DEs-Inspired Accelerated Unfolded Linearized ADMM Networks for Inverse Problems

Weixin An, Yuanyuan Liu, *Member, IEEE*, Fanhua Shang, *Senior Member, IEEE*, Hongying Liu, *Senior Member, IEEE*, and Licheng Jiao, *Fellow, IEEE*

***Abstract*— Many research works have shown that the traditional alternating direction multiplier methods (ADMMs) can be better understood by continuous-time differential equations (DEs). On the other hand, many unfolded algorithms directly inherit the traditional iterations to build deep networks. Although they achieve superior practical performance and a faster convergence rate than traditional counterparts, there is a lack of clear insight into unfolded network structures. Thus, we attempt to explore the unfolded linearized ADMM (LADMM) from the perspective of DEs, and design more efficient unfolded networks. First, by proposing an unfolded Euler LADMM scheme and inspired by the trapezoid discretization, we design a new more accurate Trapezoid LADMM scheme. For the convenience of implementation, we provide its explicit version via a prediction–correction strategy. Then, to expand the representation space of unfolded networks, we design an accelerated variant of our Euler LADMM scheme, which can be interpreted as second-order DEs with stronger representation capabilities. To fully explore this representation space, we designed an accelerated Trapezoid LADMM scheme. To the best of our knowledge, this is the first work to explore a comprehensive connection with theoretical guarantees between unfolded ADMMs and first- (second-) order DEs. Finally, we instantiate our schemes as (A-)ELADMM and (A-)TLADMM with the proximal operators, and (A-)ELADMM-Net and (A-)TLADMM-Net with convolutional neural networks (CNNs). Extensive inverse problem experiments show that our Trapezoid LADMM schemes perform better than well-known methods.**

***Index Terms*— Acceleration, differential equations (DEs), inverse problems, network structure design, unfolded LADMM networks.**

I. Introduction

INVERSE problems arise in almost any scientific and engineering applications. In the field of machine learning, inverse problems [1] generally refer to the use of low-dimensional information and prior knowledge to restore higher-dimensional information, e.g., sparse coding [2], [3], compressive sensing (CS) [4], [5], and image super-resolution [6], [7]. Formally, we focus on the following model for a linear inverse problem:

$$\mathbf{A}\mathbf{x}^* + \mathbf{y} = \mathbf{b} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{m\times d}$, $\mathbf{b} \in \mathbb{R}^m$, and $d > m$ generally. For example, given an observation vector $\mathbf{b}$, where $\mathbf{A}$ is a dictionary matrix and $\mathbf{y}$ is noise, our goal is to recover an optimal $\mathbf{x}^*$ from $\mathbf{b}$ as accurately as possible. Traditionally, to obtain good estimates of the ground truth (GT), researchers usually solve the following model with an equality constraint:

$$\min_{\mathbf{x}\in\mathbb{R}^d, \mathbf{y}\in\mathbb{R}^m} f(\mathbf{x}) + g(\mathbf{y}), \quad \text{s.t. } \mathbf{A}\mathbf{x} + \mathbf{y} = \mathbf{b} \tag{2}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ and $g : \mathbb{R}^m \to \mathbb{R}$ are closed convex but possibly nonsmooth. In this article, we focus on the model (2) with constraint $\mathbf{A}\mathbf{x} + \mathbf{y} = \mathbf{b}$. For the more complicated case of $\mathbf{A}\mathbf{x} + \mathbf{C}\mathbf{y} = \mathbf{b}$ when $\mathbf{C} \neq \tau\mathbf{I}$, there are some similar algorithms for solving Problem (2). In the machine learning community, this general formulation has a broad spectrum of applications such as norm regularized ill-posed inverse problems, including but not limited to image denoising [8], [9], image inpainting [10], and CS [5], [11]. In this article, we propose novel algorithms that achieve better performance as shown in Fig. 1.

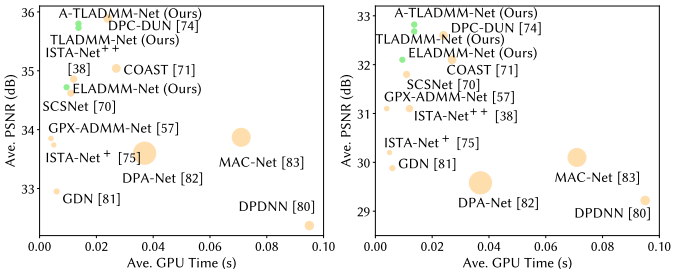


Fig. 1. Comparison of the reconstruction results for natural image CS with CS ratio $\gamma = 30\%$ on the test datasets, Set11 (left) and BSD68 (right). The size of the solid circles is proportional to the number of network parameters. Our methods (i.e., green solid circles) perform well overall, achieving faster speed, higher image quality, and lower storage costs.

Recently, diffusion models such as [12], [13], [14], and [15] have been developed for solving the inverse problems (1), but they are not based on the model (2). They have good interpretability and can generalize to different degradation operators such as [15], but their repeated sampling can lead to relatively low prediction efficiency. For solving the model (2), one popular method is the accelerated proximal gradient

Manuscript received 7 June 2023; revised 18 December 2023; accepted 15 March 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFF0906204 and in part by the National Natural Science Foundation of China under Grant 62276182 and Grant U22B2054. *(Corresponding authors: Yuanyuan Liu; Fanhua Shang; Hongying Liu.)*

Weixin An, Yuanyuan Liu, and Licheng Jiao are with the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an 710071, China (e-mail: weixinanut@163.com; yyliu@xidian.edu.cn; lchjiao@mail.xidian.edu.cn).

Fanhua Shang is with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: fhshang@tju.edu.cn).

Hongying Liu is with the Medical College, Tianjin University, Tianjin 300072, China (e-mail: hyliu2009@tju.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNNLS.2024.3382030, provided by the authors.

Digital Object Identifier 10.1109/TNNLS.2024.3382030

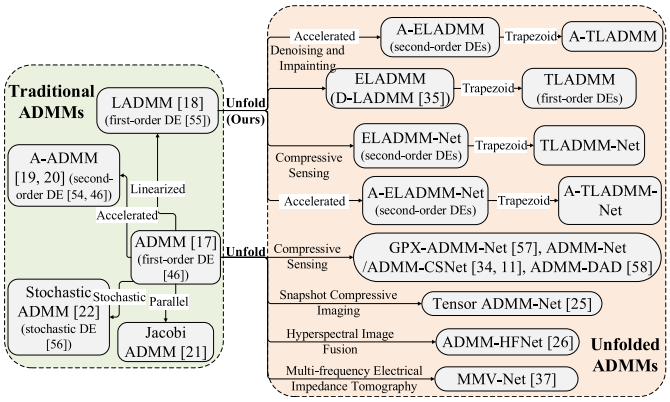


Fig. 2. There are descriptions of the connection between various ADMMs, some of which also demonstrate the connection between ADMMs and DEs. More discussion about this figure is in Section II.

(APG) algorithm [16] due to its $\mathcal{O}(1/K^2)$ convergence rate, where K is the number of iterations. The alternating direction multiplier method (ADMM) is another popular choice [17], which alternately minimizes an augmented Lagrangian function in a Gauss–Seidel fashion, followed by an update of the dual variable. However, ADMM involves the inverses of the matrices, which leads to its heavy computational complexity. To solve this problem, the work in [18] linearized the quadratic term of the augmented Lagrangian function and proposed the linearized ADMM (LADMM). The Nesterov momentum technology helps [19] and [20] to design accelerated ADMMs, which improves the convergence rate to $\mathcal{O}(1/K^2)$. Subsequently, the work in [21] proposed the Jacobian-type ADMM algorithm with multiple variable blocks to facilitate parallel and distributed computing. Besides, for solving the large-scale equality constraint problem (i.e., $f(\mathbf{x}) = (1/n)\sum_{i=1}^{n} f_i(\mathbf{x})$, where the number of samples n is large), there are some stochastic variants to reduce the complexities of deterministic ADMMs, including stochastic ADMM [22], SVRG-ADMM [23], and ASVRG-ADMM [24]. The connections between these traditional ADMMs are shown in Fig. 2. However, it is nontrivial for these traditional ADMMs to choose the values of hyper-parameters such as penalty parameters in practice and they usually require hundreds of iterations to converge, which cannot meet the needs of solving such problems quickly.

End-to-end training inspired recent works [25], [26], [27], [28] to unfold traditional optimization iterations into hierarchical deep neural networks (DNNs), which can learn some hyper-parameters from the distribution of the data instead of artificially setting them, called unfolded algorithms or learning-based optimization, which achieved better interpretability and performance when solving various inverse problems. Taking the iterative shrinkage thresholding algorithm (ISTA)-type unfolded network as an example, the work [29] proved the asymptotic coupling property of learned ISTA (LISTA) [2] and provided a rigorous linear convergence analysis, which reduced the number of learned parameters and improved the quality of sparse signal recovery. The works such as [30], [31], [32], and [33] further refined the network structure of LISTA and theoretical insights into LISTA. As for ADMM-type unfolded networks, the works such as [11], [34], [35], [36], and [37] proposed to replace the dictionary matrices with some learnable weights, or converted the proximal operator to a convolutional neural network (CNN), which achieved excellent performance in image restoration. Compared with DNNs, the work [38] verified that these unfolded algorithms often achieve better generalization capability, especially on small-scale datasets. Besides, because they are designed from the perspective of optimization, it is easier to interpret their principles than DNNs.

On the other hand, theoretical insights into unfolded algorithms become more difficult than traditional optimization algorithms owing to the introduction of learnable parameters. Many works such as [29] and [35] mainly added constraints on the learned parameters to simplify the theory, but it seems to limit the idea for designing unfolded algorithms, which prompts researchers to find a more reasonable perspective for understanding how unfolded networks work. Recently, by bridging ISTA-type algorithms and DEs, the works in [39] and [40] have made tentative progress in filling in this gap. For example, the work [40] converted one iteration of LISTA into a residual block and then treated it as the Euler discretization of a first-order differential equation.

Motivations: Nevertheless, for the widely used unfolded ADMM networks, there is still a lack of theoretical research from the perspective of DEs. To fill this gap, we need to answer the following two key questions.

1) Is there a close theoretically guaranteed comprehensive connection between unfolded ADMMs and numerical discretizations of the first- or second-order DEs?
2) Can more accurate numerical discretizations of DEs lead to superior unfolded ADMM networks?

Furthermore, we experimentally found that some existing ADMM-type unfolded algorithms cannot maintain good performance with the nondefault number of network layers. To sum up, there is an urgent need to design DEs-inspired unfolded ADMM networks with fewer parameters or layers.

A. Our Contributions

To address the above issues, we build a theoretically guaranteed comprehensive connection between the unfolded (A-)LADMM and numerical discretizations of the first- and second-order DEs. On this basis, we propose new DEs-inspired unfolded LADMM networks with better theoretical results and practical performance to solve Problem (2).

The main differences between our preliminary conference paper [41] and this article are as follows.

1) We briefly review the recent works on accelerated ADMMs and discuss the connections between traditional (accelerated) ADMMs and DEs in more detail in Section II.
2) Based on extrapolation acceleration, we generalize the DEs-inspired unfolded LADMM networks to be a more powerful network scheme in Section IV. The new scheme actually solves a system of second-order DEs, which extended design space for unfolded LADMM networks.
3) We further design a novel accelerated Trapezoid LADMM scheme based on the second-order DEs.
4) Finally, we show more experimental results, especially of accelerated LADMM schemes and several advanced methods to verify our analysis.

The main contributions are summarized as follows.

1) Firstly, we propose a new unfolded Euler LADMM scheme and derive it closely related to the Euler discretization for solving a system of first-order DEs, which opens the door for the DEs-inspired unfolded networks.

2) Then, we introduce the trapezoid discretization and design an implicit Trapezoid LADMM scheme with higher precision. Subsequently, we convert this implicit scheme into an iterable explicit version via the proposed prediction–correction strategy.
3) Moreover, we find that good performance of unfolded networks requires not only a high-precision discretization scheme but also a large representation space. To this end, we design an accelerated Euler LADMM scheme, which can be regarded as the second-order DEs with stronger representation capabilities. To fully explore the space represented by the second-order DEs, we further design an accelerated Trapezoid scheme. To the best of our knowledge, this is the first theoretically guaranteed work that analyzes the unfolded LADMM networks from the perspective of first- and second-order DEs.
4) For different inverse problems, we replace the nonlinear operators of the two nonaccelerated schemes with proximal operators or CNNs and design four specific nonaccelerated networks, named ELADMM, TLADMM, ELADMM-Net, and TLADMM-Net. Similarly, our accelerated LADMM schemes are instantiated as A-ELADMM, A-TLADMM, A-ELADMM-Net, and A-TLADMM-Net, respectively. The connection between our methods is also shown in Fig. 2.
5) Experimentally, we first perform an image denoising task, which verifies that our methods with fewer network layers outperform existing networks. Furthermore, our accelerated LADMM networks can find better solutions. Then, we evaluate the superiority of our methods over several ISTA-type networks and advanced diffusion models on image inpainting. Finally, we also perform various CS experiments to confirm the outstanding performance of our networks for generalized nonlinear transformation, where our A-TLADMM-Net can find better solutions than other methods.

The remainder of this article is organized as follows. Section II discusses recent advances in ADMM-type optimization algorithms. In Section III, we analyze the connection between unfolded LADMM networks and first-order DEs, and propose new unfolded Euler and Trapezoid LADMM schemes. Section IV discusses the connection between accelerated Euler LADMM scheme and second-order DEs, and presents an accelerated Trapezoid scheme. In Section V, we describe the details for training our networks and evaluations for various experiments. Conclusions are discussed in Section VI.

II. Related Works

Many works such as [42], [43], [44], and [45] explored the connection between traditional optimization algorithms and DEs. The researchers pointed out that many traditional optimization algorithms can be seen as numerical discretizations of DEs. Considering the gradient descent method clearly related to DEs, if we choose an infinitesimal step size to minimize the objective function $f(\mathbf{x})$, then the gradient descent method can be regarded as the Euler discretization of the following first-order differential equation:

$$\dot{\mathbf{X}} = -\nabla f(\mathbf{X}) \tag{3}$$

where $\dot{\mathbf{X}} \equiv (\mathrm{d}\mathbf{X}/\mathrm{d}t)$ and $\mathbf{X} = \mathbf{X}(t)$ is the continuous limit of $\mathbf{x}_k$. This finding helps researchers analyze optimization algorithms from a new perspective of DEs.

