A Peaceman-Rachford Splitting Approach with Deep Equilibrium Network for Channel Estimation

Dingli Yuan[1], Shitong Wu[1], Lu Yang[2], Chenghui Peng[2], Haoran Tang[1], and Hao Wu[1†]

[1]Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China
[2]Wireless Technology Lab, Central Research Institute, 2012 Labs, Huawei Tech. Co. Ltd., China
Email: hwu@tsinghua.edu.cn

***Abstract*—Multiple-input multiple-output (MIMO) is pivotal for wireless systems, yet its high-dimensional, stochastic channel poses significant challenges for accurate estimation, highlighting the critical need for robust channel estimation techniques. In this paper, we introduce a novel channel estimation method for the MIMO system. The main idea is to solve the dual form of the channel estimation problem, employing the Peaceman-Rachford (PR) splitting method with the deep equilibrium (DEQ) network. Specifically, the PR splitting method is applied to the Fenchel conjugate of the channel estimation functions to construct the non-expansive operators, which forms a fixed point equation of the optimal condition. The non-expansive operator is learned implicitly via the DEQ network and the fixed point equation is computed analytically using the learned operators, which further enables efficient iterative step with only a constant memory. Moreover, we provide a rigorous theoretical analysis using the monotone operator theory, offering the convergence proof of our proposed framework. Additionally, simulations of hybrid far- and near-field channels demonstrate that our approach yields favorable results, indicating its ability to advance channel estimation in MIMO system.**

I. Introduction

Multiple-input multiple-output (MIMO) technology is pivotal for wireless systems [1]. Recognizing its paramount importance, the channel estimation algorithm should possess high adaptability and performance for different types of channels in the MIMO systems, including the hybrid far- and near-field channels [2], [3], etc. Hence, there is a strong motivation for improvements in channel estimation algorithms with broad applicability and high accuracy.

Traditional channel estimation schemes with closed-form expressions [4], such as least square (LS) and minimum mean-squared error (MMSE), have limitations in practical applications. The LS can not ensure a high estimation accuracy for different scenarios, while the MMSE has a high computation and deployment complexity [5]. Guided by the pursuit of higher estimation accuracy and lower computation complexity, numerous iterative algorithms have been proposed, and can be classified into different categories. One of them is to add the regularization terms after least squares and utilize the optimization methods such as proximal gradient descent (PGD) or alternating direction method of multipliers (ADMM) [6]. Another one employs probabilistic models aimed at minimizing the mean square error (MSE), combining the channel prior information for algorithm design, including the approximate message passing (AMP) [7], the orthogonal AMP (OAMP) [8], and the algorithms extended based on them.

Despite using different design principles, these algorithms share a similar iteration update rule, that each iteration is composed of a linear estimator (LE) and a non-linear estimator (NLE) [9]. The LE is explicit and can be designed with the low complexity. The bottleneck of a high performance estimator design lies in the NLE, since its own complexity is higher and its design requires enough channel prior information as accurate as possible, which is difficult to acquire [10]. A heuristic way is to replace NLE with neural network (NN) [11], [12], which has advantage in capturing data features. Additionally, deep neural networks, known as powerful denoisers [13], are well-suited to replace the NLE, which motivate the thriving of model-driven channel estimators.

Model-driven deep learning-based estimators, especially those employing deep unfolding [11], encounter several systematic challenges. These estimators are constructed by truncating a classical iterative algorithm into a predefined or fixed number of layers, denoted as T, and then replacing the NLE in layer t with a deep neural network parameterized by Θ_t [11], [14]. However, this conventional deep unfolding formulation has critical issues [15]. First, its scalability and generalization ability is poor. Second, its reliability is lack of theoretical guarantees, as truncating the algorithm into T layers disrupts the convergence of a classical iterative algorithm. Third, its complexity is high, but its adaptiveness is low, as it requires full execution of T layers with unreliable intermediate states. Considering these difficulties, a reconsideration of the feasibility of the deep unfolding framework is imperative. Recent works have consider the deep equilibrium (DEQ) Model with its application in inverse problem [9], image demonising [13], video snapshot compressive imaging [16] and hybrid far- and near-field channel estimation [3]. In their work, they replace the NLE with NN and trained the model by computing the implicit gradient based on implicit function theorem [15]. All of these work requires the estimation of the Lipschitz constant [17] to ensure a linear convergence rate. However, in real applications, the measurement matrix may be ill-posed and

The first two authors contributed equally to this work and † marked the corresponding author. This work was partially supported by National Key Research and Development Program of China (2018YFA0701603) and National Natural Science Foundation of China (12271289 and 62231022).

these methods may lack theoretical foundations [3].

In this paper, we propose a novel approach for channel estimation in MIMO systems. Unlike the traditional approach of modeling channel estimation using the minimization problem with regularization terms, we take a different perspective by considering the dual problem. This choice ensures a stable convex model, enhancing robust convergence properties for different scenarios. Leveraging the advantageous convexity of the dual problem, we formulate the channel estimation algorithm using the Peaceman-Rachford (PR) splitting method. Specifically, we derive an alternative algorithm from the PR splitting algorithm for dual problem with explicit iteration and constructs a fixed point equation of an intermediate variable, ensuring the non-expansive property of the combined operator [18]. Due to the computational challenges associated with the nonlinear term in PR iteration, we employ NN to approximate this nonlinear term. We introduce a NN framework called Peaceman-Rachford (PR) splitting method implemented with Deep Equilibrium Network, (PR-DEN), which leverages the DEQ model for analytical computing the fixed point equation. This approach enables efficient iterative steps, with a same NN for each iteration, and thus the memory cost of our NN framework is a constant. Additionally, based on the monotone operator theory, our NN framework is supported by rigorous proofs of both convergence and optimality. Furthermore, extensive simulations including the hybrid far- and near-field channels MIMO system validates the exceptional performance of our methodology. These promising results underscore the potential of our proposed approach to significantly advance channel estimation in MIMO systems.

