Deep Equilibrium Models for Snapshot Compressive Imaging

Yaping Zhao¹,², Siming Zheng³,⁴, Xin Yuan¹,*

¹ Westlake University, Hangzhou, China
² The University of Hong Kong, Pokfulam, Hong Kong SAR, China
³ Computer Network Information Center, Chinese Academy of Science, Beijing, China
⁴ University of Chinese Academy of Sciences, Beijing, China
zhaoyp@connect.hku.hk, zhengsiming@cnic.cn, xyuan@westlake.edu.cn

Abstract

The ability of snapshot compressive imaging (SCI) systems to efficiently capture high-dimensional (HD) data has led to an inverse problem, which consists of recovering the HD signal from the compressed and noisy measurement. While reconstruction algorithms grow fast to solve it with the recent advances of deep learning, the fundamental issue of accurate and stable recovery remains. To this end, we propose deep equilibrium models (DEQ) for video SCI, fusing data-driven regularization and stable convergence in a theoretically sound manner. Each equilibrium model implicitly learns a nonexpansive operator and analytically computes the fixed point, thus enabling unlimited iterative steps and infinite network depth with only a constant memory requirement in training and testing. Specifically, we demonstrate how DEQ can be applied to two existing models for video SCI reconstruction: recurrent neural networks (RNN) and Plug-and-Play (PnP) algorithms. On a variety of datasets and real data, both quantitative and qualitative evaluations of our results demonstrate the effectiveness and stability of our proposed method. The code and models are available at: https://github.com/IndigoPurple/DEQSCI

Introduction

Aiming at the efficient and effective acquisition of high-dimensional (HD) visual signal, snapshot compressive imaging (SCI) systems have benefited from the advent of novel optical designs to sample the HD data as two-dimensional (2D) measurements. Considering the video SCI system, the 2D measurement of a video, i.e., a three-dimensional (3D) data-cube leads to an inverse problem. The goal of such an inverse problem is to recover a video from a collection of noisy snapshots, which could be modeled as (Yuan, Brady, and Katsaggelos 2021):

\[ y = \Phi x + e, \]  

where \( y \in \mathbb{R}^n \) is the 2D measurement with \( n \) equaling the number of each video frame’s pixels, \( \Phi \in \mathbb{R}^{n \times nB} \) is the sensing matrix, \( x \in \mathbb{R}^{nB} \) is the 3D data (by vectorizing each frame and stacking them), and \( e \) is the measurement noise; here \( B \) denotes that every \( B \) video frames are collapsed into a single 2D measurement. Though algorithms have been fully developed to reconstruct the video from its snapshot measurement in recent years, the fundamental issue remains: this inverse problem is inherently ill-posed, which makes the recovery of the signal \( x \) inaccurate and unstable for noise-affected data \( y \) (Jalali and Yuan 2019).

The rapid advancement of deep learning and artificial in-
telligence have empowered a new wave of revolutionary solutions towards these previously intractable problems. For instance, BIRNAT (Cheng et al. 2020) employed recurrent neural networks (RNNs) to reconstruct the video frames in a sequential manner and explore the temporal correlation within the video SCI signal. Inspired by particular optimization algorithms, GAP-net (Meng, Jalali, and Yuan 2020), DUN-3DUnet (Wu, Zhang, and Mou 2021) designed deep unfolding structures, which consist of a fixed number of architecturally identical blocks. The heart of RNN and deep unfolding are deep neural networks, which have posed new challenges due to their ever-growing depth and huge training memory occupation. To overcome these difficulties, inspired by (Gomez et al. 2017), a recent work (RevSCI) (Cheng et al. 2021) utilized reversible convolutional neural networks to develop a memory-efficient structure. However, all of these aforementioned algorithms inevitably suffer growing memory occupation with increasing layer depth, and thus models need to be painstakingly designed.

