ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10094712

DESIGNING TRANSFORMER NETWORKS FOR SPARSE RECOVERY OF SEQUENTIAL DATA USING DEEP UNFOLDING

Brent De Weerdt[1,2], *Yonina C. Eldar*[3], *Nikos Deligiannis*[1,2]

[1]ETRO Department, Vrije Universiteit Brussel (VUB), Pleinlaan 2, B-1050 Brussels, Belgium
[2]imec, Kapeldreef 75, B-3001 Leuven, Belgium
[3]Department of Math and Computer Science, Weizmann Institute of Science, Rehovot, Israel

ABSTRACT

Deep unfolding models are designed by unrolling an optimization algorithm into a deep learning network. These models have shown faster convergence and higher performance compared to the original optimization algorithms. Additionally, by incorporating domain knowledge from the optimization algorithm, they need much less training data to learn efficient representations. Current deep unfolding networks for sequential sparse recovery consist of recurrent neural networks (RNNs), which leverage the similarity between consecutive signals. We redesign the optimization problem to use correlations across the whole sequence, which unfolds into a Transformer architecture. Our model is used for the task of video frame reconstruction from low-dimensional measurements and is shown to outperform state-of-the-art deep unfolding RNN and Transformer models, as well as a traditional Vision Transformer on several video datasets.

Index Terms— deep unfolding, Transformer networks, sparse recovery, compressed sensing.

1. INTRODUCTION

The recovery of signals from low-dimensional noisy measurements is used in various imaging applications, e.g., dynamic magnetic resonance imaging [1], compressive video sensing [2] and high-speed hyperspectral video acquisition [3]. When recovering a signal from low-dimensional or corrupted data, prior knowledge of signal properties can be leveraged to make reconstruction possible. In the case of sequential signals, one can use both (i) the low complexity representations of individual signals, for instance, sparsity with respect to some dictionary, and (ii) the correlation between signals.

Several algorithms exist for sequential sparse recovery, e.g., [4, 5, 6, 7]. These methods use iterative optimization algorithms, resulting in computationally expensive reconstruction, especially when the problem dimensionality increases. On the other hand, deep neural networks (DNNs) move computational time to the training phase, yielding fast inference. They have achieved state-of-the-art performance on various inverse imaging problems [8], but in general, they do not make explicit use of prior domain knowledge. Hence these black-box models often lack interpretability and theoretical guarantees [8]. Furthermore, their lack of prior knowledge requires them to train on huge amounts of data to learn its specific properties or obtain efficient representations.

Deep unfolding models try to combine the fast inference and learning from data of DNNs with the domain knowledge embedded in optimization algorithms. By unrolling an optimization algorithm into a neural network that can be trained, one obtains a deep learning model with prior domain knowledge encapsulated in the structure and constraints of the model [9]. Examples include Learned ISTA (LISTA) [10], ADMM-Net [11], and DUBLIB [12]. For sequential sparse recovery, current state-of-the-art deep unfolding models consist of RNN architectures, such as SISTA-RNN [13] and reweighted-RNN [14], which take advantage of the similarity in ℓ_2 or ℓ_1 norm of consecutive signals.

Attention layers exploit similarities between signals that are farther apart and are used in the Transformer network [15], which processes a full sequence at once and outperforms state-of-the-art RNNs in language modeling. Recently, the work reported in [16] presented an energy function and corresponding optimization algorithm that unfolds into a (simplified) Transformer architecture. The authors used two text sentiment classification datasets to validate that their convergence conditions hold on real-world data, but they did not evaluate their model on any practical application.

Starting from [16], we create a new energy function tailored to the problem of sparse recovery of video frames by incorporating priors on (i) video frame sparsity in a dictionary and (ii) correlation between the frames of a video. Building upon the framework in [16], we derive a new optimization algorithm to minimize our energy function and unfold it to obtain a Transformer network. As a result, our deep unfolding Transformer model has a modified self-attention mechanism, different linear projections, and a different activation function compared to the model in [16]. Using video reconstruction experiments on different real-world datasets, we

This work was supported by the FWO Flanders Ph.D. Fellowship Strategic Basic Research under Grant 1S44523N and imec.icon Surv-AI-llance.

show that our architecture outperforms state-of-the-art deep unfolding RNNs [13, 17, 14], as well as a traditional Vision Transformer [18] and the deep unfolding Transformer in [16].