II. System Model and Problem Formulation

A. Uplink Channel Estimation Model

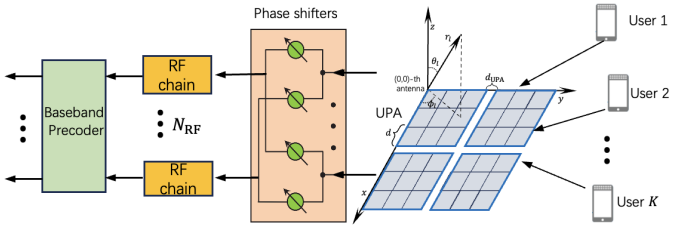


Fig. 1: System model of the uplink channel estimation.

In this work, we consider the uplink channel estimation for MIMO system as shown in Fig. 1. The base station (BS) is equipped with $\sqrt{S} \times \sqrt{S}$ uniform planar arrays (UPAs) and has a total of N antennas. Specifically, the separation between adjacent UPAs is denoted as d_{UPA}. In each UPA, the antennas are arranged in a $\sqrt{N/S} \times \sqrt{N/S}$ configuration with an antenna spacing of d. Additionally, the BS is employed with a hybrid precoding architecture [19], featuring N_{RF} radio-frequency (RF) chains and serving K single-antenna users, where $N_{\mathrm{RF}} \ll N$ and $K \leq N_{\mathrm{RF}}$. And we set the antennas in each UPA share the same RF chain. For the uplink channel estimation, we assume the K users transmit mutual orthogonal pilot sequences to the BS [20], then channel estimation for each user is independent. Thus, without loss of generality, we consider an arbitrary user.

The user transmits known pilot signals to the BS for P time slots, and we denote $s_p \in \mathbb{C}^{1\times 1}$ as the transmit pilot in time slot p. Then, the received pilot $\bar{\boldsymbol{y}}_p \in \mathbb{C}^{N_{\mathrm{RF}}\times 1}$ is

$$\bar{\boldsymbol{y}}_p = \bar{\mathbf{W}}_p \bar{\mathbf{A}}_p \mathbf{F} \bar{\boldsymbol{h}} s_p + \bar{\mathbf{W}}_p \bar{\mathbf{A}}_p \bar{\boldsymbol{n}}_p,$$

where $\mathbf{F} \in \mathbb{C}^{N\times N}$ denotes the Fourier transform matrix transforming the channel $\bar{\boldsymbol{h}} \in \mathbb{C}^{N\times 1}$ into its angular-domain, $\bar{\mathbf{A}}_p \in \mathbb{C}^{N_{\mathrm{RF}}\times N}$ denotes the analog combining matrix satisfying the constant modulus constraint $\left|\bar{\mathbf{A}}_p(i,j)\right| = \frac{1}{\sqrt{N}}$, $\bar{\mathbf{W}}_p \in \mathbb{C}^{N_{\mathrm{RF}}\times N_{\mathrm{RF}}}$ denotes the digital combining matrix which is set as the identity matrix $\mathbf{I}_{N_{\mathrm{RF}}}$, and $\bar{\boldsymbol{n}}_p \in \mathbb{C}^{N\times 1}$ denotes the Gaussian complex noise following the distribution $\mathcal{CN}(0, \sigma^2 \mathbf{I}_N)$. Assume $s_p = 1,\ p = 1, \ldots, P$, then the overall received pilot sequence $\bar{\boldsymbol{y}} = \left[\bar{\boldsymbol{y}}_1^T, \ldots, \bar{\boldsymbol{y}}_P^T\right]^T$ is denoted as

$$\bar{\boldsymbol{y}} = \bar{\mathbf{A}} \bar{\boldsymbol{h}} + \bar{\boldsymbol{n}}, \tag{1}$$

where $\bar{\boldsymbol{n}} = \left[\bar{\boldsymbol{n}}_1^T \bar{\mathbf{A}}_1^T, \ldots, \bar{\boldsymbol{n}}_P^T \bar{\mathbf{A}}_P^T\right]^T \in \mathbb{C}^{N_{\mathrm{RF}}P\times 1}$ denotes the noise and $\bar{\mathbf{A}} = \left[\mathbf{F}^T \bar{\mathbf{A}}_1^T, \ldots, \mathbf{F}^T \bar{\mathbf{A}}_P^T\right]^T \in \mathbb{C}^{N_{\mathrm{RF}}P\times N}$ denotes the overall analog combining matrix. Additionally, we transform (1) into its equivalent real-valued form, the uplink channel estimation problem can be given by:

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{h} + \boldsymbol{n}, \tag{2}$$

where $\boldsymbol{y} \in \mathbb{R}^{2N_{\mathrm{RF}}P\times 1}$, $\mathbf{A} \in \mathbb{R}^{2N_{\mathrm{RF}}P\times 2N}$, $\boldsymbol{h} \in \mathbb{R}^{2N\times 1}$ and $\boldsymbol{n} \in \mathbb{R}^{2N_{\mathrm{RF}}P\times 1}$. The detailed transformation of (1) into the real-valued form is shown in the Appendix [1].