A. ISTA-Type Unfolded Networks and Euler Discretization

We first introduce LISTA as the numerical discretization of DEs in solving problems without equality constraints. Considering LISTA: $\mathbf{x}_{k+1} = \phi(\mathbf{x}_k, \mathbf{b}; \Theta_k) \triangleq \mathrm{ST}(\mathbf{W}_1^k\mathbf{b} + \mathbf{W}_2^k\mathbf{x}_k, \theta_k)$, where $\Theta_k = \{\mathbf{W}_1^k, \mathbf{W}_2^k, \theta_k\}$ are learnable parameters, and $\mathrm{ST}(\cdot,\cdot)$ denotes a soft-thresholding function, i.e., $\mathrm{ST}(a,\theta) = \mathrm{sign}(a)\max\{0, |a| - \theta\}$. Sander et al. [40] pointed out that one iteration of LISTA can be regarded as a residual block with a residual function $r(\mathbf{x}, \mathbf{b}; \Theta) = \phi(\mathbf{x}, \mathbf{b}; \Theta) - \mathbf{x}$. Thus, the kth layer structure of LISTA can be expressed as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + r(\mathbf{x}_k, \mathbf{b}; \Theta_k) \tag{4}$$

where $r(\mathbf{x}_k, \mathbf{b}; \Theta_k)$ generalizes the expression of $-\nabla f(\mathbf{x}_k)$. Then (4) can be seen as the Euler discretization for solving the first-order differential equation (3). Based on the above analysis, the ISTA-type methods have made great progress with the help of differential equation theory. But many machine learning problems such as Problem (2) are usually constrained, and the ISTA-type methods cannot solve such problems. Thus, the ADMM-type algorithms need to be further studied with the help of differential equation theory.

B. Traditional (Extrapolation Acceleration) ADMMs as DEs

The connection between traditional optimization algorithms and DEs is bidirectional. First, from the discrete to continuous limit, i.e., what kind of differential equations (DEs) does the optimization algorithm solve? On the contrary, how to construct a discretization with faster convergence speed for solving Problem (2) from the differential equation theory? Along these two lines of thought, the work [46] proved that the continuous limit of ADMM is consistent with a first-order differential equation. He et al. [47], [48] emphasized the converse and proposed two new accelerated algorithms with the convergence rate matching the corresponding dynamic system.

The extrapolation acceleration technology, i.e., $\widetilde{\mathbf{x}}_k = \mathbf{x}_k + \rho(\mathbf{x}_k - \mathbf{x}_{k-1})$ with an extrapolation parameter ρ, has been recognized as an effective acceleration method in the optimization field [49], [50], [51], [52]. Recently, this extrapolation step has shown outstanding advantages for solving complex nonconvex minimax problems [53], not just equality constraint problems. The relationship between this technology and second-order differential dynamical systems has been studied in the traditional ADMM algorithms [46], [50], [54]. For example, the works [46], [54] proved that the second-order dynamical systems are the continuity limit of traditional accelerated ADMMs, and clarified that the solution of the dynamical system converges weakly to the minimum of the objective function. Besides, the work [50] proposed a generalized second-order dynamical system with time-dependent damping terms for solving Problem (2).

For other popular ADMM variants, there are also some works such as [55] and [56] exploring their connection with differential dynamical systems. Reviewing the iterations of

LADMM [18]

$$\begin{aligned}\mathbf{x}_{k+1} &= \text{Prox}_{\frac{f}{L_1}}\left\{\mathbf{x}_k - \tfrac{1}{L_1}\mathbf{A}^\top(\boldsymbol{\lambda}_k + \beta(\mathbf{A}\mathbf{x}_k + \mathbf{y}_k - \mathbf{b}))\right\}\\ \mathbf{y}_{k+1} &= \text{Prox}_{\frac{g}{L_2}}\left\{\mathbf{y}_k - \tfrac{1}{L_2}(\boldsymbol{\lambda}_k + \beta(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{y}_k - \mathbf{b}))\right\}\\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \beta(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{y}_{k+1} - \mathbf{b})\end{aligned} \tag{5}$$

where Prox denotes the proximal operator,[1] β is a penalty parameter, $\boldsymbol{\lambda}$ is a Lagrange multiplier, and L_1, $L_2 > 0$ are Lipschitz constants, the work [55] analyzed the convergence behavior of LADMM based on differential inclusions in the case of nonsmooth f and g. For the large-scale case, the work [56] applied the stochastic modified equation to study stochastic ADMMs, revealing important insights into their convergence behavior. These works are also shown in Fig. 2. However, to the best of our knowledge, there has been no work on designing LADMM- or SADMM-type unfolded networks by constructing discretizations of certain dynamic systems.

C. Unfolded ADMMs

Many well-known works show that the unfolded ADMMs perform much better than traditional ADMMs. Yang et al. [11], [34] reported that ADMM-CSNet achieves surprising experimental performance for the first time by rewriting the ADMM program as a learnable network for CS magnetic resonance imaging (MRI). Hu et al. [57] proposed GPX-ADMM-Net with the generalized proximal mapping and the work [37] presented the MMV-Net, which further improve the performance for image reconstruction. Kouni et al. [58] constructed a Deep Analysis Decoding network (ADMM-DAD), which achieves better performance than ISTA-type unfolded networks on speech datasets. As for LADMM, by introducing learnable matrices, the work [35] proposed the Differentiable Linearized ADMM (D-LADMM) to solve the equality constraint problems and analyzed its linear convergence. The connection between traditional ADMMs and these unfolded ADMMs is also shown in Fig. 2.

However, the connection between unfolded ADMMs or their accelerated variants and DEs remains a secret. The above facts motivate us to investigate the comprehensive connection between the unfolded ADMM-type network structures and the numerical discretizations of DEs. Not least, we hope that this comprehensive connection can inspire the design of novel and more efficient ADMM-type unfolded networks.

D. Trapezoid Discretization

As we all know, the Euler discretization uses the rectangular formula to roughly approximate the integral, while the trapezoid discretization approximates the integral of the function $f(\mathbf{x})$ with higher precision. In the field of unfolded algorithms, we extend the trapezoid discretization as the following paradigm:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{h}{2}\left[f(\mathbf{x}_k, \boldsymbol{\Theta}_k) + f(\mathbf{x}_{k+1}, \boldsymbol{\Theta}_k)\right] \tag{6}$$

where h is a DE-stepsize and $\boldsymbol{\Theta}_k$ is a learnable parameter set. Since $\mathbf{x}_{k+1}$ exists on both sides of the (6), it needs to be executed iteratively by the prediction-correction strategy.

[1]The proximal operator of the function f is defined as $\text{Prox}_{fa}(x) = \arg\min_z\{(a/2)\|z - x\|^2 + f(z)\}$.

Notation: In this article, the norm $\|\cdot\|$ denotes the ℓ_2-norm of the vector, and $\|\cdot\|_1$ denotes the ℓ_1-norm of the vector, i.e., $\|\mathbf{x}\|_1 = \sum_i |x_i|$. $\nabla f(\cdot)$ denotes the gradient of differentiable function $f(\cdot)$ and $\partial f(\cdot)$ denotes the subgradient of nondifferentiable but Lipschitz continuous function $f(\cdot)$. The L-smoothness assumption of the function f implies that $\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \le L\|\mathbf{x}_1 - \mathbf{x}_2\|$.

III. Unfolded LADMM Schemes

Existing works studied the connection between ISTA-type unfolded networks and DEs to solve unconstrained problems, but the DE analysis theory of existing works only is limited to the traditional ADMM algorithms. In this section, we further elucidate how unfolded ADMMs solve equality constraint problems through discretizations of the first-order DEs. We observe that D-LADMM [35] can be considered as the Euler discretization for solving a system of first-order DEs. Following this observation, we focus on the case from the continuous limit to discrete and construct a new scheme inspired by trapezoid discretization. The novel scheme inherits the advantages of trapezoid discretization and achieves better theoretical results.

A. Unfolded Euler LADMM Scheme as the First-Order DEs

Along the idea of unfolded algorithms, we first unfold the iterations of LADMM into a unified structure with DE-stepsize h, called the unfolded Euler LADMM scheme

$$\begin{cases}\mathbf{x}_{k+1} = \mathcal{F}_f\left(\mathbf{x}_k + \frac{h\beta_k}{\theta_k}F_k(\mathbf{x}_k)\right)\\ \mathbf{y}_{k+1} = \mathcal{G}_g\left(\mathbf{y}_k + \frac{h}{\eta_k}G_k(\mathbf{y}_k)\right)\\ \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + h\beta_k(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{y}_{k+1} - \mathbf{b})\end{cases} \tag{7}$$

where $F_k(\mathbf{x}) = -\mathbf{W}_k^\top((\boldsymbol{\lambda}_k/\beta_k) + \mathbf{A}\mathbf{x} + \mathbf{y}_k - \mathbf{b})$, $G_k(\mathbf{y}) = -((\boldsymbol{\lambda}_k/\beta_k) + \mathbf{A}\mathbf{x}_{k+1} + \mathbf{y} - \mathbf{b})$, and $\{\mathbf{W}_k, \theta_k, \eta_k, h, \beta_k\}$ are optimized by end-to-end training, while $\mathbf{W}_k \equiv \mathbf{A}$ in traditional ADMM [46]. It is especially worth noting that $\mathcal{F}_f$ and $\mathcal{G}_g$ are but not limited to nonlinear operators depending on the problem model. For example, when they are proximal operators or multiplication transformations, (7) degenerates into ELADMM and further D-LADMM [35] with DE-stepsize h=1. When they are generalized nonlinear operators such as CNNs, we call (7) ELADMM-Net. Fig. 2 also clearly shows these relationships. As for theoretical insight, Lemma 1 provides a novel first-order DEs perspective to interpret such a scheme.

Lemma 1 (Unfolded Euler LADMM Scheme as the first-Order DEs): The optimal trajectory function is defined as $\mathcal{X}(t) = (\mathbf{X}(t)^\top, \mathbf{Y}(t)^\top, \boldsymbol{\Lambda}(t)^\top)^\top$. We assume that the functions f and g are closed convex but possibly nonsmooth, $\mathcal{F}_f$ and $\mathcal{G}_g$ are both proximal operators and $\mathbf{A}$ has full column rank. Then, the continuous limit associated with the updates in (7) with timescale $t = kh$ corresponds to a system of first-order DEs

$$\dot{\mathcal{X}} = \mathscr{F}(\mathcal{X}), \quad \text{with } \mathcal{X}(0) = \begin{pmatrix}\mathbf{x}_0\\ \mathbf{y}_0\\ \boldsymbol{\lambda}_0\end{pmatrix} \tag{8}$$

where

$$\dot{\mathcal{X}} = \begin{pmatrix}\dot{\mathbf{X}}\\ \dot{\mathbf{Y}}\\ \dot{\boldsymbol{\lambda}}\end{pmatrix}, \mathscr{F}(\mathcal{X}) = \begin{pmatrix}\frac{1}{\theta(t)}\left(F(\mathbf{X}) - \nabla f_{\mu_1}(\mathbf{X})\right)\\ \frac{1}{\eta(t)}\left(G(\mathbf{Y}) - \nabla g_{\mu_2}(\mathbf{Y})\right)\\ \beta(t)(\mathbf{A}\mathbf{X} + \mathbf{Y} - \mathbf{b})\end{pmatrix}$$

and $f_{\mu_1}(\cdot)$ and $g_{\mu_2}(\cdot)$ are the Moreau-Yosida approximations[2] of $f(\cdot)$ and $g(\cdot)$, respectively. When there is no confusion, we ignore the dependence on t and denote $\mathbf{X}(t)$, $\mathbf{Y}(t)$ and $\mathbf{\Lambda}(t)$, as $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{\Lambda}$, respectively.

Proof: By the definitions of $F_k(\mathbf{x})$ and $G_k(\mathbf{y})$, and the proximal operators $\mathcal{F}_f(x) \triangleq \text{Prox}_{f(\theta_k/h)}(x)$ and $\mathcal{G}_g(y) \triangleq \text{Prox}_{g(\eta_k/h)}(y)$, our Euler LADMM scheme (7) can be reformulated as

$$\begin{cases} \mathbf{x}_{k+1} &= \underset{\mathbf{x}}{\text{argmin}}\Big\{f(\mathbf{x}) + \frac{\theta_k}{2h}\big\|\mathbf{x}-\mathbf{x}_k \\ &\quad + \frac{h\beta_k}{\theta_k}\mathbf{W}_k^\top\Big(\frac{\boldsymbol{\lambda}_k}{\beta_k} + \mathbf{A}\mathbf{x}_k + \mathbf{y}_k - \mathbf{b}\Big)\big\|^2\Big\} \\ \mathbf{y}_{k+1} &= \underset{\mathbf{y}}{\text{argmin}}\Big\{g(\mathbf{y}) + \frac{\eta_k}{2h}\big\|\mathbf{y}-\mathbf{y}_k \\ &\quad + \frac{h}{\eta_k}\Big(\frac{\boldsymbol{\lambda}_k}{\beta_k} + \mathbf{A}\mathbf{x}_{k+1} + \mathbf{y}_k - \mathbf{b}\Big)\big\|^2\Big\} \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + h\beta_k(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{y}_{k+1} - \mathbf{b}). \end{cases} \tag{9}$$

Since f and g are convex functions, and $\mathbf{A}$ has full column rank, the optimization subproblems in (9) are strongly convex such that $(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \boldsymbol{\lambda}_{k+1})$ is unique. From the optimality conditions for (9), we can obtain the following inclusions:

$$\begin{cases} 0 &\in \partial f(\mathbf{x}_{k+1}) + \frac{\theta_k}{h}(\mathbf{x}_{k+1}-\mathbf{x}_k \\ &\quad + \frac{h\beta_k}{\theta_k}\mathbf{W}_k^\top\Big(\frac{\boldsymbol{\lambda}_k}{\beta_k} + \mathbf{A}\mathbf{x}_k + \mathbf{y}_k - \mathbf{b}\Big)\Big) \\ 0 &\in \partial g(\mathbf{y}_{k+1}) + \frac{\eta_k}{h}(\mathbf{y}_{k+1}-\mathbf{y}_k \\ &\quad + \frac{h}{\eta_k}\Big(\frac{\boldsymbol{\lambda}_k}{\beta_k} + \mathbf{A}\mathbf{x}_{k+1} + \mathbf{y}_k - \mathbf{b}\Big)\Big) \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + h\beta_k(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{y}_{k+1} - \mathbf{b}). \end{cases} \tag{10}$$

