RoseCDL: Robust and Scalable Convolutional Dictionary Learning for Rare-event Detection

Jad Yehya[1] **Mansour Benbakoura**[1] **Cédric Allain**[1] **Benoît Malezieux**[1]

Matthieu Kowalski[2] **Thomas Moreau**[1]

[1]Université Paris-Saclay, Inria, CEA, Palaiseau, 91120, France
[2]Inria, Université Paris-Saclay, CNRS,
Laboratoire Interdisciplinaire des Sciences du Numériques, Gif-sur-Yvette, France

Abstract

Identifying recurring patterns and rare events in signals is a fundamental challenge in many fields including healthcare and biomedical sciences. Convolutional Dictionary Learning (CDL) provides a framework for modeling local structures, but its use in rare-event detection is unexplored. CDL also faces two challenges: high computational cost and sensitivity to outliers. We introduce RoseCDL, a scalable, robust CDL for unsupervised rare-event detection in long signals. RoseCDL couples stochastic windowing for efficient training with inline outlier detection to improve robustness. This makes CDL a practical tool for event discovery in real-world signals, extending its role beyond compression or denoising.

1 Introduction

Identifying recurring patterns and rare events is central to many domains, from ECG analysis [1] to EEG [2] and microscopy [3]. While deep supervised pipelines [4–6] are effective, their reliance on large annotated datasets is prohibitive when events are rare or ambiguous, calling for unsupervised alternatives. Convolutional Dictionary Learning (CDL; [7]) provides a powerful unsupervised framework by representing signals as convolutions of learned atoms with sparse activations, and has seen use in neuroscience, audio, and imaging [2, 8]. Yet CDL remains limited by (i) sensitivity to outliers, (ii) poor scalability, and (iii) the inability to model rare events, as these vanish in the reconstruction objective.

We address these challenges with *RObust and ScalablE CDL* (RoseCDL). RoseCDL couples stochastic windowing for scalable training with inline outlier detection via reconstruction errors. The method yields a dictionary that captures common structure while discarding anomalies, and an outlier mask that directly serves as a rare-event detection map. We validate RoseCDL on synthetic and real-world datasets, demonstrating scalability, robustness, and unsupervised rare-event discovery in noisy, high-dimensional settings.

NeurIPS 2025 Workshop on Learning from Time-Series for Health (TS4H).

Algorithm 1 CDL with stochastic windowing.

input X, N_{iter}, N_W, N_{FISTA}
  Initialize $D^{(0)}$
  for $0 \leq i \leq N_{\text{iter}} - 1$ **do**
    Sample N_W windows in the dataset: $(X_w)_{w \in [\![1, N_w]\!]}$
    for $1 \leq w \leq N_W$ **do**
      Approximate sparse code : $Z_w^{N_{\text{FISTA}}} \approx Z_w^\star(D^{(i)}; X_w)$
      Compute an outlier mask (cf. Sec. 2)
      Compute the loss F and its gradient $\nabla_D F$ outside the outlier mask
    end for
    Best step size α_i with SLS : $D^{(i+1)} \leftarrow D^{(i)} - \alpha_i \nabla_D \sum_w F_w(D^{(i)}, Z_w^{N_{\text{FISTA}}}; X_w)$
  end for
output $D^{(N_{\text{iter}})}$

2 Finding common and rare patterns in signals: the RoseCDL algorithm

Let $\mathbf{x} \in \mathbb{R}^T$ be a univariate signal of length T. CDL seeks a dictionary $D = (\mathbf{d}_k)_{k=1}^K \in \mathbb{R}^{K \times L}$ of K patterns of length $L \ll T$ and corresponding activations $Z = (\mathbf{z}_k)_{k=1}^K \in \mathbb{R}^{K \times (T-L+1)}$ such that

$$\hat{\mathbf{x}} = D * Z = \sum_{k=1}^{K} \mathbf{d}_k * \mathbf{z}_k,$$

where $*$ denotes convolution. The standard CDL optimization is

$$\min_{D,Z} \; F(D, Z; \mathbf{x}) = \tfrac{1}{2}\|\mathbf{x} - D * Z\|_2^2 + \lambda \|Z\|_1, \quad \text{s.t. } \|\mathbf{d}_k\|_2 \leq 1, \; \forall k. \tag{1}$$

However, in large datasets the aim is not to reconstruct individual training signals but to recover patterns representative of the population. This leads to

$$\min_{D} \; \mathbb{E}_{\mathbf{x}}\Big[\min_{Z} F(D, Z; \mathbf{x})\Big], \quad \text{s.t. } \|\mathbf{d}_k\|_2 \leq 1, \; \forall k, \tag{2}$$

which shifts CDL toward characterizing distributions of local patterns [9–11].