Since the matrix $\mathbf{A}$ may be ill-posed, direct use of the least square estimation does not work well [21]. Hence, we introduce a regularization term [22] to modify the least square problem by utilizing the prior information of the channel. Specifically, problem (2) can be transformed into solving the following optimal problem [23]:

$$\min_{\boldsymbol{h}\in\mathbb{R}^{2N\times 1}} \quad g(\boldsymbol{h}) + \frac{1}{2}\|\boldsymbol{y} - \mathbf{A}\boldsymbol{h}\|_2^2, \tag{3}$$

where $g(\boldsymbol{h})$ denotes the regularization function which encompasses the prior information of the channel characteristics [24].

B. Introduction of the Hybrid-Field Channel Model

We introduce the classic hybrid far- and near-field channel model. The generation of far-field or near-field channel is determined by the Rayleigh distance, i.e. $D_{\mathrm{Rayleigh}} = \frac{D^2}{\lambda_c}$, where D is the array aperture and and λ_c is the carrier wavelength. Due to limited scattering, the propagation channel is represented by a ray-based model of L rays. The 0-th ray corresponds to the line-of-sight (LoS) path, while the $l = 1, ..., L-1$, rays are non-line-of-sight (NLoS) paths. Specifically, the channel can be expressed as [25]:

$$\boldsymbol{h} = \textstyle\sum_{l=0}^{L} \beta_l \mathbf{a}\left(\phi_l, \theta_l, r_l\right) e^{-j2\pi f_c \tau_l}$$

where the parameter f_c is the carrier frequency. And the parameters $\beta_l, \phi_l, \theta_l, r_l, \mathbf{a}\left(\phi_l, \theta_l, r_l\right)$, and τ_l are respectively

[1]The derivation appendix, https://github.com/wushitong1234/PR-DEN

the path loss, azimuth angle of arrival (AoA), elevation AoA, distance between the array and the RF source/scatterer, array response vector, and time delay of the l-th path.

Under the ray-based model mentioned before, the path loss can be calculated by the following expression [25]:

$$\beta_l = |\Gamma_l| \left(\frac{c}{4\pi f_c r_1}\right) e^{-\frac{1}{2}k_{\text{abs}} r_1},$$

where Γ_l is the reflection coefficient, r_1 is the LoS path length, and k_{abs} is the molecular absorption coefficient and

$$\Gamma_l = \begin{cases} \frac{\cos\varphi_{\text{in},l} - n_t \cos\varphi_{\text{ref},l}}{\cos\varphi_{\text{in},l} + n_t \cos\varphi_{\text{ref},l}} e^{-\left(\frac{8\pi^2 f_c^2 \sigma_{\text{rough}}^2 \cos^2\varphi_{\text{in},l}}{c^2}\right)}, & \text{if } l > 1, \\ 1, & \text{if } l = 0. \end{cases}$$

Here $\varphi_{\text{in},l}$ is the angle of incidence of the l-th path, $\varphi_{\text{ref},l} = \arcsin\left(n_t^{-1}\sin\varphi_{\text{in},l}\right)$ is the angle of refraction, n_t is the refractive index and σ_{rough} is the roughness coefficient of the reflecting material.

Since the wavefront is approximately planar in the far field and spherical in the near field, the array responses, which are determined by the distance r_l, is given by [3]:

$$\mathbf{a}(\phi_l, \theta_l, r_l) = \begin{cases} \text{vec}(\mathbf{A}^{\text{far}}(\phi_l, \theta_l)), & \text{if } r_l > D_{\text{Rayleigh}}, \\ \text{vec}(\mathbf{A}^{\text{near}}(\phi_l, \theta_l, r_l)) & \text{otherwise.} \end{cases}$$

where vec() denotes the operation of unfolding a matrix into a vector. Specifically,

$$\begin{cases} \mathbf{A}^{\text{far}}(\phi_l, \theta_l)_{s,\bar{s}} = e^{-j2\pi\frac{f_c}{c}\mathbf{p}_{s,\bar{s}}^T \mathbf{t}_l}, \\ \mathbf{A}^{\text{near}}(\phi_l, \theta_l, r_l)_{s,\bar{s}} = e^{-j2\pi\frac{f_c}{c}\|\mathbf{p}_{s,\bar{s}} - r_l\mathbf{t}_l\|_2}. \end{cases}$$

where c is the speed of light, and $\mathbf{t}_l$ is the unit-length vector in the AoA direction of the l-th path, given by $\mathbf{t}_l = [\sin\theta_l\cos\phi_l, \sin\theta_l\sin\phi_l, \cos\theta_l]^T$. The coordinate $\mathbf{p}_{s,\bar{s}}$ can be expressed as [3]: $\mathbf{p}_{s,\bar{s}} = [(\bar{m}-1)d + (m-1)(\sqrt{N/S}-1)d_{\text{UPA}}, (\bar{n}-1)d + (n-1)(\sqrt{N/S}-1)d_{\text{UPA}}, 0]^T$, where $s = (m-1)(\sqrt{N/S}) + n$, $\bar{s} = (\bar{m}-1)(\sqrt{N/S}) + \bar{n}$, $1 \le m, n \le \sqrt{S}$, and $1 \le \bar{m}, \bar{n} \le \sqrt{N/S}$.

III. A Peaceman-Rachford Splitting Approach with Deep Equilibrium Network

Solving (3) directly is difficult, since the regularization term $g(\boldsymbol{h})$ might face the following two challenges. First, this term might be non-convex, leading to no theoretical convergence guarantees of numerical algorithms. Second, this term might not have an explicit formulation, due to the limited information about the prior information of the channel characteristics, leading to no explicit iterative formulae by directly computation.