We consider the first inclusion

$$0 \in \partial f(\mathbf{x}_{k+1}) + \theta_k\frac{\mathbf{x}_{k+1}-\mathbf{x}_k}{h} + \mathbf{W}_k^\top(\boldsymbol{\lambda}_k + \beta_k(\mathbf{A}\mathbf{x}_k + \mathbf{y}_k - \mathbf{b})). \tag{11}$$

Let $t = hk$ and $\mathbf{x}_k = \mathbf{X}(t)$, with the similar notation for $\mathbf{y}_k$ and $\boldsymbol{\lambda}_k$ in the limit $h \to 0$. By using the Mean Value Theorem on the ith component of $\mathbf{x}_{k+1}$ and $\mathbf{y}_{k+1}$, we have that $\mathbf{x}_{k+1,i} = \mathbf{X}_i(t+h) = \mathbf{X}_i(t) + h\dot{\mathbf{X}}_i(t+\xi_i h)$ and $\mathbf{y}_{k+1,i} = \mathbf{Y}_i(t+h) = \mathbf{Y}_i(t) + h\dot{\mathbf{Y}}_i(t+\xi_i h)$ for some $\xi_i \in [0,1]$. Thus, $\lim_{h\to 0}((\mathbf{x}_{k+1,i} - \mathbf{x}_{k,i})/h) = \lim_{h\to 0}\dot{\mathbf{X}}_i(t+\xi_i h) = \dot{\mathbf{X}}_i(t)$, $\lim_{h\to 0}((\mathbf{y}_{k+1,i} - \mathbf{y}_{k,i})/h) = \lim_{h\to 0}\dot{\mathbf{Y}}_i(t+\xi_i h) = \dot{\mathbf{Y}}_i(t)$, and $\lim_{h\to 0}\mathbf{x}_{k+1} = \mathbf{X}(t)$. We define $\theta(t)$ and $\beta(t)$, and let $\beta_k \to \beta(t)$, $\theta_k \to \theta(t)$ in the limit $h \to 0$. Then (11) can be written as the following differential inclusion:

$$0 \in \partial f(\mathbf{X}(t)) + \theta(t)\dot{\mathbf{X}}(t) - F(\mathbf{X}(t)) \tag{12}$$

where $F(\mathbf{X}) = -\mathbf{W}(t)^\top(\mathbf{\Lambda}(t) + \beta(t)(\mathbf{A}\mathbf{X}(t) + \mathbf{Y}(t) - \mathbf{b}))$. $\mathbf{W}(t)$, $\beta(t)$ and $\theta(t)$ generalize the constant coefficients in the continuous limit of the traditional ADMM algorithm (e.g., the first-order differential equation [46]). And by the same logic, we get

$$0 \in \partial g(\mathbf{Y}(t)) + \eta(t)\dot{\mathbf{Y}}(t) - G(\mathbf{Y}(t)) \tag{13}$$

where $G(\mathbf{Y}) = -((\mathbf{\Lambda}(t)/\beta(t)) + \mathbf{A}\mathbf{X}(t) + \mathbf{Y}(t) - \mathbf{b})$. Next, let us consider the third equation in (9), $\mathbf{\Lambda}(t+h) - \mathbf{\Lambda}(t) - h\beta(t)(\mathbf{A}\mathbf{X}_i(t+h) + \mathbf{Y}_i(t+h) - \mathbf{b}) = 0 \xrightarrow{h\to 0} \dot{\mathbf{\Lambda}}(t) - \beta(t)(\mathbf{A}\mathbf{X}(t) + \mathbf{Y}(t) - \mathbf{b}) = 0$. Finally, since (12) and (13) are first-order differential inclusions, we set the initial conditions $\mathbf{X}(0) = \mathbf{x}_0$, $\mathbf{Y}(0) = \mathbf{y}_0$ and $\mathbf{\Lambda}(0) = \mathbf{\Lambda}_0$, where $\mathbf{x}_0$ and $\mathbf{y}_0$ are initial solution estimations of Problem (2). Then we can obtain a differential inclusion system

$$\begin{cases} 0 \in \partial f(\mathbf{X}(t)) + \theta(t)\dot{\mathbf{X}}(t) - F(\mathbf{X}(t)) \\ 0 \in \partial g(\mathbf{Y}(t)) + \eta(t)\dot{\mathbf{Y}}(t) - G(\mathbf{Y}(t)) \\ \dot{\mathbf{\Lambda}}(t) - \beta(t)(\mathbf{A}\mathbf{X}(t) + \mathbf{Y}(t) - \mathbf{b}) = 0 \end{cases} \tag{14}$$

which generalizes the differential inclusion system in [59], where $\mathbf{W}(t) \equiv \mathbf{A}$. Following [55], we consider the Moreau–Yosida approximations $f_{\mu_1}(\mathbf{x})$ and $g_{\mu_2}(\mathbf{y})$ of functions $f(\mathbf{x})$ and $g(\mathbf{y})$ with $\mu_1, \mu_2 > 0$. Then, we can obtain a system of first-order approximating DEs as shown in (8). □

Lemma 1 indicates that the sequence $\{\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}_k\}$ generated by Euler LADMM (7) is as close as possible to the trajectory modeled by the first-order DEs (8). That is, the Euler LADMM scheme for solving Problem (2) can be considered as the Euler discretization for solving the first-order DEs (8) with the initial conditions $\mathcal{X}_0 = \mathcal{X}(0)$, which actually answers the 1st key question in our motivations. Taking D-LADMM as an example will make it clearer. D-LADMM can be seen as solving such a system of first-order DEs with $h \equiv 1$, while our Euler LADMM scheme is more flexible in choosing DE-stepsize h, so it usually performs better than D-LADMM. Overall, the connection with theoretical guarantees between unfolded LADMM networks and the first-order DEs has been clarified.

B. Trapezoid LADMM Scheme as the First-Order DEs

We have explained the connection between the unfolded Euler LADMM scheme and the system of first-order DEs, which broadens our horizon for designing unfolded LADMM networks. To further improve the precision of the Euler scheme (7) and explore the structural diversity of unfolded LADMM networks, we propose a new discretization scheme, called unfolded Trapezoid LADMM scheme, to solve Problem (2).

1) Implicit Trapezoid LADMM Scheme as DEs: By introducing the trapezoid discretization into the updates of $\mathbf{x}$ and $\mathbf{y}$, we propose a new network structure, called the implicit Trapezoid LADMM scheme

$$\begin{cases} \mathbf{x}_{k+1} = \mathcal{F}_f\Big(\mathbf{x}_k + \frac{h\beta_k}{2\theta_k}(F_k(\mathbf{x}_k) + F_k(\mathbf{x}_{k+1}))\Big) \\ \mathbf{y}_{k+1} = \mathcal{G}_g\Big(\mathbf{y}_k + \frac{h}{2\eta_k}(G_k(\mathbf{y}_k) + G_k(\mathbf{y}_{k+1}))\Big) \\ \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + h\beta_k(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{y}_{k+1} - \mathbf{b}). \end{cases} \tag{15}$$

Regarding the principle, the implicit Trapezoid LADMM scheme (15) is still recast as the DEs (8) by Lemma 2.

Lemma 2 (Unfolded Implicit Trapezoid LADMM Scheme as the First-Order DEs): Based on the same assumptions as in Lemma 1, the continuous limit associated with the implicit Trapezoid LADMM scheme (15) also corresponds to the system of first-order DEs (8).

Lemma 2 shows that the same first-order DEs (8) can also be derived by the implicit Trapezoid scheme (15), and its proof can be found in the Supplementary Material. It can be known from the work [60, Ths. 15.1–15.5] that the existence and uniqueness of the solution of DEs (8) can be guaranteed under Lipschitz continuous conditions. Thus, we can further compare the accuracy of our two schemes. Conversely, if our Euler and Trapezoid LADMM schemes can not correspond to solve the same system of first-order DEs, then our Trapezoid LADMM

[2]The Moreau-Yosida approximation of a convex function f is defined as $f_\mu(x) := \inf_z\{f(z) + (1/2\mu)\|z-x\|^2\}$. For any $\mu > 0$, the function f_μ is a convex and continuously differentiable [59].

scheme may not produce an unfolded network with a faster convergence rate despite its higher precision.

2) Advantages of Trapezoid LADMM Scheme Over Euler LADMM Scheme: We directly give theoretical results for the accuracy of our two schemes.

Theorem 1: Suppose that: 1) f and g are L_f-smooth and L_g-smooth, respectively and 2) $\mathcal{F}_f$ and $\mathcal{G}_g$ are nonexpansive mappings. Then the local and global error bounds of the implicit Trapezoid scheme (15) are $\mathcal{O}(h^3)$ and $\mathcal{O}(h^2)$, while those of the Euler scheme (7) are $\mathcal{O}(h^2)$ and $\mathcal{O}(h)$, respectively.

In fact, when functions f and g are simple nonsmooth functions, e.g., ℓ_1-norm, Theorem 1 still holds. Specifically, there are two cases to be discussed.

1) Define the set $A = \{i | x_{1i}x_{2i} \neq 0, i = 1, \ldots, d\}$, we can choose a constant $c_1 > 0$ such that $\|\nabla f(\mathbf{x}_{1a}) - \nabla f(\mathbf{x}_{2a})\| \leq c_1\|\mathbf{x}_{1a} - \mathbf{x}_{2a}\|$, and further $\|\nabla f_{\mu_1}(\mathbf{x}_{1a}) - \nabla f_{\mu_1}(\mathbf{x}_{2a})\| \leq c_1\|\mathbf{x}_{1a} - \mathbf{x}_{2a}\|$ for $\forall a \in A$.
2) Define a set $B = \{j | x_{1j}x_{2j} = 0, j = 1, \ldots, d\}$, without loss generality, we assume $x_{1j} > 0$ and $x_{2j} = 0$, i.e., considering the nondifferentiable case, Moreau-Yosida approximation allows us to choose a constant c_2 such that $\|\nabla f_{\mu_1}(\mathbf{x}_{1b}) - \nabla f_{\mu_1}(\mathbf{x}_{2b})\| \leq c_2\|\mathbf{x}_{1b} - \mathbf{x}_{2b}\|$ for $\forall b \in B$.

Thus, there exists a constant $L_{f_{\mu_1}} = \max\{c_1, c_2\}$ such that $\|\nabla f_{\mu_1}(\mathbf{x}_1) - \nabla f_{\mu_1}(\mathbf{x}_2)\| \leq L_{f_{\mu_1}}\|\mathbf{x}_1 - \mathbf{x}_2\|$ as well as g, which plays an important role in the analysis of Theorem 1.

Theorem 1 shows that the implicit Trapezoid LADMM scheme can obtain lower error bounds than the Euler scheme (7). Therefore, the implicit Trapezoid LADMM scheme can generate update points closer to the optimal trajectory $\mathcal{X}(t)$ than the Euler LADMM scheme, thereby reducing the deviation from the optimal trajectory, which actually answers the 2nd key question in our motivations.

3) Practical Solution-Explicit Trapezoid LADMM Scheme: Considering the implementation of our implicit scheme (15), there is $\mathbf{x}_{k+1}$ or $\mathbf{y}_{k+1}$ on both sides of the iterations, which requires solving such two equations, respectively. This is computationally intractable, especially when dealing with complicated nonlinear operators $\mathcal{F}_f$ and $\mathcal{G}_g$. To deal with this difficulty, we propose a more feasible scheme called the explicit Trapezoid LADMM scheme through a prediction–correction strategy, as shown in "**Case 1**" of Algorithm 1. Compared with relevant unfolded algorithms, there are two main differences in Algorithm 1 as follows.

(i) Main Idea of Prediction–Correction Strategy: About the update of $\mathbf{x}$ in (15), $\mathbf{x}_{k+1}$ can be seen as a fixed point. Then we give the following fixed point iteration to solve it:

$$\mathbf{x}_{k+1}^{i+1} = \mathcal{F}_f\left(\mathbf{x}_k + \frac{h\beta_k}{2\theta_k}\left(F_k(\mathbf{x}_k) + F_k\left(\mathbf{x}_{k+1}^i\right)\right)\right). \tag{16}$$

With an initial point $\mathbf{x}_{k+1}^0$, the sequence $\{\mathbf{x}_{k+1}^i\}$ produced by (16) can converge to the solution $\mathbf{x}_{k+1}$. To trade off the quality of the solution and the computational cost, we design an efficient prediction step to initialize $\mathbf{x}_{k+1}^0$ and correct the initial point $\mathbf{x}_{k+1}^0$ once to approximate (16) in lines 11 and 12 of Algorithm 1. Similarly, the prediction–correction strategy of variable $\mathbf{y}$ is given in lines 13 and 14 of Algorithm 1. Moreover, $\mathcal{F}_f$ and $\mathcal{G}_g$ still vary in different problem models.

▷ For the general structural regularizers, $\mathcal{F}_f$ and $\mathcal{G}_g$ are generalized nonlinear operators and the explicit Trapezoid scheme is instantiated as TLADMM-Net for solving applications such as CS inverse problems.

Algorithm 1 Explicit (Extrapolation Acceleration) Trapezoid LADMM Schemes

Input: hyper-parameter α, matrix $\mathbf{A}$, the layer numbers K, and the training dataset $\mathcal{D} = \{(\mathbf{b}_i, \mathbf{x}_i^*, \mathbf{y}_i^*)\}_{i=1}^N$.

Initialize: $\mathcal{X}_0$, $\mathbf{\Theta} = \{\mathbf{W}_k, h, \theta_k, \eta_k, \beta_k\}_{k=0}^{K-1}$.