The rest of the paper is organized as follows. Section 2 reviews the background and prior work. Section 3 describes the design of the deep unfolding Transformer for sparse recovery of sequential signals. Experiments and results are given in Section 4, and conclusions are drawn in Section 5.

2. BACKGROUND AND RELATED WORK

2.1. Deep unfolding RNNs for sequential sparse recovery

Let us reconstruct a sequence of signals $\mathbf{s}_t \in \mathbb{R}^n$, with $t = 1, \ldots, T$, from low-dimensional noisy measurements $\mathbf{x}_t = \mathbf{A}\mathbf{s}_t + \boldsymbol{\epsilon_t}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$ $(m \ll n)$ is the measurement matrix. We assume that the signal $\mathbf{s}_t$ has a sparse representation $\mathbf{h}_t$ in an overcomplete dictionary $\mathbf{D} \in \mathbb{R}^{n \times d}$, a.k.a., $\mathbf{s}_t = \mathbf{D}\mathbf{h}_t$. Such a series of signals can also be correlated in time, in which case we can improve reconstruction by adding a constraint to reflect this. The signal sequence can then be recovered by solving the following optimization problem:

$$\min_{\mathbf{h}_t} \frac{1}{2}\|\mathbf{x}_t - \mathbf{A}\mathbf{D}\mathbf{h}_t\|_2^2 + \lambda_1\|\mathbf{h}_t\|_1 + \lambda_2 C\left(\mathbf{h}_t, \mathbf{h}_{t-1}\right), \forall t \quad (1)$$

where $\|\cdot\|_p$ is the ℓ_p norm, λ_1 and λ_2 are regularization parameters and $C(\cdot)$ models the correlation in time. We can find the reconstructed signals from the obtained vectors $\mathbf{h}_t^\star$ by $\mathbf{s}_t^\star = \mathbf{D}\mathbf{h}_t^\star$. Depending on the choice of $C(\cdot)$, (1) results in different iterative optimization algorithms, which have been unrolled into deep unfolding RNN models. Specifically, when $C(\cdot)$ is $\|\mathbf{D}\mathbf{h}_t - \mathbf{F}\mathbf{D}\mathbf{h}_{t-1}\|_2^2$, (1) can be solved with the sequential iterative soft thresholding algorithm (SISTA) or its unfolded version SISTA-RNN [13]. Alternatively, when $C(\cdot)$ is $\|\mathbf{h}_t - \mathbf{G}\mathbf{h}_{t-1}\|_1$, we obtain an ℓ_1-ℓ_1 optimization problem [7], which unfolds into the ℓ_1-ℓ_1-RNN [17]. Here $\mathbf{F}$ and $\mathbf{G}$ are affine transforms promoting temporal correlation. By adding additional weighting factors to the ℓ_1-ℓ_1 problem, an improved reweighted-RNN model is obtained [14].

2.2. Transformers and deep unfolding

Transformer models have in recent years achieved state-of-the-art results in language modeling [15] and computer vision tasks [19]. Their key idea is splitting the input data into independent tokens and processing them in parallel to model long-range relationships. This is in contrast to RNNs, which process data sequentially. In Transformers, the input tokens can be word embeddings for language modeling or image patches for image or video processing. One block of a Transformer consists of a self-attention module, which lets tokens extract context from each other, followed by a one- or two-layer fully connected network (FCN) with an activation function (e.g., ReLU), which processes each token in parallel. Whereas a block is also interleaved with normalization layers, the core operation of the k-th block can be described by:

$$\begin{aligned} \mathbf{Z}^{(k+1)} &= \mathbf{Y}^{(k)} \operatorname{softmax}\left(\mathbf{Y}^{(k)T}\mathbf{W}_K^{(k)T}\mathbf{W}_Q^{(k)}\mathbf{Y}^{(k)}\right), \\ \mathbf{y}_n^{(k+1)} &= FCN\left(\mathbf{z}_n^{(k+1)}\right) \text{ for } n = 1, \ldots, N, \end{aligned} \quad (2)$$