To overcome these difficulties, in this section, we introduce a framework employing the PR splitting approach for dual form to avoid the difficulty of the non-convexity, and implementing the DEQ network to learn prior information. Specifically, we consider the dual of the channel estimation problem (3) and apply the PR splitting method to find the solution of the derived dual problem [26], [27]. The convexity of the Fenchel conjugate ensures the convergence of the PR algorithm, which remains robust enough even when the regularization term $g(\boldsymbol{h})$ is non-convex.

Based on the PR splitting framework, we obtain the a fixed point equation consisting of only non-expansive operators, whose Lipschitz constant is not larger than one. This fixed point equation inspires us to utilize the DEQ model, which leverages a fixed point approach with a NN architecture where each layer is not changing across all the iterations. In this way, the proximal operator associated with the term $g(\boldsymbol{h})$ can be implicitly learned through the equilibrium formulation, and the learned operator is fixed for each iteration.

A. Peaceman-Rachford Splitting Algorithm for Dual Problem

As presented in the previous discussion, the regularized term $g(\boldsymbol{h})$ in (3) encapsulates the prior information of channel characteristics. In order to solve the problem in a more stable and robust manner, we choose the dual form [28] of (3) the above regularized problem, which is a fundamental technique in optimization [29]. In the following, we denote $f(\boldsymbol{h}) = \frac{1}{2}\|\boldsymbol{y} - \mathbf{A}\boldsymbol{h}\|_2^2$ for brevity, and denote f^* and g^* as the Fenchel conjugate of f and g, respectively. Then, by employing the Fenchel conjugate, we derive the dual of (3) as (the derivation details are summarized in the Appendix)

$$\max_{\boldsymbol{x}\in\mathbb{R}^{2N}} \quad \{-g^*(-\boldsymbol{x}) - f^*(\boldsymbol{x})\}. \tag{4}$$

Here, the variable $\boldsymbol{x}$ is the corresponding dual variable.

Traditional methodologies [3], [9] dealing with the channel estimation problem is based on the prime form (3), where the regularization term $g(\boldsymbol{h})$ may be non-convex, leading to no convergence guarantees in theory. On the contrary, we aim to find the optimal solution of the dual problem (4). This choice can ensure a stable model, given that the Fenchel conjugate is always convex [6], thereby providing excellent robust convergence properties for different channel scenarios.

To solve the dual problem (4), the most important procedure is to construct efficient algorithm to compute the dual variable $\boldsymbol{x}$ under the formulation of Fenchel conjugate. Using first-order conditions, the optimal solution of (4) is equivalent to solving the following rooting problem [6], [30]:

$$0 \in \partial g^*(-\boldsymbol{x}) + \partial f^*(\boldsymbol{x}). \tag{5}$$

Taking $\mathbf{M}_1 = \partial(g^* \circ (-\mathbf{I}))$, $\mathbf{M}_2 = \partial(f^*)$, then $\mathbf{M}_1, \mathbf{M}_2$ are maximal monotone operators, since subgradients are maximal monotone operators [28]. Hence, the PR splitting method [26], [27] can be directly applied to solve the zeros of system (5), where the PR splitting method can be represented as:

$$\boldsymbol{x} = \mathbf{J}_{\sigma\mathbf{M}_2}\boldsymbol{\eta}, \quad \text{where } \boldsymbol{\eta} \text{ satisfies } \mathbf{R}_{\sigma\mathbf{M}_1}\mathbf{R}_{\sigma\mathbf{M}_2}\boldsymbol{\eta} = \boldsymbol{\eta}.$$

Here, $\sigma > 0$, $\mathbf{J}_{\sigma\mathbf{M}} = (\mathbf{I} + \sigma\mathbf{M})^{-1}$ is the resolvent of $\sigma\mathbf{M}$ and $\mathbf{R}_{\sigma\mathbf{M}} = 2\mathbf{J}_{\sigma\mathbf{M}} - \mathbf{I}$ is called the reflected resolvent [27], [30].

It is worth mentioning that the variable $\boldsymbol{\eta}$ in the fixed point equation (6) is an intermediate variable, which can usually be solved by fixed point iteration, i.e.,

$$\boldsymbol{\eta}^{k+1} = \mathbf{T}_\sigma^{\mathbf{PR}}(\boldsymbol{\eta}^k) = \mathbf{R}_{\sigma\mathbf{M}_1}\mathbf{R}_{\sigma\mathbf{M}_2}(\boldsymbol{\eta}^k) \tag{6}$$

In the following, we present our algorithm from the PR splitting algorithm for dual problem (4) with explicit iteration. The details of the derivation are provided in the Appendix.

Algorithm 1 PR Splitting for Dual Problem

1: **Input:** $\boldsymbol{x}^0 \in \mathbb{R}^{2N}, \boldsymbol{p}^0 = \boldsymbol{x}^0$, and $\sigma > 0$.
2: **Initialize:** $\boldsymbol{\eta}^0 = \boldsymbol{x}^0 + \sigma \boldsymbol{p}^0$.
3: **for** $k = 0, 1, \ldots L-1$ **do**
4: $\quad \boldsymbol{q}^{k+1} = (\mathbf{A}^\top \mathbf{A} + \sigma \mathbf{I})^{-1}(\mathbf{A}^\top \boldsymbol{y} + \boldsymbol{\eta}^k)$
5: $\quad \boldsymbol{p}^{k+1} = \mathrm{Prox}_{\sigma^{-1} g}(\sigma^{-1}(2\sigma \boldsymbol{q}^{k+1} - \boldsymbol{\eta}^k))$.
6: $\quad \boldsymbol{x}^{k+1} = \boldsymbol{\eta}^k + \sigma \boldsymbol{p}^{k+1} - 2\sigma \boldsymbol{q}^{k+1}$.
7: $\quad \boldsymbol{\eta}^{k+1} = \boldsymbol{\eta}^k + 2\sigma(\boldsymbol{p}^{k+1} - \boldsymbol{q}^{k+1})$.
8: **Output**: $\boldsymbol{x}^L$

Based on the derived algorithm, we prove the convergence of the sequences $\{\boldsymbol{x}^k\}$ and $\{\boldsymbol{p}^k\}$ to the corresponding optimal points of the dual and prime problem respectively.