Inference:

1: Choose a minibatch of observations $\mathbf{b}$ of size N_b from $\mathcal{D}$;
2: **for** $k = 0, 1, \ldots, K-1$ **do**
3: **Case 1:** Nonaccelerated Explicit Trapezoid LADMM
4: $\bar{\mathbf{x}}_k = \mathbf{x}_k$, $\bar{\mathbf{y}}_k = \mathbf{y}_k$, $\bar{\boldsymbol{\lambda}}_k = \boldsymbol{\lambda}_k$;
5: $a_k = \frac{\theta_k}{h}$, $c_k = \frac{\eta_k}{h}$, $e_k = h\beta_k$;
6: **Case 2:** Accelerated Explicit Trapezoid LADMM
7: $\widetilde{\mathbf{x}}_k = \mathbf{x}_k + \frac{1}{h\theta_{k+1}}(\mathbf{x}_k - \mathbf{x}_{k-1})$, $\bar{\mathbf{x}}_k = \widetilde{\mathbf{x}}_k$;
8: $\widetilde{\mathbf{y}}_k = \mathbf{y}_k + \frac{1}{h\eta_{k+1}}(\mathbf{y}_k - \mathbf{y}_{k-1})$, $\bar{\mathbf{y}}_k = \widetilde{\mathbf{y}}_k$;
9: $\widetilde{\boldsymbol{\lambda}}_k = \boldsymbol{\lambda}_k + \frac{\beta_k}{\beta_k + h}(\boldsymbol{\lambda}_k - \boldsymbol{\lambda}_{k-1})$, $\bar{\boldsymbol{\lambda}}_k = \widetilde{\boldsymbol{\lambda}}_k$;
10: $a_k = \frac{1+h\theta_k}{h^2}$, $c_k = \frac{1+h\eta_k}{h^2}$, $e_k = \frac{h^2\beta_k}{\beta_k + h}$;
11: $\mathbf{x}_{k+1}^0 = \mathcal{F}_f(\bar{\mathbf{x}}_k + \frac{\beta_k}{a_k}F_k(\bar{\mathbf{x}}_k))$; //Prediction
12: $\mathbf{x}_{k+1} = \mathcal{F}_f(\bar{\mathbf{x}}_k + \frac{\beta_k}{2a_k}[F_k(\bar{\mathbf{x}}_k) + F_k(\mathbf{x}_{k+1}^0)])$; //Correction
13: $\mathbf{y}_{k+1}^0 = \mathcal{G}_g(\bar{\mathbf{y}}_k + \frac{1}{c_k}G_k(\bar{\mathbf{y}}_k))$; //Prediction
14: $\mathbf{y}_{k+1} = \mathcal{G}_g(\bar{\mathbf{y}}_k + \frac{1}{2c_k}[G_k(\bar{\mathbf{y}}_k) + G_k(\mathbf{y}_{k+1}^0)])$; //Correction
15: $\boldsymbol{\lambda}_{k+1} = \bar{\boldsymbol{\lambda}}_k + e_k(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{y}_{k+1} - \mathbf{b})$;
16: **end for**

Training:

17: **if** the ground truth $\mathbf{x}_{N_b}^*, \mathbf{y}_{N_b}^*$ of observation $\mathbf{b}$ exists **then**
18: $Loss_1 = \min_{\mathbf{\Theta}} \frac{1}{N_b}\sum_{k=1}^K \frac{k}{\sum k}(\|\mathbf{x}_k - \mathbf{x}_{N_b}^*\|^2 + \|\mathbf{y}_k - \mathbf{y}_{N_b}^*\|^2)$;
19: **else**
20: $Loss_2 = \min_{\mathbf{\Theta}} \frac{1}{N_b}\sum_{k=1}^K \frac{k}{\sum k}(f(\mathbf{x}_k) + g(\mathbf{b} - \mathbf{A}\mathbf{x}_k))$;
21: **end if**

Output: $\mathcal{M}(\mathbf{A}, \mathcal{D}; \mathcal{X}_0; \mathbf{\Theta}) = (\mathbf{x}_K, \mathbf{y}_K)$.

▷ For ℓ_2-norm or "simple" norm constraint, in the sense that the proximal operator has a closed-form solution [49], these two nonlinear operators only involve multiplication transformations and matrix inversions or soft-thresholding, and the explicit Trapezoid scheme degenerates to TLADMM.

(ii) Network Training: Given a matrix $\mathbf{A}$ and a training dataset $\mathcal{D} = \{(\mathbf{b}_i, \mathbf{x}_i^*, \mathbf{y}_i^*)\}_{i=1}^N$, our explicit Trapezoid scheme aims to reduce the discrepancy between $(\mathbf{x}_i^*, \mathbf{y}_i^*)$ and the output $(\mathbf{x}_K, \mathbf{y}_K)$ by optimizing the parameters $\mathbf{\Theta}$. Thus, we adopt end-to-end training and design the weighted multilayer loss, i.e., Loss_1, as the training loss function, which alleviates the vanishing gradient problem and ensures better reconstruction performance. When there is no GT $(\mathbf{x}^*, \mathbf{y}^*)$, we utilize the model objective, i.e., Loss_2, as the training loss function. Note that the learned parameters $\{\theta_k, \eta_k, \beta_k, h\} > 0$ are scalars rather than matrices or vectors, which reduces redundancy especially when training on small-scale datasets. Here we keep all learnable parameters $\mathbf{\Theta}$ for our analysis.

The structure of the kth inference stage of our explicit Trapezoid scheme is shown in Fig. 3, where nonlinear operators are visualized through simple CNN modules for generality. Compared with the plug-and-play methods [62], our schemes are trained end-to-end, without relying on any pretrained network. Since the use of the forecast-correction strategy, the variables $\mathbf{x}_k$, $\mathbf{y}_k$ and $\boldsymbol{\lambda}_k$ are first updated by a forward prediction

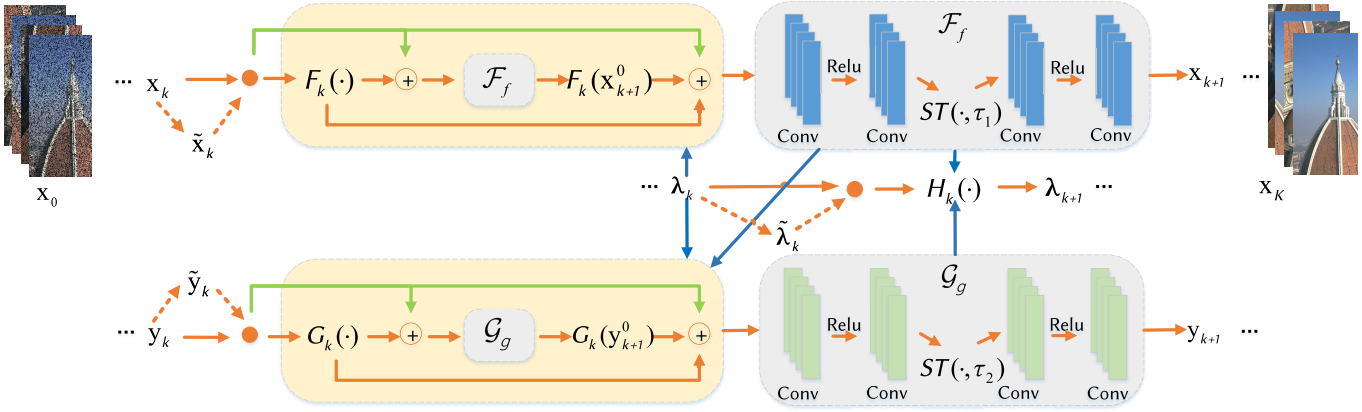


Fig. 3. Network structure of the kth inference stage of Algorithm 1. We generalize the network architecture in conference version [41]. Note that $H_k(\boldsymbol{\lambda}_k) = \tilde{\boldsymbol{\lambda}}_k + e_k(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{y}_{k+1} - \mathbf{b})$. The $(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}_k)$ replaces ● as the unfolded Trapezoid LADMM scheme and the $(\widetilde{\mathbf{x}}_k, \widetilde{\mathbf{y}}_k, \widetilde{\boldsymbol{\lambda}}_k)$ replaces ● as the accelerated unfolded Trapezoid LADMM scheme. The orange line denotes the independent update of each variable, the blue line denotes information interaction between variables, and the green line denotes skip connections, just like the identity mapping of ResNet [61].

step, followed by a further correction step, then the well-trained explicit Trapezoid scheme $\mathcal{M}(\mathbf{A}, \mathcal{D}; \mathcal{X}_0; \boldsymbol{\Theta})$ can be viewed as a traditional optimizer with better parameters $\boldsymbol{\Theta}$ with the help of training data. In fact, we have proved that there exists a set of learnable parameters for our Trapezoid LADMM scheme to generate converged solutions.

IV. Accelerated Unfolded LADMM Schemes

According to the analysis above, high-precision trapezoid discretization of first-order DEs can improve the performance of unfolded networks. While higher-precision numerical methods (e.g., the Runge–Kutta method [63]) can potentially yield improved unfolded LADMM networks, the first-order DEs (8) may impose a limitation on their upper performance limit. Thus, not only the high-precision discretization but also the function space represented by DEs influence the performance of the unfolded LADMM networks. To overcome this limitation, a question arises naturally: how can we establish the connection between unfolded LADMM networks and higher-order (e.g., second-order) DEs, thereby expanding the representation space? To answer this question, we discuss the behavior of extrapolation acceleration technology in LADMM networks and devise stronger accelerated LADMM schemes.

A. Our Accelerated Unfolded Euler LADMM Scheme

We design an accelerated unfolded Euler LADMM scheme by incorporating the extrapolation acceleration technology with different extrapolation parameters ρ for $\mathbf{x}$, $\mathbf{y}$, and $\boldsymbol{\lambda}$ as follows:

$$\begin{cases} \widetilde{\mathbf{x}}_k = \mathbf{x}_k + \frac{1}{h\theta_k+1}(\mathbf{x}_k - \mathbf{x}_{k-1}) \\ \mathbf{x}_{k+1} = \mathcal{F}_f\left(\widetilde{\mathbf{x}}_k + \frac{h^2\beta_k}{1+h\theta_k}F_k(\widetilde{\mathbf{x}}_k)\right) \\ \widetilde{\mathbf{y}}_k = \mathbf{y}_k + \frac{1}{h\eta_k+1}(\mathbf{y}_k - \mathbf{y}_{k-1}) \\ \mathbf{y}_{k+1} = \mathcal{G}_g\left(\widetilde{\mathbf{y}}_k + \frac{h^2}{1+h\eta_k}G_k(\widetilde{\mathbf{y}}_k)\right) \\ \widetilde{\boldsymbol{\lambda}}_k = \boldsymbol{\lambda}_k + \frac{\beta_k}{\beta_k+h}(\boldsymbol{\lambda}_k - \boldsymbol{\lambda}_{k-1}) \\ \boldsymbol{\lambda}_{k+1} = \widetilde{\boldsymbol{\lambda}}_k + \frac{h^2\beta_k}{\beta_k+h}(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{y}_{k+1} - \mathbf{b}). \end{cases} \tag{17}$$

Compared with the nonaccelerated Euler scheme (7), three extrapolation sequences $\{\widetilde{\mathbf{x}}_k\}$, $\{\widetilde{\mathbf{y}}_k\}$ and $\{\widetilde{\boldsymbol{\lambda}}_k\}$ are added, which play an important role in the derivation of second-order DEs. Similarly, (17) can be instantiated as A-ELADMM and A-ELADMM-Net. The clear connection between them can be also found in Fig. 2. About theoretical insights, Lemma 3 means that this accelerated scheme can be regarded as a system of second-order DEs in the continuous limit.

Lemma 3 (Accelerated Unfolded Euler LADMM Scheme as the Second-Order DEs): Based on the same assumptions as in Lemma 1, the continuous limit associated with the updates in (17) corresponds to the system of second-order DEs

$$\boldsymbol{\epsilon}\cdot\ddot{\mathcal{X}} + \dot{\mathcal{X}} = \mathscr{F}(\mathcal{X}), \mathcal{X}(0) = \begin{pmatrix}\mathbf{x}_0\\ \mathbf{y}_0\\ \boldsymbol{\lambda}_0\end{pmatrix}, \dot{\mathcal{X}}(0) = \begin{pmatrix}\mathbf{u}_0\\ \mathbf{v}_0\\ \mathbf{w}_0\end{pmatrix} \tag{18}$$

where $\boldsymbol{\epsilon} = ((1/\theta(t)), (1/\eta(t)), \beta(t))^\top$, $\ddot{\mathcal{X}} = (\ddot{\mathbf{X}}, \ddot{\mathbf{Y}}, \ddot{\boldsymbol{\lambda}})^\top$, where $\theta_k \to \theta(t)$, $\eta_k \to \eta(t)$ in the limit $h \to 0$. Note that this system of DEs generalizes the dynamic [50], where $(1/\theta(t)) = (1/\eta(t)) = \beta(t)$.

