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

SUBMISSION DATE / POSTED DATE

26-04-2022 / 02-05-2022

CITATION

Joukovsky, Boris; Deligiannis, Nikos; Eldar, Yonina C. (2022): Interpretable Neural Networks for Video Separation: Deep Unfolding RPCA with Foreground Masking. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.19658565.v1

DOI

10.36227/techrxiv.19658565.v1

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

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

***Abstract*—This paper presents 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 few data. In ROMAN-S, the correlation in time of successive binary masks is controlled with a side-information scheme 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 and comparisons are made with existing deep unfolding RPCA neural networks, which do not use a mask formulation for the foreground. The models are also compared to a U-Net baseline. Results show that our proposed models outperform other deep unfolding models, 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 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.

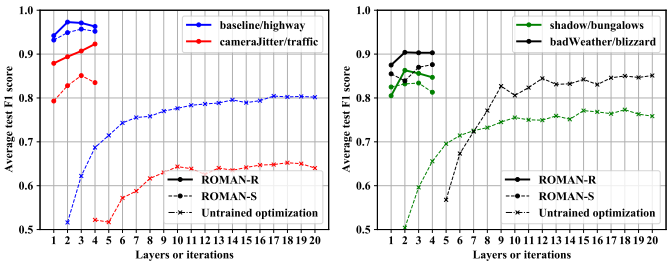


Fig. 1. Foreground detection F_1 score on test scenes of the CDNet2014 dataset: our deep unfolding models (number of layers) vs untrained optimization (number of iterations).

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] 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.

This research received funding from FWO, Belgium (research project G093817N and PhD fellowship strategic basic research 1SB5721N).

B. Joukovsky and N. Deligiannis are with the Department of Electronics and Informatics, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussels, Belgium and also with imec, Kapeldreef 75, B-3001 Leuven, Belgium.

Y. C. Eldar is with the Weizmann institute of Science, Rehovot 7610001, Israel.

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 parametrized and adaptive architecture; however, this same characteristic is responsible for their lack of interpretability and their design mostly follows empirical approaches. Most 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) [16], multi-scale segmentation networks [17], cascaded CNNs [18], [19], 3D-CNNs [20], generative adversarial networks (GANs) [21], and transformer-based networks [22]. We refer to [23], [24] 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 [25] or BMC2012 [26] datasets. In these cases, performance is measured in terms of foreground detection accuracy. In the case of the SBI dataset [27], 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 [28], [29], [30]. These models map the iterations of existing optimization algorithms to layers with learnable parameters, resulting in lightweight networks with enhanced interpretability thanks to the underlying optimization models and their ability to incoporate domain knowledge in the form of low-complexity structures in the data (e.g., sparsity and low-rankness). 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 [28]. A version of LISTA with convolution kernels has been proposed for the convolutional sparse coding task [31]. The alternating directions of multipliers method (ADMM) has also been unfolded with the ADMM-Net network [32]. Deep unfolding models have also been proposed for multi-modal data, such as the LMCSC-Net model which is based on sparse coding with side-information [33]. Models for sequential data include SISTA-RNN [34] that solves the problem of sparse signal reconstruction with correlation in time, and reweighted-RNN that solves a sequential video frame reconstruction [35].

Deep unfolding approaches have also been proposed for RPCA models: CORONA [36] is a convolutional RPCA model that learns an alternating projection algorithm applied to the task of clutter suppression in medical ultrasound imaging. Our prior work refRPCA-net [37] applies a similar technique to the task of video separation, by incoporating a side-information scheme so as to enforce the connectivity of successive sparse foregrounds. Most deep unfolding RPCA models still require to be trained in a fully-supervised manner using 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. First, unlike previous deep unfolding RPCA models [36], [37], 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. Second, ROMAN-S incorporates an efficient side-information scheme to promote the correlation of foreground masks in time, which is based on ℓ_1-ℓ_1 minimization [38], [39] and inspired by our prior refRPCA-Net model [37]. Contrary to ROMAN-S, our second model ROMAN-R takes the problems of the foreground mask and the side-information in an auxiliary transform domain using elements of convolutional sparse coding to increase its representation learning ability. By doing so, a learnable weight is assigned to each feature map via reweighted-ℓ_1-ℓ_1 minimization [40], which has been proven effective in reconstructing moving objects in video RNN models [41], [35]. Thereby, ROMAN-R provides a significant boost in performance over ROMAN-S. Third, 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. Finally, we train and evaluate our models on various categories of the CDNet2014 dataset [25] and compare with previous deep unfolding models, as well as with the U-Net [42] encoder-decoder model as a CNN baseline. We show that our models outperform existing deep unfolding RPCA models, and ROMAN-R is competitive with U-Net, while requiring substantially less training parameters.

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 [36] and our prior work refRPCA-Net [37]. 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 [25] as well as various ablation studies. We conclude in Section V.

II. Background on RPCA

A. RPCA as Principal Component Pursuit (PCP)

The original RPCA problem as described in [3], [1], [43] decomposes a data matrix $\mathbf{D}$ into a low-rank component $\mathbf{L}$ and a sparse component $\mathbf{S}$ as formulated in the following 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 Frobenious 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 [44], and $\mathbf{S}^{(k+1)}$ via the soft thresholding operator [45], since the latter subproblem effectively corresponds to a step of the iterative shrinkage-thresholding algorithm (ISTA) [45].

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 $\mathbf{s}_t$ $(t = 0,\dots,T)$ 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 can be included in the loss function, resulting in an n-ℓ_1 minimization problem. For instance, the refRPCA model [37] considers the previous signal $\mathbf{s}_{t-1}$ as side-information at time step t, which is incorporated into the model as follows:

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

where $\mathbf{S_P}$ is defined as $\mathbf{S_P} = [\mathbf{s}_1, \mathbf{P}\mathbf{s}_1, \cdots, \mathbf{P}\mathbf{s}_{m-1}]$, with $\mathbf{P}$ a correlation-promoting transform and $\mathbf{H}_1$, $\mathbf{H}_2$ generic measurement operators. 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 $\mathbf{S_P}$.

C. Deep Unfolding Methods

Deep unfolding methods outperform convex optimization methods and typically require less layers than otherwise required for optimization-based solvers [28], [46]. 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 bad generalization in the case of small training data. In this line of research, [36] proposed a deep unfolding convolutional RPCA (CORONA) network to solve the following RPCA model:

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

The matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ will be used to increase the power of the unfolded network. (3) was solved in [36] via iteratively updating $\mathbf{L}^{(k+1)}$ and $\mathbf{S}^{(k+1)}$ at iteration $k+1$ with

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

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

where $\|\cdot\|_{1,2}$ is the mixed $\ell_{1,2}$ norm, $\Gamma_{\frac{\lambda_1}{c}}(.)$ and $\Phi_{\frac{\lambda_2}{c}}(.)$ are the singular value thresholding and mixed $\ell_{1,2}$ soft thresholding [45] operators, respectively, and c is a Lipschitz constant. The corresponding deep unfolding architecture uses convolutional kernels $\mathbf{W}_1^{(k)},\cdots,\mathbf{W}_6^{(k)}$ at each layer k, as shown in Eq. (5a) and (5b). These parameters are all learned through backpropagation.