Inspired by the plug-and-play (PnP) framework (Venkatakrishnan, Bouman, and Wohlberg 2013; Sreehari et al. 2016) which has been proposed for inverse problems with provable convergence (Chan, Wang, and Elgendy 2017; Ryu et al. 2019), PnP-FFDNet (Yuan et al. 2020) and PnP-FastDVDNet (Yuan et al. 2021) bridged the gap between deep learning and conventional optimization algorithms with the plug-and-play (PnP) framework, utilizing a pre-trained denoiser as the proximal operator. While enjoying the advantages of both data-driven regularization and flexible iterative optimization steps, those algorithms still have hyperparameters to be tuned. Nevertheless, an accurate result must be guaranteed with a proper parameter setting. Due to the intrinsic unstable characteristic of the iterative recovery, even some complicated strategy needs to be employed (Wei et al. 2020). As we illustrate in Fig. 1 and Fig. 2, the hyperparameters are unavoidable to be handcrafted to achieve satisfactory performance in traditional algorithms.

An important and interesting research topic in deep learning is to train arbitrary deep networks, in which the deep equilibrium models (DEQ) (Bai, Koltn, and Kolter 2020) stands up as the leading method. A recent work (Gilton, Ongie, and Willett 2021) leverages DEQ to solve the inverse problems in imaging, which corresponds to the potentially infinite number of iteration steps in the PnP scheme.

To accommodate the state-of-the-art SCI architectures and to enable low-memory stable reconstruction, this paper sets about utilizing DEQ for solving the inverse problem of video SCI. Specifically, we applied DEQ to two existing models for video SCI reconstruction: RNN and PnP. Therefore, the former one is equivalent to an infinite-depth network using only constant memory; the latter one is tuning-free, and directly solves for the fixed point during the iterative optimization process. On a variety of simulation and real datasets, quantitative and qualitative evaluations demonstrate the effectiveness of our proposed method. As shown in Fig. 2, our reconstruction converges to stable results along with the increasing iterations during optimization.

In a nutshell, we aim to address the following two challenges which the SCI reconstruction are facing while using deep neural network and iterative optimization algorithms:

- How deep should the model be? Can it be infinite?
- Is there a tuning-free framework to be used? If yes, how to use it for SCI reconstruction?

By employing the most recent development of DEQ, we demonstrate that the answers to all the above questions are positive. Our specific contributions are as follows:

1. We firstly propose deep equilibrium models for video SCI, which fuses data-driven regularization and stable convergence in a theoretically sound manner.
2. Each equilibrium model analytically computes the fixed point, thus enabling unlimited iterative steps and infinite network depth with only a constant memory requirement in training and testing.
3. We analyze convergence for each equilibrium model, to ensure the implicit operators in our models are nonexpansive.
4. On a variety of simulations and real datasets, both quantitative and qualitative evaluations of our results demonstrate the effectiveness and stability of our proposed method.

**Related Work**

**Snapshot Compressive Imaging**

The underlying principle of SCI is to compress the 3D data cube into a 2D measurement by hardware, and then reconstruct the desired signal by algorithms. Considering video SCI, it compresses the spatio-temporal data-cube across the temporal dimension, and thus enables a low-speed camera to capture high-speed scenes. For instance, Llull et al. (Llull et al. 2013) proposed the coded aperture compressive temporal imaging (CACTI) system, which decomposes the 3D cube into its constituent 2D frames and imposes 2D masks for modulation.

Given the masks and measurements, plenty of algorithms including conventional optimization (Liu et al. 2018; Yang...
end-to-end deep learning (Qiao et al. 2020; Meng and Yuan 2021), deep unfolding (Meng, Jalali, and Yuan 2020; Wu, Zhang, and Mou 2021) and plug-and-play (Yuan et al. 2020; Yuan et al. 2021; Wu et al. 2022; Yang and Zhao 2022) are proposed for reconstruction. To solve the ill-posed problem in Eq. (1), additional regularization is usually needed to ensure accurate and stable recovery with respect to noise perturbation. To this end, these algorithms obtain the estimated value \( \hat{x} \) of \( x \) by solving the following problem:

\[
\hat{x} = \arg\min_x \frac{1}{2} \| y - \Phi x \|_2^2 + R(x),
\]

where \( \| y - \Phi x \|_2^2 \) is the fidelity term and \( R(x) \) is the regularization term.