Theorem 1. *The sequence $\{\boldsymbol{x}^k\}$ generated by Algorithm 1 converges to the optimal solution $\boldsymbol{x}^*$ of (4). Moreover, when the regularization term $g(\boldsymbol{h})$ in (3) is assumed to be convex, the sequence $\{\boldsymbol{p}^k\}$ generated by Algorithm 1 converges to the optimal solution $\boldsymbol{h}^*$ of (3) and the duality holds.*

Proof. The proof is based on the convergence property of the PR iteration [27], [31] for convex problem and the optimal condition of the system [28]. The detailed proof is moved to the Appendix due to space limitation. □

B. Deep Equilibrium Network

In the previous discussion, an unresolved issue was the specific form of the regularization term $g(\boldsymbol{h})$, which remains unknown, presenting challenges in directly applying the PR algorithm to solve the channel estimation problem. Therefore, our next step is utilizing NNs, which possess powerful feature extraction capabilities, to learn the regularization term.

Since neural networks process powerful abilities to learn statistical features, it is logical and achievable to learn this nonlinear term through neural networks. Similar to the deep unrolling approach of [11], we consider replacing $\mathrm{prox}_{\sigma^{-1} g}$ with a trainable network R_θ. Specifically, R_θ is constructed using a classical residual network [32], which facilitates efficient training of deeper models by leveraging shortcut connections for learning residual mappings. The residual network is mainly composed of four residual blocks, each of which is activated by two convolution layers and two RELU functions.

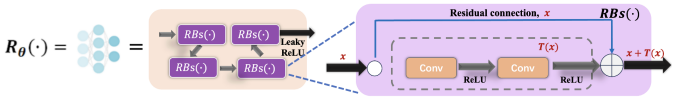


Fig. 2: The design of a CNN block R_θ to approximate the proximal operator, denoted by $R_\theta \to \mathrm{Prox}_{\sigma^{-1} g}$.

We use the DEQ model to learn the network parameter θ, since compared with traditional deep unfolding approaches that unfold distinct layers for multiple iterations [10], [16], the application of DEQ offers several advantages. First, the DEQ model uses a fixed NN for each iteration, corresponding to our fixed point iterative framework, while traditional approaches need different NNs for each layer of iteration, deviating from the fixed point framework and affecting the NN's stability due to varying depth. Second, the DEQ employs only a fixed single layer, which significantly reduces the network's memory requirements and the training overhead [15], [16].

The abstract iteration of $\boldsymbol{\eta}^k$ in DEQ can be represented as

$$\begin{aligned} \lim_{k \to +\infty} \boldsymbol{\eta}^{(k)} &= \lim_{k \to +\infty} f_\theta \left(\boldsymbol{\eta}^{(k)}; \boldsymbol{y} \right) \\ &\equiv \hat{\boldsymbol{\eta}} = f_\theta(\hat{\boldsymbol{\eta}}; \boldsymbol{y}), \end{aligned}$$

where $\hat{\boldsymbol{\eta}}$ denotes the fixed point in the network and f_θ denotes each iteration of the whole NNs with a fixed parameter θ, which contains the the trainable network R_θ.

To optimize network parameters θ, stochastic gradient descent [3] is used to minimize a loss function as follows:

$$\theta^* = \arg\min_\theta \frac{1}{m} \sum_{i=1}^{m} \ell \left(f_\theta \left(\hat{\boldsymbol{x}}_i; \boldsymbol{y}_i, \right), \boldsymbol{h}_i^\star \right),$$

where m is the number of training samples, $\hat{\boldsymbol{x}}_i$ denotes the fixed point generated by the network iteration, $\boldsymbol{h}_i^\star$ is the ground truth channel of the i-th training sample, and $\boldsymbol{y}_i$ is the paired measurement. $\ell(\cdot, \cdot)$ is the loss function, defined by the mean squared error (MSE), as

$$\ell \left(\hat{\boldsymbol{x}}, \boldsymbol{h}^\star \right) = \frac{1}{2} \left\| \hat{\boldsymbol{x}} - \boldsymbol{h}^\star \right\|_2^2.$$

Then, we calculate the loss gradient. Let ℓ be an abbreviation of $\ell \left(\hat{\boldsymbol{x}}, \boldsymbol{h}^\star \right)$, then the loss gradient is [15]:

$$\frac{\partial \ell}{\partial \theta} = \left[\frac{\partial f_\theta(\hat{\boldsymbol{x}}; \boldsymbol{y})}{\partial \theta} \right]^\top \left[\boldsymbol{I} - \left. \frac{\partial f_\theta(\boldsymbol{x}; \boldsymbol{y})}{\partial \boldsymbol{x}} \right|_{\boldsymbol{x} = \hat{\boldsymbol{x}}} \right]^{-\top} (\hat{\boldsymbol{x}} - \boldsymbol{h}^\star),$$

where $^{-\top}$ denotes the inversion followed by transpose.