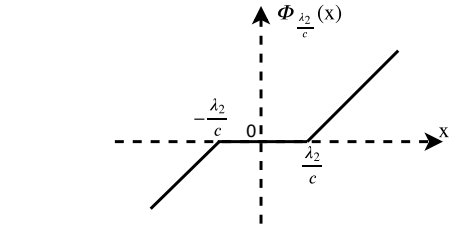
$$\mathbf{L}^{(k+1)} = \Gamma_{\lambda_1^{(k)}}\left\{\mathbf{W}_1^{(k)} * \mathbf{M} + \mathbf{W}_3^{(k)} * \mathbf{S}^{(k)} + \mathbf{W}_5^{(k)} * \mathbf{L}^{(k)}\right\}, \tag{5a}$$

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

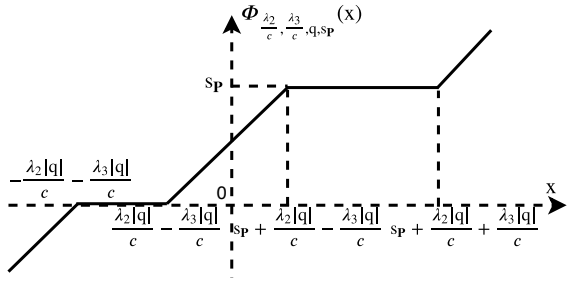
This approach was extended in [37] with the refRPCA-Net model, that solves the problem of RPCA with side-information based on (2) for the task of video separation. According to the principles of reweighted-ℓ_1 minimization [7], [39], reweighted-ℓ_1-ℓ_1 minimization with side information [6] and its deep unfolding counterpart [35], the refRPCA-Net model 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. 2(a), the activation function of refRPCA-Net in Fig. 2(b) features an additional flat region promoting the correlation with side information $\mathbf{s_P}$.

D. Masked RPCA

The Masked-RPCA (MRPCA) method [15], [47] 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



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



(b) refRPCA-Net: $\Phi_{\frac{\lambda_2}{c},\frac{\lambda_3}{c}\mathrm{q},\mathrm{s_P}}(\mathrm{x})$ for $\mathrm{s_P} \geq 0$.

Fig. 2. The soft-thresholding operator used in CORONA [36] (a) vs refRPCA-Net [37] (b). λ_2, λ_3 and c are learned globally. Note that (b) is drawn for given q and $\mathrm{s_P}$. 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 $\mathrm{s_P}$ defines the position of the non-zero plateau each time the operator is evaluated.

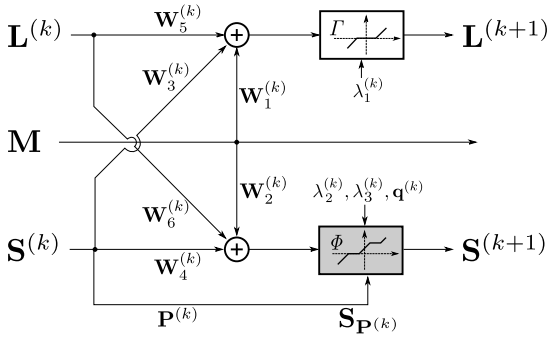


Fig. 3. refRPCA-Net architecture [37], equivalent to the CORONA model [36] with the bottom activation replaced by soft-thresholding.

without explicit computation of the true foreground. This is done by minimizing a reconstruction term which penalizes incorrect background pixels outside of the foreground mask, while constraining the mask values in the range $[0, 1]$:

$$\min_{\mathbf{L},\mathbf{M}} \|\mathbf{L}\|_* + \lambda\|\mathbf{M}\|_1 + \mathbb{1}_{[0,1]}(\mathbf{M}) + \frac{\rho}{2}\|(1-\mathbf{M}) \circ (\mathbf{L}-\mathbf{D})\|_F^2. \tag{6}$$

In (6), the masking operation is effectively realized with the Hadamard or element-wise product denoted by $\circ$. The parameters λ, ρ act as regularization coefficients and $\mathbb{1}_{[0,1]}$ is the indicator function of the interval $[0, 1]$. An algorithm to solve (6) was proposed in [15] 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:

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

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

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

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

$$\mathbf{L}^{i+1} = \Gamma\left(\mathbf{L}^i - \tau_L \nabla f(\mathbf{L})\right), \tag{10}$$

$$\mathbf{M}^{i+1} = \Pi\left(\Phi\left(\mathbf{M}^i - \tau_M \nabla g(\mathbf{M})\right)\right), \tag{11}$$

where Γ, Φ, Π correspond to the singular value thresholding, shrinkage-thresholding and $[0, 1]$-clamping operators, respectively. In particular, the gradients of the smooth parts ∇f and ∇g can be expressed as:

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

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

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 models 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 models 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 incoporated to promote the consistency of foreground mask in time. The second model (ROMAN-R) solves the mask subproblem in a transform domain via a learnable dictionnary 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 model, 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 $w \times h$, vectorized and stacked in the matrix $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_T]$, we seek to find a low-rank approximation $\mathbf{L} \in \mathbb{R}^{T \times wh}$ of $\mathbf{D}$ that models its static background and the sparse binary mask $\mathbf{M} \in \mathbb{R}^{T \times wh}$ 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 penalities, while the reconstruction term ensures that the known background pixels located outisde of the foreground mask match the original video content in $\mathbf{D}$. Similar to [37], 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 $\|\mathbf{M} - \tilde{\mathbf{M}}\|_1$ to be small, with a yet-to-be-learned linear transform $\mathbf{P}$:

