Interpretable Neural Networks for Video Separation: Deep Unfolding RPCA With Foreground Masking

Boris Joukovsky, *Graduate Student Member, IEEE*, Yonina C. Eldar, *Fellow, IEEE*, and Nikos Deligiannis, *Member, IEEE*

***Abstract*— We present two deep unfolding neural networks for the simultaneous tasks of background subtraction and foreground detection in video. Unlike conventional neural networks based on deep feature extraction, we incorporate domain-knowledge models by considering a masked variation of the robust principal component analysis problem (RPCA). With this approach, we separate video clips into low-rank and sparse components, respectively corresponding to the backgrounds and foreground masks indicating the presence of moving objects. Our models, coined ROMAN-S and ROMAN-R, map the iterations of two alternating direction of multipliers methods (ADMM) to trainable convolutional layers, and the proximal operators are mapped to non-linear activation functions with trainable thresholds. This approach leads to lightweight networks with enhanced interpretability that can be trained on limited data. In ROMAN-S, the correlation in time of successive binary masks is controlled with side-information based on ℓ_1-ℓ_1 minimization. ROMAN-R enhances the foreground detection by learning a dictionary of atoms to represent the moving foreground in a high-dimensional feature space and by using reweighted-ℓ_1-ℓ_1 minimization. Experiments are conducted on both synthetic and real video datasets, for which we also include an analysis of the generalization to unseen clips. Comparisons are made with existing deep unfolding RPCA neural networks, which do not use a mask formulation for the foreground, and with a 3D U-Net baseline. Results show that our proposed models outperform other deep unfolding networks, as well as the untrained optimization algorithms. ROMAN-R, in particular, is competitive with the U-Net baseline for foreground detection, with the additional advantage of providing video backgrounds and requiring substantially fewer training parameters and smaller training sets.**

***Index Terms*— Deep learning, deep unfolding, masked RPCA, video separation, foreground detection.**

I. Introduction

ROBUST Principal Component Analysis (RPCA) [1] is a well-known extension of Principal Component Analysis (PCA) [2]. It operates by decomposing a data matrix $\mathbf{D}$ into a compressible low-rank component $\mathbf{L}$ that contains the redundant information, and a sparse component $\mathbf{S}$ that contains the innovative information, such that $\mathbf{D} = \mathbf{L} + \mathbf{S}$. The singular value decomposition (SVD) is often used to find low-rank subspaces; thereby, RPCA addresses the sensitivity of the SVD to the presence of data outliers. Low-rank-plus-sparse (L+S) models are particularly useful for the task of background subtraction in video analysis [3], [4], [5], [6]: by constructing a matrix $\mathbf{D}$ whose columns are composed of the vectorized video frames, a decomposition is sought where the low-rank part represents the quasi-static background across time and the sparse outliers model the moving foreground in each frame.

RPCA is usually formulated as an optimization problem with convex [7] or non-convex objectives [8], [9], [10]. Common iterative solvers are based on proximal gradient descent algorithms [11] and may include augmented Lagrangian forms [12] and minimization with alternating directions [13]. Optimization models can be enhanced to account for specific features of video data: temporal continuity is enforced using additional constraints like total variation [14] or n-ℓ_1 minimization [6], and online algorithms can be used to process incoming frames sequentially [5], [6].

Nevertheless, these algorithms may require many iterations to reach convergence, increasing the computational cost related to the repeated use of SVD with high-dimensional data. Also, these subspace separation methods cannot easily perform higher-level semantic tasks such as foreground detection, since most RPCA variants estimate the foreground component based on the pixel difference with the low-rank model, making it difficult to detect objects with intermittent motion, or true foreground objects from dynamic backgrounds. The recent Masked-RPCA [15], [16] technique addresses this last drawback by replacing the sparse foreground with a sparse mask, which is multiplied point-wise with the low-rank component instead of simple addition. This non-convex variant of RPCA can be solved using alternating minimization, and the pixel foreground membership probabilities provide the location of foreground objects with higher fidelity than the simple thresholding of the sparse component. However, this model still requires many iterations and highly depends on the initialization of the optimization hyperparameters.

Deep neural networks (DNNs) are machine learning models that solely rely on the training dataset to solve a task, with the ability to model almost any physical process by training a highly parameterized and adaptive architecture; however, this same characteristic is responsible for their lack of interpretability and their design mostly follows empirical approaches. Most

Manuscript received 12 April 2022; revised 22 April 2023 and 3 October 2023; accepted 19 November 2023. Date of publication 1 December 2023; date of current version 8 December 2023. This work was supported in part by the Research Foundation Flanders (FWO), Belgium, Research Project under Grant G093817N; and in part by the Ph.D. Fellowship Strategic Basic Research under Grant 1SB5721N. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nelly Pustelnik. *(Corresponding author: Boris Joukovsky.)*

Boris Joukovsky and Nikos Deligiannis are with the Department of Electronics and Informatics, Vrije Universiteit Brussel, 1050 Ixelles, Belgium, and also with imec, 3001 Leuven, Belgium (e-mail: bjoukovs@ETROVUB.be).

Yonina C. Eldar is with the Weizmann Institute of Science, Rehovot 7610001, Israel.

Digital Object Identifier 10.1109/TIP.2023.3336176

deep learning methods treat the problem of video separation as a foreground object detection or segmentation problem, that is, by labeling the video foregrounds pixel-wise. Successful models include fully convolutional neural networks (CNNs) [17], multi-scale segmentation networks [18], cascaded CNNs [19], [20], 3D-CNNs [21], generative adversarial networks (GANs) [22], and transformer-based networks [23]. We refer to [24] and [25] for comprehensive surveys. The foreground detection paradigm is well suited for these supervised learning models since most real video datasets use foreground masks as ground truth data, such as in the CDNet2014 [26] or BMC2012 [27] datasets. In these cases, performance is measured in terms of foreground detection accuracy. In the case of the SBI dataset [28], reference backgrounds are provided, making it useful for background initialization models.

As an attempt to alleviate the interpretability issues of DNNs, a specific class of model-based neural networks has emerged referred to as *deep unfolding* neural networks [29], [30], [31], [32]. These models map the iterations of existing optimization algorithms to layers with learnable parameters, resulting in lightweight networks with enhanced interpretability. They also reach better solutions in fewer iterations (layers) than the original algorithms, thereby reducing the inference time at the expense of additional training time. Additionally, they achieve competitive or superior performance to traditional deep learning models while involving significantly less parameters and training data. Examples include the learned iterative shrinkage-thresholding algorithm (LISTA) that unfolds the corresponding sparse coding algorithm [29], and a version of LISTA with convolution kernels for the convolutional sparse coding task [33]. The alternating directions of multipliers method (ADMM) has also been unfolded with the ADMM-Net network [34]. Deep unfolding models have been proposed for multi-modal data, such as the LMCSC-Net model which is based on sparse coding with side-information [35]. Models for sequential data include SISTA-RNN [36] that solves the problem of sparse signal reconstruction with correlation in time, and reweighted-RNN that solves a sequential video frame reconstruction [37].

Deep unfolding approaches have been proposed for RPCA models: CORONA [38] is a convolutional compressive RPCA model that learns an alternating projection algorithm for the task of clutter suppression in medical ultrasound imaging. Our prior work refRPCA-net [39] applies a similar technique to the task of video separation, by incorporating a side-information scheme to enforce the connectivity of successive foregrounds. Most deep unfolding RPCA models still require fully-supervised training with ground truth background and foreground frames, the latter being composed of pixel-intensity differences with the background. However, in real scenarios, such accurate data is often unavailable since the true background is typically not known due to noise corruption and scene-specific factors such as shadows, occlusions, matching foreground-background colors and dynamic backgrounds. Furthermore, existing deep unfolding models aim at solving the background-foreground separation problem. They are thus in essence sub-optimal when it comes to predicting foreground masks based on the sparse subspace since the underlying L+S model does not explicitly account for the presence of foreground binary masks in the training set.

In this paper, we introduce two RObust MAsking Networks (ROMAN-S and ROMAN-R), which constitute deep unfolding RPCA neural networks for the simultaneous task of video background separation and foreground detection. Unlike previous deep unfolding RPCA models [38], [39], both our networks directly estimate foreground masks. This leads to superior detection performance over previous models when trained on real video data with binary foreground annotations only. First, we design ROMAN-S by unfolding the Masked-RPCA algorithm [15] and incorporating side-information to promote the correlation of foreground masks in time. This correlation scheme is based on ℓ_1-ℓ_1 minimization [40], [41] and inspired by our prior refRPCA-Net [39]. Based on the insights of ROMAN-S, we build a second network coined ROMAN-R, which takes the problems of the foreground mask and the side-information in an auxiliary transform via convolutional sparse coding; this deviates from the original Masked-RPCA formulation. By doing so, a learnable weight is assigned to each feature map via reweighted-ℓ_1-ℓ_1 minimization [42], which has been proven effective in reconstructing moving objects in video RNN models [37], [43]. Thereby, ROMAN-R provides a significant boost in performance over ROMAN-S, with an increased representation learning ability. Our models are fully convolutional, which greatly enhances their speed and memory footprint, thereby leveraging the spatial invariance nature of video frames and allowing to work on video clips of any size. The low sizes of the models allow fast training on few samples with limited risk of overfitting.

We train and evaluate our models on various categories of the CDNet2014 dataset [26] and compare with previous deep unfolding models, as well as a 3D-CNN consisting of an U-Net [44] encoder-decoder model with inflated kernels. We show that our models outperform existing deep unfolding RPCA networks, and ROMAN-R is competitive with the 3D U-Net, while requiring substantially less training parameters. Our implementation is publicly available.[1]

The remainder of the paper is organized as follows: Section II presents background on convex RPCA applied to video separation, the mask variant of RPCA for foreground detection, as well as existing deep unfolding methods including CORONA [38] and our prior work refRPCA-Net [39]. Section III motivates and derives the two proposed deep unfolding models. An experimental study is presented in Section IV and includes a thorough evaluation of our models on the CDNet2014 dataset [26] as well as various ablation studies and an analysis of the generalization to unseen video. We conclude in Section V.

II. Background on RPCA

A. RPCA as Principal Component Pursuit (PCP)

The original RPCA problem as described in [1], [3], and [45] decomposes a data matrix $\mathbf{D}$ into a low-rank component $\mathbf{L}$ and a sparse component $\mathbf{S}$ as formulated in the following

[1] https://gitlab.com/etrovub/mlsp/roman-robust-pca-masking-network

relaxed convex optimization problem:

$$\min_{\mathbf{L},\mathbf{S}} \frac{1}{2}\|\mathbf{D}-\mathbf{L}-\mathbf{S}\|_F^2+\lambda_1\|\mathbf{L}\|_*+\lambda_2\|\mathbf{S}\|_1, \tag{1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\|\cdot\|_*$ is the nuclear norm (the sum of singular values), $\|\cdot\|_1$ is the ℓ_1-norm of its argument organized in a vector, and λ_1 and λ_2 are regularizing parameters. Problem (1), also known as Principal Component Pursuit (PCP) [1], can be solved iteratively using alternating proximal gradient updates at iteration $k+1$ for $\mathbf{L}^{k+1}$ and $\mathbf{S}^{k+1}$, respectively. Specifically, $\mathbf{L}^{k+1}$ can be computed via the singular value thresholding operator [46], and $\mathbf{S}^{k+1}$ via the soft thresholding operator [47], since the latter subproblem effectively corresponds to a step of the iterative shrinkage-thresholding algorithm (ISTA) [47].

B. Video Separation Using RPCA With Side-Information

A grayscale video can formally be represented as a matrix $\mathbf{D}\in\mathbb{R}^{hw\times T}$ composed of T successive vectorized video frames of size $h\times w$. Each frame contains a redundant background, which RPCA aims to isolate into a low-rank component $\mathbf{L}$ from the remaining foreground contained in the sparse component $\mathbf{S}$. Let $\mathbf{s}_t$ $(t=1,\dots,T)$ represent the successive foregrounds, or equivalently, the columns of $\mathbf{S}$. The study in [6] shows that good estimates for $\mathbf{s}_t$ can be found by leveraging $\mathbf{s}_{t-j}$ $(j>0)$ as prior or side-information, since foreground objects are effectively correlated in time. To account for this assumption—which does not exist in the original RPCA problem—an additional penalization term is included in the loss function, resulting in an n-ℓ_1 minimization problem. For instance, the refRPCA model [39] considers the previous signal $\mathbf{s}_{t-1}$ as side-information at time step t, which is incorporated into the model as follows:

$$\begin{aligned}\min_{\mathbf{L},\mathbf{S}}\ &\frac{1}{2}\|\mathbf{D}-\mathbf{H}_1\mathbf{L}-\mathbf{H}_2\mathbf{S}\|_F^2+\lambda_1\|\mathbf{L}\|_*\\ &+\lambda_2\|\mathbf{Q}\circ\mathbf{S}\|_1+\lambda_3\|\mathbf{Q}\circ(\mathbf{S}-\mathbf{S}_\mathbf{P})\|_1,\end{aligned} \tag{2}$$

where $\mathbf{S}_\mathbf{P}$ is defined as $\mathbf{S}_\mathbf{P}=[\mathbf{s}_1,\mathbf{P}\mathbf{s}_1,\cdots,\mathbf{P}\mathbf{s}_{T-1}]$, with $\mathbf{P}$ a correlation-promoting transform. Here, $\mathbf{H}_1$, $\mathbf{H}_2$ are generic measurement operators that arise from a compressive formulation of RPCA, which enhances the recovery of sparse and low-rank components from partial or noisy observations of the true signal [48]. The inclusion of per-element weights $\mathbf{Q}=[\mathbf{q},\cdots,\mathbf{q}]$, with $\mathbf{q}\in\mathbb{R}^{hw}$, allows to use reweighted minimization, which is known to improve the accuracy of sparse estimation [7]. The additional ℓ_1 term results in a modification of the soft-thresholding operator of the ISTA algorithm by adding a second flat activation region around the reference value $\mathbf{S}_\mathbf{P}$.