where the tokens $\mathbf{y}_1^{(k)}, \ldots, \mathbf{y}_N^{(k)}$ are concatenated into the matrix $\mathbf{Y}^{(k)}$, transformed into $\mathbf{Z}^{(k)}$ by the self-attention module with learnable matrices $\mathbf{W}_K^{(k)}$ and $\mathbf{W}_Q^{(k)}$, and then the columns $\mathbf{z}_n^{(k)}$ are further processed into the output tokens of the block $\mathbf{y}_1^{(k+1)}, \ldots, \mathbf{y}_N^{(k+1)}$. The softmax function transforms the values $u_1, \ldots, u_N$ of each row in a matrix to a probability distribution of N possible outcomes with probability $\frac{\exp(u_n)}{\sum_{j=1}^N \exp(u_j)}$. Many of these blocks are then stacked on top of each other to form a full Transformer model.

The authors of [16] designed the first deep unfolding based Transformer by starting from the following optimization problem:

$$\begin{aligned} \min_{\mathbf{Y}} &\left(\sum_{i,j} -\exp\left(-\frac{1}{2}\|\mathbf{W}_a\mathbf{y}_i - \mathbf{W}_a\mathbf{y}_j\|_2^2\right) + \frac{1}{2}\|\mathbf{W}_a\mathbf{Y}\|_{\mathcal{F}}^2\right) \\ &+ \left(\frac{1}{2}\operatorname{Tr}\left(\mathbf{Y}^T\mathbf{W}_b\mathbf{Y}\right) + \frac{1}{2}\|\mathbf{Y}\|_{\mathcal{F}}^2 + \varphi(\mathbf{Y})\right), \end{aligned} \quad (3)$$

where $\mathbf{Y}$ is a matrix with the vectors $\mathbf{y}_1, \ldots, \mathbf{y}_N$ as columns, $\|\cdot\|_{\mathcal{F}}$ is the Frobenius norm, $\operatorname{Tr}(\cdot)$ is the trace of a matrix, $\varphi(u)$ is the indicator function, whose value is $+\infty$ for $u < 0$ and 0 otherwise, and $\mathbf{W}_a$ and $\mathbf{W}_b$ are arbitrary weight matrices. They proved that this total energy can be minimized by iterations of an algorithm alternating between a step that decreases the value of the first part of (3) and a proximal gradient descent step on the second part of (3). When unfolded, this first part will result in a weighted softmax self-attention layer, while the second part unfolds into a linear projection with ReLU activation as follows:

$$\mathbf{Z}^{(k+1)} = \mathbf{Y}^{(k)} \operatorname{softmax}_\beta\left(\mathbf{Y}^{(k)T}\mathbf{W}_a^{(k)T}\mathbf{W}_a^{(k)}\mathbf{Y}^{(k)}\right), \quad (4)$$

$$\mathbf{y}_n^{(k+1)} = \operatorname{ReLU}\left(\mathbf{W}_b^{(k)}\mathbf{z}_n^{(k+1)}\right) \text{ for } n = 1, \ldots, N, \quad (5)$$

where $\mathbf{W}_a^{(k)}$ and $\mathbf{W}_b^{(k)}$ are learnable matrices, $\operatorname{ReLU}(u) = \max(0, u)$ and $\operatorname{softmax}_\beta$ is a weighted softmax function with probabilities $\frac{\beta_n \exp(u_n)}{\sum_{j=1}^N \beta_j \exp(u_j)}$, where $\beta_i = \exp\left(-\frac{1}{2}\|\mathbf{y}_i\|_2^2\right)$.

3. DEEP UNFOLDING TRANSFORMER FOR SPARSE RECOVERY OF SEQUENTIAL SIGNALS

Following the notation in Section 2.1, we consider the problem of reconstructing a sequence of video frames $\mathbf{s}_t$, $t = 1, \ldots, T$ from low-dimensional noisy measurements $\mathbf{x}_t = \mathbf{A}\mathbf{s}_t + \boldsymbol{\epsilon_t}$ (where $\mathbf{A}$ is the sensing matrix) and assume that each frame $\mathbf{s}_t$ has a sparse representation $\mathbf{h}_t$ in the overcomplete dictionary $\mathbf{D} \in \mathbb{R}^{n \times d}$.