Using DEQ model, the memory complexity of our approach stays at $O(1)$ [16], independent of iteration count, and notably lower than deep unfolding's $O(T)$, enabling efficient gradient calculation for the loss term with minimal memory demand.

C. Implementation and Convergence analysis

Based on the PR algorithm and the DEQ model, we utilize the fixed point equation from (6) to construct a fixed point network operator f_θ. According to Algorithm 1, it can be seen that the algorithm consists of three linear iterations (steps 4,6,7) and one nonlinear iteration (step 5) within one loop. As the convolutional neural network (CNN) block R_θ to approximate the nonlinear proximal term, denoted as $R_\theta \to \mathrm{Prox}_{\sigma^{-1} g}$. Then, combined with other linear terms, the iteration of $\boldsymbol{\eta}^k$ in the DEQ network can be represented as

$$\boldsymbol{\eta}_R^{k+1} = f_\theta(\boldsymbol{\eta}_R^k) = f_{\mathrm{LT}_3} \circ f_{\mathrm{LT}_2} \circ R_\theta \circ f_{\mathrm{LT}_1}(\boldsymbol{\eta}_R^k). \tag{7}$$

Here, f_{LT_i} denotes the i-th linear term (LT) in Algorithm 1, and $\boldsymbol{\eta}_R^k$ denotes the intermediate variable produced by the network. We call the whole framework as Peaceman-Rachford splitting method implemented with Deep Equilibrium Network (denoted as PR-DEN for short), and Fig. 3 illustrates the iterative process of the PR-DEN approach.

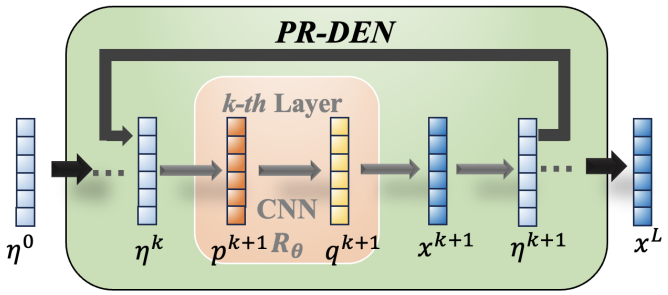


Fig. 3: Peaceman-Rachford splitting method implemented with Deep Equilibrium Network (PR-DEN) approach.

Next, we focus on demonstrating the theoretical advantages of our network architecture. We can not only strictly guarantee its convergence theoretically, but also prove the optimality of the converged solution. Our proof is based on the observation that the reflected resolvent $\mathbf{R}_{\sigma\mathbf{M}}$ is nonexpansive [30, Corollary 23.11], i.e., the Lipschitz constant of $\mathbf{R}_{\sigma\mathbf{M}}$ is not larger one. Hence, its composition $\mathbf{T}_\sigma^{\mathbf{PR}}$ is also nonexpansive. Without loss of generality, we assume $\mathrm{Lip}(\mathbf{T}_\sigma^{\mathbf{PR}}) < 1$, since when $\mathrm{Lip}(\mathbf{T}_\sigma^{\mathbf{PR}}) = 1$, a perturbation parameter ε can be added to construct a contraction mapping [30, Proposition 4.20] and the sequence of iteration converge to the same solution. Thus, we can claim that the intermediate variable $\boldsymbol{\eta}^k$ in the PR-DEN approach converges.

Theorem 2. *There exists a CNN $R_\theta(\cdot)$ which approximates the corresponding proximal operator, such that the sequence $\{\boldsymbol{\eta}_R^k\}$ produced by PR-DEN is strictly convergent, and a linear convergence rate is achieved.*

Proof. We utilize the Universal Approximation Theorem [33] to prove the existence of an network operator R_θ approximating the proximal term, and then employ the Banach Fixed Point Theorem to demonstrate convergence. The detailed proof is provided in the Appendix. □

Up to now, we have proved that our PR-DEN approach is convergent, with a linear convergence rate. In the following, we will prove that the iterative solution produced by PR-DEN is convergent to the optimal solution of dual problem (4).

Theorem 3. *There exists a CNN R_θ sufficiently approximates $\mathrm{Prox}_{\sigma^{-1}g}$, such that $\boldsymbol{x}_R$ converges to $\boldsymbol{x}_*$, where $\boldsymbol{x}_R$ is the solution produced by PR-DEN, and $\boldsymbol{x}_*$ is the optimal solution of dual problem (4).*

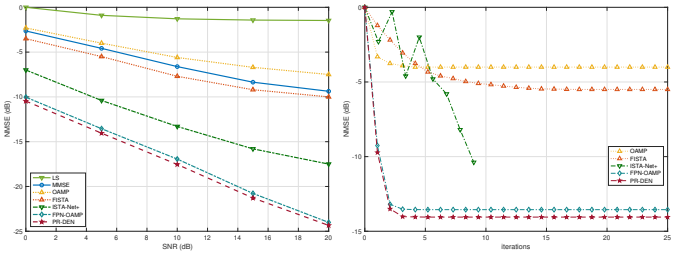
Proof. We establish the convergence by analyzing the error between the solution given by PR-DEN and the optimal solution. The detailed proof can be found in the Appendix. □

IV. Numerical Results

This section presents the experimental simulation results of our PR-DEN algorithm as shown in Section III. Specifically, the simulations are conducted using the hybrid-field channel model illustrated in subsection II-B. The detailed parameter setting is given in Table II of the Appendix. Our performance metric is the Normalized Mean Square Error (NMSE), defined as $\mathrm{NMSE} = 10\log_{10}\frac{\|\mathbf{h}-\hat{\mathbf{h}}\|}{\|\mathbf{h}\|}$, where $\mathbf{h}$ denotes the testing channel while $\hat{\mathbf{h}}$ is the estimated channel in the simulation. The source code is publicly available on GitHub [2].