C. Deep Unfolding Methods

Deep unfolding methods outperform convex optimization methods and typically require less layers than otherwise required for optimization-based solvers [29], [49]. They are also competitive with DNNs while requiring orders of magnitude less trainable parameters and can be trained on reasonably sized datasets, whereas DNNs suffer from the risk of overfitting and poor generalization in the case of small training data. In this line of research, [38] proposed a deep unfolding convolutional RPCA (CORONA) network to solve the following compressive RPCA model:

$$\min_{\mathbf{L},\mathbf{S}} \frac{1}{2}\|\mathbf{D}-\mathbf{H}_1\mathbf{L}-\mathbf{H}_2\mathbf{S}\|_F^2+\lambda_1\|\mathbf{L}\|_*+\lambda_2\|\mathbf{S}\|_{1,2}. \tag{3}$$

Problem (3) was solved in [38] via iteratively updating $\mathbf{L}^{k+1}$ and $\mathbf{S}^{k+1}$ at iteration $k+1$ with

$$\mathbf{L}^{k+1}=\Gamma_{\frac{\lambda_1}{c}}\left(\left(\mathbf{I}-\frac{1}{c}\mathbf{H}_1^\top\mathbf{H}_1\right)\mathbf{L}^k-\mathbf{H}_1^\top\mathbf{H}_2\mathbf{S}^k+\mathbf{H}_1^\top\mathbf{D}\right), \tag{4a}$$

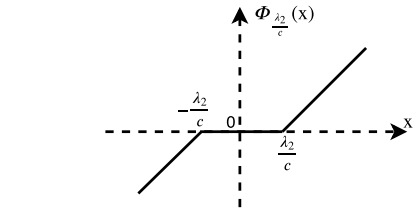
$$\mathbf{S}^{k+1}=\Phi_{\frac{\lambda_2}{c}}\left(\left(\mathbf{I}-\frac{1}{c}\mathbf{H}_2^\top\mathbf{H}_2\right)\mathbf{S}^k-\mathbf{H}_2^\top\mathbf{H}_1\mathbf{L}^k+\mathbf{H}_2^\top\mathbf{D}\right), \tag{4b}$$

where $\|\cdot\|_{1,2}$ is the mixed $\ell_{1,2}$ norm, $\Gamma_{\frac{\lambda_1}{c}}(\cdot)$ and $\Phi_{\frac{\lambda_2}{c}}(\cdot)$ are the singular value thresholding and mixed $\ell_{1,2}$ soft thresholding [47] operators, respectively, and c is a Lipschitz constant. The sensing operators $\mathbf{H}_1$ and $\mathbf{H}_2$ in Eq. (4a) and (4b) are mapped to learnable weights in the corresponding deep unfolding architecture: specifically, CORONA uses convolutional kernels $\mathbf{W}_1^k,\cdots,\mathbf{W}_6^k$ at each layer k, as shown in (5a) and (5b). These parameters are all learned through backpropagation.

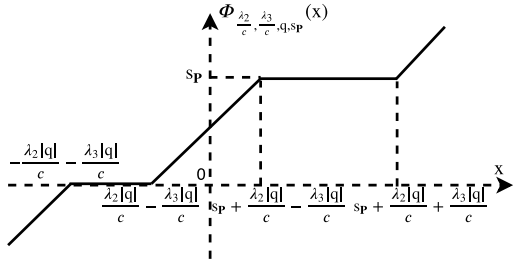
$$\mathbf{L}^{k+1}=\Gamma_{\lambda_1^k}\left\{\mathbf{W}_1^k*\mathbf{D}+\mathbf{W}_3^k*\mathbf{S}^k+\mathbf{W}_5^k*\mathbf{L}^k\right\}, \tag{5a}$$

$$\mathbf{S}^{k+1}=\Phi_{\lambda_2^k}\left\{\mathbf{W}_2^k*\mathbf{D}+\mathbf{W}_4^k*\mathbf{S}^k+\mathbf{W}_6^k*\mathbf{L}^k\right\}. \tag{5b}$$

This approach was extended in [39] with the refRPCA-Net model, which solves the problem of RPCA with side-information based on (2) for the task of video separation. According to the principles of reweighted-ℓ_1 minimization [7],



(a) CORONA: $\Phi_{\frac{\lambda_2}{c}}(x)$



(b) refRPCA-Net: $\Phi_{\frac{\lambda_2}{c}q,\frac{\lambda_3}{c}q,s_\mathbf{P}}(x)$ for $\mathbf{S}_\mathbf{P}\geq 0$.

Fig. 1. The soft-thresholding operator used in CORONA [38] (a) vs refRPCA-Net [39] (b). λ_2,λ_3 and c are learned globally. Note that (b) is drawn for given q and $\mathbf{S}_\mathbf{P}$. The weight q allows for a different proximal operator for each entry of x due to the varying length of the multiple-threshold intervals. The reference $\mathbf{S}_\mathbf{P}$ defines the position of the non-zero plateau each time the operator is evaluated.

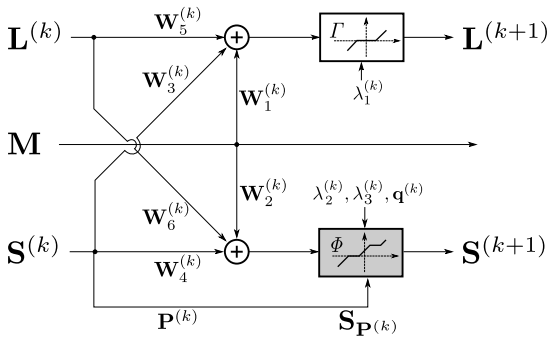


Fig. 2. One layer of the refRPCA-Net architecture [39], equivalent to the CORONA model [38] with the bottom activation replaced by soft-thresholding.

[41], reweighted-ℓ_1-ℓ_1 minimization with side-information [6] and its deep unfolding counterpart [37], the refRPCA-Net model shown in Fig. 2 uses the same update equations as CORONA, except for the soft-thresholding activation for the update of the sparse component. In comparison to the soft-thresholding operator in Fig. 1(a), the activation function of refRPCA-Net in Fig. 1(b) features an additional flat region promoting the correlation with side-information $\mathbf{S_P}$.

D. Masked RPCA

The Masked-RPCA (MRPCA) method [15], [16] changes the problem of foreground separation to a foreground detection problem, where a sparse foreground mask $\mathbf{M} \in \{0,1\}^{hw\times T}$ is predicted instead of the typical foreground $\mathbf{S}$ of RPCA. In fact, using a simple threshold on $\mathbf{S}$ to identify foreground pixels may be insufficient when the pixel difference with the background is low, or in video with high disturbances. MRPCA directly estimates the binary mask $\mathbf{M}$ through optimization without explicit computation of the true foreground. This is achieved by restricting the data fidelity constraint to pixels located outside of the foreground mask only:

$$\begin{aligned} &\min_{\mathbf{L},\mathbf{M}} \ \|\mathbf{L}\|_* + \lambda\|\mathbf{M}\|_1 \\ &s.t. \ (\mathbf{1}-\mathbf{M})\circ(\mathbf{D}-\mathbf{L}) = \mathbf{0} \\ &\quad \mathbf{M} \in [0,1]^{wh\times T}. \end{aligned} \tag{6}$$

In (6), the binary constraint on the mask is relaxed to the continuous interval $[0,1]$, the masking operation is realized with the Hadamard or element-wise product denoted by $\circ$, and λ is a regularization coefficient. An algorithm to solve (6) was proposed in [15] and is based on the alternating direction method of multipliers (ADMM). The method formulates the augmented Lagrangian of (6) and iterates between the following updates, where $\mathbf{U}$ is the dual variable, ρ is a penalization coefficient and $\mathbb{1}_{[0,1]}$ is the indicator function of the interval $[0,1]$:

$$\begin{aligned} \mathbf{L}^{k+1} = &\arg\min_{\mathbf{L}} \|\mathbf{L}\|_* \\ &+ \underbrace{\frac{\rho}{2}\|(\mathbf{1}-\mathbf{M}^k)\circ(\mathbf{D}-\mathbf{L}) + \frac{\mathbf{U}^k}{\rho}\|_F^2}_{\equiv f(\mathbf{L})}, \end{aligned} \tag{7}$$

$$\begin{aligned} \mathbf{M}^{k+1} = &\arg\min_{\mathbf{M}} \lambda\|\mathbf{M}\|_1 + \mathbb{1}_{[0,1]}(\mathbf{M}) \\ &+ \underbrace{\frac{\rho}{2}\|(\mathbf{1}-\mathbf{M})\circ(\mathbf{D}-\mathbf{L}^{k+1}) + \frac{\mathbf{U}^k}{\rho}\|_F^2}_{\equiv g(\mathbf{M})}, \end{aligned} \tag{8}$$

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \rho(\mathbf{1}-\mathbf{M}^{k+1})\circ(\mathbf{D}-\mathbf{L}^{k+1}). \tag{9}$$

The updates (7) and (8) are respectively obtained via proximal gradient descent steps:

$$\mathbf{L}^{k+1} = \Gamma_{\frac{\tau_L}{\rho}}\left(\mathbf{L}^k - \tau_L\nabla f(\mathbf{L})\right), \tag{10}$$

$$\mathbf{M}^{k+1} = \Pi\left(\Phi_{\frac{\lambda\tau_M}{\rho}}\left(\mathbf{M}^k - \tau_M\nabla g(\mathbf{M})\right)\right), \tag{11}$$

where τ_L and τ_M are proximal parameters, and Γ, Φ, Π correspond to the singular value thresholding, shrinkage-thresholding and $[0,1]$-clamping operators, respectively. The gradients of the smooth parts ∇f and ∇g are:

$$\nabla f(\mathbf{L}) = (\mathbf{1}-\mathbf{M}^k)\circ\left((\mathbf{L}-\mathbf{D})\circ(\mathbf{1}-\mathbf{M}^k) + \frac{\mathbf{U}^k}{\rho}\right), \tag{12}$$

$$\nabla g(\mathbf{M}) = (\mathbf{L}^{k+1}-\mathbf{D})\circ\left((\mathbf{L}^{k+1}-\mathbf{D})\circ(\mathbf{1}-\mathbf{M}) + \frac{\mathbf{U}^k}{\rho}\right). \tag{13}$$

In the following section, the foreground masking approach of [15] will be used as a basis for the design of our proposed deep unfolding video separation networks.

III. Proposed Deep Unfolding Networks for Video Separation

In this section, we propose two deep unfolding neural networks that solve the joint problem of background subtraction and foreground object detection in video based on low-rank plus sparse priors. Unlike the previous deep unfolding RPCA methods described in Section II-C, our models are trained to retrieve foreground masks instead of inferring the true foreground, which facilitates the training process on real video datasets. The networks are obtained by unfolding two ADMM algorithms, and compared to Masked-RPCA [15], our unfolded networks have drastically less runtime complexities than the plain optimization approach during inference, as well as superior detection performance. In the first model (ROMAN-S), the side-information scheme of our prior refRPCA-Net model is incorporated to promote the consistency of foreground masks in time. The second model (ROMAN-R) solves the mask subproblem in a transform domain via a learnable dictionary of atoms, which deviates from the original masked RPCA formulation that directly optimizes the sparsity of the foreground mask in the pixel domain. Furthermore, the subsequent deep unfolding model has additional learnable kernels compared to the first one, thereby increasing its representation learning ability and its overall foreground detection performance.

