# HNO: High-Order Numerical Architecture for ODE-Inspired Deep Unfolding Networks

**Lin Kong[1]*, Wei Sun[1]*, Fanhua Shang[1,2]†, Yuanyuan Liu[1], Hongying Liu[1,2]†**

[1]Key Lab. of Intelligent Perception and Image Understanding of Ministry of Education,
School of Artificial Intelligence, Xidian University; [2]Peng Cheng Laboratory
xdkonglin0511@163.com, sunwei9915@outlook.com, {fhshang, yyliu, hyliu}@xidian.edu.cn

## Abstract

Recently, deep unfolding networks (DUNs) based on optimization algorithms have received increasing attention, and their high efficiency has been confirmed by many experimental and theoretical results. Since this type of networks combines model-based traditional optimization algorithms, they have high interpretability. In addition, ordinary differential equations (ODEs) are often used to explain deep neural networks, and provide some inspiration for designing innovative network models. In this paper, we transform DUNs into first-order ODE forms, and propose a high-order numerical architecture for ODE-inspired deep unfolding networks. To the best of our knowledge, this is the first work to establish the relationship between DUNs and ODEs. Moreover, we take two representative DUNs as examples, apply our architecture to them and design novel DUNs. In theory, we prove the existence, uniqueness of the solution and convergence of the proposed network, and also prove that our network obtains a fast linear convergence rate. Extensive experiments verify the effectiveness and advantages of our architecture.

## Introduction

This paper mainly considers the following problem:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|y-\Phi x\|_2^2+\lambda\|Dx\|_1, \tag{1}$$

where $\lambda$ is the regularization parameter. There are many applications of this model. When $D$ is a sparse basis matrix, (1) is a compressive sensing (CS) model, while (1) degenerates to the classic Lasso model (Tibshirani 1996) for solving sparse coding (SC) when $D = I$. For CS, $\Phi$ is a measurement matrix, and the purpose of CS is to recover the original signal $x \in \mathbb{R}^n$ from an observation signal $y \in \mathbb{R}^m$, and $m \ll n$. Therefore, (1) is an ill-posed problem, and usually hard to get a numerical solution. We need to combine prior information, such as sparsity prior.

For Problem (1) which is difficult to attain the closed-form solutions, the iterative algorithms have gradually become the mainstream algorithms, such as least angle regression (LARS) (Efron et al. 2004), iterative shrinkage threshold algorithm (ISTA) (Daubechies, Defrise, and De Mol 2004; Blumensath and Davies 2008) and approcimate message passing (AMP) (Donoho, Maleki, and Montanari 2009).

With the development of deep learning, a class of networks called deep unfolding networks (DUNs) (Hershey, Roux, and Weninger 2014) or differentiable programming networks (Amos 2019) has gradually become a powerful candidate of traditional iterative algorithms (Gregor and LeCun 2010; Borgerding, Schniter, and Rangan 2017; Sun et al. 2016; Xie et al. 2019). These networks combines the prior information of the model-based algorithms and the learning ability of deep learning, thus greatly improves the convergence speed of original iterative algorithms, and also greatly reduces the number of parameters in deep networks. Moreover, since DUNs are obtained by expanding traditional iterative algorithms, they can also provide certain interpretability for deep neural networks (DNNs) (LeCun, Bengio, and Hinton 2015).

## Algorithms and Theories of Deep Unfolding

Gregor and LeCun (2010) first proposed the idea of DUN. They presented a deep unfolding network called LISTA by unfolding ISTA into a network by iterations, and set some parameters obtained by training the network. Many empircal and theoretical results (Giryes et al. 2018; Aberdam, Golts, and Elad 2020) show that LISTA can provide a more accurate solution than ISTA. Compared with ISTA, the number of iterations required by LISTA is greatly reduced, even one to two orders of magnitude less.

Since LISTA was proposed, plenty of related work appears in a spurt. On the one hand, after (Gregor and LeCun 2010), many DUNs have been proposed and successfully applied to different fields such as compressive sensing (Zhang and Ghanem 2018; Xiang, Dong, and Yang 2021), computer vision (Zheng et al. 2015; Peng et al. 2018), computational imaging (Mardani et al. 2018), signal processing (Ito, Takabe, and Wadayama 2019) and wireless communication (Cowen, Saridena, and Choromanska 2019; Balatsoukas-Stimming and Studer 2019).

On the other hand, the empirical success has also inspired theoretical research on deeper understanding of DUNs. For instance, Xin et al. (2016) discussed the LIHT (Wang, Ling, and Huang 2016) network, which is obtained by expanding IHT (Blumensath and Davies 2009) into a network, from

*Equal contribution.

†Corresponding authors.