3.1. Optimization problem

In the deep unfolding RNNs discussed in Section 2.1, only the correlation between pairs of consecutive signals is considered. When modeling a video with a static background, for example, measurements from *all* frames are useful to recover the background and could improve reconstruction quality compared to looking at only two frames at a time. The same can be argued for videos with recurring events or temporary occlusions, where similar signals in the video are not necessarily adjacent.

In order to find similar signals in the sequence and reconstruct them simultaneously, we aim to extend the term $\|\mathbf{D}\mathbf{h}_t - \mathbf{F}\mathbf{D}\mathbf{h}_{t-1}\|_2^2$ in SISTA [13] to model correlations over the whole sequence, in some form of $\sum_{\tau \neq t} \|\mathbf{F}\mathbf{D}\mathbf{h}_t - \mathbf{F}\mathbf{D}\mathbf{h}_\tau\|_2^2$. Initial experiments showed that using $\mathbf{F}$ does not improve performance and therefore we set it to the identity matrix. In order to diminish the influence of uncorrelated signals, we introduce an exponential function: $\sum_{\tau \neq t} -\exp\left(-\|\mathbf{D}\mathbf{h}_t - \mathbf{D}\mathbf{h}_\tau\|_2^2\right)$. When performing gradient descent on this expression, the gradient will be near zero for dissimilar signals while we improve the estimation of correlated signals.

Furthermore, we add the term $\|\mathbf{x}_t - \mathbf{A}\mathbf{D}\mathbf{h}_t\|_2^2$ to encode the error between the measurements and their reconstruction from $\mathbf{h}_t$, and a sparsity prior for each of the representations $\mathbf{h}_t$. Finally, we also add a constraint on the ℓ_2 norm of each signal, i.e., $\sum_t \|\mathbf{D}\mathbf{h}_t\|_2^2 = \|\mathbf{D}\mathbf{H}\|_{\mathcal{F}}^2$, where $\mathbf{H}$ is a matrix with $\mathbf{h}_1, \ldots, \mathbf{h}_T$ in its columns, to obtain the following optimization problem:

$$\min_{\mathbf{H}} \lambda_2 \left(\sum_{t,\tau} -\exp\left(-\frac{1}{2}\|\mathbf{D}\mathbf{h}_t - \mathbf{D}\mathbf{h}_\tau\|_2^2\right) + \frac{1}{2}\|\mathbf{D}\mathbf{H}\|_{\mathcal{F}}^2 \right) + \left(\sum_t \frac{1}{2}\|\mathbf{x}_t - \mathbf{A}\mathbf{D}\mathbf{h}_t\|_2^2 + \lambda_1 \sum_t \|\mathbf{h}_t\|_1 \right), \quad (6)$$

where λ_1 and λ_2 are regularization parameters. We observe a similarity between our optimization problem and (3), especially for the first part, which is nearly identical. The difference lies in the second half, where the data fidelity term $\sum_t \frac{1}{2}\|\mathbf{x}_t - \mathbf{A}\mathbf{D}\mathbf{h}_t\|_2^2$ replaces $\frac{1}{2}\operatorname{Tr}(\mathbf{Y}^T\mathbf{W}_B\mathbf{Y})$ and the ℓ_2 norm prior and indicator function $\varphi(\cdot)$ are replaced by our ℓ_1 norm prior. These differences make our optimization problem tailored to the task of sparse recovery of sequential signals.

We use the theoretical framework of [16] to derive an optimization algorithm for solving (6), analogous to the algorithm they derived for (3). The first part in (6) is minimized by steps of softmax self-attention, analogous to (3). The second part in (6) can be minimized for each $\mathbf{h}_t$ separately, resulting in T parallel LASSO problems:

$$\min_{\mathbf{h}_t} \frac{1}{2}\|\mathbf{x}_t - \mathbf{A}\mathbf{D}\mathbf{h}_t\|_2^2 + \lambda_1\|\mathbf{h}_t\|_1, \quad (7)$$

which can be solved with ISTA [20]:

$$\mathbf{h}_t^{(k+1)} = \phi_{\frac{\lambda_1}{c}} \left(\left(\mathbf{I} - \frac{1}{c}\mathbf{D}^T\mathbf{A}^T\mathbf{A}\mathbf{D} \right) \mathbf{h}_t^{(k)} + \frac{1}{c}\mathbf{D}^T\mathbf{A}^T\mathbf{x}_t \right), \quad (8)$$

with $\phi_\gamma(u) = \operatorname{sign}(u)\max(0, |u| - \gamma)$ the soft thresholding function and c an upper bound on the Lipschitz constant of the gradient of $\frac{1}{2}\|\mathbf{x}_t - \mathbf{A}\mathbf{D}\mathbf{h}_t\|_2^2$. The optimization algorithm for (6) then consists of alternating steps of softmax self-attention and parallel proximal gradient descents on (7), namely, one iteration of ISTA. The inclusion of the ISTA step (8) instead of a traditional ReLU activated linear projection (see (5)) is the main difference between our method and the one in [16].

Algorithm 1 Attention-based sequential iterative soft thresholding algorithm

```
Require: measurement matrix $\mathbf{A}$, dictionary $\mathbf{D}$, measurements $\mathbf{x}_t = \mathbf{A}\mathbf{s}_t$, with $(t = 1, \ldots, T)$
1: $\mathbf{h}_1^{(0)}, \ldots, \mathbf{h}_T^{(0)} \leftarrow \mathbf{0}$
2: for $k = 1$ to $K$ do
3:     for $t = 1$ to $T$ do
4:         $\mathbf{y}_t^{(k)} \leftarrow \lambda_2 \frac{\sum_u \beta_u \exp\left(\mathbf{h}_t^{(k-1)T}\mathbf{D}^T\mathbf{D}\mathbf{h}_u^{(k-1)}\right)\mathbf{h}_u^{(k-1)}}{\sum_u \beta_u \exp\left(\mathbf{h}_t^{(k-1)T}\mathbf{D}^T\mathbf{D}\mathbf{h}_u^{(k-1)}\right)}$
5:         $\mathbf{z}_t^{(k)} \leftarrow \left(\mathbf{I} - \frac{1}{c}\mathbf{D}^T\mathbf{A}^T\mathbf{A}\mathbf{D}\right)\mathbf{y}_t^{(k)} + \frac{1}{c}\mathbf{D}^T\mathbf{A}^T\mathbf{x}_t$
6:         $\mathbf{h}_t^{(k)} \leftarrow \phi_{\lambda_1/c}\left(\mathbf{z}_t^{(k)}\right)$
7:     end for
8: end for
9: return $\mathbf{s}_1^\star = \mathbf{D}\mathbf{h}_1^{(K)}, \ldots, \mathbf{s}_T^\star = \mathbf{D}\mathbf{h}_T^{(K)}$
```

The procedure of signal recovery is given in Algorithm 1. Since initially all $\mathbf{h}_t, t = 1, \ldots, T$, are identical (we initialize them to zero), the first iteration of the algorithm simplifies to an iteration of ISTA. After $K - 1$ more iterations of alternating self-attention and single-iteration ISTA steps, the sparse representations $\mathbf{h}_t^{(K)}$ are multiplied with the dictionary $\mathbf{D}$ to obtain the final reconstructed signals $\mathbf{s}_t^\star = \mathbf{D}\mathbf{h}_t^{(K)}$.

3.2. Proposed DUST model

By unrolling the steps of Algorithm 1 we obtain the proposed Deep Unfolding Sparse Transformer model (DUST). The model (see Fig. 1) takes as input a set of initial tokens $\mathbf{h}_t$ and the measurements $\mathbf{x}_t$, it applies self-attention (line 4 in the algorithm), followed by one layer of LISTA [10] (which we refer to as 1ℓ-LISTA) applied to each token separately (see lines 5 and 6). The main processing block of DUST then has the following form:

$$\mathbf{Z}^{(k+1)} = \lambda_2\mathbf{H}^{(k)}\operatorname{softmax}\left(\mathbf{H}^{(k)T}\mathbf{D}^T\mathbf{D}\mathbf{H}^{(k)}\right), \quad (9)$$

$$\mathbf{h}_t^{(k+1)} = \phi_{\lambda_1/c}\left(\mathbf{U}\mathbf{z}_t^{(k+1)} + \mathbf{V}\mathbf{x}_t\right) \text{ for } t = 1, \ldots, T. \quad (10)$$

where $\mathbf{U} = \mathbf{I} - \frac{1}{c}\mathbf{D}^T\mathbf{A}^T\mathbf{A}\mathbf{D}$ and $\mathbf{V} = \frac{1}{c}\mathbf{D}^T\mathbf{A}^T$. The tokens are put through such a block of self-attention (9) and