A. ROMAN-S: Robust Foreground Masking Network With Side-Information

1) Minimization Model: Given a collection of T successive video frames of size $h \times w$, vectorized and stacked in the matrix $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_T]$, we seek to find a low-rank approximation $\mathbf{L} \in \mathbb{R}^{hw \times T}$ of $\mathbf{D}$ that models its static background and a sparse binary mask $\mathbf{M} \in \mathbb{R}^{hw \times T}$ that indicates the presence of moving objects. We formulate a minimization problem in (14) that estimates $\mathbf{L}$ and $\mathbf{M}$ respectively with low-rank and sparse penalties, while the reconstruction term ensures that the known background pixels located outside of the foreground mask match the original video content in $\mathbf{D}$. Similar to [39], we construct a sequence of reference masks $\widetilde{\mathbf{M}} = [\mathbf{m}_1, \mathbf{P}\mathbf{m}_1, \cdots, \mathbf{P}\mathbf{m}_{T-1}]$ in order to promote the time correlation of successive binary masks by enforcing $\|\mathbf{M} - \widetilde{\mathbf{M}}\|_1$ to be small, with a yet-to-be-learned linear transform $\mathbf{P}$:

$$\begin{aligned} &\min_{\mathbf{L},\mathbf{M}} \ \|\mathbf{L}\|_* + \lambda_1 \|\mathbf{M}\|_1 + \lambda_2 \|\mathbf{M} - \widetilde{\mathbf{M}}\|_1 \\ &s.t. \ (\mathbf{1} - \mathbf{H}_1\mathbf{M}) \circ (\mathbf{D} - \mathbf{H}_2\mathbf{L}) = 0 \\ &\qquad \mathbf{M} \in [0,1]^{wh \times T}. \end{aligned} \tag{14}$$

Here, the low-rank constraint of $\mathbf{L}$ is relaxed to nuclear norm minimization. The sparsity of $\mathbf{M}$ and the correlation in time with $\widetilde{\mathbf{M}}$ is formulated as a ℓ_1-ℓ_1-minimization penalty term, λ_1, λ_2 are regularization parameters and $\circ$ denotes the Hadamard product. Problem (14) differs from Masked-RPCA [16] since our model uses a side-information branch and measurement operators $\mathbf{H}_1$ and $\mathbf{H}_2$, which create learnable weights in the deep unfolding steps [38], [39]. We then follow a similar approach to [15] by reformulating the non-convex problem (14) in the augmented Lagrangian form in (15) with a dual variable $\mathbf{U}$:

$$\begin{aligned} \mathcal{L}(\mathbf{M}, \mathbf{L}, \mathbf{U}) = \ &\|\mathbf{L}\|_* + \lambda_1 \|\mathbf{M}\|_1 + \lambda_2 \|\mathbf{M} - \widetilde{\mathbf{M}}\|_1 + \mathbb{1}_{[0,1]}(\mathbf{M}) \\ &+ \langle \mathbf{U}, (\mathbf{1} - \mathbf{H}_1\mathbf{M}) \circ (\mathbf{D} - \mathbf{H}_2\mathbf{L}) \rangle \\ &+ \frac{\rho}{2} \|(\mathbf{1} - \mathbf{H}_1\mathbf{M}) \circ (\mathbf{D} - \mathbf{H}_2\mathbf{L})\|_F^2. \end{aligned} \tag{15}$$

We solve (15) using the ADMM procedure to alternately update $\mathbf{L}$, $\mathbf{M}$ and $\mathbf{U}$ according to the two following convex sub-problems:

$$\begin{aligned} \mathbf{L}^{k+1} = &\arg\min_{\mathbf{L}} \|\mathbf{L}\|_* \\ &+ \frac{\rho}{2} \|(\mathbf{1} - \mathbf{H}_1\mathbf{M}^k) \circ (\mathbf{D} - \mathbf{H}_2\mathbf{L}) + \frac{\mathbf{U}^k}{\rho}\|_F^2 \end{aligned} \tag{16}$$

$$\begin{aligned} \mathbf{M}^{k+1} = &\arg\min_{\mathbf{M}} \lambda_1 \|\mathbf{M}\|_1 + \lambda_2 \|\mathbf{M} - \widetilde{\mathbf{M}}\|_1 + \mathbb{1}_{[0,1]}(\mathbf{M}) \\ &+ \frac{\rho}{2} \|(\mathbf{1} - \mathbf{H}_1\mathbf{M}) \circ (\mathbf{D} - \mathbf{H}_2\mathbf{L}^{k+1}) + \frac{\mathbf{U}^k}{\rho}\|_F^2 \end{aligned} \tag{17}$$

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \rho(\mathbf{1} - \mathbf{H}_1\mathbf{M}^{k+1}) \circ (\mathbf{D} - \mathbf{H}_2\mathbf{L}^{k+1}). \tag{18}$$

Following (10) and (11), the solutions of subproblems (16) and (17) are obtained via proximal gradient updates. The explicit solutions are given below in (19) and (20), accounting for the fact that the ℓ_1-ℓ_1 cost in the $\mathbf{M}$ subproblems is solved by choosing Φ to be the shrinkage-thresholding with side-information of Fig. 1(b) with coefficients q set to 1 (cf. [39]):

Algorithm 1 Forward Pass of ROMAN-S

Input: $\boldsymbol{D}, \boldsymbol{M}^0, \boldsymbol{L}^0, \boldsymbol{U}^0$
Output: $\widehat{\mathrm{M}}$, $\widehat{\mathrm{L}}$
for k = *1 to* K **do**
  // L branch
  $\boldsymbol{W} := \mathbf{1} - \boldsymbol{\mathcal{H}}_1^k * \boldsymbol{M}^{k-1}$
  $\boldsymbol{\Lambda}_0 := \boldsymbol{\mathcal{H}}_3^k * (\boldsymbol{W}^{\circ 2} \circ (\boldsymbol{\mathcal{H}}_2^k * \boldsymbol{L}^{k-1}))$
  $\boldsymbol{\Lambda}_1 := \boldsymbol{\mathcal{H}}_4^k * (\boldsymbol{W}^{\circ 2} \circ \boldsymbol{D})$
  $\boldsymbol{\Lambda}_2 := \frac{1}{\rho^k} \boldsymbol{\mathcal{H}}_5^k * (\boldsymbol{W} \circ \boldsymbol{U}^{k-1})$
  $\boldsymbol{L}^k = \Gamma_{\gamma^k}\big(\boldsymbol{L}^{k-1} - \tau_L^k(\boldsymbol{\Lambda}_0 - \boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2)\big)$
  // M branch
  $\boldsymbol{W} := \boldsymbol{D} - \boldsymbol{\mathcal{H}}_6^k * \boldsymbol{L}^k$
  $\boldsymbol{\Lambda}_0 := \boldsymbol{\mathcal{H}}_8^k * (\boldsymbol{W}^{\circ 2} \circ (\mathbf{1} - \boldsymbol{\mathcal{H}}_7^k * \boldsymbol{M}^{k-1}))$
  $\boldsymbol{\Lambda}_1 := \frac{1}{\rho^k} \boldsymbol{\mathcal{H}}_9^k * (\boldsymbol{W} \circ \boldsymbol{U}^{k-1})$
  $\widetilde{\boldsymbol{M}} = [\boldsymbol{M}_1^{k-1}, \boldsymbol{\mathcal{P}}^k * \boldsymbol{M}_1^{k-1}, \ldots, \boldsymbol{\mathcal{P}}^k * \boldsymbol{M}_{T-1}^{k-1}]$
  $\boldsymbol{M}^k = \sigma_{\alpha^k}\Big(\Phi_{\lambda_1^k, \lambda_2^k}\big(\boldsymbol{M}^k + \tau_M^k(\boldsymbol{\Lambda}_0 - \boldsymbol{\Lambda}_1); \widetilde{\boldsymbol{M}}\big)\Big)$
  // U branch
  $\boldsymbol{U}^k = \boldsymbol{U}^{k-1} + \rho^k(\mathbf{1} - \boldsymbol{\mathcal{H}}_{10}^k * \boldsymbol{M}^k) \circ (\boldsymbol{\mathcal{H}}_{11}^k * \boldsymbol{L}^k - \boldsymbol{D})$
end
return $\widehat{\mathrm{M}} = \boldsymbol{M}^K$, $\widehat{\mathrm{L}} = \boldsymbol{L}^K$.

$$\begin{aligned} \mathbf{L}^{k+1} = \Gamma_{\frac{\tau_L}{\rho}}\Big[\mathbf{L}^k - \tau_L \mathbf{H}_2^\top \big((\mathbf{1} - \mathbf{H}_1\mathbf{M}^k)^{\circ 2} \circ (\mathbf{H}_2\mathbf{L}^k - \mathbf{D}) \\ - (\mathbf{1} - \mathbf{H}_1\mathbf{M}^k) \circ \frac{\mathbf{U}^k}{\rho}\big)\Big], \end{aligned} \tag{19}$$

$$\begin{aligned} \mathbf{M}^{k+1} = \Pi\Bigg[\Phi_{\frac{\tau_M\lambda_1}{\rho}, \frac{\tau_M\lambda_2}{\rho}}\Big[\mathbf{M}^k + \tau_M \mathbf{H}_1^\top \big((\mathbf{D} - \mathbf{H}_2\mathbf{L}^k)^{\circ 2} \\ \circ (\mathbf{1} - \mathbf{H}_1\mathbf{M}^k) - (\mathbf{D} - \mathbf{H}_2\mathbf{L}^k) \circ \frac{\mathbf{U}^k}{\rho}\big)\Big]\Bigg]. \end{aligned} \tag{20}$$

2) Deep Unfolding Model: In order to build the deep unfolding network and apply it on entire images instead of patches, the large measurement matrices $\mathbf{H}_1, \mathbf{H}_2$ are replaced by convolutional kernels $\boldsymbol{\mathcal{H}}_1, \boldsymbol{\mathcal{H}}_2 \in \mathbb{R}^{p_1 \times p_2}$ acting on the individual frames across time. Thereby, we leverage the spatial invariance of images and drastically reduce the number of trainable parameters. Likewise, we cast the correlation matrix $\mathbf{P}$ to a 2D convolution kernel $\boldsymbol{\mathcal{P}}$. The convolutional formulation is strictly equivalent to a linear one with corresponding Toeplitz matrices; hence, the iterative model defined by (18), (19) and (20) is still applicable, and transposed matrix multiplications can be mapped to transposed 2D-convolutions.

We now build the ROMAN-S network with K layers by taking a number of iterations of the ADMM-based algorithm and unrolling them into a learnable network. The model equations and stages are detailed in Algorithm 1. During the forward pass, $\boldsymbol{L}, \boldsymbol{M}, \boldsymbol{U}$ are 3D tensors of size $T \times w \times h$ and $\{\boldsymbol{\mathcal{H}}_1^k, \ldots, \boldsymbol{\mathcal{H}}_{11}^k\}$ are individual kernels corresponding either to forward or transposed convolutions in the convolutional version of the algorithm. For practical purposes, these are implemented in the form of 3D convolutional layers with unit depth in the time axis.

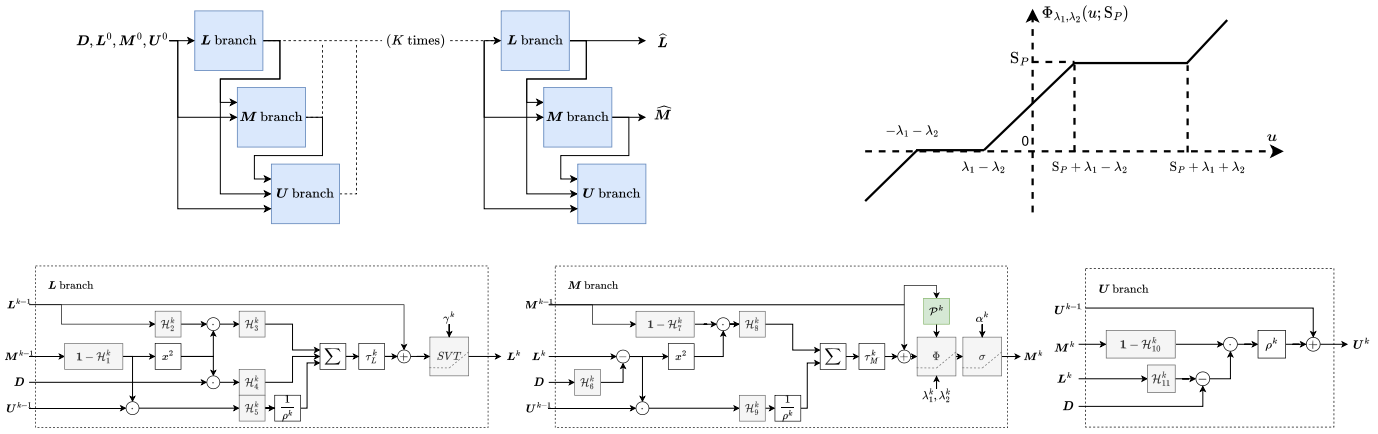


Fig. 3. Detail of the branches of the deep unfolding model **ROMAN-S**, and the soft-thresholding activation function with side-information $\Phi_{\lambda_1^k,\lambda_2^k}$.

Note that all weights are decoupled across layers, as well as within a single iteration in comparison to the original optimization model. The non-linear operations include the singular-value thresholding operator Γ_{γ^k} with learnable threshold γ^k, the low-rank component (that is, $\boldsymbol{L}$ reshaped as a 2D matrix), the shrinkage-thresholding operator $\Phi_{\lambda_1^k,\lambda_2^k}$ with learnable thresholds λ_1^k, λ_2^k and side-information $\widetilde{\boldsymbol{M}}$, as well as a reparameterized sigmoid function $\sigma_{\alpha^k}(x) \equiv \text{sigmoid}(\alpha^k(x-0.5))$ for the mask branch. This is similar to the Gumbel-Softmax activation [50] with scaling factor α^k and is used as a differentiable approximation of the clamping operator, forcing the mask distribution to follow a binary distribution better. In summary, the set of trainable parameters for K layers is:

$$\Theta = \{\boldsymbol{\mathcal{H}}_1^k, \ldots, \boldsymbol{\mathcal{H}}_{11}^k, \boldsymbol{\mathcal{P}}^k, \lambda_1^k, \lambda_2^k, \gamma^k, \alpha^k, \tau_L^k, \tau_M^k, \rho^k\}_{k=1,\ldots,K}. \tag{21}$$

The overall network structure is illustrated in Fig. 3 by following the steps in Algorithm 1. Each layer contains three interacting branches to update $\boldsymbol{L}$, $\boldsymbol{M}$ and the multiplier $\boldsymbol{U}$, respectively. In comparison, our prior refRPCA-Net of Fig. 2 only contains two branches due to its different underlying minimization algorithm. The Hadamard products are implemented as point-wise multiplications, which can be seen as adaptive masking operations during the forward pass.