By introducing an auxiliary parameter \( v \), the unconstrained optimization in Eq. (2) can be converted into:

\[
(x, v) = \arg\min_{x, v} \frac{1}{2} \| y - \Phi x \|_2^2 + R(v), \text{ s.t. } x = v.
\]

Using the alternating direction method of multipliers (ADMM) (Boyd et al. 2011) and introducing another parameter \( u \), Eq. (3) could be divided into the following sequence of sub-problems:

\[
x^{(k+1)} = \arg\min_x \frac{1}{2} \| y - \Phi x \|_2^2 + \frac{\rho}{2} \| x - (v^{(k)} - \frac{1}{\rho} u^{(k)}) \|_2^2,
\]

\[
v^{(k+1)} = \arg\min_v \mu R(v) + \frac{\rho}{2} \| v - (x^{(k)} + \frac{1}{\rho} u^{(k)}) \|_2^2,
\]

\[
u^{(k)} = v^{(k)} + \rho(\nu^{(k+1)} - v^{(k)}),
\]

where the superscript \( k \) denotes the iteration number; \( \rho \) is the penalty parameter and \( \mu \) is the regularization weight. Since Eq. (5) can be regarded as a denoising process of \( v \), implicitly we have:

\[
v^{(k+1)} = D^{(k+1)}(x^{(k+1)} + \frac{1}{\rho} u^{(k)}),
\]

where \( D \) is a denoiser.

On the other hand, generalized alternating projection (GAP) (Liao, Li, and Carin 2014) can be used as a (little bit) lower computational workload algorithm with the following two steps:

\[
x^{(k+1)} = v^{(k)} + \Phi^T (\Phi \Phi^T)^{-1}(y - \Phi v^{(k)}),
\]

\[
v^{(k+1)} = D^{(k+1)}(x^{(k+1)}).
\]

Eq. (8) can be solved efficiently due to the special structure of \( \Phi \) in SCI (Jalali and Yuan 2019).

Deep Unfolding

Inspired by optimization algorithms such as ADMM (Boyd et al. 2011) and GAP (Liao, Li, and Carin 2014), deep unfolding methods (Ma et al. 2019; Meng, Jalali, and Yuan 2020; Wu, Zhang, and Mou 2021; Yang, Zhang, and Yuan 2022) are proposed to solve inverse problems in SCI, which consist of a fixed number of architecturally identical blocks. Each of those blocks represents a single iterative step in conventional optimization algorithms. Though deep unfolding successfully assimilate the advantages of the iterative optimization algorithms and could be trained in an end-to-end manner, the fixed number of network blocks in deep unfolding is needed to be kept small for two reasons: i) these systems should be concise to keep a high inference speed for real-time reconstruction; ii) it is challenging to train deep unfolding networks for numerous stages due to memory limitations.

Plug-and-Play

The latest trend is to bridge the gap between deep learning and optimization with the PnP framework. Yuan et al. (Yuan et al. 2021) proposed PnP-ADMM framework and PnP-GAP framework, using a pre-trained denoiser as the proximal operator in Eq. (5) and Eq. (9), respectively. In contrast to deep unfolding, PnP relieves itself from the limited memory by integrating a flexible denoising module into the iterative optimization process. Nevertheless, it suffers manual parameter tuning in addition to the time-consuming reconstruction process. That is, its performance is highly sensitive to the internal parameter selection, including but not limited to the penalty parameter, the denoising level, and the terminal step number. Moreover, the optimal parameter setting differs image-by-image, depending on the modulation process, noise level, noise type, and the unknown image itself.

Memory-Efficient Deep Networks

Since the important factor that limits the development of deep learning and deep unfolding for SCI is limited memory on hardware devices used for training, to address this issue, RevSCI (Cheng et al. 2021) developed a memory-efficient network for large-scale video SCI. Using reversible neural networks, where each layer’s input can be calculated from the layer’s activation during back-propagation, which means the activation during training is not needed to be stored. Nevertheless, it still suffers growing memory occupation along with the increasing depth of the network. In contrast, DEQ reduces memory consumption to a constant (i.e., independent of network depth) by directly differentiating through the equilibrium point and thus circumvents the construction and maintenance of layers. Moreover, DEQ can solve stable estimation, easily extended to larger computing in the test time, while reversible neural networks cannot.