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

In (14), 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 the Masked-RPCA model [47] 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 [36], [37]. We then follow a similar approach to [15] by reformulating the non-convex problem (14) in the augmented Lagrangian form with a dual variable $\mathbf{U}$. It is then solved using the ADMM procedure to alternately update $\mathbf{L}$, $\mathbf{M}$ and $\mathbf{U}$ according to the two following convex sub-problems:

$$\begin{aligned} \mathbf{L}^{i+1} =& \arg\min_{\mathbf{L}} \|\mathbf{L}\|_* \\ & + \rho \frac{1}{2} \|(\mathbf{1} - \mathbf{H}_1\mathbf{M}^i) \circ (\mathbf{D} - \mathbf{H}_2\mathbf{L}) + \frac{\mathbf{U}^i}{\rho}\|_F^2 \end{aligned} \tag{15}$$

$$\begin{aligned} \mathbf{M}^{i+1} =& \arg\min_{\mathbf{M}} \lambda_1 \|\mathbf{M}\|_1 + \lambda_2 \|\mathbf{M} - \widetilde{\mathbf{M}}\|_1 + \mathbb{1}_{[0,1]}(\mathbf{M}) \\ & + \rho \frac{1}{2} \|(\mathbf{1} - \mathbf{H}_1\mathbf{M}) \circ (\mathbf{D} - \mathbf{H}_2\mathbf{L}^{i+1}) + \frac{\mathbf{U}^i}{\rho}\|_F^2 \end{aligned} \tag{16}$$

$$\mathbf{U}^{i+1} = \mathbf{U}^i + \rho(\mathbf{1} - \mathbf{H}_1\mathbf{M}^{i+1}) \circ (\mathbf{D} - \mathbf{H}_2\mathbf{L}^{i+1}). \tag{17}$$

Following (10) and (11), the solutions of subproblems (15) and (16) can be obtained via proximal gradient updates. The explicit solutions are given below in (18) and (19), 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. 2(b) with coefficients q set to 1 (cf. [37]):

$$\begin{aligned} \mathbf{L}^{i+1} =& \Gamma_{\frac{\tau_L}{\rho}} \Big[\mathbf{L}^i - \tau_L \mathbf{H}_2^T \big((\mathbf{1} - \mathbf{H}_1\mathbf{M}^i)^{\circ 2} \circ (\mathbf{H}_2\mathbf{L}^i - \mathbf{D}) \\ & - (\mathbf{1} - \mathbf{H}_1\mathbf{M}^i) \circ \frac{\mathbf{U}^i}{\rho} \big) \Big] \end{aligned} \tag{18}$$

$$\begin{aligned} \mathbf{M}^{i+1} =& \Pi \Big[\Phi \Big[\mathbf{M}^i + \tau_M \mathbf{H}_1^T \big((\mathbf{D} - \mathbf{H}_2\mathbf{L}^i)^{\circ 2} \circ (\mathbf{1} - \mathbf{H}_1\mathbf{M}^i) \\ & - (\mathbf{D} - \mathbf{H}_2\mathbf{L}^i) \circ \frac{\mathbf{U}^i}{\rho} \big) \Big] \Big]. \end{aligned} \tag{19}$$

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 accross 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 (17), (18) and (19) is still applicable, and transposed matrix multiplications can be mapped to transposed 2D-convolutions.

We now build the ROMAN-S model 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.

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 an reparametrized sigmoid function $\sigma_{\alpha^k}(x) \equiv \text{sigmoid}(\alpha^k(x - 0.5))$ for the mask branch. This last activation function is similar to the gumbel-softmax activation [48] 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:

$$\Theta = \{\boldsymbol{\mathcal{H}}_1^k, \ldots, \boldsymbol{\mathcal{H}}_{11}^k, \boldsymbol{\mathcal{P}}^k, \lambda_1^k, \lambda_2^k, \gamma^k, \alpha^k, \tau_L^k, \tau_M^k, \rho^k\}_{k=1,\ldots,K}. \tag{20}$$

The overall network structure is illustrated in Fig. 4 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 model of Fig. 3 only contains two branches due its different underlying minimization algorithm. The Hadamard products are imple-

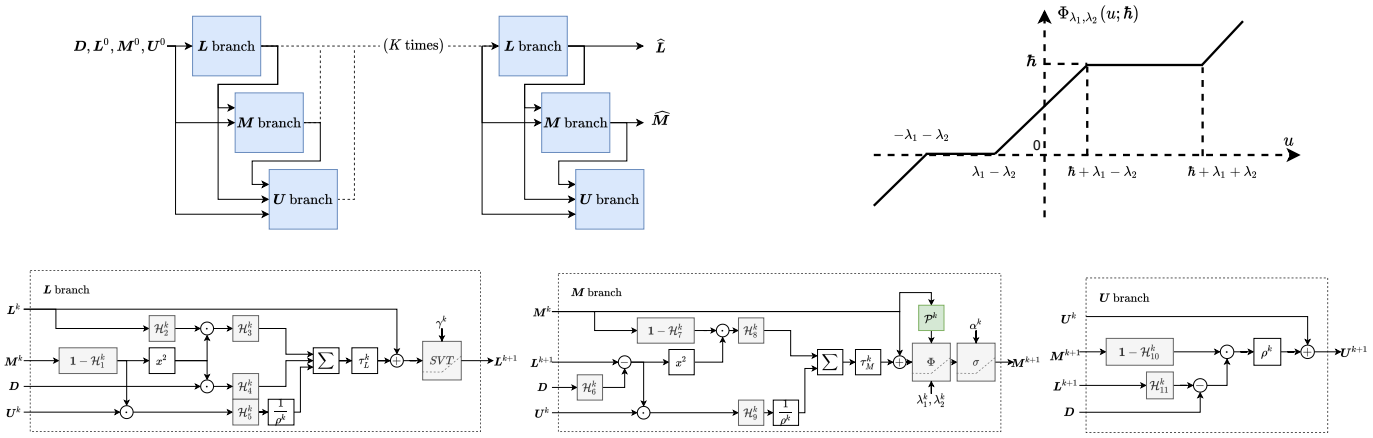


Fig. 4. Detail of the branches of the deep unfolding model **ROMAN-S**, and the soft-thresholding activation function with side-information $\Phi_{\lambda_1^k,\lambda_2^k}$.