We calculate the NMSE of comprehensive testing datasets comprising 5,000 samples. The benchmarks include (a) the classical channel estimation methods with closed-form expression: **LS**, **MMSE** [4], (b) classical iterative algorithm including **OAMP** [8], **FISTA** [22], and (c) two NN-based iterative algorithms: **ISTA-Net+** [11], a classical deep unfolding method and **FPN-OAMP** [3], a recently developed NN-based estimator for THz Ultra-Massive MIMO Channel Estimation.

In our approach, the convolutional layers are equipped with 3×3 kernels and a fixed number of 64 feature maps. For a comprehensive comparison with the NN-based algorithms, we train all networks with an 80,000-sample training set and a 5,000-sample validation set. Additionally, we conduct training for 150 epochs utilizing the Adam optimizer, initialized with a learning rate of 0.001 and a batch size of 128.



(a) NMSE Performance versus SNR. (b) NMSE versus iterations.

Fig. 4: NMSE performance versus SNR in two scenarios, the curves represent: LS (Green), MMSE (Blue), OAMP (Orange), Fista (Dark Green), ISTA-Net+ (Red dashed), FPN-OAMP (Maroon dash-dot), and PR-DEN (Brown dash-dot).

Fig. 4 (a) illustrates the NMSE performance versus SNR of our PR-DEN and six benchmarks in MIMO system with hybrid channels. The results clearly illustrate that PR-DEN significantly outperforms LS and MMSE for different SNR values. Moreover, in comparison to network-based unfolding algorithms such as ISTA-Net+ and FPN-OAMP, we observe a notable improvement over ISTA-Net+, and approximately a 3%-5% performance gain compared to FPN-OAMP.

Fig. 4 (b) illustrates the NMSE performance over iteration steps at an SNR of 5 dB, compared with the benchmarks, excluding LS and MMSE since these two algorithms are not iterative algorithms. The results demonstrate that PR-DEN achieves rapid convergence within 4 iterations. Furthermore, when compared to classical algorithms such as OAMP, FISTA, and ISTA-Net, our approach exhibits superior NMSE performance as early as the second iteration. Additionally, compared to FPN-OAMP, our approach converges faster and has better performance in terms of the NMSE validation.

Table I illustrates a comparison on the computational times between six benchmarks and the PR-DEN algorithm. The

[2]https://github.com/wushitong1234/PR-DEN

TABLE I: Running time of different algorithms

	OAMP	FISTA	ISTA-Net+	FPN-OAMP	PR-DEN
Center Processor	0.982	0.103	0.0492	0.00439	0.00912
Graphics Processor	/	/	0.00552	0.000361	0.000506

Notes: a) The computation time (in seconds) for all algorithms is averaged over five instances at different SNR levels (0:5:20 dB). b) We set a uniform convergence criterion across all algorithms, i.e., $\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^{k}\| < 10^{-2}$.

computational time is calculated respectively, using a center processor (typical examples known as Central Processing Units) and a graphics processor (typical examples known as Graphics Processing Units). Compared to traditional iterative algorithms such as OAMP and FISTA, PR-DEN demonstrates significantly lower computational time. In comparison with the network unfolding algorithms, our approach exhibits lower execution time than ISTA-Net+ and is comparable to FPN-OAMP. These experimental findings validate the efficiency of our algorithm in handling the channel estimation problem.

V. Conclusion

This paper introduces a novel channel estimation methodology for MIMO system, addressing the dual problem through the Peaceman-Rachford (PR) splitting approach with the deep equilibrium (DEQ) network. This approach constructs non-expansive operators using PR splitting on the Fenchel conjugate of the channel estimation problem, forming a fixed point equation. The DEQ model implicitly learns the proximal operators, enabling efficient iterative steps with constant memory. Theoretical analysis based on monotone operator theory provides a convergence proof. The simulations demonstrate the approach's effectiveness, that our approach has high accuracy and efficiency for channel estimation in MIMO system, compared with six benchmarks.

References

[1] H. Sarieddeen, M.-S. Alouini, and T. Y. Al-Naffouri, "An overview of signal processing techniques for terahertz communications," *Proceedings of the IEEE*, vol. 109, no. 10, pp. 1628–1665, 2021.

[2] X. Wei and L. Dai, "Channel estimation for extremely large-scale massive mimo: Far-field, near-field, or hybrid-field?" *IEEE Communications Letters*, vol. 26, no. 1, pp. 177–181, 2021.

[3] W. Yu, Y. Shen, H. He, X. Yu, J. Zhang, and K. B. Letaief, "Hybrid far- and near-field channel estimation for thz ultra-massive mimo via fixed point networks," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 5384–5389.

[4] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, *MIMO-OFDM wireless communications with MATLAB*. John Wiley & Sons, 2010.

[5] M. Myllyla, J.-M. Hintikka, J. R. Cavallaro, M. Juntti, M. Limingoja, and A. Byman, "Complexity analysis of mmse detector architectures for mimo ofdm systems," in *Conference Record of the Thirty-Ninth Asilomar Conference onSignals, Systems and Computers, 2005.* IEEE, 2005, pp. 75–81.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[7] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[8] J. Ma and L. Ping, "Orthogonal amp," *IEEE Access*, vol. 5, pp. 2020–2033, 2017.