B. ROMAN-R: Robust Masking Network With Reweighted Minimization and Sparse Coding

1) Minimization Model: Our second approach takes the mask estimation problem into the transform domain by taking inspiration from the learned convolutional sparse coding technique [33]. For each video frame t, we compute a set of n feature maps $\boldsymbol{\mathcal{M}}_i^t$ using a learnable convolutional dictionary of atoms $\boldsymbol{\Psi}_i$, $i = 1, \ldots, n$, such that each 2D mask at every frame is given by $\mathbf{M}_t = \sum_i^n \boldsymbol{\Psi}_i * \boldsymbol{\mathcal{M}}_i^t$. In what follows, we define $\boldsymbol{\mathcal{M}}_i$ as the 3D-tensor composed of the feature maps $[\boldsymbol{\mathcal{M}}_i^1, \ldots, \boldsymbol{\mathcal{M}}_i^T]$, and $\boldsymbol{\Psi}_i * \boldsymbol{\mathcal{M}}_i$ is a convolution distributed across time. In this case, the reference signal $\widetilde{\boldsymbol{\mathcal{M}}}_i$ is constructed as $[\boldsymbol{\mathcal{M}}_i^1, \boldsymbol{\mathcal{P}}_i * \boldsymbol{\mathcal{M}}_i^1, \ldots, \boldsymbol{\mathcal{P}}_i * \boldsymbol{\mathcal{M}}_i^{T-1}]$. It can be observed that a different correlation operator $\boldsymbol{\mathcal{P}}_i$ corresponds to each feature map. Also, we may reweight the contribution of each feature map in the cost function by using a positive coefficient g_i, enabling the use of reweighted minimization, which is known to improve the accuracy of sparse signal reconstruction. We also penalize the difference of successive representations by another ℓ_1 cost. As a result, (22) becomes:

$$\begin{aligned} \min_{\mathbf{L},\mathbf{M}} \; & \|\mathbf{L}\|_* + \lambda_1 \sum_i g_i \|\boldsymbol{\mathcal{M}}_i\|_1 + \lambda_2 \sum_i g_i \|\boldsymbol{\mathcal{M}}_i - \widetilde{\boldsymbol{\mathcal{M}}}_i\|_1 \\ s.t. \; & (\mathbf{1}-\mathbf{M}) \circ (\mathbf{D}-\mathbf{L}) = 0 \\ & \mathbf{M} = \text{reshape}\left(\sum_i \boldsymbol{\Psi}_i * \boldsymbol{\mathcal{M}}_i\right) \\ & \mathbf{M} \in [0,1]^{wh \times T}. \end{aligned} \tag{22}$$

In order to simplify the derivations, we use the notation $\boldsymbol{\Psi}\boldsymbol{\mathcal{M}}$ as a replacement for $\mathbf{M} = \sum_i \boldsymbol{\Psi}_i * \boldsymbol{\mathcal{M}}_i$, where $\boldsymbol{\Psi} \in \mathbb{R}^{hw \times hwn}$ is the equivalent Toeplitz matrix and $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{hwn \times T}$ a vectorized version of the feature maps. We use the Moore-Penrose pseudoinverse $\boldsymbol{\Psi}^\dagger$ to transform $\mathbf{M}$ into the feature space, with the actual transformation being learned via specific convolution kernels during the deep unfolding steps. Next, we reformulate (22) in the augmented Lagrangian form:

$$\begin{aligned} \mathcal{L}(\mathbf{L}, \boldsymbol{\mathcal{M}}, \mathbf{U}) = \; & \|\mathbf{L}\|_* + \lambda_1 \|\mathbf{G} \circ \boldsymbol{\mathcal{M}}\|_1 + \lambda_2 \|\mathbf{G} \circ (\boldsymbol{\mathcal{M}} - \widetilde{\boldsymbol{\mathcal{M}}})\|_1 \\ & + \mathbb{1}_{[0,1]}(\boldsymbol{\Psi}\boldsymbol{\mathcal{M}}) + \langle \mathbf{U}, (\mathbf{1} - \boldsymbol{\Psi}\boldsymbol{\mathcal{M}}) \circ (\mathbf{D} - \mathbf{L}) \rangle \\ & + \frac{\rho}{2} \|(\mathbf{1} - \boldsymbol{\Psi}\boldsymbol{\mathcal{M}}) \circ (\mathbf{D} - \mathbf{L})\|_F^2, \end{aligned} \tag{23}$$

where $\mathbf{U}$ is a dual variable, $\mathbf{G}$ is a matrix formed by the corresponding weights g_i, and $\mathbb{1}_{[0,1]}$ is the indicator function.

A fundamental difference with CORONA, refRPCA-Net and the previous model (14) is the absence of measurement operators $\mathbf{H}_1$ and $\mathbf{H}_2$; from the perspective of deep unfolding, the introduction of $\boldsymbol{\Psi}$ will automatically result in learnable convolution kernels, thus rendering the use of additional operators unnecessary. Similar to the previous derivations in Section III-A, we may write the following update equations for $\mathbf{L}$, $\boldsymbol{\mathcal{M}}$ and $\mathbf{U}$:

$$\begin{aligned} \mathbf{L}^{k+1} = \Gamma_{\frac{\tau_L}{\rho}} \Big[& \mathbf{L}^{k+1} - \tau_L (\mathbf{1} - \boldsymbol{\Psi}\boldsymbol{\mathcal{M}}^k)^{\circ 2} \circ (\mathbf{L}^{k+1} - \mathbf{D}) \\ & - \tau_L (\mathbf{1} - \boldsymbol{\Psi}\boldsymbol{\mathcal{M}}^k) \circ \frac{\mathbf{U}^k}{\rho} \Big], \end{aligned} \tag{24}$$

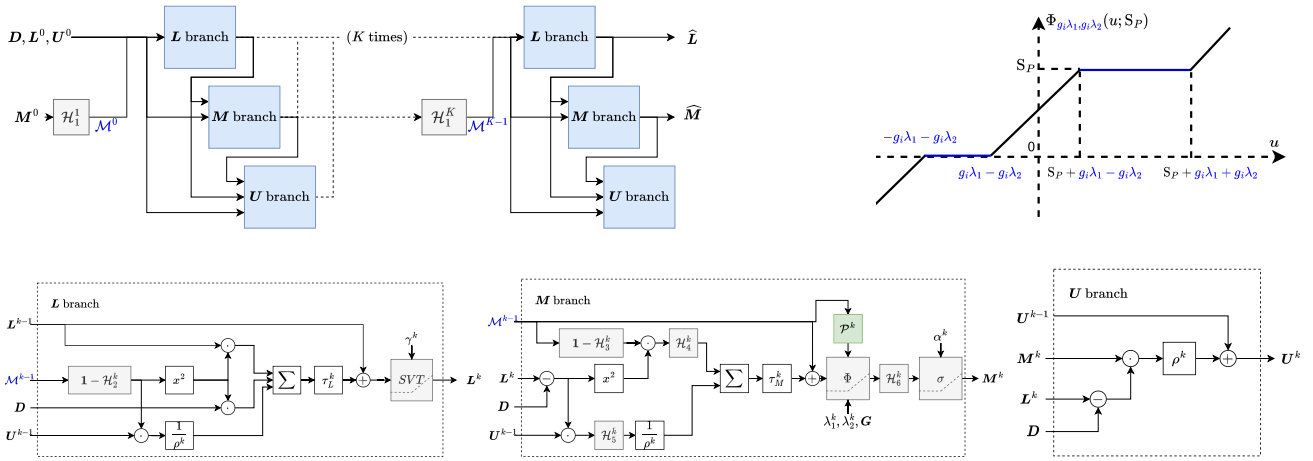


Fig. 4. Detail of the branches of the deep unfolding **ROMAN-R** model. The learnable activation function is the reweighted soft-thresholding operator with side-information. Each channel has a different threshold scaling factor g_i.

$$\boldsymbol{\mathcal{M}}^{k+1} = \boldsymbol{\Psi}^{\dagger}\Pi\Big[\boldsymbol{\Psi}\Phi_{\frac{\tau_M\lambda_1}{\rho},\frac{\tau_M\lambda_2}{\rho},\mathbf{g}}\Big[\boldsymbol{\mathcal{M}}^k+\tau_M\boldsymbol{\Psi}^{\top}\big((\mathbf{D}-\mathbf{L}^{k+1})^{\circ 2} \circ (\mathbf{1}-\boldsymbol{\Psi}\boldsymbol{\mathcal{M}}^k) - (\mathbf{D}-\mathbf{L}^{k+1})\circ\frac{\mathbf{U}^k}{\rho}\big)\Big]\Big], \tag{25}$$

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \rho(\mathbf{1}-\boldsymbol{\Psi}\boldsymbol{\mathcal{M}}^{k+1})\circ(\mathbf{D}-\mathbf{L}^{k+1}). \tag{26}$$

Here, Γ still refers to the SVT operator, while Φ is the soft-thresholding operator with side-information and conditioned on weights $\mathbf{G}$, which can be derived from reweighted-ℓ_1-ℓ_1 minimization, and Π is the clamping operator.

2) Deep Unfolding Model: The approach to building the ROMAN-R network results from unfolding of the iterations (24), (25) and (26). As opposed to ROMAN-S, the trainable convolutional kernels $\boldsymbol{\mathcal{H}}_i^k$ arise from $\boldsymbol{\Psi}$, $\boldsymbol{\Psi}^{\top}$ and $\boldsymbol{\Psi}^{\dagger}$. Also, in the mask branch, most operations are performed in the transform domain, including the processing of side-information, after which the mask is transformed back into the image domain for the remaining non-linearity. For a K-layer network, the set of trainable parameters is:

$$\Theta=\{\boldsymbol{\mathcal{H}}_1^k,\ldots,\boldsymbol{\mathcal{H}}_6^k,\boldsymbol{\mathcal{P}}^k,\mathbf{g}^k,\lambda_1^k,\lambda_2^k,\gamma^k,\alpha^k,\tau_L^k,\tau_M^k,\rho^k\}_{k=1,\ldots,K}. \tag{27}$$

The forward pass of this second deep unfolding network is detailed in Algorithm 2 and the corresponding flowchart is given in Fig. 4. In this model, a feature map $\boldsymbol{\mathcal{M}}^k$ is computed at each layer k with the multi-channel convolution kernel $\boldsymbol{\mathcal{H}}_1^k$, which is then given as input to the $\mathbf{L}$ and $\boldsymbol{\mathcal{M}}$ branches. It is only at the output of the $\boldsymbol{\mathcal{M}}$ branch that the foreground mask is converted back to the image domain, before entering the $\boldsymbol{U}$ branch.

Algorithm 2 Forward Pass ROMAN-R

Input: $\boldsymbol{D}, \boldsymbol{M}^0, \boldsymbol{L}^0, \boldsymbol{U}^0$
Output: $\widehat{\mathbf{M}}$, $\widehat{\mathbf{L}}$.
for k = *1 to* K **do**
  // Compute previous mask features
  $\boldsymbol{\mathcal{M}}^{k-1} = \boldsymbol{\mathcal{H}}_1^k * \boldsymbol{M}^{k-1}$
  // L branch
  $\boldsymbol{W} := \mathbf{1} - \boldsymbol{\mathcal{H}}_2^k * \boldsymbol{\mathcal{M}}^{k-1}$
  $\boldsymbol{\Lambda}_0 := \boldsymbol{W}^{\circ 2} \circ (\boldsymbol{L}^{k-1} - \boldsymbol{D}) + \frac{1}{\rho^k}\boldsymbol{W} \circ \boldsymbol{U}^{k-1}$
  $\boldsymbol{L}^k = \Gamma_{\gamma^k}\big(\boldsymbol{L}^{k-1} - \tau_L^k\boldsymbol{\Lambda}_0\big)$
  // M branch
  $\boldsymbol{W} := \boldsymbol{D} - \boldsymbol{L}^k$
  $\boldsymbol{\Lambda}_0 := \boldsymbol{\mathcal{H}}_4^k * (\boldsymbol{W}^{\circ 2} \circ (\mathbf{1} - \boldsymbol{\mathcal{H}}_3^k * \boldsymbol{\mathcal{M}}^{k-1}))$
  $\boldsymbol{\Lambda}_1 := \frac{1}{\rho^k}\boldsymbol{\mathcal{H}}_5^k * (\boldsymbol{W} \circ \boldsymbol{U}^{k-1})$
  $\widetilde{\boldsymbol{\mathcal{M}}} = [\boldsymbol{\mathcal{M}}_1^{k-1}, \boldsymbol{\mathcal{P}}^k * \boldsymbol{\mathcal{M}}_1^{k-1}, \ldots, \boldsymbol{\mathcal{P}}^k * \boldsymbol{\mathcal{M}}_{T-1}^{k-1}]$
  $\boldsymbol{\mathcal{M}}^k = \Phi_{\lambda_1^k,\lambda_2^k,\mathbf{G}^k}\big(\boldsymbol{\mathcal{M}}^{k-1} + \tau_M^k(\boldsymbol{\Lambda}_0 - \boldsymbol{\Lambda}_1); \widetilde{\boldsymbol{\mathcal{M}}}\big)$
  $\boldsymbol{M}^k = \sigma_{\alpha^k}\big(\boldsymbol{\mathcal{H}}_6^k * \boldsymbol{\mathcal{M}}^k\big)$
  // U branch
  $\boldsymbol{U}^k = \boldsymbol{U}^{k-1} + \rho^k(\mathbf{1} - \boldsymbol{M}^k) \circ (\boldsymbol{L}^k - \boldsymbol{D})$
end
return $\widehat{\mathbf{M}} = \boldsymbol{M}^K$, $\widehat{\mathbf{L}} = \boldsymbol{L}^K$.