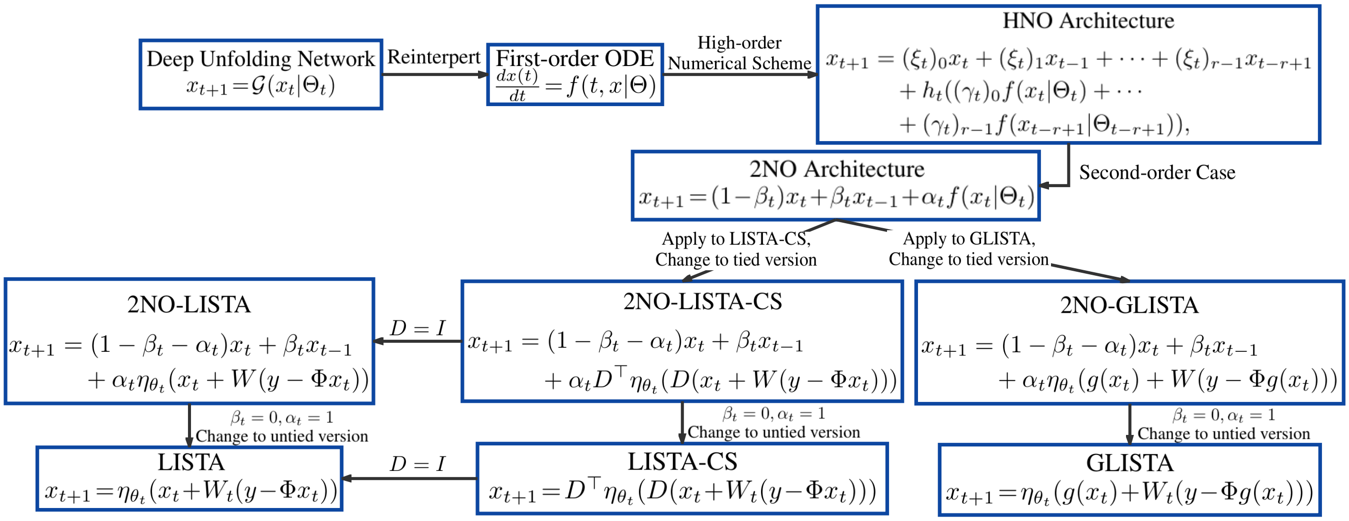
Deep Unfolding Network
$x_{t+1} = \mathcal{G}(x_t|\Theta_t)$
Reinterpert
First-order ODE
$\frac{dx(t)}{dt} = f(t, x|\Theta)$
High-order
Numerical Scheme
HNO Architecture
$x_{t+1} = (\xi_t)_0 x_t + (\xi_t)_1 x_{t-1} + \cdots + (\xi_t)_{r-1} x_{t-r+1}$
$+ h_t((\gamma_t)_0 f(x_t|\Theta_t) + \cdots$
$+ (\gamma_t)_{r-1} f(x_{t-r+1}|\Theta_{t-r+1})),$
Second-order Case
2NO Architecture
$x_{t+1} = (1-\beta_t)x_t + \beta_t x_{t-1} + \alpha_t f(x_t|\Theta_t)$
Apply to LISTA-CS,
Change to tied version
Apply to GLISTA,
Change to tied version
2NO-LISTA
$x_{t+1} = (1 - \beta_t - \alpha_t)x_t + \beta_t x_{t-1}$
$+ \alpha_t \eta_{\theta_t}(x_t + W(y - \Phi x_t))$
$D = I$
2NO-LISTA-CS
$x_{t+1} = (1 - \beta_t - \alpha_t)x_t + \beta_t x_{t-1}$
$+ \alpha_t D^\top \eta_{\theta_t}(D(x_t + W(y - \Phi x_t)))$
2NO-GLISTA
$x_{t+1} = (1 - \beta_t - \alpha_t)x_t + \beta_t x_{t-1}$
$+ \alpha_t \eta_{\theta_t}(g(x_t) + W(y - \Phi g(x_t)))$
$\beta_t = 0, \alpha_t = 1$
Change to untied version
$\beta_t = 0, \alpha_t = 1$
Change to untied version
$\beta_t = 0, \alpha_t = 1$
Change to untied version
LISTA
$x_{t+1} = \eta_{\theta_t}(x_t + W_t(y - \Phi x_t))$
$D = I$
LISTA-CS
$x_{t+1} = D^\top \eta_{\theta_t}(D(x_t + W_t(y - \Phi x_t)))$
GLISTA
$x_{t+1} = \eta_{\theta_t}(g(x_t) + W_t(y - \Phi g(x_t)))$

Figure 1: Summary of this paper. $\mathcal{G}(x_t|\Theta_t)$ is the $t$-layer of a DUN. The details will be described later. We remark that the 2NO-based algorithms shown here are just a few examples of applying our method, which can be applied to more algorithms.

the perspective of improving the RIP constant for the first time. Moreau and Bruna (2017) explained the mechanism of LISTA by re-factorizing the Gram matrix of dictionary. Chen et al. (2018b) reduced the number of parameters of the network by proving the coupling relationship of the matrix parameters in the iteration of LISTA, and the linear convergence of LISTA is proved for the first time. Liu et al. (2019) further reduced the number of learnable parameters by proposing a tied network, called ALISTA, whose matrix parameter is obtained by solving a data-independent optimization problem. Inspired by gated recurrent unit (GRU) (Cho et al. 2014; Chung et al. 2015), Wu et al. (2020) proposed GLISTA, which can gain the estimation obtained through LISTA by improving the soft-thresholding function, thereby enhancing the performance. Hosseini et al. (2020) presented a history-cognizant unrolling of the optimization algorithm, called HC-PGD, with dense connections across iterations for improved performance. Li et al. (2021b) proposed ELISTA based on the extragradient descent method (Nguyen et al. 2018), and established the relationship with the well-known Res-Net (He et al. 2016). In addition, there are also many other theoretical studies on DUNs, such as (Giryes et al. 2018; Ablin et al. 2019; Takabe and Wadayama 2020; Meng et al. 2020).

As discussed above, we know that the research on DUNs has attracted increasing attention, and a series of related work has appeared. However, the idea of existing DUNs is generally to unfold existing traditional iterative algorithms. We know that in addition to using traditional optimization algorithms to explain deep networks, differential equations including ordinary differential equations (ODEs) and partial differential equations (PDEs) are also often used to explain deep models. A series of works (Haber and Ruthotto 2017; Weinan 2017; Chen et al. 2018a; Ruthotto and Haber 2020) discussed the connection between ODEs/PDEs and existing DNNs. For instance, some works (Lu et al. 2018; Li et al. 2021a) were also proposed to guide the construction of a new deep network structure through the knowledge of ODE. However, so far, ODEs have not been effectively applied to the research of DUNs, which is a gap in the research of existing DUNs, and further study is needed.

## Motivation and Main Contributions

The research of numerical differential equations has been mature. The application of differential equations to explain deep networks and to propose networks with novel structures has begun to emerge. Therefore, for the study of DUNs, we ask a natural question:

*Can we establish a connection between the DUNs and ODEs, so as to guide us to design innovative, effective, and interpretable DUNs?*

Through our studies in this work, the answer to the above question is "yes". By theoretical derivation, we show that all the DUNs with a single variable can be converted into the form of first-order ODE. In this way, it is easy to introduce various methods in ODEs to improve the efficiency and performance of state-of-the-art (SOTA) DUNs such as LISTA, LAMP, LIHT, ALISTA, GLISTA and ELISTA. We show the flow chart of the idea of this paper in Figure 1, and the main contributions of this paper are summarized as follows:

- Firstly, we show that the DUNs can be regarded as the numerical schemes approximating ODE $\frac{dx}{dt} = f(t, x)$. By rewriting the original update rule of the DUN and converting it into an ODE, we consider this class of networks from a novel perspective. To the best of our knowledge, this is the first time to systematically establish the relationship between DUNs and ODE.
- We also construct a framework called High-order Numerical methods for ODE-inspired DUNs (HNO) by introducing high-order numerical methods in ODEs. Besides, we take the second-order Numerical method for ODE-inspired DUNs (2NO) as a special case and apply it to

the two classic DUNs, LISTA-CS and GLISTA for CS problems and SC problems, respectively, and obtain innovative networks, called 2NO-LISTA-CS and 2NO-GLISTA. Moreover, we also propose a third-order Numerical ODE (3NO) architecture, and apply the 2NO and 3NO architectures to more DUNs.

- In theory, we prove the existence and uniqueness of the solution of 2NO-LISTA-CS from the perspective of ODE, and prove that our network can achieve linear convergence from the perspective of the deep unfolding network. Moreover, we find that the convergence rate of our 2NO-LISTA-CS with the 2NO architecture has an almost square improvement over that of original LISTA-CS.
- Finally, in order to verify the effectiveness of our designed networks and high-order framework, we conduct extensive experiments, including synthetic data experiments, image inpainting, and natural image CS. The results show that the HNO architecture can effectively improve the performance of original networks and is superior to existing SOTA methods.