Classical CDL solvers alternate between a sparse coding step (e.g., FISTA) and a dictionary update. Their primary bottleneck is the need to process the entire signal at each iteration, rendering them impractical for large datasets. While online methods [12] offer an improvement, they still rely on full-signal coding and its high memory cost. This motivates our move to a stochastic, localized approximation, which is crucial for achieving both scalability and robustness to outliers.

Stochastic windowing Due to the convolutional structure of CDL, distant points in the signal are only weakly dependent [13]. In practice, the sparse code at time t is rarely influenced by entries at $t + s$ once $s > L$, where L is the dictionary length. Thus, the expected loss (2) can be approximated by restricting the optimization to windows of size W_{win}:

$$\min_{D} \mathbb{E}_\tau \Big[\min_{Z} F(D, Z; \mathbf{x}_\tau) \Big], \quad \text{s.t.} \quad \|\mathbf{d}_k\|_2^2 \leq 1, \; \forall k \in [\![1, K]\!], \tag{3}$$

where windows $\mathbf{x}_\tau$ start at $\tau \in [\![1, T - W_{\text{win}} + 1]\!]$, with $L \leq W_{\text{win}} \ll T$. Although windowing introduces border effects and approximate sparse codes, these errors vanish when activations are sparse and W_{win} is large. Using overlapping windows further mitigates residual boundary artifacts.

To minimize (3), we propose *RoseCDL*, a stochastic gradient descent algorithm designed to learn the distribution of patches. At each outer iteration, we sample N_W overlapping windows $(\mathbf{x}_w)_{1 \leq w \leq N_W}$ uniformly to reduce border bias. For each window, we compute an approximate sparse code $Z_w^\star$ by running N_{FISTA} iterations of FISTA, yielding $Z^{N_{\text{FISTA}}}(D; \mathbf{x}_w)$. Following [14, 15], precise sparse codes are unnecessary for updating D, so these approximations suffice. The dictionary is updated using a single stochastic gradient step. To stabilize training, we employ the Stochastic Line Search (SLS) algorithm [16] to adaptively select the learning rate. The full procedure is outlined in Alg. 1.

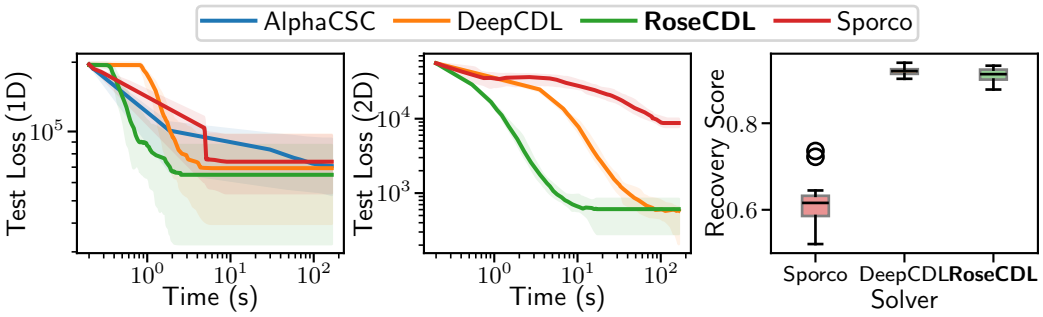

Figure 1: RoseCDL achieves orders-of-magnitude faster convergence without sacrificing solution quality. **(Left, Center)** Test loss versus runtime on large 1D and 2D signals shows RoseCDL's superior convergence speed. **(Right)** The final dictionary recovery score confirms solution quality is competitive with far slower methods.

Inline outlier detection RoseCDL's second component is an inline outlier detection module. We model a signal as

$$\mathbf{x} = \mathbf{d}_a * \mathbf{z}_a + \mathbf{d}_b * \mathbf{z}_b + \mathbf{n}, \tag{4}$$

where $\mathbf{d}_a$ is a common pattern, $\mathbf{d}_b$ a rare one, and $\mathbf{n}$ an artifact. Classical CDL often fails here, as both high-variance artifacts ($\mathbf{n}$) and rare patterns ($\mathbf{d}_b$) prevent stable recovery of $\mathbf{d}_a$, while preprocessing is unreliable at scale [2]. Our key insight is that reconstruction error provides a natural discriminator: patches containing the frequent pattern $\mathbf{d}_a$ will be well-reconstructed, whereas those with anomalies ($\mathbf{n}$) or rare events ($\mathbf{d}_b$) will yield systematically higher errors. To summarize, the distribution of patch reconstruction errors $F(D, Z; \mathbf{x}_\tau)$ is expected to form three modes: (i) low errors from chunks of $\mathbf{x}_a = \mathbf{d}_a * \mathbf{z}_a$, (ii) moderately higher errors from $\mathbf{x}_b = \mathbf{d}_b * \mathbf{z}_b$, and (iii) large errors from artifacts $\mathbf{n}$. Thus, reconstruction error discriminates informative from corrupted patches.