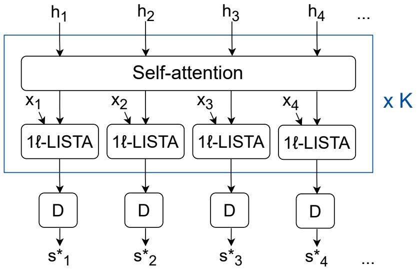

Fig. 1. Proposed deep unfolding Transformer (DUST) architecture for sequential sparse recovery.

Table 1. Average video reconstruction quality (PSNR) on three real-world datasets.

	Avenue	UCSD	ST
SISTA-RNN	35.73	34.13	34.90
ℓ_1-ℓ_1-RNN	36.51	34.34	35.56
Reweighted-RNN	36.94	35.22	**36.03**
ViT [18]	36.04	34.79	35.91
Unfolded Transformer [16]	34.36	32.94	34.25
DUST (proposed)	**37.61**	**35.98**	35.94

1ℓ-LISTA (10) K times, followed by a final projection on $\mathbf{D}$ to obtain the reconstructed signals.

The learnable parameters of DUST are the weights $\mathbf{U}$ and $\mathbf{V}$, the parameters λ_1, λ_2 and c, which are shared across all blocks, and the dictionary $\mathbf{D}$ (initialized with the DCT), which is also the same matrix for each block and the output projection. The model is trained by minimizing the empirical loss $\frac{1}{JT}\sum_{j=1}^{J}\sum_{t=1}^{T}\|\mathbf{s}_{j,t}-\mathbf{s}^{\star}_{j,t}\|_2^2$ between the original and reconstructed time-series signals, where J is the number of training data. The sensing matrix $\mathbf{A}$ (randomly initialized) is also learned during training.

We note that the difference between the Transformer in [16] and DUST is that our model promotes sparsity in the processed signals through the ISTA steps and takes the original measurements $\mathbf{x}_t$ as input in each block (see (5) vs. (10)).

4. EXPERIMENTS AND RESULTS

To assess the performance of DUST, we use the same setup as in [14] to reconstruct video frames from low-dimensional measurements. Three datasets are considered, namely CUHK Avenue, UCSD Anomaly Detection and ShanghaiTech Campus. Videos are downscaled and converted to grayscale, then non-overlapping clips of 20 frames are extracted from the videos and further split into clips of patches of 16×16 pixels with 50% overlap. We use the original testing split from each dataset, while the training sets are split into approximately 80% for training and 20% for validation. To keep the amount of data manageable, videos are cut off after 200 frames.

Table 2. Average video reconstruction quality (PSNR) on the Avenue dataset for different compression rates.

	50%	40%	30%	10%
SISTA-RNN	41.89	39.92	37.99	32.01
ℓ_1-ℓ_1-RNN	42.86	40.90	38.89	32.98
Reweighted-RNN	43.23	41.16	39.12	33.88
ViT [18]	39.53	38.28	37.12	33.85
Unfold. Transf. [16]	39.66	37.93	36.07	32.11
DUST (proposed)	**43.32**	**41.47**	**39.67**	**34.71**

Our DUST model is compared to SISTA-RNN [13], ℓ_1-ℓ_1-RNN [17] and reweighted-RNN [14], as well as a vanilla Vision Transformer (ViT) model [18] and our implementation of [16]. Each model has $K = 3$ layers, a learnable $m \times 256$ sensing matrix $\mathbf{A}$, and where applicable a 256×1024 dictionary $\mathbf{D}$, initialized with the discrete cosine transform. The size $m < 256$ of $\mathbf{A}$ corresponds to the chosen compression rate. Other parameters are initialized according to the respective papers. For DUST we initialize $c = 1$, $\lambda_1 = 0.1$ and $\lambda_2 = 0.4$. All models are trained for 100 epochs on Avenue and UCSD, but 40 epochs on ShanghaiTech Campus given its large size. Regarding the size of the models, DUST has 1.38M parameters, SISTA-RNN 341K, ℓ_1-ℓ_1-RNN 1.32M, reweighted-RNN 2.37M, the ViT 2.46M and the Transformer in [16] 485K.