## Related Work

In this section, we introduce some related work, including two representative DUNs and the well-known ODE numerical methods.

### LISTA-CS and GLISTA

ISTA is a popular first-order proximal method, which is very suitable for solving many large-scale linear inverse problems. From (Zhang and Ghanem 2018), we know that for the CS problem (1), the update rule of ISTA is

$$x_{t+1} = D^\top \eta_{\theta_t}\Big(D\Big(x_t + \frac{1}{L}\Phi^\top(y - \Phi x_t)\Big)\Big),$$

where $D$ is a fixed orthogonal basis matrix, $\eta_\theta(\cdot)$ is the soft-thresholding function, and $L$ is the largest singular value of $\Phi$. Then if we regard the matrix $\frac{1}{L}\Phi^\top$ as a learnable matrix parameter $W_t$, and make $\theta_t$ also learnable, we can obtain the ISTA-based DUN for solving the CS problems. In order to distinguish it from LISTA, we name this network for CS problems as LISTA-CS. Moreover, when the basis matrix is replaced by the convolution operation $\mathcal{F}$, that is, Problem (1) is transformed into the following form:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|y - \Phi x\|_2^2 + \lambda\|\mathcal{F}(x)\|_1, \tag{2}$$

ISTA-Net (Zhang and Ghanem 2018) was proposed to solve Problem (2).

Recently, Wu et al. (2020) proposed that due to the disadvantage of the soft-thresholding function, the code components in LISTA estimations may be lower than expected values, i.e., the algorithms require gains. Therefore, inspired by gated recurrent unit (GRU) (Cho et al. 2014; Chung et al. 2015), they presented a gain gate operator, which is equivalent to multiplying the soft-thresholding function by a coefficient greater than 1 to improve it. Finally, by combining this gain gate with LISTA, they proposed GLISTA, whose update rule is

$$x_{t+1} = \eta_{\theta_t}(g(x_t) + W_t(y - \Phi g(x_t))), \tag{3}$$

where $g(x_t) = g_t(x_t, y|\Lambda_t^g) \odot x_t$. $g_t(\cdot,\cdot|\Lambda_t^g)$ is a gate function to output an $n$-dimensional vector, and $\Lambda_t^g$ is the set of its learnable parameters. $\odot$ denotes that each element is multiplied by coordinates. Note that this network is used to solve the Lasso problem (Problem (1) with $D = I$).

### Well-known Numerical Schemes of ODE

In this subsection, we introduce two commonly used ODE numerical estimation schemes: the Runge-Kutta scheme and linear multi-step (LM) scheme.

- **Runge-Kutta Scheme:** For the ODE $\frac{dy}{dx} = f(x, y)$, we can use the following scheme to find a numerical solution:

$$y_{t+1} = y_t + h\sum_{i=1}^{r} v_i k_i, \tag{4}$$

where $h$ is the step size, $v_i$ is the weighting factor, $k_i$ is the slope of the $i$-th segment, and there are $r$ segments in total. By taking the slope of the first segment $k_1 = f(x_t, y_t)$, we can use the following formula to recursively obtain other slopes:

$$k_j = f\Big(x_t + d_j h, y_t + h\sum_{l=1}^{j-1} p_{jl} k_j\Big),\ j = 2, 3, \cdots, r, \tag{5}$$

where $d_j$ and $p_{jl}$ are undetermined constants. (4) and (5) are called the $r$-order Runge-Kutta scheme, which is one of the most classic ODE numerical methods.

- **Linear Multi-step Scheme:** Another well-known ODE numerical scheme is the linear multi-step (LM) method, and the iterative scheme of the $r$-step LM method is

$$\begin{aligned} y_{t+1} = {}& \xi_0 y_t + \xi_1 y_{t-1} + \cdots + \xi_{r-1} y_{t-r+1} \\ & + h(\gamma_{-1} f_{t+1} + \gamma_0 f_t + \cdots + \gamma_{r-1} f_{t-r+1}), \end{aligned} \tag{6}$$

where $f_i = f(x_i, y_i)$, $\xi_i$ and $\gamma_i$ are coefficients, and $\sum_{i=0}^{r-1} \xi_i = 1$. When $\gamma_{-1} = 0$, the above scheme is explicit, otherwise it is implicit.

These two methods are the two most classic ODE numerical schemes. Since the calculation process of Runge-Kutta is more complicated than that of LM, we apply the LM scheme that is easier to apply to DUNs in our study.

## Our High-order Numerical Architecture for ODE-Inspired Deep Unfolding Networks

In this section, we first reinterpret the DUNs from the perspective of ODEs. Inspired by the well-known LM method, which is a high-order numerical scheme of ODE, we propose a novel architecture that is applicable to all the DUNs with a singe variable, called the High-order Numerical methods for ODE-inspired DUNs (HNO). Finally, we apply the proposed HNO architecture to the classic DUNs, LISTA-CS and GLISTA, and design new efficient DUNs.

### From Deep Unfolding to ODEs

The expression of the $t$-th layer of a DUN is as follows:

$$x_{t+1} = \mathcal{G}(x_t|\Theta_t), \tag{7}$$

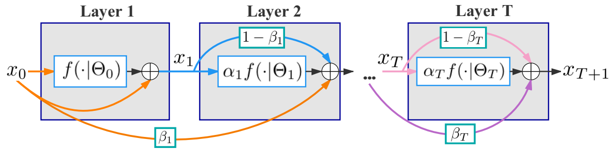

Figure 2: Our 2NO architecture for deep unfolding networks. Note that the structure diagram of our third-order architecture is shown in the Appendix.

where $\mathcal{G}(\cdot|\Theta)$ is the operation of one layer of any DUN, and $\Theta$ is the set of all learnable parameters of $\mathcal{G}(\cdot|\Theta)$. Thus,

$$x_{t+1} - x_t = \mathcal{G}(x_t|\Theta_t) - x_t, \tag{8}$$

which can be viewed as a difference. If we set the step size of the difference tending to 0 and transform (7) into a continuous form, then we obtain

$$\frac{dx(t)}{dt} = \mathcal{G}(x(t)|\Theta) - x(t).$$

Next, we define $f(t, x|\Theta) = \mathcal{G}(x(t)|\Theta) - x(t)$, then we have

$$\frac{dx(t)}{dt} = f(t, x|\Theta), \tag{9}$$

which is an ordinary differential equation. Moreover, if we set $f(x_t|\Theta) = \mathcal{G}(x_t|\Theta_t) - x_t$, then from (8), which is obtained after a simple change of the original DUN, we attain $x_{t+1} = x_t + f(x_t|\Theta)$, which can be interpreted as an approximation to one step of the forward Euler scheme with step size 1. Thus, by simply deriving the iterative rule of the DUN, we can draw the following conclusion:

*All the DUNs with a single variable can be regarded as a step of the forward Euler numerical estimation of ODE,* $\frac{dx}{dt} = f(x, t)$.