Algorithm 1: Forward pass of ROMAN-S

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

mented 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 model takes the mask estimation problem into the transform domain by taking inspiration from the learned convolutional sparse coding technique [31]. 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, (21) becomes:

$$\begin{aligned} \min_{\mathbf{L},\mathbf{M}} \quad & \|\mathbf{L}\|_* + \lambda_1 \sum_i g_i \|\boldsymbol{\mathcal{M}}_i\|_1 + \lambda_2 \sum_i g_i \|\boldsymbol{\mathcal{M}}_i - \widetilde{\boldsymbol{\mathcal{M}}}_i\|_1 \\ s.t. \quad & (\mathbf{1} - \mathbf{M}) \circ (\mathbf{D} - \mathbf{L}) = 0 \\ & \mathbf{M} = \text{reshape}\left(\sum_i \boldsymbol{\Psi}_i * \boldsymbol{\mathcal{M}}_i\right) \\ & \mathbf{M} \in [0,1]^{wh \times T} . \end{aligned} \tag{21}$$

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. Moreover, $\mathbf{G}$ is a matrix formed by the corresponding weights g_i. Then, (21) can be reformulated in the augmented Lagrangian form:

$$\begin{aligned} \min_{\mathbf{L},\boldsymbol{\mathcal{M}}} \quad & \|\mathbf{L}\|_* + \lambda_1 \|\mathbf{G} \circ \boldsymbol{\mathcal{M}}\|_1 + \lambda_2 \|\mathbf{G} \circ (\boldsymbol{\mathcal{M}} - \widetilde{\boldsymbol{\mathcal{M}}})\|_1 \\ & + \mathbb{1}_{[0,1]}(\boldsymbol{\Psi}\boldsymbol{\mathcal{M}}) + \langle \mathbf{U}, (\mathbf{1} - \boldsymbol{\Psi}\boldsymbol{\mathcal{M}}) \circ (\mathbf{D} - \mathbf{L}) \rangle \\ & + \frac{\rho}{2} \|(\mathbf{1} - \boldsymbol{\Psi}\boldsymbol{\mathcal{M}}) \circ (\mathbf{D} - \mathbf{L})\|_F^2, \end{aligned} \tag{22}$$

where $\mathbf{U}$ is a dual variable 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

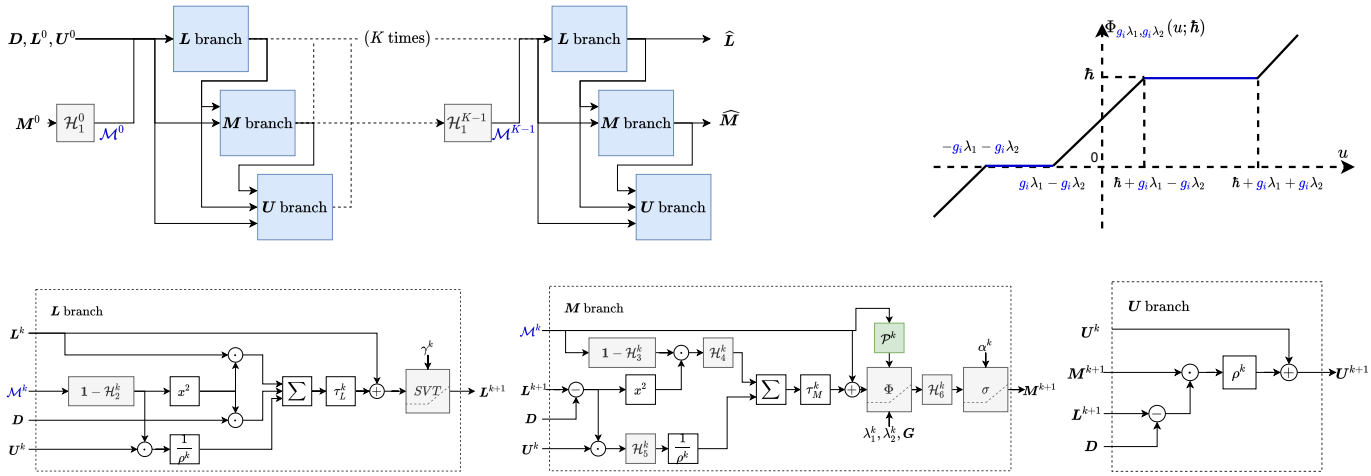


Fig. 5. Detail of the branches of the deep unfolding **ROMAN-R** model. The learnable activation functions is the reweighted soft-thresholding operator with side-information. Each channel has a different threshold scaling factor g_i.

Section III-A, we may write the following update equations for $\mathbf{L}, \boldsymbol{\mathcal{M}}$ and $\mathbf{U}$:

$$\mathbf{L}^{i+1} = \Gamma_{\frac{\tau_L}{\rho}}\Big[\mathbf{L}^i - \tau_L(\mathbf{1} - \boldsymbol{\Psi}\boldsymbol{\mathcal{M}}^i)^{\circ 2} \circ (\mathbf{L}^i - \mathbf{D}) - \tau_L(\mathbf{1} - \boldsymbol{\Psi}\boldsymbol{\mathcal{M}}^i) \circ \frac{\mathbf{U}^i}{\rho}\Big], \tag{23}$$

$$\boldsymbol{\mathcal{M}}^{i+1} = \boldsymbol{\Psi}\Pi\Bigg[\boldsymbol{\Psi}^T\Phi\Big[\boldsymbol{\mathcal{M}}^i + \tau_M\boldsymbol{\Psi}^T\big((\mathbf{D} - \mathbf{L}^i)^{\circ 2} \circ (\mathbf{1} - \boldsymbol{\Psi}\boldsymbol{\mathcal{M}}^i) - (\mathbf{D} - \mathbf{L}^i) \circ \frac{\mathbf{U}^i}{\rho}\big)\Big]\Bigg], \tag{24}$$

$$\mathbf{U}^{i+1} = \mathbf{U}^i + \rho(\mathbf{1} - \boldsymbol{\Psi}\boldsymbol{\mathcal{M}}^{i+1}) \circ (\mathbf{D} - \mathbf{L}^{i+1}), \tag{25}$$

there, 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 model results from unfolding of the iterations (23), (24) and (25). As opposed to ROMAN-S, the trainable convolutional kernels $\boldsymbol{\mathcal{H}}_i^k$ arise from $\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}^T$. Also, in the mask branch, most operations are performed in the transform domain, including the processing 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:

$$\Theta = \{\boldsymbol{\mathcal{H}}_1^k, \ldots, \boldsymbol{\mathcal{H}}_6^k, \boldsymbol{\mathcal{P}}^k, \mathbf{g}^k, \lambda_1^k, \lambda_2^k, \gamma^k, \alpha^k, \tau_L^k, \tau_M^k, \rho^k\}_{k=1,\ldots,K}. \tag{26}$$

The forward pass of this second deep unfolding network is detailed in Algorithm 2 and the corresponding flowchart is given in Fig. 5. 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 $\mathbf{U}$ branch.

Algorithm 2: Forward pass ROMAN-R

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

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 [49]. 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

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

low-rank background is generated as in [6], which is, by setting $\mathbf{L} \doteq \mathbf{U}\mathbf{V}^T$, 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 (27) that optimizes the reconstruction of the background using the mean-square-error (MSE) as well as the estimation of the foreground mask using the binary cross-entropy loss (BCE):

$$\begin{aligned}\mathcal{L}_{\text{fs}}(\mathbf{M}, \mathbf{L}, \widehat{\mathbf{M}}, \widehat{\mathbf{L}}) \\ &= \alpha\ \text{BCE}(\mathbf{M}, \widehat{\mathbf{M}}) + \text{MSE}(\mathbf{L}, \widehat{\mathbf{L}}) \\ &= \alpha \sum_{x,y,t} -M_{x,y,t} \log(\widehat{M}_{x,y,t}) + \frac{1}{2}\|\mathbf{L} - \widehat{\mathbf{L}}\|_F^2. \end{aligned} \tag{27}$$

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.