With trimmed loss, RoseCDL is more robust to outliers than classical CDL. With non-overlapping activations of two patterns corrupted by Gaussian noise, classical CDL keeps the frequent pattern $\mathbf{d}_a$ as a fixed point only when the inter-pattern correlation c does not exceed the sparsity threshold λ. RoseCDL also retains the rare pattern $\mathbf{d}_b$ whenever either $c \leq \lambda$ or the trimming step discards more windows than the pattern's activation rate ρ. The full proposition and proof are given in App. A.

Threshold selection. A key remaining design choice is the statistic used for the trimming threshold β. We tested standard thresholding strategies from outlier detection (quantile, z-score, and MAD), and show that the MAD-based criterion provides the most reliable separation in our setting; we therefore adopt it throughout.

Role of the outlier mask for rare-event detection. A key feature of RoseCDL is its inline outlier detection module, which produces an outlier mask during training. This mask serves two purposes: (i) excluding corrupted patches from the loss, thereby improving dictionary robustness, and (ii) enabling unsupervised rare-event detection by interpreting the mask as a detection map.

3 Numerical experiments

RoseCDL[1] achieves scalable, robust, and stable performance on both synthetic and real-world datasets. The method is implemented in `PyTorch` [17] , and our analysis highlights its computational efficiency and consistent recovery of meaningful patterns under varying conditions. For synthetic data, we evaluate performance using the convolutional dictionary recovery metric from [13], detailed in App. C. This metric computes the best assignment between the true and estimated dictionaries, with similarity measured via convolutional cosine similarity to account for shift invariance.

On RoseCDL scalability. In Fig. 1, we compare RoseCDL against three CDL baselines: `AlphaCSC` [2], Sporco [18], and DeepCDL (an unrolled variant of RoseCDL following [15]). Unlike

[1]Code is available at : https://github.com/tomMoral/RoseCDL

DeepCDL, which requires backpropagating through sparse codes, RoseCDL leverages alternating minimization, decoupling updates and reducing overhead.

We evaluate runtime and optimization cost on two large-scale datasets: (i) 1D multivariate signals of length 50,000 with two channels (App. B), and (ii) 2D semi-synthetic images of 2000×2000 pixels. Cost is measured as the objective $F(D, Z^*(D); \mathbf{x})$ on a test set.

optimizer	**Adam**				**SLS**			
window size	$10L$	$20L$	$50L$	$100L$	$10L$	$20L$	$50L$	$100L$
validation loss	3.0%	3.8%	2.7%	2.7%	−0.2%	−0.2%	−0.2%	−0.4%
runtime	15.4%	12.3%	12.5%	12.7%	21.7%	21.2%	16.3%	17.6%

Table 1: Comparison of Adam and SLS with different window sizes.

Results highlight RoseCDL's scalability. GPU-optimized training, `fftconv`-based convolutions and alternate minimization yield substantial speedups. To further examine scaling, we vary window sizes for 1D signals with $T = 100{,}000$ and $\lambda = 0.8$, using both Adam and SLS while fixing (window $\times$ batch) for full GPU utilization. As reported in Tab. 1, RoseCDL achieves validation losses within 4% of `AlphaCSC` across settings, yet runs in only 12–22% of its time (roughly $5\times$ faster). For signal length scalability, we evaluate performance on signals ranging from 10k to 1M time samples. As detailed in Tab. 2, RoseCDL demonstrates sublinear scaling, while `AlphaCSC` fails beyond 100k samples. This superior scalability enables RoseCDL to process signals substantially larger than existing full-signal sparse coding methods. It is important to note that `AlphaCSC` does not support two-dimensional data and was excluded from image-based experiments.

T	10k	30k	100k	300k	1M
Runtime RoseCDL (s)	15.2	16.4	32.6	68.0	202.8
Runtime `AlphaCSC` (s)	67.5	102.8	198.5	N/A	N/A

Table 2: Runtime comparison between RoseCDL and `AlphaCSC` for varying signal sizes T.