IV. Experiments

A. Foreground Detection and Background Modeling on Synthetic Data

We first assess the performance of our models in the task of video background separation and foreground detection on the synthetic moving MNIST dataset [51]. We work on 20 frames long sequences of size 32×32 pixels. Out of the 10,000 video sequences of moving digits, we create validation and test sets of 1,000 samples each. The synthetic low-rank background is generated as in [6], which is, by setting $\mathbf{L} \doteq \mathbf{U}\mathbf{V}^{\top}$, with $\mathbf{U}\in\mathbb{R}^{n\times r}$ and $\mathbf{V}\in\mathbb{R}^{m\times r}$ sampled from a standard Gaussian distribution and the rank set to $r=5$. The ground truth for the foreground mask is generated by applying a threshold of 0.2 to the original digits in the video.

1) Training: Since the video background is perfectly known in this case, we consider a fully-supervised loss function $\mathcal{L}_{\text{fs}}$ in (28) that optimizes the reconstruction of the background using the mean squared error (MSE) as well as the estimation of the foreground mask using the binary cross-entropy loss (BCE):

$$\begin{aligned}\mathcal{L}_{\text{fs}}(\mathbf{M},\mathbf{L},\widehat{\mathbf{M}},\widehat{\mathbf{L}}) &= \alpha\ \text{BCE}(\mathbf{M},\widehat{\mathbf{M}}) + \text{MSE}(\mathbf{L},\widehat{\mathbf{L}}) \\ &= \alpha\sum_{x,y,t} -M_{x,y,t}\log(\widehat{M}_{x,y,t}) + \frac{1}{2}\|\mathbf{L}-\widehat{\mathbf{L}}\|_F^2.\end{aligned} \tag{28}$$

TABLE I
MOVING-MNIST WITH SUPERVISED $\mathcal{L}_{\text{fs}}$ LOSS

	1 layer		2 layers		3 layers		4 layers		5 layers	
Model	MSE L	F_1	MSE L	F_1	MSE L	F_1	MSE L	F_1	MSE L	F_1
CORONA	3.58×10^{-3}	0.952	7.14×10^{-4}	0.968	5.41×10^{-4}	0.970	3.99×10^{-4}	0.973	1.99×10^{-3}	0.975
refRPCA-net	3.58×10^{-3}	0.955	5.56×10^{-4}	0.971	2.21×10^{-3}	0.974	3.59×10^{-4}	0.973	1.79×10^{-3}	0.975
ROMAN-S	2.83×10^{-3}	0.971	2.40×10^{-3}	0.976	1.66×10^{-3}	0.981	1.60×10^{-3}	0.986	1.54×10^{-3}	0.981
ROMAN-R	4.17×10^{-3}	0.971	4.53×10^{-4}	0.989	1.54×10^{-3}	0.982	9.30×10^{-5}	0.995	6.80×10^{-5}	0.995

The relative weight of the two loss components can be controlled via a parameter α, which is set to 1 in our experiments. We use the ADAM optimizer with an initial learning rate of 0.005 by decreasing it every 25 epochs by a factor of 0.3, for a total of 75 epochs. The batch size is set to 64. We set the number of channels in the transform domain to 8 for ROMAN-R, which is the number of filters used in the corresponding 3D convolutional layers. The convolution kernels $\boldsymbol{\mathcal{H}}_i^k$ of our models are initialized with a uniform distribution. $\boldsymbol{\mathcal{P}}_i^k$ are initially set to Gaussian kernels with small variance, promoting local correlations. $\mathbf{g}^k$ are initialized to the all-ones. ρ^k, τ_M^k and τ_L^k were all initialized to 1.0 and the sigmoid scaling parameters α^k to 5.0. The initial values of the thresholds are set to $\lambda_1^k = 0.05$, $\lambda_2^k = 0.001$ and $\gamma^k = 0.1$.

2) Evaluation: The test performance is evaluated using the MSE loss on the background component. We also compute the F_1 score defined by

$$F_1 = 2\frac{\text{precison} \times \text{recall}}{\text{precision} + \text{recall}}, \tag{29}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{30}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{31}$$

which is a measure typically used to assess the quality of the foreground mask in such applications [24], [26]. This score is computed per sequence and then averaged over the number of test samples.

3) Experimental Results: We compare ROMAN-S and ROMAN-R against CORONA [38] and our prior refRPCA-Net [39]. These existing deep unfolding RPCA networks directly estimate the foreground component as the pixel difference with the low-rank background model. Therefore, to calculate the F_1 score for these models, we add a 3×3 convolutional layer with softmax activation to predict a probabilistic foreground mask. Table I reports the MSE and the F_1 scores obtained on the test set for different number of hidden layers. We observe a systematic improvement on the estimation of the foreground mask with our proposed models, thereby corroborating the efficacy of the sparse mask formulation. Overall, the F_1 score increases with the number of layers with a peak performance at 4 layers for ROMAN-S and ROMAN-R, and 5 layers for CORONA and refRPCA-Net.

B. Results on Real Video Sequences

1) Dataset: We train and evaluate our models on various videos from the CDNet2014 dataset [26]. This dataset contains 11 video categories, corresponding to different challenges in background subtraction, with 4 to 6 videos per category. Compared to other real video datasets, CDNet2014 provides ground truth pixel-wise foreground masks for every frame, with integer labels corresponding to the background, foreground, unknown, hard-shadow and outside-of-ROI classes. However, no reference backgrounds are available. We rule out 2 categories from the dataset, which are the Intermittent Object Motion (IOM) and Pan-Tilt-Zoom (PTZ) categories since the former mostly contains sequences with very small ROIs—thus, leaving only few labeled objects to train on—and the latter contains continuous camera motion, which is outside of the scope of our model. Also, we intentionally remove the "port" sequence from the Low-Framerate (LFR) category due to its very small ROI, as well as the "fountain01" sequence from the Dynamic Background (DB) category. When fed to the neural network, the video sequences are first converted to the gray color scale, split into 50 frame long segments and resized to a maximum width of 128 pixels using bilinear interpolation. Likewise, the ground truth masks used for supervised training are downscaled using nearest neighbor interpolation, and pixels corresponding to hard-shadow regions are relabeled as background. The "unknown motion" pixels are treated stochastically by converting them to background or foreground regions for each video segment with a probability of 0.5, which acts as some kind of data augmentation and results in slightly more robust performance.

We choose to evaluate our deep unfolding models in a scene-specific setting, where 40% of the available video frames are selected for training and hyperparameter selection, and the remaining 60% unseen frames are used for testing. In this setting, the test performance may fluctuate depending on the presence of challenging video segments or the absence of motion within the test set; hence, we always report metrics averaged over 5 runs on different dataset splits. Per-video training is especially suitable for deep unfolding models since they are able to generalize well with only few training examples and because optimal sparsity and SVT thresholds are largely dependent on the video content. Later, in Section IV-C, we also study the generalization performance of the ROMAN networks and a 3D U-Net baseline on unseen video of different scenes, with similar and dissimilar properties.

2) Training: Since we do not have access to true video background in the real video setting, we opt for a semi-supervised composite loss $\mathcal{L}_{\text{ss}}$ defined as,

$$\begin{aligned}\mathcal{L}_{\text{ss}}(\mathbf{M}, \mathbf{D}, \widehat{\mathbf{M}}, \widehat{\mathbf{L}}) \\ = \alpha_1\ \text{BCE}(\mathbf{M}, \widehat{\mathbf{M}}) + \alpha_2\ \text{Tversky}(\mathbf{M}, \widehat{\mathbf{M}}) \\ + \text{MSE}((\mathbf{1} - \mathbf{M}) \circ \mathbf{D}, (\mathbf{1} - \mathbf{M}) \circ \widehat{\mathbf{L}}),\end{aligned} \tag{32}$$

where the foreground mask is optimized using a combination of the BCE and Tversky losses, and the background is optimized with the MSE loss w.r.t the background of the original video located outside of the ground truth mask, that is, by masking $\mathbf{D}$ with $(\mathbf{1}-\mathbf{M})$. Semi-supervision occurs since the missing pixels are estimated via the low-rank parametrization of the background in the ROMAN models. However, as a potential side-effect, background pixels that are never visible during the chosen time span may be wrongly inferred.

The Tversky loss [52], which is formulated in Eq. (33) below, allows for a better control of the precision and recall in segmentation applications in the case of unbalanced classes,

$$\text{Tversky}(\mathbf{M}, \widehat{\mathbf{M}}) = 1 - \frac{\sum_i M_i \widehat{M}_i}{\sum_i M_i \widehat{M}_i + \eta_1 \sum_i M_i (1 - \widehat{M}_i) + \eta_2 \sum_i (1 - M_i) \widehat{M}_i}. \tag{33}$$

We empirically set η_1 and η_2 to 0.5, which is effectively equivalent to the binary Dice loss [53]. For the ROMAN models, we find that using the combination of losses in Eq. (32), with $\alpha_1 = \alpha_2 = 0.5$, yields better results than optimizing the Tversky or BCE losses alone, and the trained models achieve optimal F_1 scores at a fixed threshold of 0.5 on the foreground probability masks. However, this is not the case for refRPCA-Net and CORONA, where training with the Tversky loss would prevent learning convergence. Hence, it is best to train these models using the BCE loss only and optimize the decision threshold after training.

The models are trained using the ADAM optimizer for 90 epochs with an initial learning rate of 0.003, which is decreased by a factor of 0.3 every 30 epochs. We use 3-layers models and the number of channels in the transform domain is set to 32 for ROMAN-R. We use the same initialization as in Section IV-A1, except for the thresholds. Specifically, for every scene, $\lambda_1^k = 0.01$ was found to be a good initial value, while γ^k and λ_2^k are respectively initialized within the ranges $[0.25, 0.8]$ and $[0.001, 0.01]$ using grid search and by partially training the model to ensure stable outputs and early convergence on the training set.

Finally, we also train a conventional deep 3D-CNN following the U-Net architecture [44] and inflating the convolution kernels to three-dimensional ones to capture temporal features. Training and evaluation are performed per sequence using the same 5 dataset folds for fair comparison. U-Nets have been extensively used in semantic segmentation tasks, both on two-dimensional and three-dimensional data. Consequently, we only train the 3D U-Net to predict the foreground mask by optimizing a combination of the Tversky and the cross-entropy losses, contrary to the RPCA-based models that also estimate the sequence background.

3) Experimental Results: We compute the per-sequence precision, recall and F_1 score metrics, averaged over the 5 test set splits. The "unknown motion" and outside-of-ROI pixels are ignored during the count, following the CDNet2014 evaluation protocol. Since we average over 5 runs, we deliberately ignore models that lead to an F_1 score lower than 0.5, which can happen in exceptional occasions due to a bad selection of training or testing samples within the split. All results are reported in Table II. As for the deep unfolding models, we notice that ROMAN-R outperforms the other alternatives in almost all cases, followed by ROMAN-S, refRPCA-Net and CORONA in order of decreasing performance. Results indicate that using the proposed mask formulation is better suited than the traditional deep unfolding RPCA models for the task of foreground detection, especially when training samples consist of binary masks. Moreover, the higher representation learning power of ROMAN-R along with its reweighting scheme leads to superior performance compared to ROMAN-S. 3D U-Net offers comparable performance to ROMAN-R for the foreground detection task, although this network is not trained to reconstruct the video background.

In Fig. 5, we provide a series of test samples over different categories and for each model, which are comprised of the estimated background and the raw non-thresholded foreground probability map (except 3D U-Net that is only trained to segment the foreground). In complement, we provide receiver operating characteristics (ROC) for two example scenes in Fig. 6. From both figures, we observe that ROMAN-R and ROMAN-S are more robust than the other deep unfolding models that classify objects based on foreground pixel intensity, since the foreground membership probabilities in the ROMAN networks are less dependent on the foreground pixel values and object textures than refRPCA-net and CORONA. This is more apparent for difficult scenes (e.g.: dynamic backgrounds and camera jitter) where these models struggle to provide accurate masks and clear backgrounds. An increase in AUC is also observed for the ROMAN models when using the side-information scheme, compared to their counterparts without side-information, demonstrating the effectiveness of the foreground-correlation scheme.

A demonstration of the learning convergence of the ROMAN models is given in Fig. 7 and shows the progress of the training and test metrics after each epoch. For each layer k, we also report the norm of the gap between the side-information kernels $\boldsymbol{\mathcal{P}}^k$ with their initial values (that is, $\|\boldsymbol{\mathcal{P}}^k_{\text{init}} - \boldsymbol{\mathcal{P}}^k_e\|_F$ at training epoch e), as well as the norm of their update after every epoch (that is, $\|\boldsymbol{\mathcal{P}}^k_{e-1} - \boldsymbol{\mathcal{P}}^k_e\|_F$). Likewise, we also report similar metrics for the other convolution kernels $\boldsymbol{\mathcal{H}}^k_i$ of both models, which are averaged over all kernels. We observe that ROMAN-R and ROMAN-S are able to reach good levels of performance in few epochs when trained with ADAM.