Proof: In a similar way, our accelerated Euler LADMM scheme (17) can be rewritten as one minimizing problem, which has the following first-order optimality conditions with proximal parameters $((1+h\theta_k)/h^2)$ and $((1+h\eta_k)/h^2)$:

$$\begin{cases} 0 \in \partial f(\mathbf{x}_{k+1}) + \frac{1+h\theta_k}{h^2}\left(\mathbf{x}_{k+1} - \widetilde{\mathbf{x}}_k - \frac{h^2\beta_k}{1+h\theta_k}F_k(\widetilde{\mathbf{x}}_k)\right) \\ 0 \in \partial g(\mathbf{y}_{k+1}) + \frac{1+h\eta_k}{h^2}\left(\mathbf{y}_{k+1} - \widetilde{\mathbf{y}}_k - \frac{h^2}{1+h\eta_k}G_k(\widetilde{\mathbf{y}}_k)\right) \\ \boldsymbol{\lambda}_{k+1} = \widetilde{\boldsymbol{\lambda}}_k + \frac{h^2\beta_k}{\beta_k+h}(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{y}_{k+1} - \mathbf{b}). \end{cases} \tag{19}$$

Consider Taylor's Theorem for the ith component of $\mathbf{x}_{k\pm1}$

$$\begin{aligned} \mathbf{x}_{k\pm1,i} &= \mathbf{X}_i(t \pm h) \\ &= \mathbf{X}_i(t) \pm h\dot{\mathbf{X}}_i(t) + \frac{1}{2}h^2\ddot{\mathbf{X}}_i(t \pm \xi_i^{\pm}h) \end{aligned} \tag{20}$$

for some $\xi_i^{\pm} \in [0, 1]$. Hence, we have

$$\begin{aligned} \mathbf{x}_{k+1,i} - \widetilde{\mathbf{x}}_{k,i} &= \mathbf{x}_{k+1,i} - \mathbf{x}_{k,i} - \frac{1}{1+h\theta_k}(\mathbf{x}_{k,i} - \mathbf{x}_{k-1,i}) \\ &= \frac{h^2\theta_k}{1+h\theta_k}\dot{\mathbf{X}}_i(t) + \frac{1}{2}h^2\ddot{\mathbf{X}}_i(t + \xi_i^{+}h) \\ &\quad + \frac{h^2}{2(1+h\theta_k)}\ddot{\mathbf{X}}_i(t - \xi_i^{-}h). \end{aligned} \tag{21}$$

Substituting (21) into (19) and letting $h \to 0$, we obtain

$$0 \in \ddot{\mathbf{X}}(t) + \partial f(\mathbf{X}(t)) + \theta(t)\dot{\mathbf{X}}(t) - F(\mathbf{X}(t)). \tag{22}$$

Similarly, $0 \in \ddot{\mathbf{Y}}(t) + \partial g(\mathbf{Y}(t)) + \eta(t)\dot{\mathbf{Y}}(t) - G(\mathbf{Y}(t))$ can be also obtained. About $\boldsymbol{\lambda}$, we can obtain $\beta(t)\dot{\boldsymbol{\Lambda}}(t) + \dot{\boldsymbol{\Lambda}}(t) - \beta(t)(\mathbf{A}\mathbf{X}(t) + \mathbf{Y}(t) - \mathbf{b}) = 0$. We also consider the Moreau-Yosida approximations $f_{\mu_1}(\mathbf{x})$ and $g_{\mu_2}(\mathbf{y})$ of objectives $f(\mathbf{x})$ and $g(\mathbf{y})$ with $\mu_1, \mu_2 > 0$. Then the accelerated Euler LADMM scheme (17) corresponds to solve the system of second-order approximating DEs (18). □

Furthermore, we turn to discuss the representation capabilities of our LADMM schemes in the continuous limit.

1) Representation Capabilities: According to [40, Proposition 3], we know that the first-order DEs (8) is not a universal approximator, while our accelerated unfolded Euler LADMM scheme is treated as second-order DEs (18) in the continuous limit, so it can represent a strictly larger class of functions. In other words, our accelerated unfolded Euler LADMM scheme can search the solution trajectory in a larger function space than nonaccelerated schemes. Thus, our accelerated Euler scheme can break the barrier caused by the first-order DEs (8) and generate better solutions than nonaccelerated schemes.

2) High-Level Idea: The first- and second-order DEs can be interpreted as continuous equivalent formulations of unfolded LADMM schemes and the accelerated unfolded Euler LADMM scheme, respectively. More convincingly, we can further intuitively understand this fact like Neural ODEs [64]. In fact, the variable at time T, i.e., $\mathcal{X}(T)$, corresponds to the output of our schemes. An analogy of DEs to our (accelerated) unfolded LADMM schemes can make it more explicit. Our LADMM schemes map an input $(\mathbf{x}_0, \mathbf{y}_0, \boldsymbol{\lambda}_0)$ to some output $(\mathbf{x}_K, \mathbf{y}_K, \boldsymbol{\lambda}_K)$ through a forward propagation of the schemes. We then adjust the weights (e.g., θ_k, η_k and proximal parameters) by backward propagation to match $(\mathbf{x}_K, \mathbf{y}_K, \boldsymbol{\lambda}_K)$ with $(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*)$. From the perspective of DEs, they map the initial values $(\mathbf{x}_0, \mathbf{y}_0, \boldsymbol{\lambda}_0)$ to $(\mathbf{x}_K, \mathbf{y}_K, \boldsymbol{\lambda}_K)$ by solving the DEs starting from this initial conditions. We then adjust the dynamics (encoded by $\mathscr{F}$) such that the DEs transforms $(\mathbf{x}_0, \mathbf{y}_0, \boldsymbol{\lambda}_0)$ into an output $(\mathbf{x}_K, \mathbf{y}_K, \boldsymbol{\lambda}_K)$ that is as close as possible to $(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*)$. The above analysis also explains that more precise numerical discretizations for solving DEs usually lead to more efficient unfolded LADMM networks.

B. Accelerated Unfolded Trapezoid LADMM Scheme

According to representation capabilities, the second-order DEs can represent a larger class of functions. Specifically, for any sequence $\{\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}_k\}$ in the first-order DEs solution space, there is at least one sequence in the solution space represented by the second-order DEs with a similar optimization performance and even better. However, our scheme (17) sometimes fails to find this ideal sequence. For example, as shown in Fig. 4, although the scheme (17) is superior to the nonaccelerated Euler scheme, it can not achieve better performance than the nonaccelerated Trapezoid LADMM scheme, thus there is still a gap between the sequences $\{\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}_k\}$ generated by (17) and optimal trajectory. Therefore, good performance requires not only a larger function space but also an unfolded network discretization scheme with higher accuracy, which motivates us to further improve the scheme (17). In other words, we need to design a new scheme to solve second-order DEs (18) better. Following the idea of the Trapezoid LADMM scheme (15), we design an accelerated unfolded Trapezoid scheme, which is implicit. Due to space limitations, it can be found in the Supplementary Material.

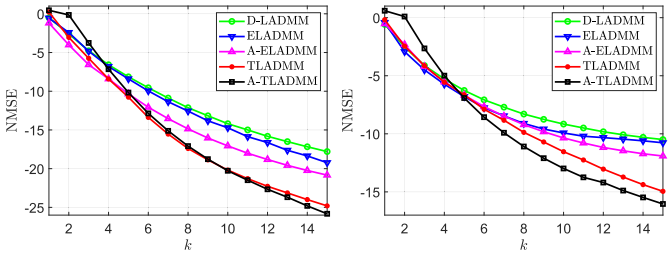


Fig. 4. Comparison of NMSE results at sampling probability $p = 0.08$ (left) and $p = 0.1$ (right) for unfolded LADMM networks.

Subsequently, we provide a more effective explicit version through the proposed prediction–correction strategy, of which the procedure can be found in "**Case 2**" of Algorithm 1 and the kth block structure is also shown in Fig. 3. Here, we added dotted lines to denote extrapolation acceleration steps. In the next section, extensive experimental results show that this explicit scheme can find solutions closer to the optimal trajectory and obtain better performance.

V. Experiments

We perform extensive inverse problem experiments to verify the feasibility and effectiveness of our schemes. In Sections V-A–V-C, we test our schemes by instantiating $\mathcal{F}_f$ and $\mathcal{G}_g$ as multiple transformation or soft-thresholding operators, and we name our methods as (A-)ELADMM and (A-)TLADMM, respectively. In Sections V-D–V-F, we simulate generalized nonlinear operators $\mathcal{F}_f$ and $\mathcal{G}_g$ by using simple CNNs for solving the CS inverse problems, and we name them as (A-)ELADMM-Net and (A-)TLADMM-Net, respectively. For fair comparison, we set K as the number of layers for matching the compared methods instead of the default. Our source codes are available: https://github.com/Weixin-An/A-TLADMM-Net. Due to the page limit, more experimental results are provided in the Supplementary Material.

A. Simulation Experiments

We first evaluate our methods on synthetic datasets. We consider the specific equality constraint model

$$\min_{\mathbf{x}\in\mathbb{R}^{500}, \mathbf{y}\in\mathbb{R}^{250}} \alpha\|\mathbf{x}\|_1 + \|\mathbf{y}\|_1, \quad \text{s.t. } \mathbf{A}\mathbf{x} + \mathbf{y} = \mathbf{b} \tag{23}$$

where $\mathbf{x}$ is the data to be restored, $\mathbf{y}$ is the noise to be removed, α is a hyper-parameter to balance the restoration results and denoising performance, and $\mathbf{b} \in \mathbb{R}^m$ is an observation vector. Each entry in the dictionary matrix $\mathbf{A}$ is sampled from independent and identical Gaussian distribution, namely $A_{i,j} \sim \mathcal{N}(0, 1/250)$, and then we normalize its columns to have ℓ_2 unit norm. We apply the Bernoulli sampling with probabilities $p = 0.08$ and 0.1 to the variables $\mathbf{x}$ and $\mathbf{y}$. In this problem, both nonlinear operators $\mathcal{F}_f$ and $\mathcal{G}_g$ become soft-thresholding operators.

We set the number of training and testing samples to 10 000 and 1000, respectively. We train our four methods and

D-LADMM with the layer numbers $K=15$ by using the SGD algorithm optimizing Loss_1. For testing, we choose normalized mean square error (NMSE $= 10\log_{10}((\|\mathbf{x}_k-\mathbf{x}^*\|^2/\|\mathbf{x}^*\|^2)+(\|\mathbf{y}_k-\mathbf{y}^*\|^2/\|\mathbf{y}^*\|^2))$) to evaluate the performance of all the networks. The experimental results are shown in Fig. 4.

Discussion on Convergence Rate: Fig. 4 shows that our ELADMM achieves a similar linear decay rate as D-LADMM [35] and it converges faster than D-LADMM. Moreover, we have proved in our conference version [41] that our Trapezoid LADMM scheme achieves a similar linear decay rate to D-LADMM, and Fig. 4 also verifies this conclusion. As for the accelerated LADMM schemes, they achieve better convergence performance than the linearly convergent ELADMM, which also provides support for their consistent linear decay rates with our nonaccelerated LADMM schemes.

As shown by the curves in Fig. 4, we also observed that our ELADMM converges slightly faster than D-LADMM because ELADMM is more flexible in the choice of h. TLADMM outperforms significantly ELADMM and D-LADMM, which confirms our conclusions in Theorem 1. Besides, A-TLADMM performs distinctively, which shows that it has found a better optimization trajectory in a larger function space. Note that ISTA-type unfolded algorithms can not solve this problem. This simulation experiment verifies that the Trapezoid-guided schemes can effectively improve convergence speed and further the accuracy of data recovery, and our accelerated networks can find better optimization trajectories.

B. Natural Image Denoising

We further evaluate the denoising performance of our methods on natural images and verify the stability of our methods with fewer network layers. The Waterloo BragZone Greyscale (WBZG) and FFHQ 256 × 256-1k [65] datasets are used for testing and $r\%$ salt-and-pepper noise is added to each dataset. $\mathbf{b}$ contains 10 000 and 1024 noisy image blocks with size 16 × 16 in the training and testing sets for our methods, respectively. We use the patch-dictionary method [66] to initialize dictionary matrix $\mathbf{A}$ in (23). We evaluate denoising performance by using peak signal-to-noise ratio (PSNR) as shown in Tables I and II, where we implemented the source code of D-LADMM and MPRNet [9] as baselines. For practicality, we consider the case of unknown GT and use the loss function (24) to train our methods

$$\text{Loss}_2 = \min_{\boldsymbol{\Theta}} \frac{1}{N_b}\sum_{k=1}^{K}\frac{k}{\sum k}(\|\mathbf{A}\mathbf{x}_k-\mathbf{b}\|_1+\alpha\|\mathbf{x}_k\|_1). \tag{24}$$

From Table I and the results in the Supplementary Material, it can be seen that our TLADMM improves the denoising performance by about 2.2/2.3 dB over ELADMM/D-LADMM, which is because our TLADMM corresponds to the Trapezoid discretization and can generate points closer to the optimal trajectory of $\mathbf{x}$- and $\mathbf{y}$-subproblems in each iteration, while D-LADMM and ELADMM only correspond to the Euler discretization with lower precision. Our A-TLADMM further improves the denoising result of our TLADMM by 0.4 dB, which also confirms our intuition. From Tables I and II, and Fig. 3, both our TLADMM and A-TLADMM still outperform D-LADMM at almost the same time cost despite their two additional correction steps. In short, our (non)accelerated Trapezoid LADMM schemes can improve denoising performance with the limited number of network layers. These all confirm that more accurate numerical discretizations can lead to better unfolded LADMM networks.

TABLE I

COMPARISON OF THE PSNR (DB) RESULTS ON 12 IMAGES IN THE WBZG DATASETS AT SALT-AND-PEPPER NOISE RATES 5%, 10%, AND 15%. THE BEST, SECOND BEST, AND THIRD BEST RESULTS ARE HIGHLIGHTED IN RED, BLUE, AND GREEN, RESPECTIVELY

Algorithms	5%	10%	15%	Ave.	Time(s)
D-LADMM ($K=15$ [35])	33.53	31.97	29.70	31.73	0.2748
A-ELADMM ($K=15$, Ours)	35.40	33.32	30.79	33.17	0.2841
TLADMM ($K=15$, Ours)	36.39	34.04	31.67	34.03	0.5218
A-TLADMM ($K=8$, Ours)	36.01	33.54	31.91	33.82	0.2836
A-TLADMM ($K=15$, Ours)	36.72	34.21	32.30	34.41	0.5429

TABLE II

COMPARISON OF THE PSNR RESULTS ON THE FFHQ 256×256-1K DATASET AT SALT-AND-PEPPER NOISE RATIOS 5%, 10% AND 15%

Algorithms	5%	10%	15%	#Params (Millions)
MPRNet [9]	27.03	24.27	22.65	20.1
D-LADMM [35]	34.37	32.01	28.77	2.09
TLADMM (Ours)	36.45	33.25	30.35	1.97
A-TLADMM (Ours)	37.02	33.61	29.62	1.97

TABLE III

COMPARISON OF DENOISING RESULTS WITH DIFFERENT K ON WATERLOO BRAGZONE GREYSCALE SET2 AT 10% SALT-AND-PEPPER NOISE

Ave. PSNR / Layers / Algorithms	$K=9$	$K=12$	$K=15$	$K=18$
Original D-LADMM [35]	31.10	31.64	31.97	31.69
D-LADMM, $Loss_2$ (24)	31.53	31.93	32.01	32.23
TLADMM, loss of D-LADMM	33.48	33.67	33.79	33.90
TLADMM, $Loss_2$ (24)	33.63	33.86	34.04	34.11

Ablation Study: To assess how much our loss Loss_2 and trapezoid structure each contribute, we performed the following ablation study. We replace the training loss function of original D-LADMM [35] and TLADMM with Loss_2 and the loss of D-LADMM, respectively. Setting different layer numbers K, we train the networks on these four cases with $\alpha = 0.2$. Table III summarizes the PSNR results on the dataset "WBZG set2." It can be found that only changing the loss function will also improve the performance of 0.4 dB of D-LADMM. However, such improvement is far less significant than that of our TLADMM with the loss of D-LADMM (2.0 dB). Moreover, comparing the TLADMM with the loss of D-LADMM and our Loss_2, TLADMM with Loss_2 can improve the denoising result by 0.3 dB on average. Thus, taking the objective function as the training loss can impose strict constraints on the training procedure, which can be regarded as a substitute for no GT. Moreover, it is also verified that the trapezoid structure plays a more important role than our loss functions.