To the best of our knowledge, this is the first time to systematically establish the connection between the DUNs and ODEs. This conclusion, which means that the rich knowledge in the field of ODE can be introduced into the study of DUNs, is very valuable, and the new networks obtained in this way are also interpretable, to a certain extent.

## Proposal of HNO Architecture

After converting the DUNs into numerical estimation of ODEs, there is an obvious advantage that we can introduce high-order numerical methods in numerical differential equations, such as the well-known Runge-Kutta and LM methods. By taking the LM method as an example, we propose an innovative architecture based on high-order numerical methods of ODE.

Recall that the definition of the $r$-step LM scheme in (6), which is actually a high-order generalization method of the classic Euler scheme. For the DUN (7), since $x_{t+1}$ is not yet known when it is iteratively calculated, we cannot use the implicit LM scheme, but can only use the explicit LM scheme, that is, we need to fix $\gamma_{-1} = 0$ in (6). Then, for DUNs, according to (6), we can improve (8) and obtain the following ODE-inspired $r$-order numerical architecture based on the $r$-order LM scheme.

$$\begin{aligned} x_{t+1} = {} & (\xi_t)_0 x_t + (\xi_t)_1 x_{t-1} + \cdots + (\xi_t)_{r-1} x_{t-r+1} \\ & + h_t((\gamma_t)_0 f(x_t|\Theta_t) + \cdots \\ & + (\gamma_t)_{r-1} f(x_{t-r+1}|\Theta_{t-r+1})), \end{aligned} \tag{10}$$

where $(\xi_t)_i$, $(\gamma_t)_i$ and $h_t$ correspond to $\xi_i$, $\gamma_i$ and $h$ in (6), but the difference is that $(\xi_t)_i$, $(\gamma_t)_i$ and $h_t$ are learnable parameters, and $f(x_t|\Theta_t) = \mathcal{G}(x_t|\Theta_t) - x_t$. Finally, for DUNs, we propose an innovative architecture, called High-order Numerical architecture for ODE-inspired DUNs (HNO).

Note that this architecture can be easily applied to any DUNs with a single variable. Moreover, since our proposed method is a high-order generalization method of the original Euler method-based network, it will effectively improve the convergence rate and performance, which will be verified by both theoretical analysis and numerical results.

In order to provide a relatively concise and practical network, we also propose a second-order Numerical method for ODE-inspired DUNs (2NO), which is a second-order special case of our HNO architecture, and the iteration rule of our 2NO architecture is

$$x_{t+1} = (1 - \beta_t) x_t + \beta_t x_{t-1} + \alpha_t f(x_t|\Theta_t), \tag{11}$$

whose structure diagram is shown in Figure 2. For the first layer which is a special case, we specifically fix $\alpha_0 = 1$ and $\beta_0 = 0$, that is, $x_1 = x_0 + f(x_0|\Theta_0)$.

Similarly, we can also obtain the following third-order numerical ODE (3NO) architecture,

$$x_{t+1} = (1 - \beta_t^1 - \beta_t^2) x_t + \beta_t^1 x_{t-1} + \beta_t^2 x_{t-1} + \alpha_t f(x_t|\Theta_t), \tag{12}$$

whose structure diagram can be found in the Appendix. In fact, the higher-order HNO architecture can also derive iterative rules and structure diagrams accordingly, and the process is similar, so we will not elaborate them in detail here.

Besides, we note that Lu et al. (2018) also used the 2-step LM method to improve the network structure of ResNet (He et al. 2016), but they only simply introduced a 2-step LM method into ResNet and ResNeXt, while we propose a more general architecture, which can be generalized to higher-order cases, such as the 3NO architecture in (12), and the 2NO architecture is just a second-order special case of the proposed architecture. In addition, our architecture is for DUNs, while the 2-step LM method in (Lu et al. 2018) is proposed for ResNet-style networks.

## 2NO-LISTA-CS and 2NO-GLISTA

Below, we take two classic DUNs, LISTA-CS and GLISTA as examples, apply our 2NO architecture to the two networks, and give specific update rules. As mentioned above, these two networks can be applied to the CS model and the Lasso model, respectively. We first consider LISTA-CS, whose iteration formula for Problem (1) is

$$x_{t+1} = D^\top \eta_{\theta_t}(D(x_t + W_t(y - \Phi x_t))), \tag{13}$$

where $\theta_t$, $W_t$ are learnable parameters. Then we apply our 2NO architecture to LISTA-CS, and obtain a novel network with the following update rule:

$$
\begin{aligned}
x_{t+1} &= (1-\beta_t)x_t + \beta_t x_{t-1} \\
&\quad + \alpha_t(D^\top \eta_{\theta_t}(D(x_t + W(y-\Phi x_t))) - x_t) \\
&= (1-\beta_t-\alpha_t)x_t + \beta_t x_{t-1} \\
&\quad + \alpha_t D^\top \eta_{\theta_t}(D(x_t + W(y-\Phi x_t))),
\end{aligned}
\tag{14}
$$

where $\alpha_t$, $\beta_t$ and $W$ are data-driven. Besides, from the calculation of $x_1$ in (11), we can get the expression of $x_1$, i.e., $x_1 = D^\top \eta_{\theta_0}(D(x_0 + W_0(y - \Phi x_0)))$, which is the same as LISTA-CS in (13).

For this network, we have the following remarks:

- We note that the proposed deep network is tied, if we set the learnable matrix $W$ different for each layer, i.e., $W_t$, like LISTA. Then we will get an untied network. From (Liu et al. 2019), we can draw a fact that the tied network is usually better than the untied network.
- Moreover, by applying our 2NO architecture, we introduce a learnable step size parameter $\alpha_t$ outside the nonlinear operator. However, for LISTA, which is equivalent to the forward Euler method, this parameter degenerate to $\alpha_t = 1$. Therefore, for the tied variant of our network, the parameter $\alpha_t$ can also make the parameters between different layers diversified to a certain extent, thereby improving the learning ability of the network. Besides, the tied network can effectively reduce the number of learnable parameters. Thus, we infer that the tied variant of our network has better performance, which will be verified in our experiments.
- In addition, compared to LISTA, our proposed network uses more information including the information in $x_{t-1}$, which is similar to the idea of momentum in accelerated algorithms. Therefore, this network can solve the objective problem more accurately and fast.