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

$$\text{F}_1 = 2\frac{\text{precison} \times \text{recall}}{\text{precision} + \text{recall}}, \tag{28}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{29}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{30}$$

which is a measure typically used to assess the quality of the foreground mask in such applications [23], [25]. 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 [36] and our prior refRPCA-Net [37]. 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 Data

1) Dataset: We train and evaluate our models on various videos from the CDNet2014 dataset [25]. 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 frames 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 relabelled 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% frames are used for testing. In this

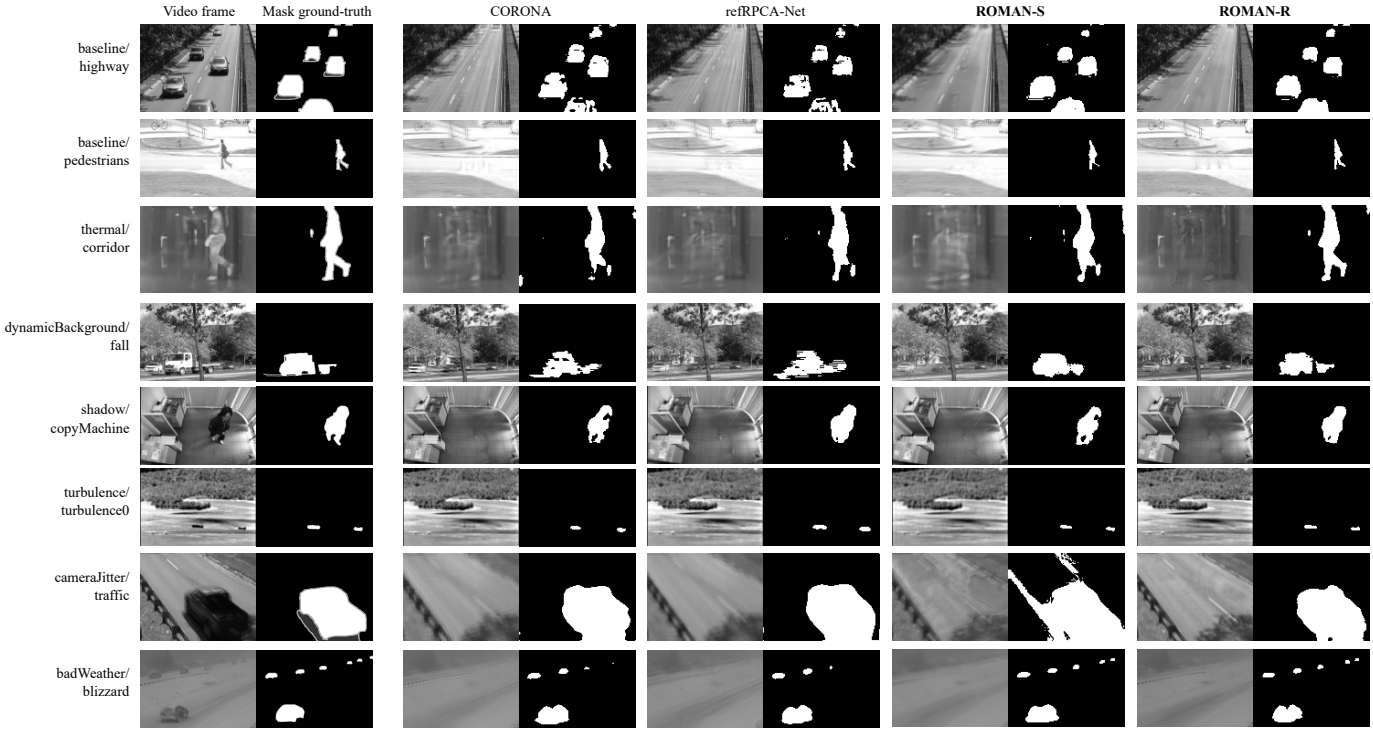


Fig. 6. Real video samples (with adaptive threshold). Pixels in black, dark grey, light gray and white in the ground truth masks correspond to background, unknown motion, shadow and foreground pixels, respectively.

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.

2) Training: Since we do not have access to true video background, we opt for a semi-supervised composite loss $\mathcal{L}_{\rm ss}$ defined in (31) that optimizes the BCE loss on the foreground mask and the MSE loss only on the known pixels outside of the foreground mask, which renders the task of background subtraction semi-supervised. These known pixels can be obtained directly from the input sequence by masking $\mathbf{D}$ with the negative of the ground-truth mask $(\mathbf{1}-\mathbf{M})$.

$$\begin{aligned}\mathcal{L}_{\rm ss}(\mathbf{M},\mathbf{D},\widehat{\mathbf{M}},\widehat{\mathbf{L}})&\\ &= \alpha\ \mathrm{BCE}(\mathbf{M},\widehat{\mathbf{M}}) + \mathrm{MSE}((\mathbf{1}-\mathbf{M})\circ\mathbf{D},(\mathbf{1}-\mathbf{M})\circ\widehat{\mathbf{L}})\\ &= \alpha\sum_{x,y,t} -M_{x,y,t}\log(\widehat{M}_{x,y,t}) + \frac{1}{2}\|(\mathbf{1}-\mathbf{M})\circ(\widehat{\mathbf{L}}-\mathbf{D})\|_F^2.\end{aligned} \tag{31}$$

The proposed loss, $\mathcal{L}_{\rm ss}$, is useful in the real video setting since the truth background cannot be known a priori. Nevertheless, we expect that the low-rank model will be successfully able to estimate the value of missing pixels in the background. However, a potential side-effect of this scheme is that background pixels which are never visible during the 50 frames span may be wrongly inferred. The models are trained using the ADAM optimizer for 80 epochs with an initial learning rate of 0.005, which is decreased by a factor 0.3 every 30 epochs. We use 3-layers models and the number of channels in the transform domain is set to 32 for Model 2. We also fine-tune a decision threshold τ on the training video sequences to classify the mask pixels as background if $M_{x,y,t} < \tau$, or as foreground otherwise. This selection is performed so as to maximize the F_1 score in (28), which is the standard evaluation metric adopted for the CDNet2014 dataset [25].

Finally, we also train a conventional deep CNN following the U-Net architecture [42]. 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 U-Net to predict the foreground mask by optimizing the cross-entropy loss, 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 oustide-of-ROI pixels are ignored during 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

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.