Deep Equilibrium Models

Motivated by the surprisingly recent works (Bai, Kolter, and Koltun 2018; Dehghani et al. 2018; Dabre and Fujita 2019) that employ the same transformation in each layer and still achieve competitive results with the state-of-the-art, Bai et al. (Bai, Kolter, and Koltun 2019) proposed a new approach to model this process and directly computed the fixed point. To leverage ideas from DEQ, Gilton et al. (Gilton, Ongie, and Willett 2021) proposed DEQ for inverse problems in imaging, which corresponds to a potentially infinite number of iteration steps in the PnP reconstruction scheme.
Recurrent Neural Networks

this equilibrium point. Considering SCI reconstruction, we result of DEQ is the equilibrium point itself. Therefore, the

Unlike the conventional optimization method where the ter-

Convergence for specific

network wights

and Willett 2021), for gradient calculation, we optimize the

terms of the implicit infinite-depth RNN architecture and in-

imization iteration or neural network as:

where \( \theta \) denotes the weights of embedded neural networks;

\( x^{(k)} \in \mathbb{R}^{n_B} \) is the output of the \( k \)th iterative step or hidden layer, and \( x^{(0)} = \Phi^\top y; f_0(\cdot; y, \Phi) \) is an iteration map

\( \mathbb{R}^{n_B} \rightarrow \mathbb{R}^{n_B} \) towards a stable equilibrium:

Therefore, as illustrated in Fig. 4, the iteration map is:

\[
\begin{align*}
D_\theta (x; y, \Phi) = & \quad f_\theta (x; y, \Phi) \\
\text{Generalized Alternating Projection } & \quad \text{Regarding the optimization iterations in the GAP method, represented in Eq. (8)-(9), we iteratively update } x \text{ by:}
\end{align*}
\]

\[
\begin{align*}
x^{(k+1)} = & \quad D_\theta^{(k+1)} \left[ x^{(k)} + \Phi^\top (\Phi \Phi^\top)^{-1} (y - \Phi x^{(k)}) \right].
\end{align*}
\]

\[
\begin{align*}
\text{Therefore, as illustrated in Fig. 4, the iteration map is:}
\end{align*}
\]

\[
\begin{align*}
D_\theta (x; y, \Phi) = & \quad f_\theta (x + \Phi^\top (\Phi \Phi^\top)^{-1} (y - \Phi x)).
\end{align*}
\]
calculating an inverse Jacobian-vector product, it avoids the backpropagation through many iterations of \( f_\theta(x; y, \Phi) \). To approximate the inverse Jacobian-vector product, we define the vector \( \hat{a}^{(\infty)} \) as:

\[
a^{(\infty)} = \left[ 1 - \frac{\partial f_\theta(x; y, \Phi)}{\partial x} \right]_{x = \hat{x}} \cdot (\hat{x} - x^*). \tag{21}
\]

Following (Gilton, Ongie, and Willett 2021), it is noted that \( a^{(\infty)} \) is a fixed point of the equation:

\[
a^{(k+1)} = \left[ \frac{\partial f_\theta(x; y, \Phi)}{\partial x} \right]_{x = \hat{x}} \cdot a^{(k)} + (\hat{x} - x^*), \quad \forall k = 0, 1, \ldots, \infty. \tag{22}
\]

Therefore, the same algorithm used to calculate the fixed point \( \hat{x} \) could also be used to calculate \( a^{(\infty)} \). The limit of fixed-point iterations for solving Eq. (22) with initial iterate \( a^{(0)} = 0 \) is denoted equivalently to the Neumann series:

\[
a^{(\infty)} = \sum_{p=0}^{\infty} \left\{ \left[ \frac{\partial f_\theta(x; y, \Phi)}{\partial x} \right]_{x = \hat{x}} \cdot (\hat{x} - x^*) \right\}^p. \tag{23}
\]