RoseCDL on real-world data We evaluated RoseCDL on real-world and benchmark datasets. We used the Physionet Apnea-ECG dataset [19] without preprocessing, ensuring evaluation directly on raw signals and multiple datasets from the TSB-UAD benchmark [20].

On Apnea-ECG, we trained a three-atom dictionary from a 10 min ECG segment with outlier blocks and tested on clean segments. Without outlier detection, the model converges to noise-like atoms. In contrast, RoseCDL reliably filters high-variance blocks, enabling recovery of characteristic ECG patterns.

On the TSB-UAD benchmark, we compared RoseCDL with Matrix Profile (MP), Autoencoder (AE), and One-Class SVM (OC-SVM). As reported in Tab. 3, RoseCDL consistently matches or outperforms these methods in AUC while achieving significantly lower runtimes.

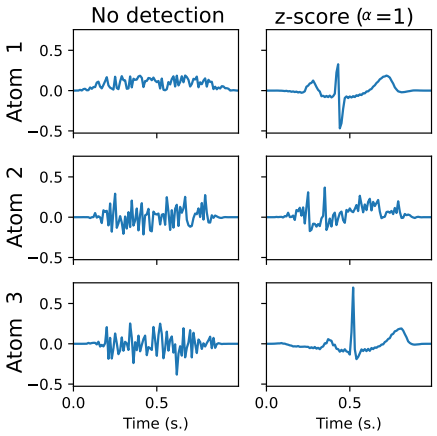

Figure 2: Learned atoms with and without outliers detection method, on 10 bad trials of subject a02 of dataset Physionet Apnea-ECG.

These results demonstrate that RoseCDL robustly extracts meaningful patterns from corrupted, unprocessed signals across diverse datasets.

4 Conclusion

We introduced *RoseCDL*, a scalable and robust framework for Convolutional Dictionary Learning (CDL) enabling unsupervised rare-event detection. By modeling local patch distributions, RoseCDL

	RoseCDL	AE	MP	OC-SVM
Simulated	**0.97**	0.94	0.91	0.61
MITDB	**0.81**	0.59	0.76	0.66
MGAB	0.58	0.61	**0.91**	0.52
ECG	**0.92**	0.91	0.47	**0.92**

Table 3: AUC ROC scores of different anomaly detection methods across datasets.

couples stochastic windowing with inline outlier detection, yielding a simple pipeline that learns common patterns and recovers rare ones from residuals. This scalable, interpretable approach extends CDL to noisy real-world data, opening new directions for large empirical studies and scientific applications.

5 Acknowledgements

This project was supported by the French National Research Agency (ANR) through the BenchArk project (ANR-24-IAS2-0003).

Mansour Benbakoura was supported from a national grant attributed to the ExaDoST project of the NumPEx PEPR program, under the reference ANR-22-EXNU-0004.

This work was performed using HPC resources from GENCI–IDRIS (Grant 2025-AD011015308R1).

References

[1] Luz Luz, Eduardo José da S, William Robson Schwartz, Guillermo Cámara-Chávez, and David Menotti. ECG-based heartbeat classification for arrhythmia detection: A survey. *Computer Methods and Programs in Biomedicine*, 127:144–164, 2016. 1

[2] Tom Dupré la Tour, Thomas Moreau, Mainak Jas, and Alexandre Gramfort. Multivariate convolutional sparse coding for electromagnetic brain signals. *Advances in Neural Information Processing Systems*, 31:3292–3302, 2018. 1, 3

[3] Florence Yellin, Benjamin D. Haeffele, and René Vidal. Blood cell detection and counting in holographic lens-free imaging by convolutional sparse dictionary learning and coding. In *IEEE International Symposium on Biomedical Imaging (ISBI)*, Melbourne, Australia, 2017. 1

[4] Fatma Murat, Ozal Yildirim, Muhammed Talo, Ulas Baran Baloglu, Yakup Demir, and U. Rajendra Acharya. Application of deep learning techniques for heartbeats detection using ECG signals-analysis and review. *Computers in Biology and Medicine*, 120:103726, May 2020. 1

[5] Shilpa Choudhary, Sandeep Kumar, Pammi Sri Siddhaarth, and Guntu Charitasri. Transforming Blood Cell Detection and Classification with Advanced Deep Learning Models: A Comparative Study, October 2024.