		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	0.94	0.96	0.95	**0.97**	0.97	**0.97**
	office	**0.96**	**0.98**	**0.97**	0.49	0.57	0.53	0.55	0.67	0.60	0.75	0.74	0.74	0.87	0.86	0.86
	pedestrians	0.66	**0.98**	0.79	0.93	0.81	0.87	0.95	0.81	0.87	**0.96**	0.95	0.95	**0.96**	0.94	**0.95**
	PETS2006	0.78	**0.96**	0.86	0.76	0.56	0.63	0.79	0.59	0.67	0.87	0.86	0.86	**0.91**	0.89	**0.90**
lowFramerate	tramCrossroad_1fps	-	-	-	0.77	0.52	0.60	0.53	0.65	0.58	0.83	0.59	0.66	**0.88**	**0.80**	**0.84**
	tunnelExit_0_35fps	0.82	**0.83**	0.82	0.73	0.59	0.65	0.70	0.69	0.69	0.82	0.57	0.66	**0.89**	**0.83**	**0.85**
	turnpike_0_5fps	0.77	**0.99**	0.84	0.84	0.80	0.82	0.81	0.77	0.79	0.83	0.79	0.81	**0.96**	0.87	**0.91**
thermal	corridor	0.92	**0.98**	**0.95**	0.88	0.75	0.81	0.89	0.85	0.87	0.92	0.90	0.91	**0.95**	0.94	**0.95**
	diningRoom	**0.93**	**0.99**	**0.96**	0.77	0.45	0.57	0.55	0.58	0.56	0.88	0.84	0.86	**0.93**	0.88	0.90
	lakeSide	0.75	**0.96**	**0.84**	0.47	0.66	0.55	0.55	0.58	0.56	0.62	0.59	0.60	**0.81**	0.80	0.80
	library	**0.98**	**1.00**	**0.99**	0.94	0.84	0.88	0.97	0.95	0.96	0.97	0.95	0.96	0.99	0.98	0.98
	park	0.71	**0.87**	0.78	0.85	0.62	0.71	0.75	0.73	0.73	**0.90**	0.85	0.87	0.89	**0.87**	**0.88**
shadow	backdoor	0.72	0.90	0.80	0.86	0.77	0.81	0.83	0.82	0.82	**0.91**	0.84	0.87	**0.91**	**0.93**	**0.92**
	bungalows	0.71	**0.98**	0.83	0.69	0.74	0.72	0.73	0.81	0.77	0.78	0.96	0.86	**0.85**	0.90	**0.88**
	busStation	0.74	**0.97**	**0.84**	0.75	0.44	0.56	0.63	0.80	0.70	0.68	0.80	0.73	**0.80**	0.82	0.81
	copyMachine	0.93	**1.00**	**0.96**	0.91	0.84	0.87	0.91	0.88	0.90	0.92	0.84	0.88	**0.95**	0.95	0.95
	cubicle	0.78	**1.00**	0.87	0.69	0.75	0.72	0.74	0.74	0.73	0.84	0.90	0.86	**0.88**	0.92	**0.90**
	peopleInShade	0.78	**0.99**	**0.88**	0.71	0.72	0.70	0.87	0.68	0.75	0.67	0.85	0.74	**0.91**	0.86	**0.88**
cameraJitter	badminton	0.73	**0.96**	**0.83**	0.66	0.54	0.59	0.83	0.67	0.74	0.71	0.71	0.71	**0.76**	0.75	0.75
	boulevard	0.87	**0.99**	**0.93**	0.68	0.66	0.66	0.73	0.70	0.71	0.84	0.78	0.80	**0.93**	0.90	0.92
	sidewalk	0.73	**0.93**	**0.82**	-	-	-	0.68	0.49	0.56	0.76	0.46	0.57	**0.85**	0.78	0.81
	traffic	0.77	**0.98**	0.87	0.80	0.81	0.80	0.84	0.80	0.82	0.80	0.91	0.85	**0.94**	0.91	**0.92**
dynamicBackground	boats	0.69	**1.00**	0.82	0.70	0.50	0.58	0.80	0.76	0.78	0.83	0.51	0.63	**0.86**	0.89	**0.83**
	canoe	0.79	**0.99**	0.88	0.70	0.68	0.68	0.81	0.66	0.72	0.84	0.75	0.79	**0.90**	0.91	**0.90**
	fall	0.82	**0.90**	**0.85**	0.78	0.55	0.64	0.76	0.67	0.71	0.79	0.75	0.76	**0.87**	0.76	0.81
	fountain02	0.52	**0.95**	0.67	0.75	0.66	0.70	0.73	0.53	0.60	0.74	0.76	0.73	**0.88**	0.92	**0.90**
	overpass	0.67	**1.00**	0.80	0.69	0.53	0.59	0.73	0.53	0.61	0.73	0.69	0.71	**0.81**	0.91	**0.86**
turbulence	turbulence0	0.64	**0.87**	0.73	0.75	0.80	0.74	0.83	0.72	0.76	0.86	0.81	0.83	**0.89**	0.86	**0.87**
	turbulence1	0.61	**0.75**	0.66	0.72	0.81	0.64	0.82	0.61	0.70	0.82	0.64	0.72	**0.86**	0.64	**0.73**
	turbulence2	**0.98**	0.75	0.83	0.80	0.95	0.70	0.94	0.73	0.81	0.88	0.65	0.73	0.94	**0.76**	**0.84**
	turbulence3	0.81	**0.88**	**0.84**	0.66	0.81	0.59	0.72	0.59	0.65	0.87	0.82	**0.84**	**0.89**	0.81	**0.84**
badWeather	blizzard	0.87	0.81	0.84	0.83	0.83	0.83	0.85	0.79	0.81	0.90	0.82	0.86	**0.93**	**0.84**	**0.88**
	skating	0.89	0.65	0.74	**0.94**	0.73	0.82	0.90	0.68	0.77	0.93	0.86	0.89	**0.94**	**0.87**	**0.91**
	snowfall	**0.81**	**0.88**	**0.85**	0.76	0.51	0.60	0.64	0.57	0.59	0.78	0.76	0.77	0.69	0.58	0.61
	wetSnow	**0.87**	**0.80**	**0.83**	0.70	0.59	0.63	0.69	0.54	0.59	0.64	0.57	0.60	0.73	**0.80**	0.76

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 lead to superior performance compared to ROMAN-S. 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. 6, we provide a series of test samples over different categories and for each model. The output masks are obtained by thresholding the raw output probability map with the optimal threshold found on the training set. We also provide the estimated background models for the considered frames.