To quickly calculate the vector-Jacobian products in Eq. (22) and Eq. (23), a lot of auto-differentiation tools (e.g., autograd packages in Pytorch/Passke et al. 2019) could be utilized. After the accurate approximation of \( a^{(\infty)} \) is calculated, the gradient in Eq. (18) is given by:

\[
\frac{\partial}{\partial \theta} = \left( \frac{\partial f_\theta(x; y, \Phi)}{\partial x} \right) \cdot a^{(\infty)}. \tag{24}
\]

**Convergence Analysis**

Given the iteration map \( f_\theta(:, y, \Phi) : \mathbb{R}^{nB} \to \mathbb{R}^{nB} \), in this section, we discuss conditions that guarantee the convergence of the proposed deep equilibrium models \( x^{(k+1)} = f_\theta(x^{(k)}; y, \Phi) \) to a fixed-point \( \hat{x} \) as \( k \to \infty \).

**Assumption 1 (Convergence of DE-RNN).** For all \( x, x' \in \mathbb{R}^{nB} \), if there exists a constant \( 0 < c < 1 \) satisfies:

\[
\| R_{\theta}(x, y, \Phi) - R_{\theta}(x', y, \Phi) \| \leq c \| x - x' \|, \tag{25}
\]

then the DE-RNN iteration map \( f_\theta(x; y, \Phi) \) is contractive.

**Assumption 2 (Convergence of DE-GAP).** For all \( x, x' \in \mathbb{R}^{nB} \), if there exists a \( \varepsilon > 0 \) such that the denoiser \( \mathcal{D}_\theta : \mathbb{R}^{nB} \to \mathbb{R}^{nB} \) satisfies:

\[
\|(\mathcal{D}_\theta - I)(x) - (\mathcal{D}_\theta - I)(x')\| \leq \varepsilon \| x - x' \|, \tag{26}
\]

where \( (\mathcal{D}_\theta - I)(x) := \mathcal{D}_\theta(x) - x \), that is, we assume the map \( \mathcal{D}_\theta - I \) is \( \varepsilon \)-Lipschitz, then the DE-GAP iteration map \( f_\theta(:, y, \Phi) \) defined in Eq. (15) satisfies:

\[
\| f_\theta(x; y, \Phi) - f_\theta(x'; y, \Phi) \| \leq \eta \| x - x' \| \tag{27}
\]

for all \( x, x' \in \mathbb{R}^{nB} \). The coefficient \( \eta \) is less than 1, in which case the DE-GAP iteration map \( f_\theta(x; y, \Phi) \) is contractive.

Following (Gilton, Ongie, and Willett 2021), to prove \( f_\theta(:, y, \Phi) \) is contractive it suffices to show \( \| \partial_x f_\theta(x; y, \Phi) \| < 1 \) for all \( x \in \mathbb{R}^{nB} \), where \( \| \cdot \| 
\]
Table 1: The results in terms of PSNR (dB) and SSIM by different algorithms on six datasets for video SCI reconstruction.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Kobe</th>
<th>Traffic</th>
<th>Runner</th>
<th>Drop</th>
<th>Vehicle</th>
<th>Aerial</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAP-net-AE-S9</td>
<td>24.20</td>
<td>0.570</td>
<td>21.13</td>
<td>0.685</td>
<td>32.21</td>
<td>0.907</td>
<td>24.19</td>
</tr>
<tr>
<td>GAP-TV</td>
<td>26.46</td>
<td>0.885</td>
<td>20.89</td>
<td>0.715</td>
<td>34.63</td>
<td>0.970</td>
<td>24.82</td>
</tr>
<tr>
<td>E2E-CNN</td>
<td>29.02</td>
<td>0.861</td>
<td>23.45</td>
<td>0.838</td>
<td>34.43</td>
<td>0.958</td>
<td>26.40</td>
</tr>
<tr>
<td>PnP-FFDnet</td>
<td>30.50</td>
<td>0.926</td>
<td>24.18</td>
<td>0.828</td>
<td>32.15</td>
<td>0.933</td>
<td>40.70</td>
</tr>
<tr>
<td>DE-RNN</td>
<td>21.46</td>
<td>0.697</td>
<td>19.47</td>
<td>0.715</td>
<td>27.85</td>
<td>0.818</td>
<td>30.16</td>
</tr>
<tr>
<td>DE-GAP-Unet-3D</td>
<td>26.76</td>
<td>0.866</td>
<td>21.42</td>
<td>0.786</td>
<td>30.45</td>
<td>0.894</td>
<td>33.82</td>
</tr>
<tr>
<td>DE-GAP-RSN-CNN</td>
<td>27.33</td>
<td>0.887</td>
<td>22.58</td>
<td>0.829</td>
<td>30.74</td>
<td>0.903</td>
<td>35.95</td>
</tr>
<tr>
<td>DE-GAP-RSN-Unet</td>
<td>28.92</td>
<td>0.939</td>
<td>23.68</td>
<td>0.869</td>
<td>32.37</td>
<td>0.951</td>
<td>36.54</td>
</tr>
<tr>
<td>DE-GAP-CNN</td>
<td>28.79</td>
<td>0.935</td>
<td>23.55</td>
<td>0.864</td>
<td>32.35</td>
<td>0.950</td>
<td>38.14</td>
</tr>
<tr>
<td>DE-GAP-FFDnet</td>
<td>29.32</td>
<td>0.952</td>
<td>24.71</td>
<td>0.907</td>
<td>33.06</td>
<td>0.971</td>
<td>39.89</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5: Comparison of selected reconstruction results with the spatial size of $256 \times 256 \times 8$.