[6] D. Cornu, P. Salomé, B. Semelin, A. Marchal, J. Freundlich, S. Aicardi, X. Lu, G. Sainton, F. Mertens, F. Combes, and C. Tasse. YOLO-CIANNA: Galaxy detection with deep learning in radio data - I. A new YOLO-inspired source detection method applied to the SKAO SDC1. *Astronomy & Astrophysics*, 690:A211, October 2024. 1

[7] Roger Grosse, Rajat Raina, Helen Kwong, and Andrew Y. Ng. Shift-Invariant Sparse Coding for Audio Classification. *Cortex*, 8:9, 2007. 1

[8] Vardan Papyan, Yaniv Romano, Jeremias Sulam, and Michael Elad. Convolutional dictionary learning via local processing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1

[9] Meyer Scetbon, Michael Elad, and Peyman Milanfar. Deep k-svd denoising. *IEEE Transactions on Image Processing*, 30:5944–5955, 2021. 2

[10] Hongyi Zheng, Hongwei Yong, and Lei Zhang. Deep convolutional dictionary learning for image denoising. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 630–641, 2021.

[11] Xin Deng, Jingyi Xu, Fangyuan Gao, Xiancheng Sun, and Mai Xu. Deep M^2cdl: Deep multi-scale multi-modal convolutional dictionary learning network. *IEEE transactions on pattern analysis and machine intelligence*, 46(5):2770–2787, 2023. 2

[12] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(1), 2010. 2

[13] Thomas Moreau and Alexandre Gramfort. DiCoDiLe: Distributed Convolutional Dictionary Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2, 3, 9

[14] Benoît Malézieux, Thomas Moreau, and Matthieu Kowalski. Understanding approximate and unrolled dictionary learning for pattern recovery. *International Conference on Learning Representations*, 2022. 2

[15] Bahareh Tolooshams and Demba E. Ba. Stable and interpretable unrolled dictionary learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=e3S0Bl2RO8. 2, 3

[16] Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates. *Advances in neural information processing systems*, 32:3732–3745, 2019. 2

[17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. 2017. 3

[18] Brendt Wohlberg. SPORCO: A Python package for standard and convolutional sparse representations. In *Proceedings of the 15th Python in Science Conference*, pages 1–8, Austin, TX, USA, July 2017. doi: 10.25080/shinma-7f4c6e7-001. 3

[19] Thomas Penzel, George B Moody, Roger G Mark, Ary L Goldberger, and J Hermann Peter. The apnea-ecg database. In *Computers in Cardiology 2000. Vol. 27 (Cat. 00CH37163)*, pages 255–258. IEEE, 2000. 4

[20] John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S Tsay, Themis Palpanas, and Michael J Franklin. Tsb-uad: an end-to-end benchmark suite for univariate time-series anomaly detection. *Proceedings of the VLDB Endowment*, 15(8):1697–1711, 2022. 4

[21] David F Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016. 9

A Analytical study

Proposition A.1 (Stability of the common pattern)**.** *Let X be a population of signals composed of two patterns $\mathbf{d}_a$ and $\mathbf{d}_b$, with non-overlapping activations, corrupted by additive Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma I)$. Define $c = \mathbf{d}_a^\top \mathbf{d}_b$ and let ρ be the proportion of $\mathbf{d}_b$ activations. In the noiseless case:*

i. $\mathbf{d}_a$ is a fixed point of classical CDL with $K = 1$ if $c \leq \lambda$.

ii. $\mathbf{d}_b$ is a fixed point of RoseCDL with $K = 1$ either when $c \leq \lambda$ or when the trimming threshold discards a fraction of windows greater than ρ.

Proof. We consider a dictionary $D \in \mathbb{R}^{1\times L}$ with a single atom $\mathbf{d}$. As we consider signals composed of patterns with no overlap, we can separate each segment and we have a population of signals $X = zd_i + \epsilon$, with $z \in \mathbb{R}$, $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ and $d_i = \mathbf{d}_a$ with probability $1-\rho$ and $\mathbf{d}_b$ with probability ρ, with $\|\mathbf{d}_a\|_2 = \|\mathbf{d}_b\|_2 = 1$. We consider all atoms $\mathbf{d}, \mathbf{d}_a, \mathbf{d}_b$ to be unit norm. Wlog, we can consider $z = 1$, as this amounts to rescaling the value of λ_{max}, and we consider that $c_a = \mathbf{d}^\top \mathbf{d}_a$ and $c_b = \mathbf{d}^\top \mathbf{d}_b$ are positive, as we can consider $-\mathbf{d}$ otherwise. We also consider that the noise level is small enough such that $\sigma^2 < c_j$.