C. Natural Image Inpainting

Image inpainting is one of the most typical ill-posed inverse problems. We also perform our (A-)ELADMM and (A-)TLADMM to solve this task, which will indicate advantages over advanced methods and identify that the trapezoid

TABLE IV
COMPARISON OF NATURAL IMAGE INPAINTING RESULTS IN TERMS OF PSNR (DB) ON THE SET11 DATASET AT PIXEL MISSING RATIO 50%

Algorithms	Barb	Boat	C.man	House	Monarch	Lena	Flinstones	Peppers	Foreman	Finger	Parrot	Ave. (dB)
LFISTA (ICLR2019, [67], [68])	26.13	30.02	27.24	32.55	28.17	32.65	26.53	28.84	30.65	28.35	31.03	29.29
GLISTA (ICLR2019, [31])	25.52	28.76	25.85	31.04	26.07	31.18	24.32	27.64	29.24	26.35	29.44	27.76
D-LADMM (ICML2019, [35])	26.51	30.91	27.44	34.11	28.82	34.31	27.04	29.84	31.84	29.95	32.44	30.29
ELISTA (AAAI2021, [32])	26.75	30.55	27.68	32.84	26.45	32.94	24.40	29.64	30.85	28.52	30.95	29.23
ELADMM (Ours)	26.54	30.84	27.48	34.15	28.99	34.55	27.11	29.75	31.95	29.85	32.62	30.35
A-ELADMM (Ours)	28.36	31.86	27.65	33.90	29.03	31.31	27.45	29.87	35.57	29.66	31.24	30.53
TLADMM (Ours)	27.08	31.58	27.94	34.75	29.45	35.08	27.69	30.09	32.45	30.38	33.61	30.91
A-TLADMM (Ours)	30.00	32.86	28.35	35.26	30.02	32.36	28.01	30.75	36.80	30.11	33.09	31.59

Fig. 5. Comparison of visual and PSNR (dB) results for natural image inpainting at pixel missing ratio 50% on Cameraman (top) and Barbara (bottom) images. From left to right are the GT, corrupted image, and the recovery results of LFISTA [67], GLISTA [31], ELISTA [32], D-LADMM [35], ELADMM (Ours), A-ELADMM (Ours), TLADMM (Ours), and A-TLADMM (Ours).

discretization and extrapolation acceleration technology can enhance the performance of unfolded LADMM networks in different problem models. Assuming images corrupted by a mask $\mathbf{M}$ with a missing pixel ratio $r\%$, the image inpainting problem can be expressed as

$$\min_{\mathbf{x},\mathbf{y}} \frac{1}{2}\|\mathbf{y}\|^2 + \alpha\|\mathbf{x}\|_1, \quad \text{s.t. } \mathbf{MDx} + \mathbf{y} = \mathbf{b} \tag{25}$$

where $\mathbf{b}$ are corrupted image patches, α balances the noise and sparsity, and the dictionary matrix $\mathbf{D}$ is generated by training clean images. In this problem, $\mathcal{F}_f$ becomes a soft-thresholding operator and $\mathcal{G}_g$ becomes a multiple transformation $(\beta_k/(1+\beta_k))\mathcal{I}(\cdot)$. By the end-to-end training, the learned linearized optimizer $\mathcal{M}(\mathbf{A}, \mathcal{D}; \mathcal{X}_0; \boldsymbol{\Theta})$ has a natural advantage, i.e., preventing $\mathcal{O}(d^3)$ computational complexity caused by matrix inversion and matrix–matrix multiplication in existing ADMM-type unfolded algorithms.

We divide the images in the BSD500 dataset into 16×16 size image blocks and randomly select $N = 50\,000$ blocks for training with a batch size of 256. Based on the source code of LFISTA [68], we implement our methods and other compared algorithms by ourselves. The dataset Set11 is used to evaluate the performance of all the methods. For a fair comparison, we set the same layer numbers $K = 20$ in all the methods and train them by using our Loss_1. All numerical results with $r = 50$ are listed in Table IV. We further choose PSNR and Normalized Root Mean Square Error (NRMSE $= (1/|\mathcal{S}|)\sum_{(\mathbf{b},\mathbf{x}^*)\in\mathcal{S}}(\|\mathcal{M}(\mathbf{A}, \mathbf{b}; \mathbf{x}_0; \boldsymbol{\Theta}) - \mathbf{x}^*\|/\|\mathbf{x}^*\|)$) to evaluate test performance at different missing ratios, where $\mathcal{S}$ contains the pairs of testing data. The inpainting visual results are shown in Fig. 5 and the inpainting results at different missing ratios (e.g., 30%, 40%, 50%, 60%, and 70%) are shown in Fig. 6.

From Table IV, we observe that our (A-)ELADMM and (A-)TLADMM perform consistently much better than the

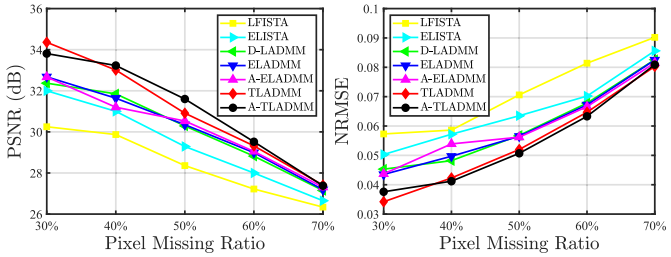


Fig. 6. Natural image inpainting results in terms of PSNR and NRMSE with missing ratios 30%, 40%, 50%, 60% and 70% on the Set11 dataset.

ISTA-type networks, including LFISTA, GLISTA, and ELISTA on all images. Besides, our A-TLADMM and A-ELADMM find better solutions than their nonaccelerated counterparts. As we can see, the average PSNR results of TLADMM and A-TLADMM are about 0.6 and 1.1 dB higher than those of ELADMM and A-ELADMM, respectively, which implies that the trapezoid strategy can also improve the acceleration LADMM networks. Fig. 5 displays the inpainting effect for all the networks on Barbara and Cameraman (abbreviated as C.man) images, where our A-TLADMM can recover higher quality images than other methods. For a more comprehensive comparison, Fig. 6 indicates that our Trapezoid LADMM schemes consistently outperform the compared methods, even at high pixel missing ratios, and the lower the pixel missing ratio, the more obvious the improvement.

Comparison With Advanced Methods: To provide a more encompassing perspective on the practical utility of our methods, we compare them with the recent MMES [69] and the diffusion model DPS [12] on the natural image inpainting task. Note that the DPS method depends on a pretrained denoising network. We test our above-trained TLADMM, A-TLADMM networks, the MMES method and the DPS model on the color FFHQ 256×256.1k dataset [65] and the results of PSNR and

TABLE V
COMPARISON OF THE PSNR AND LPIPS RESULTS AND GPU TIME (S) FOR INFERENCE ON THE COLOR FFHQ 256 × 256.1K DATASET AT PIXEL RANDOM MISSING RATIOS $\gamma = 30\%, 50\%$ AND 70%

Algorithms	30%		50%		70%		GPU
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	Time
DPS [12]	32.74	0.159	31.46	0.172	29.57	0.192	122.7
MMES [69]	35.97	0.093	34.19	0.120	31.51	0.173	116.6
TLADMM	37.78	0.061	34.29	0.105	30.67	0.201	48.19
A-TLADMM	36.82	0.062	34.74	0.093	30.68	0.189	49.11

learned perceptual image patch similarity (LPIPS) are shown in Table V. It is clear that our TLADMM and A-TLADMM almost all outperform the diffusion model DPS and the MMES method at different pixel-missing ratios.

D. CS for Natural Image

Here, we expand our methods to solve the CS inverse problem and show the advantages of our schemes in the case of generalized nonlinear operators $\mathcal{F}_f$ and $\mathcal{G}_g$. By introducing an auxiliary variable $\mathbf{y}$, the CS inverse problem can be formulated as

$$\min_{\mathbf{x},\mathbf{y}} \frac{1}{2}\|\mathbf{c} - \mathbf{\Phi}\mathbf{x}\|^2 + \alpha\|\mathbf{\Psi}\mathbf{y}\|_1, \quad \text{s.t. } \mathbf{x} = \mathbf{y} \tag{26}$$

where the CS measurement of $\mathbf{x}$ is denoted by $\mathbf{c}$, $\mathbf{x}$ is the vectorized image, $\mathbf{\Phi} \in \mathbb{R}^{p\times d}$ is an under-sampling matrix, and $\mathbf{\Psi}$ is the transformation matrix of filtering operations such as Discrete Wavelet Transform or Discrete Cosine Transform. In this experiment, $\mathbf{\Psi}$ is replaced by a simple nonlinear transformation $\mathcal{T}(\cdot)$ as in [75] to maintain the sparsity of natural images. Thus, the nonlinear operators $\mathcal{F}_f(\mathbf{z})$ and $\mathcal{G}_g(\cdot)$ become $(\mathbf{\Phi}^\top\mathbf{\Phi} + \beta_k\mathbf{I})^{-1}(\mathbf{z} + \mathbf{c})$ and $\widetilde{\mathcal{T}}(ST(\mathcal{T}(\cdot)))$, respectively, where $\widetilde{\mathcal{T}}(\cdot)$ is the inverse transformation of $\mathcal{T}(\cdot)$ such that $\widetilde{\mathcal{T}} \circ \mathcal{T} = \mathcal{I}$. In fact, we have also successfully avoided such matrix inversions by linearizing quadratic terms in the Supplementary Material, whereas existing ADMM-type unfolded networks require computing this inversion. Thus, the computational complexity and time cost of our networks can be further reduced.

Datasets: We train our unfolded LADMM networks on the same Train400 dataset as in [38], which contains a large number of various scenarios. Its distribution of different scenes is more balanced than Train91 [76] and DIV2K [77]. As for testing, we utilize the classical BSD68 [78], Set11 [76], and ImageNet-1k [79] datasets to evaluate reconstruction performance.

For training, we initialize $\mathbf{\Phi}$ as an orthogonal column matrix, and $\mathbf{x}_0 = \mathbf{\Phi}^\top\mathbf{c}$ as well as $\mathbf{y}_0$. For a fair comparison, we set $K = 20$ for the compared algorithms. Then, by using the Adam optimizer, we train all the methods to 400 epochs with a batch size of 64. We also performed a simple ablation experiment in the Supplementary Material to select loss functions for compared algorithms, which shows that their original losses can always lead to better performance than our Loss_1. Thus, we compared the results of their source code here.

In Fig. 7, the successive output visual images in layers 1–10 by A-TLADMM-Net with $K = 10$ are increasingly clearer. Compared algorithms include ISTA-Net$^+$ [75], DPDNN [80], GDN [81], SCSNet [70], DPA-Net [82], MAC-Net [83], COAST [71], ISTA-Net^{++} [38], HSSE [72], LGSR [73], and DPC-DUN [74]. Due to space limitations, some results are shown in Tables VI and VII, and more detailed results can be found in the Supplementary Material. As we can see, if we train our A-TLADMM-Net with $K = 10$, at least 0.2/0.9 dB improvement can be achieved on the Set11/BSD68 datasets compared with existing methods. If we train our A-TLADMM-Net with $K = 20$, the reconstruction performance will be further improved, which fully confirms the feasibility and efficiency of our accelerated Trapezoid LADMM scheme. From Tables VI and VII, our Trapezoid-inspired LADMM networks consistently outperform the Euler-inspired LADMM networks. By comparing ELADMM-Net with A-ELADMM-Net or TLADMM-Net with A-TLADMM-Net, it is clear that the extrapolation acceleration technology also improves the performance of the LADMM networks, mainly because the acceleration methods can search for better solution trajectories in a larger function space. For a more intuitive comparison with several well-known networks, Fig. 1 shows the results of all the methods for the number of network parameters, time cost, and PSNR at CS ratio $\gamma = 30\%$ in more detail.

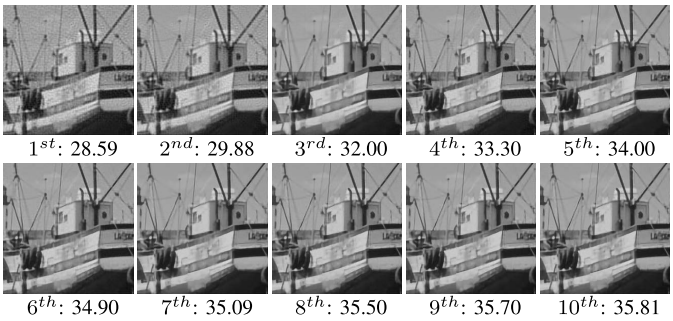


Fig. 7. Visualization and PSNR (dB) results of reconstructed boat images of successive layers 1–10 for our A-TLADMM-Net at CS ratio $\gamma = 10\%$.

E. CS on Speech Data

We also consider the application to another type of data, compressed sensing for speech data, with the same problem model as (26), but $\mathbf{x}$ is vectorized speech segments.

Datasets: We perform the same preprocessing as ADMM-DAD [58] on two classical speech datasets, SpeechCommands [84] (including 85 511 training samples and 4890 test samples) and TIMIT [85] (including 144 188 training samples and 52 712 test samples). All phonemes are sampled at 16 kHz and we randomly take 70% and 30% of them for training and testing, respectively.