4) Study of the Side-Information: We study the effectiveness of the side-information scheme based on ℓ_1-ℓ_1 minimization for ROMAN-S and reweighted-ℓ_1-ℓ_1 minimization for ROMAN-R. To do so, we train model alternatives by removing the side-information branches; this can be done by changing the non-linear activations to the simple soft-thresholding activations presented in Figs. 3 and 4 (and by keeping the weights g_i for ROMAN-R). This effectively cancels the side-information branches. Table III reports the gains in precision, recall and F_1 scores obtained respectively for both models when using the proposed side-information scheme. These gains are averaged for each video category and the same subsets of video sequences are taken from the base

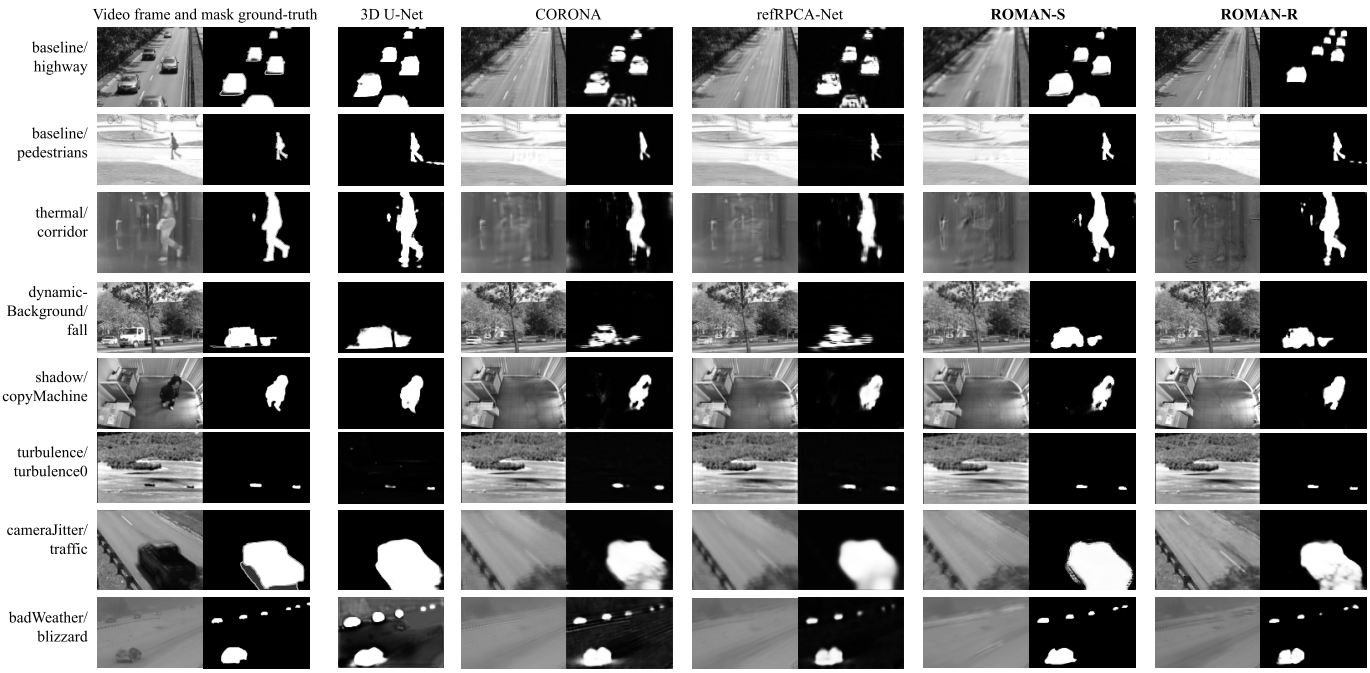


Fig. 5. Real video samples with raw output masks. Pixels in black, dark grey, light gray and white in the ground truth masks correspond to background, unknown motion, shadow and foreground pixels, respectively.

TABLE II

PRECISION, RECALL AND F_1 SCORES AVERAGED OVER THE TEST FRAMES OF THE 5 SPLITS FOR EACH SEQUENCE FROM THE CDNET2014 DATASET. BOLD INDICATES HIGHEST F_1 SCORES

		3D U-Net			**CORONA**			**refRPCA-Net**			**ROMAN-S**			**ROMAN-R**		
Category	Scene	pre	rec	F_1	pre	rec	F_1	pre	rec	F_1	pre	rec	F_1	pre	rec	F_1
baseline	highway	0.88	1.00	0.93	0.80	0.86	0.83	0.84	0.88	0.86	0.88	0.99	0.93	0.98	0.95	**0.97**
	office	0.96	0.98	**0.97**	0.49	0.57	0.53	0.55	0.67	0.60	0.80	0.77	0.78	0.89	0.87	0.88
	pedestrians	0.72	0.97	0.83	0.93	0.81	0.87	0.95	0.81	0.87	0.92	0.97	**0.95**	0.88	0.97	0.92
	PETS2006	0.80	0.96	0.87	0.76	0.56	0.63	0.79	0.59	0.67	0.92	0.85	0.88	0.92	0.88	**0.90**
lowFramerate	tramCrossroad_1fps	0.52	0.50	0.54	0.77	0.52	0.60	0.53	0.65	0.58	0.87	0.80	**0.82**	0.86	0.76	0.79
	tunnelExit_0_35fps	0.79	0.70	0.73	0.73	0.59	0.65	0.70	0.69	0.69	0.86	0.41	0.55	0.97	0.90	**0.93**
	turnpike_0_5fps	0.69	0.95	0.78	0.84	0.80	0.82	0.81	0.77	0.79	0.93	0.88	0.90	0.98	0.78	**0.96**
thermal	corridor	0.93	0.97	**0.95**	0.88	0.75	0.81	0.89	0.85	0.87	0.93	0.89	0.91	0.96	0.94	**0.95**
	diningRoom	0.94	0.99	**0.97**	0.77	0.45	0.57	0.55	0.58	0.56	0.88	0.86	0.87	0.93	0.88	0.91
	lakeSide	0.81	0.98	**0.89**	0.47	0.66	0.55	0.55	0.58	0.56	0.70	0.71	0.70	0.83	0.82	0.82
	library	0.97	0.98	0.98	0.94	0.84	0.88	0.97	0.95	0.96	0.98	0.94	0.96	0.99	0.98	**0.99**
	park	0.75	0.80	0.76	0.85	0.62	0.71	0.75	0.73	0.73	0.80	0.93	0.86	0.84	0.91	**0.87**
shadow	backdoor	0.72	0.95	0.82	0.86	0.77	0.81	0.83	0.82	0.82	0.98	0.85	**0.91**	0.91	0.91	**0.91**
	bungalows	0.70	0.98	0.82	0.69	0.74	0.72	0.73	0.81	0.77	0.78	0.91	0.84	0.91	0.86	**0.88**
	busStation	0.75	0.94	**0.83**	0.75	0.44	0.56	0.63	0.80	0.70	0.57	0.84	0.75	0.78	0.89	**0.83**
	copyMachine	0.94	0.99	**0.97**	0.91	0.84	0.87	0.91	0.88	0.90	0.94	0.86	0.89	0.95	0.93	0.94
	cubicle	0.79	0.98	0.87	0.69	0.75	0.72	0.74	0.74	0.73	0.84	0.92	0.88	0.89	0.94	**0.91**
	peopleInShade	0.81	0.99	**0.89**	0.71	0.72	0.70	0.87	0.68	0.75	0.65	0.84	0.72	0.86	0.88	0.86
cameraJitter	badminton	0.73	0.91	**0.81**	0.66	0.54	0.59	0.83	0.67	0.74	0.86	0.66	0.75	0.84	0.62	0.71
	boulevard	0.85	0.96	**0.90**	0.68	0.66	0.66	0.73	0.70	0.71	0.80	0.83	0.81	0.89	0.90	0.89
	sidewalk	0.68	0.94	0.79	-	-	-	0.68	0.49	0.56	0.82	0.67	0.74	0.85	0.83	**0.84**
	traffic	0.75	0.98	0.85	0.80	0.81	0.80	0.84	0.80	0.82	0.89	0.87	0.88	0.93	0.90	**0.92**
dynamicBackground	boats	0.67	0.99	0.80	0.70	0.50	0.58	0.80	0.76	0.78	0.90	0.59	0.70	0.97	0.79	**0.87**
	canoe	0.82	0.96	**0.88**	0.70	0.68	0.68	0.81	0.66	0.72	0.92	0.83	0.87	0.83	0.93	**0.88**
	fall	0.83	0.89	**0.85**	0.78	0.55	0.64	0.76	0.67	0.71	0.89	0.67	0.76	0.91	0.79	0.84
	fountain02	0.60	0.85	0.69	0.75	0.66	0.70	0.73	0.53	0.60	0.89	0.87	**0.88**	0.87	0.90	**0.88**
	overpass	0.80	0.90	**0.84**	0.69	0.53	0.59	0.73	0.53	0.61	0.78	0.81	0.79	0.85	0.84	**0.84**
turbulence	turbulence0	0.72	0.78	0.75	0.75	0.80	0.74	0.83	0.72	0.76	0.85	0.94	0.89	0.92	0.91	**0.92**
	turbulence1	0.48	0.97	0.65	0.72	0.81	0.64	0.82	0.61	0.70	0.76	0.78	**0.77**	0.87	0.72	**0.77**
	turbulence2	0.58	0.90	0.70	0.80	0.95	0.70	0.94	0.73	0.81	0.97	0.60	0.73	0.97	0.74	**0.84**
	turbulence3	0.50	0.92	0.64	0.66	0.81	0.59	0.72	0.59	0.65	0.94	0.75	0.83	0.93	0.85	**0.89**
badWeather	blizzard	0.40	0.92	0.56	0.83	0.83	0.83	0.85	0.79	0.81	0.95	0.87	**0.91**	0.96	0.79	0.87
	skating	0.69	0.88	0.76	0.94	0.73	0.82	0.90	0.68	0.77	0.97	0.89	**0.93**	0.94	0.85	0.89
	snowfall	0.71	0.89	**0.79**	0.76	0.51	0.60	0.64	0.57	0.59	0.85	0.70	0.75	0.86	0.61	0.70
	wetSnow	0.52	0.62	0.55	0.70	0.59	0.63	0.69	0.54	0.59	0.66	0.69	0.67	0.82	0.76	**0.79**

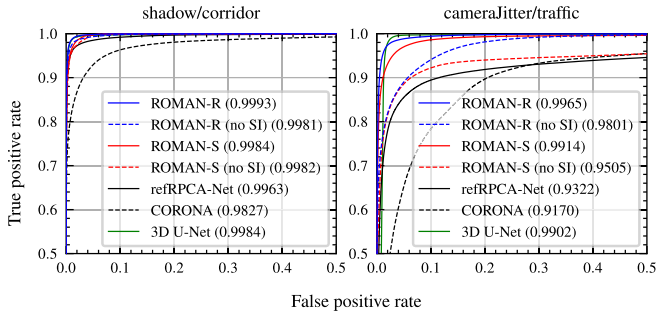


Fig. 6. ROC curve and AUC (in legend) for the thermal/corridor and cameraJitter/boulevard sequences. "no SI" stands for no side-information. Axis have been zoomed to the $[0, 0.5] \times [0.5, 1.0]$ range for clarity (on the left graph, the curve for U-Net is superimposed to ROMAN-R).

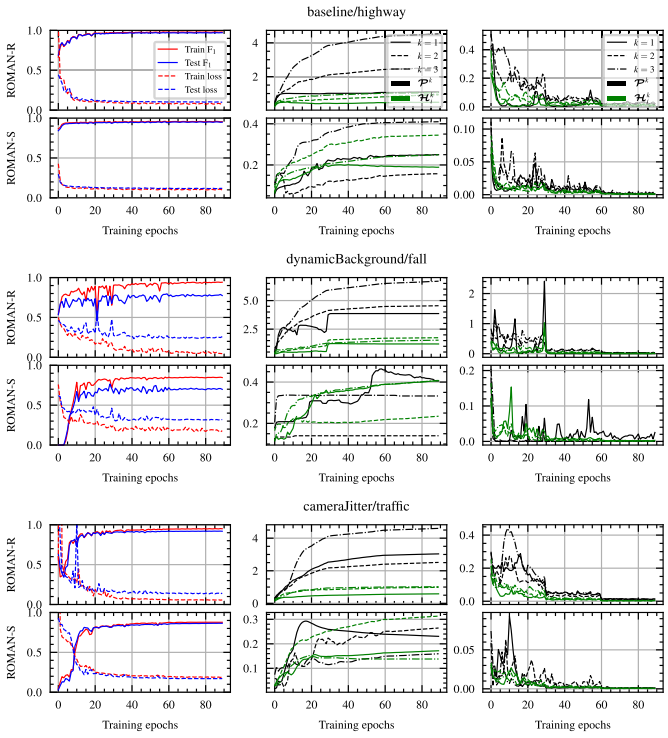


Fig. 7. Convergence of ROMAN-R and ROMAN-S. Left: training and test losses and F_1 scores. Middle: Frobenius norm of the difference between $\mathcal{P}^k_{\text{init}}$ and $\mathcal{P}^k$ after every epoch, for every layer $k = 1, 2, 3$, and also for the convolution kernels $\mathcal{H}^k_i$ (each curve reports the average across all kernels). Right: Frobenius norm of the difference between the previous value and the update of $\mathcal{P}^k$ after every epoch, and also for the convolution kernels $\mathcal{H}^k_i$.

simulations to train the models in a 5-fold cross-validation setting. We observe an overall gain in performance in most categories, with a higher overall gap for ROMAN-R as a result of the more efficient side-information scheme, when going to a higher-dimensional representation domain along with the feature reweighting coefficients g_i.