This model is a simplified model in which we have a population of signals where we want to identify the pattern of an event $\mathbf{d}_a$ from the pattern of a rare event $\mathbf{d}_b$.

In this setting, if we further have that the auto-correlation of $\mathbf{d}$ with $\mathbf{d}_a$ and $\mathbf{d}_b$ is maximal when they are aligned, then the sparse coding of a signal X can be computed with the following formula:

$$z^*(X, \mathbf{d}) = \begin{cases} 0 & \text{if} \quad c + \epsilon^\top \mathbf{d} \leq \lambda \\ c + \epsilon^\top \mathbf{d} - \lambda & \text{otherwise} \end{cases} \tag{5}$$

with $c = \mathbf{d}^\top d_i$, which has value c_a with probability $1-\rho$ and c_b otherwise.

We can compute the loss value for this $z^*(X, \mathbf{d})$ for X where z^* is non-zero:

$$F(\mathbf{d}, z^*; X) = \frac{1}{2}\|X - z^*\mathbf{d}\|_2^2 + \lambda\|z^*\|_1 \tag{6}$$

$$= \frac{1}{2}\left(\|X\|_2^2 - 2(c + \epsilon^\top \mathbf{d} - \lambda)(c + \epsilon^\top \mathbf{d}) + \|(c + \epsilon^\top \mathbf{d} - \lambda)\mathbf{d}\|_2^2\right) + \lambda|c + \epsilon^\top \mathbf{d} - \lambda| \tag{7}$$

$$= \frac{1}{2}\left(\|X\|_2^2 - 2(c + \epsilon^\top \mathbf{d} - \lambda)(c + \epsilon^\top \mathbf{d}) + (c + \epsilon^\top \mathbf{d} - \lambda)^2 + 2\lambda(c + \epsilon^\top \mathbf{d} - \lambda)\right) \tag{8}$$

$$= \frac{1}{2}\left(\|X\|_2^2 - 2(c + \epsilon^\top \mathbf{d} - \lambda)(c + \epsilon^\top \mathbf{d} - \lambda) + (c + \epsilon^\top \mathbf{d} - \lambda)^2\right) \tag{9}$$

$$= \frac{1}{2}\left(\|X\|_2^2 - (c + \epsilon^\top \mathbf{d} - \lambda)^2\right) \tag{10}$$

$$= \frac{1}{2}\left(\|d_i\|_2^2 - (c - \lambda)^2 + \|\epsilon\|_2^2 - (\epsilon^\top \mathbf{d})^2 - 2(1 - (c - \lambda))\epsilon^\top \mathbf{d}\right) \tag{11}$$

$$\tag{12}$$

Taking the expectation over the noise yields:

$$\mathbb{E}_\epsilon\left[F(\mathbf{d}, z^*; X)\right] = \frac{1}{2}\left(1 - (c - \lambda)^2 + (L-1)\sigma^2\right) \tag{13}$$

For c between λ and 1, this function is decreasing in c, meaning that for two samples constructed with $\mathbf{d}_a$ and $\mathbf{d}_b$, if the correlation $c_0 = \mathbf{d}^\top \mathbf{d}_a$ is larger than the correlation $c_b = \mathbf{d}^\top \mathbf{d}_b$, then the reconstruction loss for sample 0 is smaller in expectation than the reconstruction loss for a sample 1.

We can also compute the gradient of this function with respect to $\mathbf{d}$. Note that with the KKT condition defining z^*, we have that the $\nabla_z F(\mathbf{d}, z*; X) = 0$, and thus we do not need to compute the Jacobian of z^* when computing the derivative of F with respect to $\mathbf{d}$. The gradient reads:

$$\nabla_{\mathbf{d}} F(\mathbf{d}, z^*; X) = z^*(z^*\mathbf{d} - X) \tag{14}$$

$$= (z^*)^2\mathbf{d} - z^*X \tag{15}$$

$$= (c + \epsilon^\top \mathbf{d} - \lambda)^2\mathbf{d} - (c + \epsilon^\top \mathbf{d} - \lambda)(d_i + \epsilon) \tag{16}$$