To sum up, we finally propose an innovative network that applies our 2NO architecture to LISTA-CS, called 2NO-LISTA-CS. Here, we emphasize that we define 2NO-LISTA-CS as a tied network, that is, the learnable matrix parameters of each layer are the same, and for the untied variant, we name it 2NO-LISTA-CS(u). Moreover, the network proposed below is also named according to this rule.

Besides, for ISTA-Net that uses convolution operation instead of base matrix multiplication, we also propose the 2NO-ISTA-Net, and its iteration is as follows:

$$
\begin{aligned}
x_{t+1} &= (1-\beta_t-\alpha_t)x_t + \beta_t x_{t-1} \\
&\quad + \alpha_t \tilde{\mathcal{F}}_t \eta_{\theta_t}(\mathcal{F}_t(x_t + \rho_t \Phi^\top(y - \Phi x_t))),
\end{aligned}
$$

where $\alpha_t$, $\beta_t$, $\rho_t$, $\tilde{\mathcal{F}}_t$ and $\mathcal{F}_t$ are data-driven. $\tilde{\mathcal{F}}_t$ is the left inverse of $\mathcal{F}_t$, which satisfies $\tilde{\mathcal{F}}_t \circ \mathcal{F}_t = \mathcal{I}$, where $\circ$ denotes the combination of operators, and $\mathcal{I}$ is the identity operator. Similar to 2NO-LISTA-CS, the iteration of the first layer of 2NO-ISTA-Net is the same as ISTA-Net.

Similarly, from (3), we can obtain the update rule of 2NO-GLISTA, by introducing our 2NO architecture into GLISTA.

$$
\begin{aligned}
x_{t+1} &= (1-\beta_t-\alpha_t)x_t + \beta_t x_{t-1} \\
&\quad + \alpha_t \eta_{\theta_t}(g(x_t) + W(y - \Phi g(x_t))).
\end{aligned}
$$

The first layer of 2NO-GLISTA is the same as GLISTA, and we obtain the untied network 2NO-GLISTA(u) when the weight matrix $W$ becomes different $W_t$ layer by layer.

Finally, we present some innovative DUNs for different problem models (CS and Lasso): 2NO-LISTA-CS and 2NO-GLISTA, as well as their untied variants, based on our 2NO architecture. In fact, for (14), when $D = I$, we can also get a variant of LISTA to solve the Lasso problem, that is, 2NO-LISTA obtained by the specialization of 2NO-LISTA-CS.

The two proposed networks are only two special cases of our HNO architecture. Moreover, we can also apply our higher-order HNO architectures to more DUNs, such as LIHT, LAMP (Borgerding, Schniter, and Rangan 2017), ALISTA and SLISTA (Ablin et al. 2019), and get a wealth of innovative DUNs . More examples of higher-order HNO architecture for other DUNs are provided in the Appendix.

## Theoretical Analysis

In this section, we provide the theoretical analysis of the proposed 2NO-LISTA-CS, including the proof of the existence and uniqueness of the solution and the convergence analysis of the network. Firstly, from the perspective of ODE, we prove that the solution of Problem (1) solved by 2NO-LISTA-CS exists and is unique. In order to provide the convergence guarantee and explore the convergence speed of the network, we analyze the convergence rate of 2NO-LISTA-CS from the perspective of DUN , and show the convergence property.

Firstly, we provide the existence and uniqueness of the solution of 2NO-LISTA-CS through Theorem 1 below.

**Theorem 1 (Existence and Uniqueness)** *For Problem* (1)*, the solution obtained by 2NO-LISTA-CS, whose update rule is shown in* (14)*, exists and is unique.*

The detailed proof of Theorem 1 can be found in the Appendix. After proving the existence and uniqueness of the solution of 2NO-LISTA-CS, we also analyze the convergence and the convergence rate of 2NO-LISTA-CS.

**Assumption 1 (Basic assumption)** *The optimal solution of Problem* (1) *$x_*$ satisfies*

$$
x_* \in \mathcal{X}(B,s) \overset{\triangle}{=} \{x_* \,|\, |[x_*]_i| \le B, \forall i, \|x_*\|_0 \le s\},
$$

*which means $x_*$ is bounded and $s$-sparse.*

This assumption is a basic assumption, and almost all related work such as (Liu et al. 2019; Wu et al. 2020; Li et al. 2021b) made this assumption.

**Definition 1 (Liu et al. (2019))** *Given $\Phi \in \mathbb{R}^{m\times n}$ whose columns are normalized, we define its generalized mutual coherence:*

$$
\mu(\Phi) = \inf_{\substack{W\in\mathbb{R}^{n\times m} \\ W_{i,:}\Phi_{:,i}=1,\forall i}} \left\{ \max_{\substack{i\neq j \\ 1\le i,j\le n}} W_{i,:}\Phi_{:,j} \right\}.
$$

*Furthermore, based on the definition of $\mu(\Phi)$, we define the set $\mathcal{W}(\Phi)$ as follows:*

$$
\mathcal{W}(\Phi) = \left\{ W | \max_{\substack{i\neq j \\ 1\le i,j\le n}} W_{i,:}\Phi_{:,j} = \mu(\Phi), W_{i,:}\Phi_{:,i} = 1, \forall i \right\}.
$$

*A weight matrix $W$ is "good" if $W \in \mathcal{W}(\Phi)$.*

This definition was first proposed in (Liu et al. 2019). From Lemma 1 in (Chen et al. 2018b), we have $\mathcal{W}(\Phi) \neq \varnothing$.

**Definition 2** *Given a model with the learnable parameter set $\Theta$, in which $\theta_t = \Gamma\mu(\Phi)\sup_{x_*}\|x_t - x_*\|_1$, we employ $\omega_{t+1}(k_{t+1}|\Theta)$ to characterize its relationship with the "false positive" rate, which is*

$$\begin{aligned}&\omega_{t+1}(k_{t+1}|\Theta)\\ =&\sup_{\forall x_*, |\mathrm{supp}(\check{x}_{t+1})\bigcup \mathrm{supp}(x_*)| \leq |\mathrm{supp}(x_*)| + k_{t+1}} \Gamma,\end{aligned}$$

*where $\check{x}_{t+1} = (1-\beta_t-\alpha_t)x_t + \beta_t x_{t-1} + \alpha_t D^\top \eta_{\theta_t}(D((I - W\Phi)(x_t - x_*)))$, and $k_{t+1}$ is the desired maximal number of "false positive" of $x_{t+1}$.*

This definition is similar to Definition 2 in (Wu et al. 2020), but our Definition 2 is applicable to 2NO-LISTA-CS.

Based on the above assumption and definitions, we give the convergence property of 2NO-LISTA-CS.