As a practical example, we provide a sample frame from the "traffic" sequence from the Camera Jitter category in Fig. 8 (top). There, the side-information branch shows to be useful to better discriminate between the actual object in motion and the background scene affected by the chaotic motion of the camera, which can be seen from the uncertain foreground probability maps in locations around the fence when no side-information is incorporated. Still, an exception is made

TABLE III

AVERAGE GAIN PER-CATEGORY IN PRECISION, RECALL AND F_1 SCORES OVER ROMAN-S AND ROMAN-R WITHOUT SIDE-INFORMATION BRANCHES

	ROMAN-S			ROMAN-R		
categories	pre	rec	F_1	pre	rec	F_1
baseline	+0.02	+0.05	+0.04	+0.06	+0.04	+0.05
lowFramerate	-0.01	+0.03	+0.00	-0.02	+0.04	+0.05
thermal	+0.03	+0.06	+0.04	+0.04	+0.06	+0.05
shadow	-0.06	+0.04	-0.00	+0.03	+0.08	+0.06
cameraJitter	-0.01	+0.09	+0.05	+0.02	+0.08	+0.05
dynamicBackground	+0.04	+0.12	+0.09	+0.02	+0.16	+0.11
turbulence	-0.01	+0.01	+0.00	+0.05	+0.01	+0.04
badWeather	+0.05	+0.02	+0.03	+0.01	+0.10	+0.07

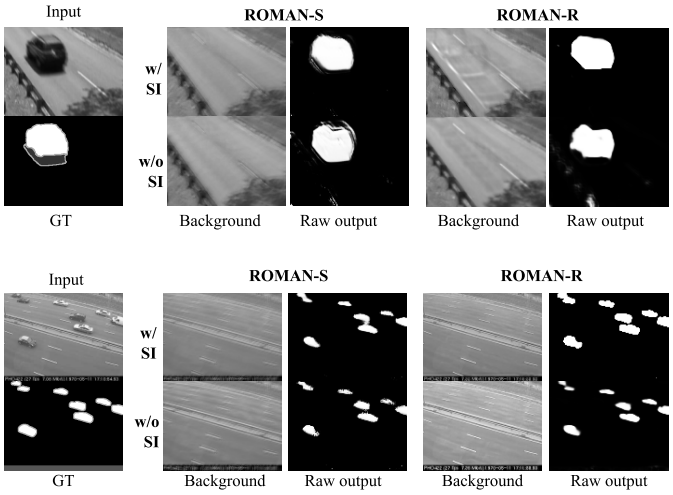


Fig. 8. Comparison of the outputs (raw mask outputs) with and without using side-information (SI) on the cameraJitter/traffic and lowFramerate/turnpike_0_5fps sequences.

for the low-framerate sequences, where the time-correlation between foreground objects is less relevant and renders the side-information branches useless; such an example is given in Fig. 8 (bottom) for the "turnpike" sequence that is acquired at 0.5 fps. In this case, the outputs are similar with and without side-information.

5) Hyperparameter Study: To show the influence of model hyperparameters on performance, we select some of the traffic-related videos and train ROMAN-R with varying number of layers. The depth of the convolutional dictionary is set to 8. The averaged F_1 scores for the 5 dataset splits are reported in Table IV. These results show that multi-layer models reach better performance, although simpler video scenes do not require a high layer count to reach peak accuracy. A second experiment is performed by using 3-layer models and changing the depth of the convolutional dictionary from single-channel, 8 and 32 channels. This directly impacts the reweighting and side-information schemes, since the number of feature maps $\mathcal{M}_i$ directly relates to the number of weights g_i. Table V shows a systematic drop in performance when using the single-channel architecture over the multiple-channels ones.

C. Generalization to Unseen Videos

We study the ability of the proposed models to generalize on unseen video, across various scenes. We follow a similar methodology as in [17] by creating training sets with different sizes and properties that affect the natural sparsity of

TABLE IV

F_1 SCORES ON TEST CLIPS FOR ROMAN-R WITH VARYING KERNEL DEPTHS (LAYERS=3)

layers:	1	2	3	4
baseline/highway	0.934	0.958	0.956	0.950
lowFramerate/tramCrossroad	0.844	0.800	0.835	0.827
cameraJitter/traffic	0.868	0.891	0.895	0.918
shadow/bungalows	0.816	0.839	0.854	0.850
badWeather/blizzard	0.881	0.899	0.877	0.878

TABLE V

F_1 SCORES ON TEST CLIPS FOR ROMAN-R WITH VARYING KERNEL DEPTHS (LAYERS=3)

kernel depth:	1	8	32
baseline/highway	0.933	0.956	0.967
lowFramerate/tramCrossroad	0.773	0.835	0.790
cameraJitter/traffic	0.877	0.895	0.918
shadow/bungalows	0.780	0.854	0.884
badWeather/blizzard	0.865	0.877	0.866

foregrounds and spectral norms of backgrounds. The videos in Table VI are selected for their common characteristic of having few to no intermittent objects in motion, unlike other clips of the CDNet2014 dataset. These are grouped into three categories according to the following image properties: Basic (B) clips, which are exempt of background noise and camera motion, Noisy (N) clips that feature dynamic and noisy backgrounds outside of the ground truth masks, and Jitter (J) video with constant camera vibrations. Additionally, we combine these three categories into a joined training set (B+N+J) containing a larger number of training samples and all kinds of perturbations. For training, we use identical hyper-parameters, optimizers and loss functions as in Section IV-B, except that entire videos are used for training and testing is performed on unseen clips both from within and outside the category considered for training. Model initialization differs only with the thresholds: γ^k is initialized to 0.5 and λ_1^k to 0.01, and as a rule of thumb, λ_2^k are initialized to 0.01 in the absence of noisy and jittering videos, and to 0.001 otherwise.

Generalization results are shown in Fig. 9, where each point corresponds to a video from the combined B+N+J test set and is placed vertically according to the model's F_1 performance on the unseen video, while the horizontal coordinate corresponds to the F_1 generalization score reported in Section IV-B in the intra-scene setting. Therefore, a point located close to the diagonal line reflects a good generalization performance. As a first observation, it appears that no model can outperform its counterpart trained in the intra-video setting, leading to an overall degradation of the testing performance. Still, ROMAN-R achieves the overall best generalization to unseen video, with an average degradation of 18% for testing video belonging to the chosen training set variation, compared to 37% and 46% for ROMAN-S and 3D U-Net, respectively. Second, ROMAN-R achieves similar generalization regardless of the training set variation and size, since the 3D U-Net does not generalize well if trained on clips from the Noisy or Jitter sets alone, which is also the case for ROMAN-S to a lesser extent. This also reflects how traditional deep models usually generalize better with larger and more diverse training datasets,

TABLE VI

TRAINING AND TESTING VIDEOS FOR THE STUDY OF GENERALIZATION TO UNSEEN VIDEOS

Datasets	Training videos	Testing videos
Basic (B)	baseline/PETS2006, shadow/backdoor, shadow/bungalows	baseline/highway, baseline/pedestrians, shadow/peopleInShade
Noisy (N)	badWeather/skating, dynamicBackground/fall, dynamicBackground/canoe, turbulence/turbulence3	badWeather/snowFall, turbulence/turbulence2
Jitter (J)	cameraJitter/traffic, cameraJitter/badminton	cameraJitter/boulevard

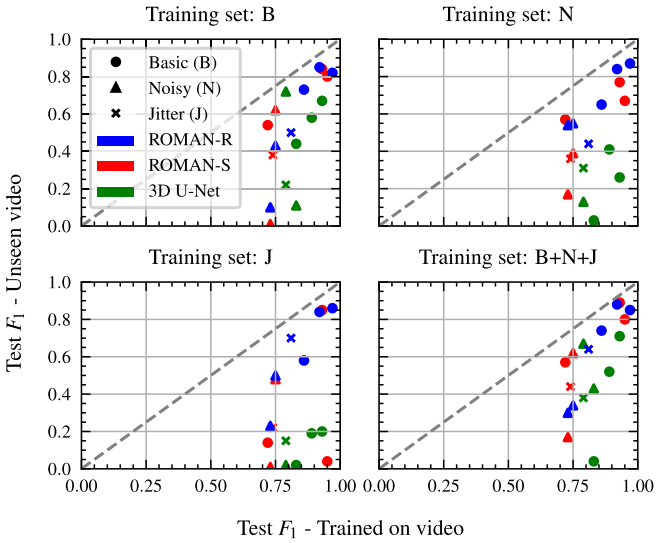


Fig. 9. Intra-scene (i.e., trained on video) v.s. cross-scene (i.e., unseen video) generalization performance of ROMAN-R, ROMAN-S and the 3D U-Net baseline and for each of the training set categories (B, N, J and B+N+J). Each marker type represents a video from corresponding testing categories.

Fig. 10. Example of predictions for shadow/peopleInShade: intra-scene (i.e., trained on partial video) v.s. cross-scene generalization (i.e., training on the Basic subset and testing on unseen video).

whereas the model-driven structures of deep unfolding networks are beneficial for smaller datasets with low-complexity priors. A visual example is provided in Fig. 10, showing the difference between the predictions of seen versus unseen clips, which also illustrate the transferability of the learned parameters of the proposed models to a different scene.

D. Comparison With Untrained Optimization

One advantage of deep unfolding neural networks is their ability to reach peak performance with fewer layers than the number of iterations required with the original untrained optimization algorithm. As a comparison, we evaluate the untrained version of ROMAN-S by removing the learnable

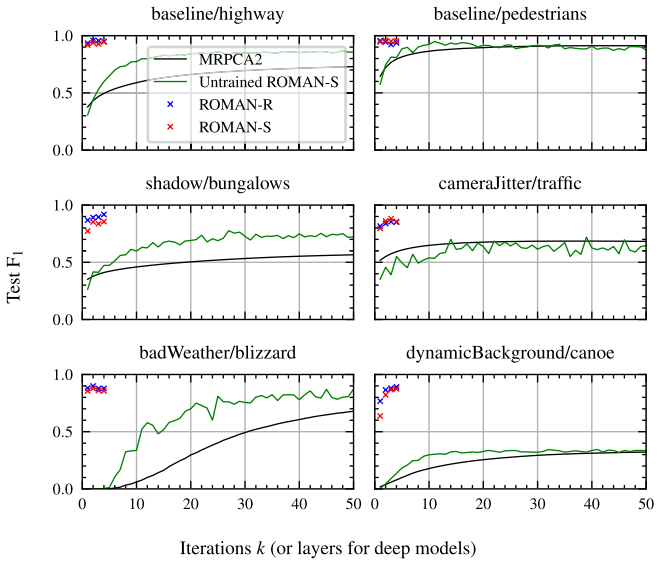


Fig. 11. Foreground detection F_1 score on test scenes of the CDNet2014 dataset: our deep unfolding models (each k corresponds to a number of layers) vs untrained ROMAN-S and MRPCA2 algorithm [16] (k corresponds to the number of iterations).

convolution kernels (corresponding to setting all measurement operators $\mathbf{H}_1$, $\mathbf{H}_2$ to $\mathbf{I}$), and by performing a grid search on λ_1, λ_2, ρ, α, τ_L and τ_M of algorithm 1 to select the best configuration on the training set. These parameters are kept constant for every iteration. Fig. 11 reports the F_1 score obtained with the deep unfolding models versus the untrained optimization model for various number of layers or iterations, on 4 different video sequences. ROMAN-S and ROMAN-R reach higher performance in very few iterations, while the untrained optimization method requires at least 10 iterations to settle with lower scores on the test sets. Fig. 11 also reports the F_1 score for the Masked-RPCA algorithm of [16] (referred to as MRPCA2), which consists of a Douglas-Rachford (DR) splitting algorithm to solve the non-convex version of Problem. (6) and without side-information. For a fair comparison with our methods, we evaluate MRPCA2 on whole video by splitting the test clips into sequences of 50 contiguous frames, which differs from the original implementation of [16]. Its performance follows a similar trend to the untrained ROMAN-S, although the untrained version of our algorithm is able to perform better on some scenes.

E. Complexity Analysis

In our experimental configurations, ROMAN-S and ROMAN-R have 391 and 6,408 trainable parameters per layer, respectively, including the trainable thresholds for the activation functions. These numbers increase linearly with the number of layers. The higher parameter count for the second model is due to the dictionary size in the transform domain, which translates to convolutional kernels with 32 channels in the proposed configuration. In contrast, CORONA and refRPCA-Net have approximately 290 trainable parameters each, and the U-Net baseline has substantially more trainable parameters (87.5 million), which is typical of deep models that perform feature extraction. The main computational bottleneck of the ROMAN models resides in the SVD computation; however, compared to the untrained RPCA optimization methods (which also require an SVD), the computational load is reduced at inference time due to the small number of layers than the number of optimization steps required to reach peak accuracy, as seen in Fig. 11.