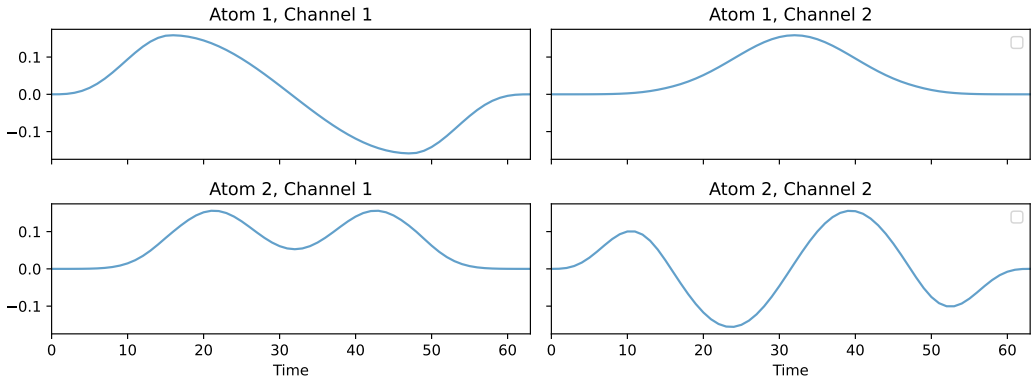

Figure B.3: True dictionary in experiments on synthetic data.

Taking the expectation over the noise yields:

$$\mathbb{E}_\epsilon \left[\nabla_{\mathbf{d}} F(\mathbf{d}, z^*; X)\right] = ((c-\lambda)^2 + \sigma^2)\mathbf{d} - (c-\lambda)d_i + \underbrace{\mathbb{E}_\epsilon\left[\epsilon^\top \mathbf{d} \epsilon\right]}_{\sigma^2 \mathbf{d}} \tag{17}$$

$$= ((c-\lambda)^2 + 2\sigma^2)\mathbf{d} - (c-\lambda)d_i \tag{18}$$

This yields

$$\mathbb{E}[\nabla_{\mathbf{d}} F(\mathbf{d}, z^*; X)] = \left((1-\rho)(c_a-\lambda)^2 + \rho(c_b-\lambda)^2 + 2\sigma^2\right)\mathbf{d} - (1-\rho)(c_a-\lambda)\mathbf{d}_a - \rho(c_b-\lambda)\mathbf{d}_b$$

In the noiseless case, if $\mathbf{d} = \mathbf{d}_a$, and $\lambda \leq c_b = (\mathbf{d}_b)^\top \mathbf{d}_a < 1$, with the classical algorithm, the expected gradient reads

$$\mathbb{E}_X \left[\nabla_{\mathbf{d}} F(\mathbf{d}_a, z^*; X)\right] = -(1-\rho)\lambda(1-\lambda)\mathbf{d}_a + \rho((c_a-\lambda)^2\mathbf{d}_a - (c_b-\lambda)\mathbf{d}_b) \tag{19}$$

$$= \left(\rho(c_a-\lambda)^2 - (1-\rho)\lambda(1-\lambda)\right)\mathbf{d}_a - \rho(c_b-\lambda)\mathbf{d}_b \tag{20}$$

This gradient is not colinear with $\mathbf{d}_a$, showing that $\mathbf{d}_a$ is not a fixed point of the projected gradient descent algorithm in this context. Even in a noiseless and very simple setting, the $\mathbf{d}_a$ is not a solution of the Classical CDL algorithm.

In contrast, when using the least trimmed square procedure with a trimming threshold rejecting a proportion ρ of the samples, we can show that $\mathbf{d}_a$ is a fixed point in the noiseless setting. As seen in (13), the loss for samples X associated with $\mathbf{d}_b$ is smaller than the loss for samples associated with $\mathbf{d}_a$, and therefore rejecting ρ samples from the gradient computation leads to:

$$\mathbb{E}_X \left[\nabla_{\mathbf{d}} F(\mathbf{d}_a, z^*; X)\right] = -(1-\rho)\lambda(1-\lambda)\mathbf{d}_a \tag{21}$$

as the gradient is colinear with $\mathbf{d}_a$, thus $\mathbf{d}_a$ is a fixed point of the projected gradient descent and of the learning procedure. □

B Data simulation

The synthetic multivariate 1D signals $X \in \mathbb{R}^{P \times T}$ used in Sect. 3 are generated from a dictionary $D \in \mathbb{R}^{K \times P \times L}$, a sparse activation vector $Z \in \mathbb{R}^{K \times (T-L+1)}$, and a random Gaussian noise $\varepsilon \sim \mathcal{N}\left(0, \sigma^2\right)$ as $X = D * Z + \varepsilon$. In this definition,

- P is the number of channels,
- T is the length of the signal,
- K is the number of atoms,
- L is the length of the atoms.