**Theorem 2 (Linear Convergence)** *If Assumption 1 holds, $W \in \mathcal{W}(\Phi)$ can be satisfied by choosing $W$ properly, $\theta_t = \omega_{t+1}(k_{t+1}|\Theta)\mu(\Phi)\sup_{x_*}\|x_t - x_*\|_1$ is achieved, $\frac{1-u}{2} + \frac{1}{2\mu(\Phi)^2+2\mu(\Phi)} \leq s < \frac{1-u}{2} + \frac{1}{2\mu(\Phi)}$, $\frac{1}{\mu(\Phi)+1} \leq \alpha_t \leq kc$, where $k$ is a constant that satisfies $\frac{1}{(\mu(\Phi)+1)c} \leq k \leq 1$ and $c = (2s + u - 1)\mu(\Phi)$, $\hat{\beta} \leq \frac{(1-k)c^4}{c^2+1}$, where $\hat{\beta} = \max_t |\beta_t|$, then the sequence generated by 2NO-LISTA-CS satisfies the following result:*

$$\|x_{t+1} - x_*\|_2 \leq c^{2t} sCB,$$

*where $c = (2s + u - 1)\mu(\Phi) < 1$, and $C = (2s + u - 1)\mu(\Phi) + \frac{\hat{\beta}}{((2s+u-1)\mu(\Phi))^2}$ is a constant.*

The detailed proof of Theorem 2 is provided in the Appendix. In Theorem 2, $u$ is a small parameter, whose definition is given in the proof. This theorem shows that 2NO-LISTA-CS achieves linear convergence. Besides, we note that the convergence rate of the network obtains an almost square improvement over that of LISTA, after applying our 2NO architecture, which is consistent with the fact that the LM method used in the HNO architecture is a high-order numerical method for ODE.

Due to page limit, we only give the theoretical analysis of 2NO-LISTA-CS. The theoretical analysis of 2NO-GLISTA can be obtained by combining the theories in (Wu et al. 2020) and this paper, and the analysis of the corresponding untied variants of the networks is similar to that of the tied variant, thus we omit them here.

# Experimental Results

In this section, we first verify the effectiveness of our HNO architecture by comparing the performance of different DUNs on synthetic data. Then we evaluate our 2NO-LISTA, 2NO-GLISTA and 2NO-LISTA-CS for natural image inpainting and image CS tasks, respectively. All experimental settings and all training follow the previous work (Chen et al. 2018b; Zhang and Ghanem 2018; Wu et al. 2020; Aberdam, Golts, and Elad 2020). For the learnable parameters, $\alpha_t$, $\beta_t$, $\theta_t$ and $W$ are initialized as 1.0, -0.5, $\frac{\lambda}{L}$ and $\frac{1}{L}\Phi^\top$ respectively. We run all the experiments ten times and show the results after averaging.

## Sparse Representation on Synthetic Data

In order to verify the effectiveness of HNO architecture, we perform sparse representation experiments for the Lasso model (Problem (1) with $D = I$) on the synthetic data.

We evaluate LISTA, GLISTA, 2NO-LISTA and 2NO-GLISTA (and their untied variants) that apply our HNO architecture, with three different settings of noise levels expressed by SNR (Signal-to-Noise Ratio) as the indicator and condition numbers $\kappa$ of ill conditioned matrix $\Phi$ on sparse coding problems. We set $m = 250$, $n = 500$ and $T = 16$. The procedure of generating synthetic data is provided in the Appendix. Besides, to evaluate the ability of different orders of HNO architecture to improve the DUNs, we also compare the performance of 2NO-LISTA and 3NO-LISTA whose update rule can be found in the Appendix.

All the results are shown in Figure 3, where NMSE (in dB) is defined as follows:

$$\mathrm{NMSE}(\hat{x}, x_*) = 10\log_{10}\left(\frac{\mathbb{E}\|\hat{x} - x_*\|^2}{\mathbb{E}\|x_*\|^2}\right),$$

where $\hat{x}$ represents the output of the networks. From the comparison of LISTA, 2NO-LISTA(u), GLISTA and 2NO-GLISTA(u) in Figure 3, we can see that our 2NO architecture can greatly improve the convergence speed and accuracy in all the cases[1], which implies that our HNO architecture can effectively enhance the performance of DUNs. In addition, from the comparison of the tied and untied variants of the two proposed networks in the Appendix, we can verify the inference: the tied network usually outperforms the untied variant. Therefore, in all subsequent experiments, we use the tied variant of our networks. Moreover, from the comparison of 2NO-LISTA and 3NO-LISTA in Figure 3, we know that higher-order HNO architecture can further improve the performance of the networks to a certain extent, but not particularly obvious. Therefore, we mainly use the 2NO-based network for subsequent experiments

## Natural Image Inpainting

In order to verify the effectiveness of our proposed networks in practical problems, in this subsection, we apply our networks, 2NO-LISTA and 2NO-GLISTA to image inpainting problems (Aberdam, Golts, and Elad 2020) with 50% pixels missing, and compare them with many algorithms, ISTA, LISTA and GLISTA.[2] We use BSD500 as the training set, Set 11 as the test set, and randomly extract 100,000 and 5,000 8×8 patches from the images in the BSD500 training set and validation set, respectively, for training. Besides, for the dictionary matrix $\Phi$, we use the same dictionary as in (Aberdam, Golts, and Elad 2020).

Table 1 shows PSNR results of different algorithms with 50% pixels missing. More qualitative results can be found in

[1] The iterative forms of this kind of network (i.e., DUNs) are very similar, thus the time of one iteration of different DUNs is basically the same. Therefore, our experiments can verify that our 2NO architecture can improve the convergence speed.

[2] In order to compare the networks fairly, we change the iterative rules of all the networks into the same settings as in (Aberdam, Golts, and Elad 2020), which is also provided in the Appendix.

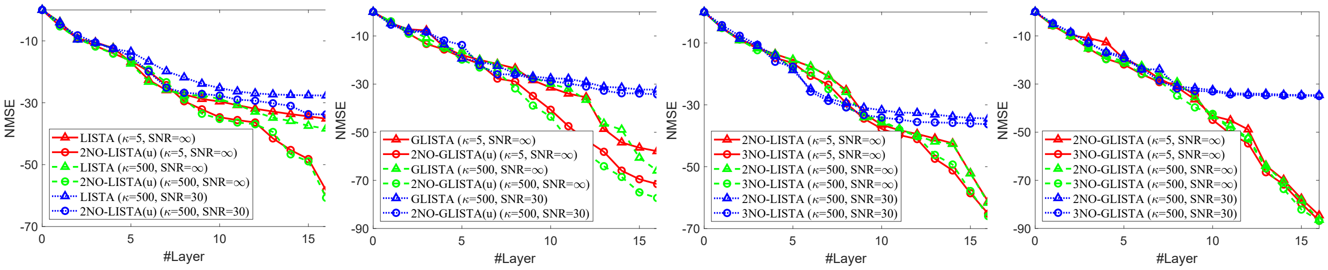

Figure 3: Comparison of the DUNs with different layers under different SNR and $\kappa$. In the two subfigures on the left, we compare the performance before and after applying our 2NO architecture on LISTA and GLISTA to verify its effectiveness. In the two subfigures on the right, we mainly compare the performance of our 2-order and 3-order architectures.