V. Conclusion

Supervised learning of background separation models often relies on the estimation of foreground masks in the case of real data. For this aim, we proposed a family of deep unfolding neural networks that learns the iterations of alternating minimization algorithms for a masked RPCA model with side-information. The proposed unfolded networks require less layers than traditional optimization models, and our second model achieves competitive performance with semantic networks like the 3D U-Net, while requiring few parameters, small amount of data for training (a few labeled clips for each sequence), and is able to estimate the video background thanks to the low-rank model. Furthermore, the side-information scheme based on reweighted-ℓ_1-ℓ_1 minimization proves to be effective to promote the temporal correlation of foreground masks.

References

[1] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis," *J. ACM*, vol. 58, no. 3, pp. 1–37, Jun. 2011.
[2] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.
[3] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 22. Vancouver, BC, Canada, Dec. 2009, pp. 2080–2088.
[4] X. Liu, G. Zhao, J. Yao, and C. Qi, "Background subtraction based on low-rank and structured sparse decomposition," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2502–2514, Aug. 2015.
[5] S. Javed, T. Bouwmans, and S. K. Jung, "Improving OR-PCA via smoothed spatially-consistent low-rank modeling for background subtraction," in *Proc. Symp. Appl. Comput.*, Apr. 2017, pp. 89–94.
[6] H. Van Luong, N. Deligiannis, J. Seiler, S. Forchhammer, and A. Kaup, "Compressive online robust principal component analysis via n-ℓ_1 minimization," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4314–4329, Sep. 2018.
[7] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, nos. 5–6, pp. 877–905, Dec. 2008.
[8] Z. Kang, C. Peng, and Q. Cheng, "Robust PCA via nonconvex rank approximation," in *Proc. IEEE Int. Conf. Data Mining*, Nov. 2015, pp. 211–220.
[9] X. Yi, D. Park, Y. Chen, and C. Caramanis, "Fast algorithms for robust PCA via gradient descent," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 4159–4167.
[10] P. Netrapalli, U. N. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain, "Non-convex robust PCA," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27. Montreal, QC, Canada, Dec. 2014, pp. 1107–1115.
[11] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," Coordinated Science Lab., Univ. Illinois Urbana-Champaign, Urbana, IL, USA, Tech. Rep. DC-246, 2009.
[12] Z. Lin, M. Chen, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," 2010, *arXiv:1009.5055*.
[13] X. Yuan and J. Yang, "Sparse and low-rank matrix decomposition via alternating direction methods," *Pacific J. Optim.*, vol. 9, no. 1, p. 167, Jan. 2009.

[14] W. Cao et al., "Total variation regularized tensor RPCA for background subtraction from compressive measurements," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4075–4090, Sep. 2016.
[15] A. Khalilian-Gourtani, S. Minaee, and Y. Wang, "Masked-RPCA: Sparse and low-rank decomposition under overlaying model and application to moving object detection," 2019, *arXiv:1909.08049*.
[16] A. Khalilian-Gourtani, S. Minaee, and Y. Wang, "Masked-RPCA: Moving object detection with an overlaying model," *IEEE Open J. Signal Process.*, vol. 1, pp. 274–286, 2020.
[17] M. O. Tezcan, P. Ishwar, and J. Konrad, "BSUV-Net: A fully-convolutional neural network for background subtraction of unseen videos," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 2763–2772.
[18] L. A. Lim and H. Y. Keles, "Learning multi-scale features for foreground segmentation," *Pattern Anal. Appl.*, vol. 23, no. 3, pp. 1369–1380, Aug. 2020.
[19] J. Liao, G. Guo, Y. Yan, and H. Wang, "Multiscale cascaded scene-specific convolutional neural networks for background subtraction," in *Proc. Pacific Rim Conf. Multimedia*, Heifei, China, Sep. 2018, pp. 524–533.
[20] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recognit. Lett.*, vol. 96, pp. 66–75, Sep. 2017.
[21] D. Sakkos, H. Liu, J. Han, and L. Shao, "End-to-end video background subtraction with 3D convolutional neural networks," *Multimedia Tools Appl.*, vol. 77, no. 17, pp. 23023–23041, Sep. 2018.
[22] M. C. Bakkay, H. A. Rashwan, H. Salmane, L. Khoudour, D. Puig, and Y. Ruichek, "BSCGAN: Deep background subtraction with conditional generative adversarial networks," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 4018–4022.
[23] I. Osman, M. Abdelpakey, and M. S. Shehata, "TransBlast: Self-supervised learning using augmented subspace with transformer for background/foreground separation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 215–224.
[24] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, "Deep neural network concepts for background subtraction: A systematic review and comparative evaluation," *Neural Netw.*, vol. 117, pp. 8–66, Sep. 2019.
[25] J. H. Giraldo, H. T. Le, and T. Bouwmans, "Deep learning based background subtraction: A systematic survey," in *Handbook of Pattern Recognition and Computer Vision*. Singapore: World Scientific, Mar. 2020, pp. 51–73.
[26] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDNet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 393–400.
[27] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequievre, "A benchmark dataset for outdoor foreground/background extraction," in *Proc. Asian Conf. Comput. Vis. Workshop*, Daejeon, South Korea, Nov. 2012, pp. 291–300.
[28] L. Maddalena and A. Petrosino, "Towards benchmarking scene background initialization," in *Proc. Int. Conf. Image Anal. Process.*, Aug. 2015, pp. 469–476.
[29] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, Jun. 2010, pp. 399–406.
[30] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," *Proc. IEEE*, vol. 111, no. 5, pp. 465–499, May 2023.
[31] N. Shlezinger, Y. C. Eldar, and S. P. Boyd, "Model-based deep learning: On the intersection of deep learning and optimization," *IEEE Access*, vol. 10, pp. 115384–115398, 2022.
[32] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.
[33] H. Sreter and R. Giryes, "Learned convolutional sparse coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2191–2195.
[34] Y. Yan, S. Jian, H. Li, and Z. Xu, "Deep ADMM-Net for compressive sensing MRI," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, Dec. 2016, pp. 10–18.
[35] I. Marivani, E. Tsiligianni, B. Cornelis, and N. Deligiannis, "Multimodal deep unfolding for guided image super-resolution," *IEEE Trans. Image Process.*, vol. 29, pp. 8443–8456, 2020.
[36] S. Wisdom, T. Powers, J. Pitton, and L. Atlas, "Interpretable recurrent neural networks using sequential sparse recovery," 2016, *arXiv:1611.07252*.
[37] H. V. Luong, B. Joukovsky, and N. Deligiannis, "Designing interpretable recurrent neural networks for video reconstruction via deep unfolding," *IEEE Trans. Image Process.*, vol. 30, pp. 4099–4113, 2021.
[38] O. Solomon et al., "Deep unfolded robust PCA with application to clutter suppression in ultrasound," *IEEE Trans. Med. Imag.*, vol. 39, no. 4, pp. 1051–1063, Apr. 2020.
[39] H. Van Luong, B. Joukovsky, Y. C. Eldar, and N. Deligiannis, "A deep-unfolded reference-based RPCA network for video foreground-background separation," in *Proc. 28th Eur. Signal Process. Conf. (EUSIPCO)*, Jan. 2021, pp. 1432–1436.
[40] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *J. Roy. Stat. Soc. Ser. B, Stat. Methodol.*, vol. 67, no. 1, pp. 91–108, Feb. 2005.
[41] J. F. C. Mota, N. Deligiannis, and M. R. D. Rodrigues, "Compressed sensing with prior information: Strategies, geometry, and bounds," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4472–4496, Jul. 2017.
[42] J. F. C. Mota, L. Weizman, N. Deligiannis, Y. C. Eldar, and M. R. D. Rodrigues, "Reference-based compressed sensing: A sample complexity approach," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 4687–4691.
[43] H. D. Le, H. Van Luong, and N. Deligiannis, "Designing recurrent neural networks by unfolding an L1–L1 minimization algorithm," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2329–2333.
[44] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Oct. 2015, pp. 234–241.
[45] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM J. Optim.*, vol. 21, no. 2, pp. 572–596, Apr. 2011.
[46] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010.
[47] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, Jan. 2009.
[48] J. Wright, A. Ganesh, K. Min, and Y. Ma, "Compressive principal component pursuit," *Inf. Inference, A J. IMA*, vol. 2, no. 1, pp. 32–68, Jun. 2013.
[49] J. Liu and X. Chen, "ALISTA: Analytic weights are as good as learned weights in LISTA," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–33. [Online]. Available: https://openreview.net/forum?id=B1lnzn0ctQ
[50] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel–Softmax," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=rkE3y85ee
[51] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using LSTMs," in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 37, F. Bach and D. Blei, Eds. Lille, France: PMLR, Jul. 2015, pp. 843–852.
[52] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, "Tversky loss function for image segmentation using 3D fully convolutional deep networks," in *Machine Learning in Medical Imaging*. Cham, Switzerland: Springer, 2017, pp. 379–387.
[53] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 565–571.

Boris Joukovsky (Graduate Student Member, IEEE) received the joint master's degree in electrical engineering from Université Libre de Bruxelles (ULB) and Vrije Universiteit Brussel (VUB), Belgium, in 2019, where he is currently pursuing the Ph.D. degree with the Department of Electronics and Informatics (ETRO), under the supervision of Prof. Nikos Deligiannis. In 2020, he obtained a Ph.D. Strategic Basic Research Fellowship from the Research Foundation Flanders (FWO), Belgium. His research interests include interpretable deep learning, deep unfolding, explainable AI, and learning theory with applications in image and video processing.

Yonina C. Eldar (Fellow, IEEE) received the B.Sc. degree in physics and the B.Sc. degree in electrical engineering from Tel-Aviv University (TAU), Tel-Aviv, Israel, in 1995 and 1996, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2002.

She is currently a Professor with the Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel, where she holds the Dorothy and Patrick Gorman Professorial Chair and heads the Center for Biomedical Engineering. Previously, she was a Professor with the Department of Electrical Engineering, Technion, where she held the Edwards Chair in Engineering. She is also a Visiting Professor with MIT, a Visiting Scientist with the Broad Institute, a Visiting Research Collaborator with Princeton, an Adjunct Professor with Duke University, an Advisory Professor with Fudan University, and a Distinguished Visiting Professor with Tsinghua University. She was a Visiting Professor with Stanford. She is a member of the Israel Academy of Sciences and Humanities (elected 2017) and the Academia Europaea (elected 2023), a EURASIP Fellow, a fellow of the Asia–Pacific Artificial Intelligence Association, and a fellow of the 8400 Health Network. Her research interests include the broad areas of statistical signal processing, sampling theory and compressed sensing, learning and optimization methods, and their applications to biology, medical imaging and optics.

Dr. Eldar has received many awards for excellence in research and teaching, including the IEEE Signal Processing Society Technical Achievement Award in 2013, the IEEE/AESS Fred Nathanson Memorial Radar Award in 2014, and the IEEE Kiyo Tomiyasu Award in 2016. She was a Horev Fellow of the Leaders in Science and Technology Program with Technion and an Alon Fellow. She received the Michael Bruno Memorial Award from the Rothschild Foundation, the Weizmann Prize for Exact Sciences, the Wolf Foundation Krill Prize for Excellence in Scientific Research, the Henry Taub Prize for Excellence in Research (twice), the Hershel Rich Innovation Award (three times), the Award for Women with Distinguished Contributions, the Andre and Bella Meyer Lectureship, the Career Development Chair at the Technion, the Muriel & David Jacknow Award for Excellence in Teaching, and the Technionâs Award for Excellence in Teaching (two times). She received several best paper awards and best demo awards together with her research students and colleagues, including the SIAM Outstanding Paper Prize, the UFFC Outstanding Paper Award, the Signal Processing Society Best Paper Award, and the IET Circuits, Devices and Systems Premium Award. She was selected as one of the 50 most influential women in Israel and Asia and she is a highly cited researcher. She was a member of the Young Israel Academy of Science and Humanities and the Israel Committee for Higher Education. She is the Editor-in-Chief of *Foundations and Trends in Signal Processing*, a member of the IEEE Sensor Array and Multichannel Technical Committee, and serves on several other IEEE committees. In the past, she was a Signal Processing Society Distinguished Lecturer and a member of the IEEE Signal Processing Theory and Methods and Bio Imaging Signal Processing technical committees. She served as an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING, *EURASIP Journal of Signal Processing*, *SIAM Journal on Matrix Analysis and Applications*, and *SIAM Journal on Imaging Sciences*. She was the co-chair and the technical co-chair of several international conferences and workshops. She is the author of the book *Sampling Theory: Beyond Bandlimited Systems* and the coauthor of seven other books.

Nikos Deligiannis (Member, IEEE) received the Diploma degree in electrical and computer engineering from the University of Patras, Patras, Greece, in 2006, and the Ph.D. degree (Hons.) in engineering sciences from Vrije Universiteit Brussel (VUB), Brussels, Belgium, in 2012.

From 2013 to 2015, he was a Senior Researcher with the Department of Electronic and Electrical Engineering, University College London, London, U.K. He is currently an Associate Professor with the Department of Electronics and Informatics (ETRO), VUB, and a Principal Investigator with imec, Leuven, Belgium. Since 2021, he has been serving as the Chair for the EURASIP Technical Area Committee on Signal and Data Analytics for Machine Learning. His current research interests include interpretable and explainable machine learning, image and video processing, computer vision, and distributed deep learning.

Dr. Deligiannis is a member of the EURASIP. He serves as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING and he was the Lead Guest Editor of the Special Issue on Understanding and Designing Deep Neural Networks of *EURASIP Journal on Advances in Signal Processing*.