In the experiments conducted in Sect. 3, we generated signals of length $T = 50\,000$ with $P = 2$ channels from dictionaries with $K = 2$ atoms of length $L = 64$. The atoms were generated from sine and gaussian waveforms, as illustrated in Fig. B.3. The activations Z were randomly generated sparse Dirac combs with sparsity 0.4 % and the noise level was set to $\sigma = 0.1$.

C Dictionary evaluation

In our methodology, we evaluate the effectiveness of a learned dictionary, denoted as $\widehat{\mathbf{D}} \in \mathbb{R}^{K' \times P \times L'}$, by comparing it against a set of true dictionary patterns, represented as $\mathbf{D} \in \mathbb{R}^{K \times P \times L}$ and computing a "recovery score", using the convolutional cosine similarity following optimal assignment, as defined by [13]. The learned dictionary and the true patterns are structured as three-dimensional arrays, where dimensions correspond to the number of atoms, channels, and atoms' duration. The learned dictionary may differ from the true dictionary in terms of the number of atoms and the length of time atoms, typically featuring more atoms and extended durations.

The evaluation process involves a computational step known as multi-channel correlation. In this step, each atom of the learned dictionary is systematically compared with each pattern in the true dictionary. This comparison is carried out channel by channel, aggregating the results to capture the overall similarity between the dictionary atom and the pattern.

After performing these comparisons for all combinations of atoms and patterns, we create a matrix that represents the correlation strengths between each pair. To objectively assess the quality of the learned dictionary, we use an optimization technique called the Hungarian algorithm. This algorithm finds the best possible "matching" between the learned dictionary atoms and the true patterns, aiming to maximize the overall correlation.

The final score, which quantifies the performance of the learned dictionary, is derived by averaging the values of these optimal matchings. This score is scaled between 0 and 1, where 1 represents the best possible performance. A higher score indicates that the learned dictionary more accurately represents the true dictionary patterns, providing a measure of its quality and effectiveness in capturing the essential features of the data.

Mathematically, the recovery score between the dictionaries $\widehat{\mathbf{D}}$ and $\mathbf{D}$ can be expressed as follow:

$$\text{score} = \frac{1}{K} \sum_{i=1}^{K} C_{i,j^*(i)} \quad , \tag{22}$$

where $j^*(i), i = 1, \dots, K$ denote the results of the linear sum assignment problem [21][2] on correlation matrix $C \coloneqq \text{Corr}\left(\mathbf{D}, \widehat{\mathbf{D}}\right) \in \mathbb{R}^{K \times K'}$, with $\forall i \in [\![1\,, K]\!]$, $\forall j \in [\![1\,, K']\!]$,

$$C_{i,j} = \max_{l=1,\dots,L+L'-1} \text{Corr}_{\text{2D}}\left(D_i, \widehat{D}_j\right)[l] \in \mathbb{R} \quad , \tag{23}$$

where $D_i \in \mathbb{R}^{P \times L}$ and $\widehat{D}_j \in \mathbb{R}^{P \times L'}$. The multivariate "2D" correlation between the two matrices D and $\widehat{D}$ is defined as follow:

$$\text{Corr}_{\text{2D}}\left(D, \widehat{D}\right) = \sum_{p=1}^{P} \text{Corr}_{\text{1D}}\left(d_p, \hat{d}_p\right) \in \mathbb{R}^{L+L'-1} \quad , \tag{24}$$

where $d_p \in \mathbb{R}^L$ and $\hat{d}_p \in \mathbb{R}^{L'}$. The 1D "*full*" correlation between the two vectors d and $\hat{d}$ is defined as follow, $\forall t \in [\![1\,, L + L' - 1]\!]$:

$$\text{Corr}_{\text{1D}}\left(d, \hat{d}\right)[t] = \left(d * \hat{d}\right)[t - T + 1] = \sum_{l=1}^{L} d[l]\hat{d}\,[l - t + T] \in \mathbb{R} \quad , \tag{25}$$

where $T \coloneqq \max(L, L')$.

D Experiments setup

[2] We use `SciPy`'s implementation.

Experiment	Data size	Number of runs	Hardware
Runtime 1D	20 signals, 50,000 data points, 2 channels	50	GPU NVIDIA A40, 30 CPUs
Runtime 2D	2000×2000 grayscale images	20	GPU NVIDIA A40
Regularization impact 1D	10 signals, 30,000 data points, 1 channel	10	GPU NVIDIA A40
Regularization impact 2D	2000×2000 grayscale images	10	GPU NVIDIA A40
Inline vs After	2D image data	10	GPU NVIDIA A40
Physionet	10 trials of subject a02	10	GPU NVIDIA A40 and CPUs