| | Barbara | Boat | House | Lena | Peppers | C.man | Couple | Fingerprint | Hill | Man | Montage |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ISTA | 23.26 | 24.94 | 26.32 | 27.32 | 23.13 | 22.32 | 24.93 | 19.95 | 26.86 | 25.91 | 22.09 |
| LISTA | 27.19 | 31.98 | 35.55 | 35.46 | 30.74 | 28.34 | 32.35 | 31.13 | 33.45 | 32.67 | 29.21 |
| GLISTA | 27.36 | 32.11 | 35.93 | 35.73 | 30.70 | 28.44 | 32.55 | 31.28 | 33.49 | 32.83 | 28.88 |
| 2NO-LISTA | **28.13** | 32.62 | 36.38 | **36.28** | 31.78 | **28.91** | 33.19 | 32.16 | **33.82** | 33.27 | **29.80** |
| 2NO-GLISTA | 28.11 | **32.64** | **36.65** | 36.26 | **31.92** | 28.69 | **33.29** | **32.19** | 33.80 | **33.29** | **29.80** |

Table 1: PSNR (dB) results of natural image inpainting. The best is marked in bold and the second best is underlined.

| Algorithms | CS Ratio | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1% | 4% | 10% | 25% | 30% | 40% | 50% |
| TVAL3 | 16.43 | 18.75 | 22.99 | 27.92 | 29.23 | 31.46 | 33.55 |
| D-AMP | 5.21 | 18.40 | 22.64 | 28.46 | 30.39 | 33.56 | 35.92 |
| IRCNN | 7.70 | 17.56 | 24.02 | 30.07 | 31.18 | 34.06 | 36.23 |
| SDA | 17.29 | 20.12 | 22.65 | 25.34 | 26.63 | 27.79 | 28.95 |
| ReconNet | 17.27 | 20.63 | 24.28 | 25.60 | 28.74 | 30.58 | 31.50 |
| LISTA-CS | 18.83 | 22.08 | 25.20 | 29.96 | 31.21 | 33.70 | 36.01 |
| 2NO-LISTA-CS | **19.25** | **22.35** | **25.60** | **30.46** | **31.90** | **34.60** | **37.24** |

Table 2: Comparisons of average PSNR (dB) performance on Set11 with different CS ratios. The best performance is marked in bold and the second best is underlined.

the Appendix. From all the results, we can indicate that after applying our 2NO architecture, the networks obtain better performance than their basic networks, respectively, and our networks outperform all the other methods.

## Natural Image Compression Sensing

In this subsection, we perform a traditional CS based on a sparse basis matrix on natural images to evaluate 2NO-LISTA-CS and many other methods, TVAL3 (Li et al. 2013), D-AMP (Metzler, Maleki, and Baraniuk 2016), IRCNN (Zhang et al. 2017), SDA (Mousavi, Patel, and Baraniuk 2015), ReconNet (Kulkarni et al. 2016) and LISTA-CS. We produce the results of LISTA-CS by ourselves, and refer to the results in (Zhang and Ghanem 2018) for other compared algorithms. We also adopt the same BSD500 for the training set, but a different Set 11 as the test set, and randomly extract 30,000 and 5,000 patches with size 32×32 from the images in the BSD500 training set and validation set, respectively, for training. Besides, as in (Abo-Zahhad et al. 2015), we utilize the DCT transformation matrix as the sparse base matrix $D$ in the CS problem model (1).

The results with different compression ratios are reported in Table 2. Moreover, we make a more comprehensive comparison between LISTA-CS and 2NO-LISTA-CS. For these two networks, we give the PSNR results and the qualitative results of each image on Set11 under different compression ratios in the Appendix. From all the results, we know that our 2NO architecture can effectively improve the performance of LISTA-CS. Moreover, we obtain that our 2NO-LISTA-CS network outperforms the other methods.

## Conclusion and Further Work

In this paper, in order to introduce the rich knowledge on ODEs into the study of DUNs, we reinterpreted each DUN into an ODE, systematically established the relationship between ODE and DUNs. Besides, we constructed a High-order Numerical architecture for ODE-inspired DUNs, called HNO architecture, that can be applied to any DUNs with a single variable by using the classical linear multi-step method, and we applied it to existing DUNs to obtain innovative HNO-based deep unfolding networks. The linear convergence with an improved rate of 2NO-LISTA-CS, which is one of our proposed networks, was proved. Extensive experimental results verified the high efficiency of our HNO architecture and improved deep DUNs. In this work, we only introduced the linear multi-step method and studied the DUNs with a single variable. How to construct a more general architecture to establish the relationship between ODEs and DUNs is our future work. Besides, we only apply the numerical methods for the first-order ODE into DUNs. Recently, Sander et al. (2021) presented Momentum ResNet and successfully built the connection between the second-order ODE and the network. Therefore, how to introduce higher-order ODE related theories into the research of DUNs is also an important part of our future work.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61876220, 61876221 and 61976164), the Project supported the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 61621005), the Program for Cheung Kong Scholars and Innovative Research Team in University (No. IRT_15R53), and the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048).

## References

Aberdam, A.; Golts, A.; and Elad, M. 2020. Ada-LISTA: Learned Solvers Adaptive to Varying Models. *arXiv preprint arXiv:2001.08456*.

Ablin, P.; Moreau, T.; Massias, M.; and Gramfort, A. 2019. Learning step sizes for unfolded sparse coding. In *Advances in Neural Information Processing Systems*, 13100–13110.

Abo-Zahhad, M. M.; Hussein, A. I.; Mohamed, A. M.; et al. 2015. Compressive sensing algorithms for signal processing applications: A survey. *International journal of communications, network and system sciences*, 8(06): 197.

Amos, B. 2019. *Differentiable optimization-based modeling for machine learning*. Ph.D. thesis, PhD thesis. Carnegie Mellon University.

Balatsoukas-Stimming, A.; and Studer, C. 2019. Deep unfolding for communications systems: A survey and some new directions. In *2019 IEEE International Workshop on Signal Processing Systems (SiPS)*, 266–271. IEEE.

Blumensath, T.; and Davies, M. E. 2008. Iterative thresholding for sparse approximations. *Journal of Fourier analysis and Applications*, 14(5-6): 629–654.

Blumensath, T.; and Davies, M. E. 2009. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3): 265–274.