Figure 6: Comparison of selected reconstruction results of real data Water Balloon with the spatial size of $512 \times 512 \times 10$. Reconstruction of the real data is more difficult than simulations due to the inevitable measurement noise. As shown in this figure, GAP-TV, DeSCI, and PnP-FFDnet (GAP) have more artifacts and distortions around margins. Our model can maintain a clear and accurate image structure, thus leading to higher performance.
Table 2: Average running time per measurement in seconds by different algorithms on classical six datasets.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>DeSCI</th>
<th>PnP-FFDnet</th>
<th>RevSCI</th>
<th>DE-RNN</th>
<th>DE-GAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAP-TV</td>
<td>4.2</td>
<td>6180</td>
<td>3.0</td>
<td>0.19</td>
<td>4.68</td>
</tr>
</tbody>
</table>

Figure 7: Comparison of reconstruction results of real data Chopper Wheel with the spatial size of $256 \times 256 \times 3$.

**Future Work**

Since DEQ under exact gradients may suffer from training time and stability issues, we will incorporate inexact gradients (Geng et al. 2021) and fixed point correction (Bai et al. 2022) to solve these issues and improve the performance. Our preliminary experiments found that inexact gradients could accelerate the backward passes in training our models by roughly $1.3 \sim 1.5 \times$. Another direction is to integrate DEQ with semantic analysis in SCI (Zhang et al. 2022).

**Conclusion**

In this paper, to solve the problems of memory requirement and unstable recovery in existing methods, we propose deep equilibrium models for video SCI. Fusing data-driven regularization and stable convergence in a theoretically sound manner, we combine DEQ with existing methods and design two novel models, i.e., DE-RNN and DE-GAP. Each equilibrium model implicitly learns a nonexpansive operator by training the embedded neural network and analytically computes the fixed point, thus enabling unlimited iterative steps and infinite network depth with only a constant memory requirement in the training and inference process. Furthermore, we analyze the convergence conditions for each equilibrium model to ensure the results of our models converge to equilibrium. We evaluate our proposed models using different neural networks as the implicit operator on a variety of simulations and real datasets. In comprehensive comparisons with existing algorithms, both quantitative and qualitative evaluations of our results demonstrate the effectiveness and stability of our proposed method.
Acknowledgements

This work was supported by the National Natural Science Foundation of China [62271414], Zhejiang Provincial Natural Science Foundation of China [LR23F010001] and Westlake Foundation [2021B1 501-2]. The authors would like to thank Research Center for Industries of the Future (RCIF) at Westlake University for supporting this work and the funding from Lochn Optics. The support and funding from Research Postgraduate Student Innovation Award (The University of Hong Kong) is also appreciated.

References


Sreehari, S.; Venkatakrisnan, S. V.; Wohlberg, B.; Buzzard, G. T.; Drummy, L. F.; Simmons, J. P.; and Bouman, C. A.