We report the average video reconstruction quality for a compression rate of 20% on Avenue, UCSD, and ShanghaiTech in Table 1. DUST yields a 0.7 dB improvement over the best model on Avenue and UCSD, while being only slightly behind on ShanghaiTech. The Vision Transformer performs well ShanghaiTech, but less so on Avenue and UCSD; probably due to the smaller size of the datasets, for which it is difficult to train a traditional Transformer. The deep unfolding Transformer in [16] is significantly worse than the ViT since it is a simpler architecture with less modeling capacity, and does not incorporate the priors embedded in the RNN models and the proposed DUST. We also evaluated our model for different compression ratios on the Avenue dataset, from 50% to 10%, as shown in Table 2, where DUST improves over reweighted-RNN in all settings.

5. CONCLUSION

In this paper, we design an optimization problem for sparse recovery of sequential signals, including correlations between all signals instead of pairs of consecutive time steps. The resulting optimization algorithm can be unfolded into a Transformer architecture that outperforms several state-of-the-art deep unfolding RNN and Transformer models, as well as a traditional Vision Transformer on the task of video reconstruction from compressed measurements.

6. REFERENCES

[1] L. Weizman, Y. C. Eldar, and D. Ben Bashat, "Compressed sensing for longitudinal MRI: An adaptive-weighted approach," *Medical Physics*, vol. 42, no. 9, pp. 5195–5208, 2015.

[2] R. G. Baraniuk, T. Goldstein, A. C. Sankaranarayanan, C. Studer, A. Veeraraghavan, and M. B. Wakin, "Compressive Video Sensing: Algorithms, architectures, and applications," *IEEE Signal Processing Magazine*, vol. 34, no. 1, pp. 52–66, 2017.

[3] L. Wang, Z. Xiong, H. Huang, G. Shi, F. Wu, and W. Zeng, "High-Speed Hyperspectral Video Acquisition By Combining Nyquist and Compressive Sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 4, pp. 857–870, 2019.

[4] D. M. Malioutov, S. R. Sanghavi, and A. S. Willsky, "Sequential Compressed Sensing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 435–444, 2010.

[5] A. Charles, M. S. Asif, J. Romberg, and C. Rozell, "Sparsity penalties in dynamical system estimation," in *45th Annual Conference on Information Sciences and Systems*, 2011, pp. 1–6.

[6] J. Zhan and N. Vaswani, "Time Invariant Error Bounds for Modified-CS-Based Sparse Signal Sequence Recovery," *IEEE Transactions on Information Theory*, vol. 61, no. 3, pp. 1389–1409, 2015.

[7] J. F. C. Mota, N. Deligiannis, A. C. Sankaranarayanan, V. Cevher, and M. R. D. Rodrigues, "Adaptive-Rate Reconstruction of Time-Varying Signals With Application in Compressive Foreground Extraction," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3651–3666, 2016.

[8] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos, "Using Deep Neural Networks for Inverse Problems in Imaging: Beyond Analytical Methods," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 20–36, 2018.

[9] 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, 2021.

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

[11] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-Net for compressive sensing MRI," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 10–18.

[12] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. C. Eldar, "Efficient and Interpretable Deep Blind Image Deblurring Via Algorithm Unrolling," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 666–681, 2020.

[13] S. Wisdom, T. Powers, J. Pitton, and L. Atlas, "Building recurrent networks by unfolding iterative thresholding for sequential sparse recovery," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4346–4350.

[14] 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, 2021.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, vol. 30.

[16] Y. Yang, Z. Huang, and D. Wipf, "Transformers from an Optimization Perspective," arXiv:2205.13891, May 2022.

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

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2022.

[19] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in Vision: A Survey," *ACM Computing Surveys*, vol. 54, no. 10s, 2022.

[20] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.