For training, we choose Loss_1 as the loss function and train all the methods with $K = 10$ for 100 epochs by using the Adam optimizer. We set Mean Square Error, $\text{MSE} = (1/|\mathcal{S}|)\sum_{(\mathbf{b},\mathbf{x}^*)\in\mathcal{S}} \|\mathcal{M}(\mathbf{A},\mathbf{b};\mathbf{x}_0;\mathbf{\Theta}) - \mathbf{x}^*\|^2$ as a test criterion. We consider two CS ratios $p/d \in \{25\%, 40\%\}$ and use ADMM-DAD, ISTA-Net$^+$, and HSSE as baseline methods. The recovery results are shown in Table VIII. We can find that the MSE results of our networks are always lower than those of the baselines, and our A-TLADMM-Net can obtain the lowest MSE in all the cases. Furthermore, we plot the spectrograms of an example in the TIMIT dataset, as shown in Fig. 8, from which we can see that our networks distinguish more frequencies than

TABLE VI

COMPARISON OF NATURAL IMAGE CS RESULTS IN TERMS OF PSNR (DB) WITH DIFFERENT SAMPLED RATIOS γ =10%, 20%, 30%, 40%, AND 50% ON THE TEST DATASETS, BSD68, AND SET11

Algorithms \ Datasets	BSD68						Set11					
	10%	20%	30%	40%	50%	Avg.	10%	20%	30%	40%	50%	Avg.
SCSNet (CVPR2019, [70])	27.28	29.01	31.87	33.86	35.77	31.56	28.48	31.95	34.62	36.92	39.01	34.20
ISTA-Net^{++} (ICME2021, [38])	26.25	29.00	31.10	33.00	34.85	30.84	28.34	32.33	34.86	36.94	38.73	34.24
COAST (TIP2021, [71])	26.28	29.00	32.10	32.93	34.74	31.01	28.69	32.53	35.04	37.13	38.94	34.47
HSSE (TNNLS2022, [72])	26.29	28.99	32.01	32.75	34.75	30.96	28.69	-	34.92	37.04	38.92	-
LGSR (TNNLS2023, [73])	26.33	29.78	32.30	34.32	36.33	31.81	28.24	-	34.93	37.10	38.99	-
DPC-DUN (TIP 2023, [74])	26.82	29.66	31.81	33.75	35.68	31.54	29.33	32.86	35.80	37.88	39.79	35.13
TLADMM-Net ($K=10$, Ours)	27.76	30.38	32.68	34.78	36.82	32.48	28.95	32.81	35.73	38.18	40.32	35.19
A-TLADMM-Net ($K=10$, Ours)	27.67	30.60	32.82	34.90	36.95	32.59	29.20	33.02	35.87	38.24	40.42	35.35
A-TLADMM-Net ($K=20$, Ours)	27.94	30.86	33.21	35.36	37.41	32.96	29.43	33.35	36.14	38.61	40.89	35.68

TABLE VII

COMPARISON OF NATURAL IMAGE CS RESULTS IN TERMS OF PSNR (SSIM) AT DIFFERENT SAMPLED RATIOS γ =10%, 20%, 30%, AND 40% ON THE IMAGENET-1K DATASET

Algorithms	10%	20%	30%	40%
ISTA-Net [80]	24.62(0.670)	28.16(0.803)	30.37(0.865)	32.27(0.904)
ISTA-Net^{++} [38]	27.03(0.759)	29.75(0.843)	31.75(0.889)	33.48(0.920)
COAST [71]	27.11(0.764)	29.85(0.847)	31.88(0.892)	33.65(0.922)
DPC-DUN [74]	27.59(0.781)	30.20(0.865)	32.51(0.903)	34.33(0.931)
ELADMM-Net	27.66(0.800)	30.21(0.872)	32.29(0.914)	34.06(0.939)
A-ELADMM-Net	27.76(0.801)	30.30(0.873)	32.34(0.915)	34.10(0.939)
TLADMM-Net	27.93(0.806)	30.53(0.879)	32.61(0.918)	34.47(0.942)
A-TLADMM-Net	27.82(0.804)	30.79(0.882)	32.85(0.921)	34.58(0.944)

TABLE VIII

COMPARISON OF THE TEST MSE RESULTS ($\times 10^{-2}$ AND $\times 10^{-4}$) ON THE TIMIT AND SPEECHCOMMANDS DATASETS AT CS RATIOS $\gamma = 25\%, 40\%$

Algorithms \ Datasets	TIMIT		SpeechCommands	
	25%	40%	25%	40%
ISTA-Net^{+} [75]	2.232	2.029	0.584	0.462
HSSE [72]	0.911	0.829	0.476	0.347
ADMM-DAD [58]	0.791	0.424	0.252	0.134
ELADMM-Net (Ours)	0.681	0.403	0.173	0.078
A-ELADMM-Net (Ours)	0.491	0.222	0.161	0.076
TLADMM-Net (Ours)	0.507	0.215	0.158	0.080
A-TLADMM-Net (Ours)	0.470	0.202	0.154	0.075

ADMM-DAD. Moreover, (A-)TLADMM-Net further reduces the noise of the reconstruction results of (A-)ELADMM-Net, and A-TLADMM-Net preserves the most details.

F. CS MRI

To test the applicability of our methods, we perform our (A-)ELADMM-Net and (A-)TLADMM-Net to resolve the CS MRI. Following previous work [75], we set the matrix $\mathbf{\Phi}$ in Problem (26) to be the product of an under-sampling matrix and the discrete Fourier transform. For each sampling ratio, we train our networks separately with $K = 10$. We also performed a simple ablation experiment in the Supplementary Material to determine the loss functions for compared algorithms. Based on it, we also compared the results of their source code here. The accuracy and efficiency of reconstruction are measured by PSNR and GPU time, and all the test results are shown in

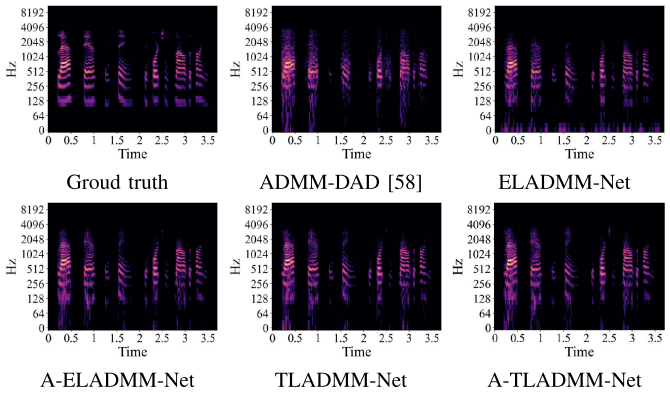


Fig. 8. Comparison of the visualization results for CS at $\gamma = 40\%$ on an example in the speech dataset, TIMIT. The reconstructed spectrograms of GT, ADMM-DAD [58] and our four networks are displayed in sequence.

TABLE IX

COMPARISON OF TEST PSNR (DB) AND RUNTIME (S) RESULTS FOR CS MRI ON A BRAIN DATASET. THE LAST COLUMN LISTS THE AVERAGE GPU TIME TO RECONSTRUCT A 256 × 256 IMAGE

Algorithms	CS Ratio γ				GPU Time
	20%	30%	40%	50%	
ADMM-Net [34]	37.17	39.84	41.56	43.00	0.046
ISTA-Net [75]	38.30	40.52	42.12	43.60	0.006
ISTA-Net^{+} [75]	38.73	40.89	42.52	44.09	0.007
ELADMM-Net (Ours)	38.31	40.21	42.12	43.66	0.007
A-ELADMM-Net (Ours)	38.40	40.31	42.19	43.77	0.007
TLADMM-Net (Ours)	38.72	40.81	42.57	44.15	0.012
A-TLADMM-Net (Ours)	38.78	40.84	42.60	44.28	0.012

Table IX. We also provide the comparison of these methods at the same time cost in the Supplementary Material.

Datasets: Our networks are trained on the same brain MR images as in ADMM-Net [34] and ISTA-Net^{+} [75]. 100 and 50 images are used for training and testing, respectively.

At relatively high CS ratios, all our networks outperform all the compared methods. Moreover, our networks always perform much better than ADMM-Net in terms of runtime and PSNR. Our TLADMM-Net enjoys better performance than ELADMM-Net regardless of the same number of network layers or time cost and our accelerated scheme (A-TLADMM-Net) almost always reconstructs higher quality images than other networks. At relatively low CS ratios, our TLADMM-Net and A-TLADMM-Net are competitive with the reconstruction results of ISTA-Net^{+}. Compared with ADMM-Net,

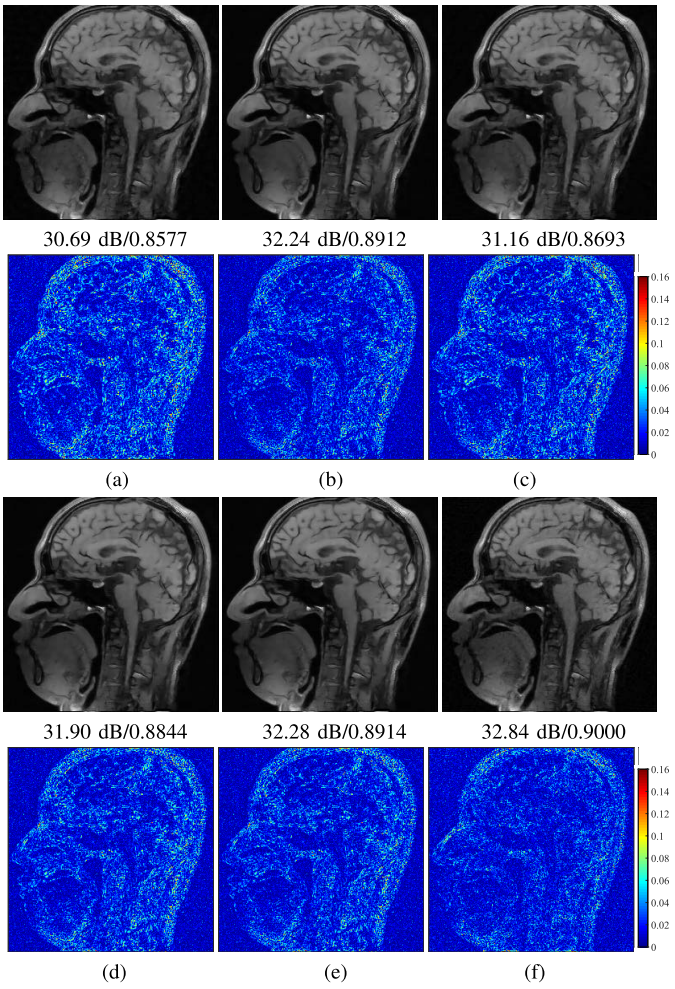


Fig. 9. Comparison of PSNR/SSIM and error results on a brain image for the CS MRI task at sampled ratio $\gamma = 20\%$. (a)–(f) Reconstructed images and errors of (a) ADMM-Net, (b) ISTA-Net$^+$, (c) ELADMM-Net, (d) A-E-LADMM-Net, (e) TLADMM-Net, and (f) A-TLADMM-Net, respectively.

(A-)TLADMM-Net reduces the reconstruction time by more than two-thirds because ADMM-Net needs to solve the matrix inversions. We further evaluate the structure similarity index measure (SSIM) [86] between the network output $\mathbf{x}_K$ and the GT. Fig. 9 shows the reconstruction images and errors on a brain image, which suggests that our A-TLADMM-Net can recover better results in terms of both detail preservation and reconstruction accuracy.

G. Summary of Experiments

By conducting the above experiments, we found that our experimental results match our theoretical analysis. Moreover, some common phenomena are summarized as follows.

1) Comparing the performance of our accelerated Euler and nonaccelerated trapezoid LADMM schemes, we find that the trapezoid structure can improve network performance better than extrapolation acceleration technology.

2) In all inverse problem experiments, our accelerated Trapezoid LADMM scheme almost always performs optimally, which is also the best scheme we designed through theoretical analysis. Considering both time consumption and data recovery accuracy, the Trapezoid LADMM and accelerated Trapezoid LADMM schemes are both good choices.

VI. Conclusion

In this article, we comprehensively designed the unfolded LADMM networks from the perspective of DEs for solving various inverse problems. We clarified the connection between existing unfolded networks and Euler discretization, and proposed a more efficient Trapezoid LADMM scheme. We find that good performance of unfolded networks requires high-precision discretization and a large representation space. Thus, we analyzed that the extrapolation accelerated Euler scheme has a close connection with second-order DEs, thereby expanding the representation space, and designed a better accelerated Trapezoid LADMM scheme. Extensive inverse problem experiments verified that our Trapezoid LADMM schemes can be used to solve inverse problems better.

References

[1] C. R. Vogel, *Computational Methods for Inverse Problems*. Philadelphia, PA, USA: SIAM, 2002.

[2] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Jun. 2010, pp. 399–406.

[3] Y. Zhang, X. Wang, F. Wang, and J. Wang, "Image denoising using convolutional sparse coding network with dry friction," in *Proc. Asian Conf. Comput. Vis.*, 2022, pp. 2322–2336.

[4] R. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007.

[5] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[6] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[7] X. Mei, Y. Yang, M. Li, C. Huang, K. Zhang, and P. Lió, "A feature reuse framework with texture-adaptive aggregation for reference-based super-resolution," 2023, *arXiv:2306.01500*.

[8] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 84–98, Mar. 2017.

[9] S. W. Zamir et al., "Multi-stage progressive image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 1–11.

[10] G. Teschke and R. Ramlau, "An iterative algorithm for nonlinear inverse problems with joint sparsity constraints in vector-valued regimes and an application to color image inpainting," *Inverse Problems*, vol. 23, no. 5, pp. 1851–1870, Oct. 2007.

[11] Y. Yang, J. Sun, H. Li, and Z. Xu, "ADMM-CSNet: A deep learning approach for image compressive sensing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 521–538, Mar. 2020.

[12] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems," in *Proc. Int. Conf. Learn. Represent.*, 2022.

[13] Y. Wang, J. Yu, and J. Zhang, "Zero-shot image restoration using denoising diffusion null-space model," in *Proc. Int. Conf. Learn. Represent.*, 2022.

[14] H. Chung, B. Sim, D. Ryu, and J. C. Ye, "Improving diffusion models for inverse problems using manifold constraints," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 25683–25696.