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. 4 and 5 (and by keeping the weights g_i for ROMAN-R). This effectively cancels the side-information branches. Tables 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 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. 7 (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 and road markings when no side-information is incorporated. Still, an exception is made for the low-framerate sequences,

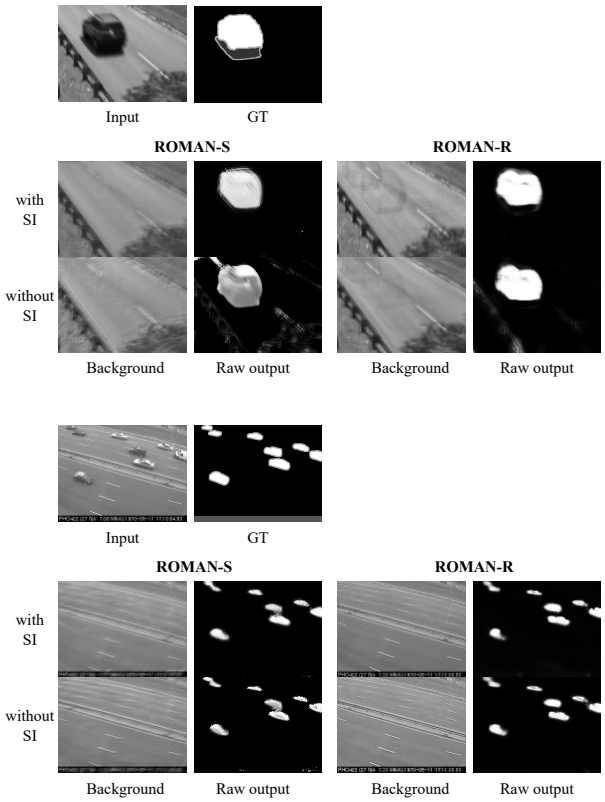


Fig. 7. Comparison of the outputs (raw mask outputs) with and without using side-information (SI) on the cameraJitter/traffic and lowFramerate/turnpike_0_5fps sequences.

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	+0.04	+0.02	+0.03	+0.05	+0.05	+0.05
lowFramerate	-0.01	-0.01	-0.01	+0.03	-0.02	+0.01
thermal	+0.03	+0.01	+0.02	+0.05	+0.02	+0.04
shadow	+0.00	+0.00	+0.00	+0.15	+0.09	+0.13
cameraJitter	+0.05	+0.07	+0.06	+0.09	+0.02	+0.06
dynamicBackground	+0.01	+0.04	+0.04	+0.09	+0.11	+0.11
turbulence	+0.03	+0.01	+0.02	+0.05	+0.06	+0.06
badWeather	-0.01	-0.01	-0.01	+0.08	+0.04	+0.06

where the time-correlation between foreground objects is less relevant and renders the side-information branches useless; such an example is given in Fig. 7 (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) Decision Threshold: Figure 8 depicts the receiver operating characteristic (ROC) for two example scenes. The masked-based models show to be more robust than the previous models that classify objects based on foreground pixel intensity. In these examples, ROMAN-R reaches the highest area under curve (AUC) score. In the same figure, we also show examples of raw mask outputs for ROMAN-R,

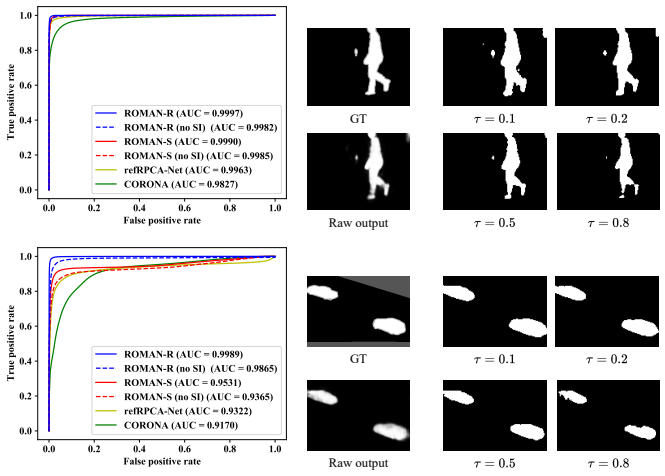


Fig. 8. ROC curve and AUC for the thermal/corridor and cameraJitter/boulevard sequences (no SI stands for no side-information), followed by a mask output for different decision thresholds τ using ROMAN-R

as well as various binarized masks obtained by changing the decision threshold τ. In the second model, the foreground membership probability is less dependent on the foreground pixel intensities and object textures compared to refRPCA-net and CORONA; hence, the effect of changing the decision threshold mainly affects the object boundaries. Also, a slight increase in AUC is observed with both models when using the side-information scheme, compared to their counterparts without side-information.

6) 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 $\boldsymbol{\mathcal{M}}_i$ 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. 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 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

TABLE IV
F_1 SCORES ON TEST CLIPS FOR ROMAN-R WITH VARYING NUMBER OF LAYERS (KERNEL DEPTH=8).

layers:	1	2	3	4
baseline/highway	0.942	0.973	0.971	0.963
lowFramerate/tramCrossroad	0.756	0.795	0.822	0.840
cameraJitter/traffic	0.879	0.894	0.907	0.923
shadow/bungalows	0.805	0.863	0.856	0.847
badWeather/blizzard	0.875	0.904	0.903	0.903

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

kernel depth:	1	8	32
baseline/highway	0.957	0.971	0.973
lowFramerate/tramCrossroad	0.785	0.822	0.837
cameraJitter/traffic	0.901	0.907	0.923
shadow/bungalows	0.805	0.863	0.856
badWeather/blizzard	0.724	0.903	0.880

best configuration on the training set. These parameters are kept constant for every iteration. Fig. 1 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.

D. 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. The main computational bottleneck of these models resides in the SVD computation; however, compared to the untrained RPCA optimization methods, 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. 1. For comparison, the U-Net baseline has substantially more trainable parameters (31×10^6), which is typical of deep models that perform feature extraction.

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 deep unfolding models require less layers than traditional optimization models, and our second model achieves competitive performance with semantic networks like U-Net, while requiring few parameters, small amount of data for training (a few labelled 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?," *Journal of the ACM*, vol. 58, no. 3, pp. 11:1–11:37, June 2011.

[2] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, Aug 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 *Advances in Neural Information Processing Systems*, Vancouver, Dec 2009, vol. 22, pp. 2080–2088.

[4] X. Liu, G. Zhao, J. Yao, and C. Qi, "Background subtraction based on low-rank and structured sparse decomposition," *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2502–2514, Apr 2015.

[5] S. Javed, T. Bouwmans, and S. K. Jung, "Improving or-pca via smoothed spatially-consistent low-rank modeling for background subtraction," in *ACM Symposium on Applied Computing*, Marrakech, Apr 2017, pp. 89–94.