Borgerding, M.; Schniter, P.; and Rangan, S. 2017. AMP-inspired deep networks for sparse linear inverse problems. *IEEE Transactions on Signal Processing*, 65(16): 4293–4308.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018a. Neural ordinary differential equations. *Advances in neural information processing systems*, 31: 6571–6583.

Chen, X.; Liu, J.; Wang, Z.; and Yin, W. 2018b. Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. In *Advances in Neural Information Processing Systems*, 9061–9071.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2015. Gated feedback recurrent neural networks. In *International Conference on Machine Learning*, 2067–2075.

Cowen, B.; Saridena, A. N.; and Choromanska, A. 2019. LSALSA: accelerated source separation via learned sparse coding. *Machine Learning*, 108(8): 1307–1327.

Daubechies, I.; Defrise, M.; and De Mol, C. 2004. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11): 1413–1457.

Donoho, D. L.; Maleki, A.; and Montanari, A. 2009. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45): 18914–18919.

Efron, B.; Hastie, T.; Johnstone, I.; Tibshirani, R.; et al. 2004. Least angle regression. *The Annals of statistics*, 32(2): 407–499.

Giryes, R.; Eldar, Y. C.; Bronstein, A. M.; and Sapiro, G. 2018. Tradeoffs between convergence speed and reconstruction accuracy in inverse problems. *IEEE Transactions on Signal Processing*, 66(7): 1676–1690.

Gregor, K.; and LeCun, Y. 2010. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 399–406.

Haber, E.; and Ruthotto, L. 2017. Stable architectures for deep neural networks. *Inverse problems*, 34(1): 014004.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hershey, J. R.; Roux, J. L.; and Weninger, F. 2014. Deep unfolding: Model-based inspiration of novel deep architectures. *arXiv preprint arXiv:1409.2574*.

Hosseini, S. A. H.; Yaman, B.; Moeller, S.; Hong, M.; and Akçakaya, M. 2020. Dense recurrent neural networks for accelerated mri: History-cognizant unrolling of optimization algorithms. *IEEE Journal of Selected Topics in Signal Processing*, 14(6): 1280–1291.

Ito, D.; Takabe, S.; and Wadayama, T. 2019. Trainable ISTA for sparse signal recovery. *IEEE Transactions on Signal Processing*, 67(12): 3113–3125.

Kulkarni, K.; Lohit, S.; Turaga, P.; Kerviche, R.; and Ashok, A. 2016. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 449–458.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature*, 521(7553): 436–444.

Li, B.; Du, Q.; Zhou, T.; Zhou, S.; Zeng, X.; Xiao, T.; and Zhu, J. 2021a. ODE Transformer: An Ordinary Differential Equation-Inspired Model for Neural Machine Translation. *arXiv preprint arXiv:2104.02308*.

Li, C.; Yin, W.; Jiang, H.; and Zhang, Y. 2013. An efficient augmented Lagrangian method with applications to total variation minimization. *Computational Optimization and Applications*, 56(3): 507–530.

Li, Y.; Kong, L.; Shang, F.; Liu, Y.; Liu, H.; and Lin, Z. 2021b. Learned Extragradient ISTA with Interpretable Residual Structures for Sparse Coding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10): 8501–8509.

Liu, J.; Chen, X.; Wang, Z.; and Yin, W. 2019. Alista: Analytic weights are as good as learned weights in lista. In *Proceedings of the International Conference on Learning Representations*.

Lu, Y.; Zhong, A.; Li, Q.; and Dong, B. 2018. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *International Conference on Machine Learning*, 3276–3285. PMLR.

Mardani, M.; Sun, Q.; Vasawanala, S.; Papyan, V.; Monajemi, H.; Pauly, J.; and Donoho, D. 2018. Neural proximal gradient descent for compressive imaging. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 9596–9606.

Meng, T.; Chen, X.; Jiang, Y.; and Wang, Z. 2020. A Design Space Study for LISTA and Beyond. In *International Conference on Learning Representations*.

Metzler, C. A.; Maleki, A.; and Baraniuk, R. G. 2016. From denoising to compressed sensing. *IEEE Transactions on Information Theory*, 62(9): 5117–5144.

Moreau, T.; and Bruna, J. 2017. Understanding trainable sparse coding via matrix factorization. In *Proceedings of the International Conference on Learning Representations*.

Mousavi, A.; Patel, A. B.; and Baraniuk, R. G. 2015. A deep learning approach to structured signal recovery. In *2015 53rd annual allerton conference on communication, control, and computing (Allerton)*, 1336–1343. IEEE.

Nguyen, T. P.; Pauwels, E.; Richard, E.; and Suter, B. W. 2018. Extragradient method in optimization: Convergence and complexity. *Journal of Optimization Theory and Applications*, 176(1): 137–162.

Peng, X.; Tsang, I. W.; Zhou, J. T.; and Zhu, H. 2018. k-meansnet: When k-means meets differentiable programming. *arXiv preprint arXiv:1808.07292*.

Ruthotto, L.; and Haber, E. 2020. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision*, 62(3): 352–364.

Sander, M. E.; Ablin, P.; Blondel, M.; and Peyré, G. 2021. Momentum residual neural networks. *arXiv preprint arXiv:2102.07870*.

Sun, J.; Li, H.; Xu, Z.; et al. 2016. Deep ADMM-Net for compressive sensing MRI. In *Advances in Neural Information Processing Systems*, 10–18.

Takabe, S.; and Wadayama, T. 2020. Theoretical interpretation of learned step size in deep-unfolded gradient descent. *arXiv preprint arXiv:2001.05142*.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1): 267–288.

Wang, Z.; Ling, Q.; and Huang, T. S. 2016. Learning deep $\ell_0$ encoders. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Weinan, E. 2017. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1): 1–11.

Wu, K.; Guo, Y.; Li, Z.; and Zhang, C. 2020. SPARSE CODING WITH GATED LEARNED ISTA. In *Proceedings of the International Conference on Learning Representations*.

Xiang, J.; Dong, Y.; and Yang, Y. 2021. FISTA-Net: Learning A fast iterative shrinkage thresholding network for inverse problems in imaging. *IEEE Transactions on Medical Imaging*, 40(5): 1329–1339.

Xie, X.; Wu, J.; Zhong, Z.; Liu, G.; and Lin, Z. 2019. Differentiable linearized ADMM. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*.

Xin, B.; Wang, Y.; Gao, W.; Wipf, D.; and Wang, B. 2016. Maximal sparsity with deep networks? In *Advances in Neural Information Processing Systems*, 4340–4348.

Zhang, J.; and Ghanem, B. 2018. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1828–1837.

Zhang, K.; Zuo, W.; Gu, S.; and Zhang, L. 2017. Learning deep CNN denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3929–3938.

Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; and Torr, P. H. 2015. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, 1529–1537.