[15] H. Chung, S. Lee, and J. Chul Ye, "Decomposed diffusion sampler for accelerating large-scale inverse problems," 2023, *arXiv:2303.05754*.

[16] K. C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," *Pacific J. Optim.*, vol. 6, nos. 615–640, p. 15, 2010.

[17] S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Boston, MA, USA: Now, 2011.

[18] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. Adv. Neural. Inf. Process. Syst. (NIPS)*, 2011, pp. 612–620.

[19] C. Lu, H. Li, Z. Lin, and S. Yan, "Fast proximal linearized alternating direction method of multiplier with parallel splitting," in *Proc. Conf. Artif. Intell. (AAAI)*, Feb. 2016, pp. 739–745.

[20] T. Goldstein, B. O'Donoghue, S. Setzer, and R. G. Baraniuk, "Fast alternating direction optimization methods," *SIAM J. Imag. Sci.*, vol. 7, no. 3, pp. 1588–1623, 2014.

[21] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with $o(1/k)$ convergence," *J. Sci. Comput.*, vol. 71, no. 2, pp. 712–736, May 2017.

[22] H. Ouyang, N. He, L. Tran, and A. Gray, "Stochastic alternating direction method of multipliers," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 80–88.

[23] S. Zheng and J. T. Kwok, "Fast-and-light stochastic admm," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2407–2613.

[24] Y. Liu, F. Shang, H. Liu, L. Kong, L. Jiao, and Z. Lin, "Accelerated variance reduction stochastic ADMM for large-scale machine learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4242–4255, Dec. 2021.

[25] J. Ma, X.-Y. Liu, Z. Shou, and X. Yuan, "Deep tensor ADMM-Net for snapshot compressive imaging," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10223–10232.

[26] D. Shen, J. Liu, Z. Wu, J. Yang, and L. Xiao, "ADMM-HFNet: A matrix decomposition-based deep approach for hyperspectral image fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022, Art. no. 5513417.

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

[28] T. Chen et al., "Learning to optimize: A primer and a benchmark," *J. Mach. Learn. Res.*, vol. 23, no. 189, pp. 1–59, 2022.

[29] X. Chen, J. Liu, Z. Wang, and W. Yin, "Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9079–9089.

[30] J. Liu and X. Chen, "ALISTA: Analytic weights are as good as learned weights in LISTA," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[31] K. Wu, Y. Guo, Z. Li, and C. Zhang, "Sparse coding with gated learned ISTA," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[32] Y. Li, L. Kong, F. Shang, Y. Liu, H. Liu, and Z. Lin, "Learned extragradient ISTA with interpretable residual structures for sparse coding," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 8501–8509.

[33] Z. Zheng, W. Dai, D. Xue, C. Li, J. Zou, and H. Xiong, "Hybrid ISTA: Unfolding ISTA with convergence guarantees using free-form deep neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3226–3244, Mar. 2023.

[34] Y. Yang, J. Sun, H. B. Li, and Z. B. Xu, "Deep ADMM-Net for compressive sensing MRI," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 10–18.

[35] X. Xie, J. Wu, G. Liu, Z. Zhong, and Z. Lin, "Differentiable linearized ADMM," in *Proc. Int. Conf. Mach. Learn.*, May 2019, pp. 6902–6911.

[36] J. M. Ramirez, J. I. Martínez-Torre, and H. Arguello, "LADMM-Net: An unrolled deep network for spectral image fusion from compressive data," *Signal Process.*, vol. 189, Dec. 2021, Art. no. 108239.

[37] Z. Chen, J. Xiang, P.-O. Bagnaninchi, and Y. Yang, "MMV-Net: A multiple measurement vector network for multifrequency electrical impedance tomography," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 8938–8949, Nov. 2023.

[38] D. You, J. Xie, and J. Zhang, "ISTA-NET++: Flexible deep unfolding network for compressive sensing," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2021, pp. 1–6.

[39] W. Su, S. Boyd, and E. J. Candes, "A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 5312–5354, 2016.

[40] M. E. Sander, P. Ablin, M. Blondel, and G. Peyré, "Momentum residual neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 9276–9287.

[41] W. An, Y. Yue, Y. Liu, F. Shang, and H. Liu, "A numerical design perspective on unfolded linearized ADMM networks for inverse problems," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 5065–5073.

[42] A. A. Brown and M. C. Bartholomew-Biggs, "Some effective methods for unconstrained optimization based on the solution of systems of ordinary differential equations," *J. Optim. Theory Appl.*, vol. 62, pp. 211–224, Aug. 1989.

[43] A. Bloch, *Hamiltonian and Gradient Flows, Algorithms and Control*. Providence, RI, USA: American Mathematical Society, 1994.

[44] J. Schropp and I. Singer, "A dynamical systems approach to constrained minimization," *Numer. Funct. Anal. Optim.*, vol. 21, nos. 3–4, pp. 537–551, Jan. 2000.

[45] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*. New York, NY, USA: Springer, 2012.

[46] G. Franca, D. Robinson, and R. Vidal, "ADMM and accelerated ADMM as continuous dynamical systems," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1559–1567.

[47] X. He, R. Hu, and Y.-P. Fang, "'Second-order primal' + 'first-order dual" dynamical systems with time scaling for linear equality constrained convex optimization problems," 2021, *arXiv:2103.12931*.

[48] X. He, R. Hu, and Y.-P. Fang, "Fast primal–dual algorithm via dynamical system for a linearly constrained convex optimization problem," *Automatica*, vol. 146, Dec. 2022, Art. no. 110547.

[49] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imag. Vis.*, vol. 40, no. 1, pp. 120–145, May 2011.

[50] X. He, R. Hu, and Y. P. Fang, "Convergence rates of inertial primal-dual dynamical methods for separable convex optimization problems," *SIAM J. Control Optim.*, vol. 59, no. 5, pp. 3278–3301, Jan. 2021.

[51] D. Kovalev, A. Gasnikov, and P. Richtárik, "Accelerated primal-dual gradient method for smooth and convex-concave saddle-point problems with bilinear coupling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 21725–21737.

[52] K. K. Thekumparampil, N. He, and S. Oh, "Lifted primal-dual method for bilinearly coupled smooth minimax optimization," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2022, pp. 4281–4308.

[53] J. Zhang, P. Xiao, R. Sun, and Z.-Q. Luo, "A single-loop smoothed gradient descent-ascent algorithm for nonconvex-concave min-max problems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 7377–7389.

[54] G. França, D. P. Robinson, and R. Vidal, "A nonsmooth dynamical systems perspective on accelerated extensions of ADMM," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 2966–2978, May 2023.

[55] H. Yuan, Y. Zhou, C. J. Li, and Q. Sun, "Differential inclusions for modeling nonsmooth ADMM variants: A continuous limit theory," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7232–7241.

[56] X. Zhou, H. Yuan, C. Junchi Li, and Q. Sun, "Stochastic modified equations for continuous limit of stochastic ADMM," 2020, *arXiv:2003.03532*.

[57] S.-W. Hu, G.-X. Lin, and C.-S. Lu, "GPX-ADMM-Net: ADMM-based neural network with generalized proximal operator," in *Proc. 28th Eur. Signal Process. Conf. (EUSIPCO)*, Jan. 2021, pp. 2055–2059.

[58] V. Kouni, G. Paraskevopoulos, H. Rauhut, and G. C. Alexandropoulos, "ADMM-DAD Net: A deep unfolding network for analysis compressed sensing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 1506–1510.

[59] A. Vassilis, A. Jean-François, and D. Charles, "The differential inclusion modeling FISTA algorithm and optimality of convergence rate in the case b $\leq$ 3," *SIAM J. Optim.*, vol. 28, no. 1, pp. 551–574, Jan. 2018.

[60] R. P. Agarwal and D. O'Regan, *An Introduction to Ordinary Differential Equations*. Cham, Switzerland: Springer, 2008.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[62] H. Gupta, K. H. Jin, H. Q. Nguyen, M. T. McCann, and M. Unser, "CNN-based projected gradient descent for consistent CT image reconstruction," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1440–1453, Jun. 2018.

[63] J. R. Dormand and P. J. Prince, "A family of embedded Runge–Kutta formulae," *J. Comput. Appl. Math.*, vol. 6, no. 1, pp. 19–26, Mar. 1980.

[64] E. Dupont, A. Doucet, and Y. W. Teh, "Augmented neural ODEs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 3140–3150.

[65] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4401–4410.

[66] Y. Xu and W. Yin, "A fast patch-dictionary method for whole image recovery," *Inverse Problems Imag.*, vol. 10, no. 2, pp. 563–583, May 2016.

[67] T. Moreau and J. Bruna, "Understanding trainable sparse coding via matrix factorization," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[68] A. Aberdam, A. Golts, and M. Elad, "Ada-LISTA: Learned solvers adaptive to varying models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9222–9235, Dec. 2022.

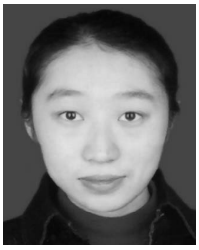
[69] T. Yokota, H. Hontani, Q. Zhao, and A. Cichocki, "Manifold modeling in embedded space: An interpretable alternative to deep image prior," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 1022–1036, Mar. 2022.

[70] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Scalable convolutional neural network for image compressed sensing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12282–12291.

[71] D. You, J. Zhang, J. Xie, B. Chen, and S. Ma, "COAST: Controllable arbitrary-sampling network for compressive sensing," *IEEE Trans. Image Process.*, vol. 30, pp. 6066–6080, 2021.
[72] Z. Zha, B. Wen, X. Yuan, J. Zhou, C. Zhu, and A. C. Kot, "A hybrid structural sparsification error model for image restoration," *IEEE Trans. Neural Netw. Learn. Syst. (TNNLS)*, vol. 33, no. 9, pp. 4451–4465, Sep. 2022.
[73] Z. Zha, B. Wen, X. Yuan, J. Zhou, C. Zhu, and A. C. Kot, "Low-rankness guided group sparse representation for image restoration," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 7593–7607, Oct. 2023.
[74] J. Song, B. Chen, and J. Zhang, "Dynamic path-controllable deep unfolding network for compressive sensing," *IEEE Trans. Image Process.*, vol. 32, pp. 2202–2214, 2023.
[75] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proc. IEEE CVPR*, Jun. 2018, pp. 1828–1837.
[76] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "ReconNet: Non-iterative reconstruction of images from compressively sensed measurements," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 449–458.
[77] R. Timofte et al., "NTIRE 2017 challenge on single image super-resolution: Methods and results," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 114–125.
[78] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, Jul. 2001, pp. 416–423.
[79] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[80] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, "Denoising prior driven deep neural network for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2305–2318, Oct. 2019.
[81] D. Gilton, G. Ongie, and R. Willett, "Neumann networks for linear inverse problems in imaging," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 328–343, 2019.
[82] Y. Sun, J. Chen, Q. Liu, B. Liu, and G. Guo, "Dual-path attention network for compressed sensing image reconstruction," *IEEE Trans. Image Process. (TIP)*, vol. 29, pp. 9482–9495, 2020.
[83] J. Chen, Y. Sun, Q. Liu, and R. Huang, "Learning memory augmented cascading network for compressed sensing of images," in *Proc. ECCV*, 2020, pp. 513–529.
[84] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018, *arXiv:1804.03209*.
[85] J. S. Garofolo, *TIMIT: Acoustic-Phonetic Continuous Speech Corpus*. Philadelphia, PA, USA: Linguistic Data Consortium, 1993.
[86] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

Weixin An received the B.S. degree majoring in information and computation science from Xi'an University of Technology, Xi'an, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Artificial intelligence, Xidian University, China.

His current research interests include trustworthy machine learning, model optimization and generalization, and image processing.

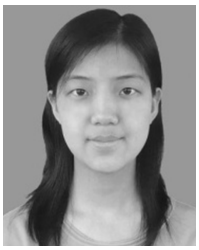


Yuanyuan Liu (Member, IEEE) received the Ph.D. degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2013.

She is currently a Professor with the School of Artificial Intelligence, Xidian University. Before joining Xidian University, she was a Post-Doctoral Research Fellow at the Department of Computer Science and Engineering, The Chinese University of Hong Kong (CUHK), Hong Kong. From 2013 to 2014, she was a Post-Doctoral Research Fellow at the Department of Systems Engineering and Engineering Management, CUHK. Her current research interests include machine learning, pattern recognition, and image processing.

Fanhua Shang (Senior Member, IEEE) received the Ph.D. degree in circuits and systems from Xidian University, Xi'an, China, in 2012.

From 2012 to 2013, he was a Post-Doctoral Research Associate at the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA. From 2013 to 2015, he was a Post-Doctoral Research Fellow at the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, where he was a Research Associate, from 2016 to 2018. From 2018 to 2022, he was a Professor at the School of Artificial Intelligence, Xidian University. He is currently a Professor with the School of Computer Science and Technology, College of Computing and Intelligence, Tianjin University, Tianjin, China. His current research interests include machine learning, deep learning, data mining, computer vision, and stochastic optimization.



Hongying Liu (Senior Member, IEEE) received the B.E. and M.S. degrees in computer science and technology from Xi'an University of Technology, Xi'an, China, in 2006 and 2009, respectively, and the Ph.D. degree in engineering from Waseda University, Tokyo, Japan, in 2012.

From 2013 to 2023, she was an Associate Professor at the School of Artificial Intelligence, Xidian University, Xi'an. Currently, she is an Associate Professor with Tianjin University, Tianjin, China. Her major research interests include image processing, intelligent signal processing, and machine learning.

Licheng Jiao (Fellow, IEEE) received the B.S. degree from Shanghai Jiaotong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

Currently, he is the Director of the Key Laboratory of Intelligent Perception and Image Understanding, Ministry of Education, Xi'an. He has charged about 40 important scientific research projects and published more than 20 monographs and 100 papers in international journals and conferences. His current research interests include image processing, natural computation, machine learning, and intelligent information processing.

Dr. Jiao is the Chairperson of the Awards and Recognition Committee, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, a Councilor of the Chinese Institute of Electronics, a Committee Member of the Chinese Committee of Neural Networks, and an Expert of the Academic Degrees Committee of the State Council.