[6] H. V. Luong, N. Deligiannis, J. Seiler, S. Forchhammer, and A. Kaup, "Compressive online robust principal component analysis via n-ℓ_1 minimization," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4314–4329, Sept 2018.

[7] E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, Dec 2008.

[8] Z. Kang, C. Peng, and Q. Cheng, "Robust pca via nonconvex rank approximation," in *IEEE International Conference on Data Mining*, Atlantic City, Nov 2015, IEEE, pp. 211–220.

[9] X. Yi, D. Park, Y. Chen, and C. Caramanis, "Fast algorithms for robust pca via gradient descent," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Barcelona, Dec 2016, pp. 4159–4167.

[10] P. Netrapalli, U. N. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain, "Non-convex robust pca," in *Advances in Neural Information Processing Systems*, Montréal, Dec 2014, vol. 27, 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," Tech. Rep., Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois, 2009.

[12] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv:1009.5055*, Sept 2010.

[13] X. Yuan and J. Yang, "Sparse and low-rank matrix decomposition via alternating direction methods," *Pacific Journal of Optimization*, vol. 9, no. 1, pp. 167, 01 Jan 2009.

[14] W. Cao, Y. Wang, J. Sun, D. Meng, C. Yang, A. Cichocki, and Z. Xu, "Total variation regularized tensor rpca for background subtraction from compressive measurements," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4075–4090, Sept 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," *arXiv:1909.08049*, Sept 2019.

[16] O. Tezcan, P. Ishwar, and J. Konrad, "BSUV-Net: A fully-convolutional neural network for background subtraction of unseen videos," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, Snowmass Village, Mar 2020, pp. 2774–2783.

[17] L. A. Lim and H. Y. Keles, "Learning multi-scale features for foreground segmentation," *Pattern Analysis and Applications*, vol. 23, no. 3, pp. 1369–1380, Aug 2020.

[18] J. Liao, G. Guo, Y. Yan, and H. Wang, "Multiscale cascaded scene-specific convolutional neural networks for background subtraction," in *Pacific Rim Conference on Multimedia*, Heifei, Sept 2018, Springer, pp. 524–533.

[19] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recognition Letters*, vol. 96, no. C, pp. 66–75, Sept 2017.

[20] D. Sakkos, H. Liu, J. Han, and L. Shao, "End-to-end video background subtraction with 3d convolutional neural networks," *Multimedia Tools and Applications*, vol. 77, no. 17, pp. 23023–23041, Sept 2018.

[21] 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 *25th IEEE International Conference on Image Processing*, Athens, Sept 2018, pp. 4018–4022.

[22] I. Osman, M. Abdelpakey, and M. S. Shehata, "TransBlast: Self-supervised learning using augmented subspace with transformer for background/foreground separation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Montréal, Oct 2021, pp. 215–224.

[23] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, "Deep neural network concepts for background subtraction:a systematic review and comparative evaluation," *Neural Networks*, vol. 117, pp. 8–66, Sept 2019.

[24] 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*, pp. 51–73. World Scientific, Mar 2020.

[25] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, Colombus, Sept 2014, pp. 387–394.

[26] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequievre, "A benchmark dataset for outdoor foreground/background extraction," in *Asian Conference on Computer Vision*, Daejeon, Nov 2012, pp. 291–300.

[27] L. Maddalena and A. Petrosino, "Towards benchmarking scene background initialization," in *International conference on image analysis and processing*, Genova, Sept 2015, pp. 469–476.

[28] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, June 2010, pp. 399–406.

[29] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," *arXiv:2012.08405*, Dec 2020.

[30] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, Mar 2021.

[31] H. Sreter and R. Giryes, "Learned convolutional sparse coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Calgary, Apr 2018, pp. 2191–2195.

[32] J. Sun, H. Li, and Z. Xu, "Deep ADMM-Net for compressive sensing MRI," *Advances in neural information processing systems*, vol. 29, pp. 10–18, Dec 2016.

[33] I. Marivani, E. Tsiligianni, B. Cornelis, and N. Deligiannis, "Multimodal deep unfolding for guided image super-resolution," *IEEE Transactions on Image Processing*, vol. 29, pp. 8443–8456, Aug 2020.

[34] S. Wisdom, T. Powers, J. Pitton, and L. Atlas, "Interpretable recurrent neural networks using sequential sparse recovery," *arXiv:1611.07252*, Nov 2016.

[35] H. V. Luong, B. Joukovsky, and N. Deligiannis, "Designing interpretable recurrent neural networks for video reconstruction via deep unfolding," *IEEE Transactions on Image Processing*, vol. 30, pp. 4099–4113, Apr 2021.

[36] O. Solomon, R. Cohen, Y. Zhang, Y. Yang, Q. He, J. Luo, R. van Sloun, and Y. C. Eldar, "Deep unfolded robust pca with application to clutter suppression in ultrasound," *IEEE Transactions on Medical Imaging*, vol. 39, no. 4, pp. 1051–1063, Sep 2019.

[37] V. H. Luong, B. Joukovsky, Y. C. Eldar, and N. Deligiannis, "A deep-unfolded reference-based rpca network for video foreground-background separation," in *2020 28th European Signal Processing Conference (EUSIPCO)*, Amsterdam, Jan 2021, pp. 1432–1436.

[38] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, Dec 2005.

[39] J. F. C. Mota, N. Deligiannis, and M. R. D. Rodrigues, "Compressed sensing with prior information: Strategies, geometry, and bounds," *IEEE Transactions on Information Theory*, vol. 63, no. 7, pp. 4472–4496, July 2017.

[40] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing*, Shanghai, Mar 2016, pp. 4687–4691.

[41] H. D. Le, H. V. Luong, and N. Deligiannis, "Designing recurrent neural networks by unfolding an l1-l1 minimization algorithm," in *IEEE International Conference on Image Processing*, Taipei, Sept 2019, pp. 2329–2333.

[42] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Munich, Oct 2015, pp. 234–241.

[43] Venkat C., Sujay S., P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, no. 2, pp. 572–596, 2011.

[44] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010.

[45] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2(1), pp. 183–202, 2009.

[46] J. Liu, X. Chen, Z. Wang, and W. Yin, ," in *International Conference on Learning Representations*, New Orleans, May 2019.

[47] A. Khalilian-Gourtani, S. Minaee, and Y. Wang, "Masked-RPCA: Moving object detection with an overlaying model," *IEEE Open Journal of Signal Processing*, vol. 1, pp. 274–286, Nov 2020.

[48] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv:1611.01144*, Nov 2016.

[49] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using LSTMs," in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, July 2015, vol. 37, pp. 843–852.