSN: 1053587X, DOI: 10.1109/TSP.2003.822356.

[15] L. Gan and K.-K. Ma, "A simplified lattice factorization for linear-phase perfect reconstruction filter bank", *IEEE Signal Processing Letters*, 8(7), 2001, 207–9, DOI: 10.1109/97.928679.

[16] Y. Godage and S. Muramatsu, "Tangent Space Sampling of Video Sequence with Locally Structured Unitary Network", in *2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, IEEE, 2023, 1–5.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, MIT press, 2016.

[18] P. L. Hagemann, J. Hertrich, and G. Steidl, *Generalized normalizing flows via Markov chains*, Cambridge University Press, 2023.

[19] C. R. Helmrich and B. Edler, "Audio coding using overlap and kernel adaptation", *IEEE Signal Processing Letters*, 23(5), 2016, 590–4.

[20] N. Kambhatla and T. K. Leen, "Dimension reduction by local principal component analysis", *Neural computation*, 9(7), 1997, 1493–516.

[21] Z. Ke, Z.-X. Cui, W. Huang, J. Cheng, S. Jia, L. Ying, Y. Zhu, and D. Liang, "Deep manifold learning for dynamic MR imaging", *IEEE Transactions on Computational Imaging*, 7, 2021, 1314–27.

[22] K. U. Kiran, E. Kalpana, P. S. Nair, S. H. Ahammad, and K. Saikumar, "MRI High Dimensional Data and Statistical Analysis on Spinal Cord Injury Detection", in *Handbook of Research on Advancements in AI and IoT Convergence Technologies*, IGI Global, 2023, 131–44.

[23] R. D. Koilpillai and P. P. Vaidyanathan, "Cosine-Modulated FIR Filter Banks Satisfying Perfect Reconstruction", *IEEE Transactions on Signal Processing*, 40(4), 1992, 770–83.

[24] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*, SIAM, 2016.

[25] A. Liu, Y. Zhang, E. Gehan, and R. Clarke, "Block principal component analysis with application to gene microarray data classification", *Statistics in Medicine*, 21(22), November 2002, 3465–74, ISSN: 02776715, DOI: 10.1002/sim.1263.

[26] B. D. Liu, Y. X. Wang, Y. J. Zhang, and B. Shen, "Learning dictionary on manifolds for image classification", *Pattern Recognition*, 46(7), July 2013, 1879–90, ISSN: 00313203, DOI: 10.1016/j.patcog.2012.11.018.

[27] H. S. Malvar, "Extended Lapped Transforms: Properties Applications, and Fast Algorithms", *IEEE Transactions on Signal Processing*, 40(11), 1992, 2703–14, ISSN: 19410476, DOI: 10.1109/78.165657.

[28] H. S. Malvar and D. H. Staelin, "The LOT: Transform coding without blocking effects", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(4), 1989, 553–9, ISSN: 00963518, DOI: 10.1109/29.17536.

[29] J. Masci, U. Meier, D. C. Ciresan, and J. Schmidhuber, "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction", *Artificial Neural Networks and Machine Learning–ICANN 2011*, 2011, 52–9.

[30] S. Muramatsu, K. Furuya, and N. Yuki, "Multidimensional Nonseparable Oversampled Lapped Transforms: Theory and Design", *IEEE Transactions on Signal Processing*, 65(5), March 2017, 1251–64, ISSN: 1053587X, DOI: 10.1109/TSP.2016.2633240.

[31] S. Muramatsu, T. Kobayashi, M. Hiki, and H. Kikuchi, "Boundary operation of 2-D nonseparable linear-phase paraunitary filter banks", *IEEE Transactions on Image Processing*, 21(4), April 2012, 2314–8, ISSN: 10577149, DOI: 10.1109/TIP.2011.2181527.

[32] S. Muramatsu, A. Yamada, and H. Kiya, "A design method of multidimensional linear-phase paraunitary filter banks with a lattice structure", *IEEE Transactions on Signal Processing*, 47(3), 1999, 690–700.

[33] D. H. Ochoa, K. R. Rao, and, *Discrete Cosine Transform*, 2nd ed., CRC Press, 2019.

[34] V. Papyan, Y. Romano, J. Sulam, and M. Elad, "Convolutional Dictionary Learning via Local Processing", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, 5296–304.

[35] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition", in *Proceedings of 27th Asilomar conference on signals, systems and computers*, IEEE, 1993, 40–4.

[36] G. Peyré, "Manifold models for signals and images", *Computer Vision and Image Understanding*, 113(2), 2009, 249–60, ISSN: 1090235X, DOI: 10.1016/j.cviu.2008.09.003.

[37] J. Princen and A. Bradley, "Analysis/synthesis filter bank design based on time domain aliasing cancellation", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(5), 1986, 1153–61.

[38] J. Princen, A. Johnson, and A. Bradley, "Subband/transform coding using filter bank designs based on time domain aliasing cancellation", in *ICASSP'87. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 12, IEEE, 1987, 2161–4.

[39] A. Raid, W. Khedr, E.-d. M.A, and W. Ahmed, "JPEG Image compression using discrete cosine transform - A survey", *International Journal of Computer Science & Engineering Survey*, 5(2), April 2014, 39–47, ISSN: 09763252, DOI: 10.5121/ijcses.2014.5204.

[40] P. Shamsolmoali and M. Zareapoor, "Entropy Transformer Networks: A Learning Approach via Tangent Bundle Data Manifold", in *2023 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2023, 1–8.

[41] S. Shlien, "The modulated lapped transform, its time-varying forms, and its applications to audio coding standards", *IEEE Transactions on speech and audio processing*, 5(4), 1997, 359–66.

[42] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, 2nd ed., SIAM, 1996.

[43] T. Tanaka, Y. Hirasawa, and Y. Yamashita, "Variable-length lapped transforms with a combination of multiple synthesis filter banks for image coding", *IEEE transactions on image processing*, 15(1), 2005, 81–8.

[44] H. Tyagi, E. Vural, and P. Frossard, "Tangent space estimation for smooth embeddings of Riemannian manifolds", *Information and Inference: A Journal of the IMA*, 2(1), 2013, 69–114.

[45] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Pearson Education India, 2006.

[46] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Prentice-hall, 1995.

[47] K. Warwick and M. Karny, *Computer Intensive Methods in Control and Signal Processing: The Curse of Dimensionality*, Springer Science & Business Media, 1997.

[48] H. Xiaofei and L. Binbin, "Tangent space learning and generalization", *Frontiers of Electrical and Electronic Engineering in China*, 6, 2011, 27–42.

[49] K. Yuan, X. Zhuang, G. Schaefer, J. Feng, L. Guan, and H. Fang, "Deep-Learning-Based Multispectral Satellite Image Segmentation for Water Body Detection", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 2021, 7422–34, ISSN: 21511535, DOI: 10.1109/JSTARS.2021.3098678.

[50] T. Zhang, J. Yang, D. Zhao, and X. Ge, "Linear local tangent space alignment and application to face recognition", *Neurocomputing*, 70(7-9), 2007, 1547–53.