[9] D. Gilton, G. Ongie, and R. Willett, "Deep equilibrium architectures for inverse problems in imaging," *IEEE Transactions on Computational Imaging*, vol. 7, pp. 1123–1133, 2021.

[10] W. Yu, Y. Shen, H. He, X. Yu, S. Song, J. Zhang, and K. B. Letaief, "An adaptive and robust deep learning framework for thz ultra-massive mimo channel estimation," *IEEE Journal of Selected Topics in Signal Processing*, 2023.

[11] J. Zhang and B. Ghanem, "Ista-net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1828–1837.

[12] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for mimo detection," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1702–1715, 2020.

[13] J. Liu, X. Xu, W. Gan, U. Kamilov *et al.*, "Online deep equilibrium learning for regularization by denoising," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 363–25 376, 2022.

[14] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "A model-driven deep learning network for mimo detection," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2018, pp. 584–588.

[15] S. Bai, J. Z. Kolter, and V. Koltun, "Deep equilibrium models," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[16] Y. Zhao, S. Zheng, and X. Yuan, "Deep equilibrium models for video snapshot compressive imaging," *arXiv preprint arXiv:2201.06931*, 2022.

[17] C. Pabbaraju, E. Winston, and J. Z. Kolter, "Estimating lipschitz constants of monotone deep equilibrium models," in *International Conference on Learning Representations*, 2020.

[18] P.-L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM Journal on Numerical Analysis*, vol. 16, no. 6, pp. 964–979, 1979.

[19] A. Alkhateeb, G. Leus, and R. W. Heath, "Limited feedback hybrid precoding for multi-user millimeter wave systems," *IEEE transactions on wireless communications*, vol. 14, no. 11, pp. 6481–6494, 2015.

[20] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[21] S. Boyd and L. Vandenberghe, *Introduction to applied linear algebra: vectors, matrices, and least squares*. Cambridge university press, 2018.

[22] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[23] Z. Hu, G. Liu, Q. Xie, J. Xue, D. Meng, and D. Gündüz, "A learnable optimization and regularization approach to massive mimo csi feedback," *IEEE Transactions on Wireless Communications*, 2023.

[24] S. Rangan, P. Schniter, and A. K. Fletcher, "Vector approximate message passing," *IEEE Transactions on Information Theory*, vol. 65, no. 10, pp. 6664–6684, 2019.

[25] K. Dovelos, M. Matthaiou, H. Q. Ngo, and B. Bellalta, "Channel estimation and hybrid combining for wideband terahertz massive mimo systems," *IEEE Journal on selected Areas in communications*, vol. 39, no. 6, pp. 1604–1620, 2021.

[26] G. Zhang, Y. Yuan, and D. Sun, "An Efficient HPR Algorithm for the Wasserstein Barycenter Problem with O(Dim(P)/ε) computational complexity," *arXiv preprint arXiv:2211.14881*, 2022.

[27] P. L. Combettes, "The geometry of monotone operator splitting methods," *arXiv preprint arXiv:2310.08443*, 2023.

[28] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970, vol. 18.

[29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[30] H. H. Bauschke, P. L. Combettes, and et al., *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2011, vol. 408.

[31] A. Davydov, S. Jafarpour, A. V. Proskurnikov, and F. Bullo, "Non-euclidean monotone operator theory with applications to recurrent neural networks," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 6332–6337.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[33] P. Kidger and T. Lyons, "Universal approximation with deep narrow networks," in *Conference on learning theory*. PMLR, 2020, pp. 2306–2327.

APPENDIX

Parameter	Value
Number of RF chain/ UPA	$N_{\text{RF}}/\quad S = 4$
Number of BS antennas	$N = 1024$
Number of antennas in each UPA	$N/S = 256$
Carrier frequency	$f_c = 300\,\text{GHz}$
Antenna spacing	$d = 5.0 \times 10^{-4}\,\text{m}$
UPA spacing	$d_{\text{UPA}} = 5.6 \times 10^{-2}\,\text{m}$
Pilot length	$P = 128$
Azimuth AoA	$\theta_l \sim \mathcal{U}(-\pi/2, \pi/2)$
Elevation AoA	$\phi_l \sim \mathcal{U}(-\pi, \pi)$
Angle of incidence	$\varphi_{\text{in},l} \sim \mathcal{U}(0, \pi/2)$
Number of paths	$L = 5$
Rayleigh distance	$D_{\text{Rayleigh}} = 20\,\text{m}$
LoS path length	$r_1 = 30\,\text{m}$
Scatterer distance ($l > 1$)	$r_l \sim \mathcal{U}(10, 25)\,\text{m}$
Time delay of LoS path	$\tau_1 = 100\,\text{nsec}$
Time delay of NLoS paths ($l > 1$)	$\tau_l \sim \mathcal{U}(100, 110)\,\text{nsec}$
Absorption coefficient	$k_{\text{abs}} = 0.0033\,\text{m}^{-1}$
Refractive index	$n_t = 2.24 - j0.025$
Roughness factor	$\sigma_{\text{rough}} = 8.8 \times 10^{-5}\,\text{m}$

TABLE II: Parameter Setting

TRANSFORMATION DETAILS

Let $\boldsymbol{y} = \left[\Re(\bar{\boldsymbol{y}})^T, \Im(\bar{\boldsymbol{y}})^T\right]^T \in \mathbb{R}^{2N_{\text{RF}}P\times 1}$, $\boldsymbol{h} = \left[\Re(\bar{\boldsymbol{h}})^T, \Im(\bar{\boldsymbol{h}})^T\right]^T \in \mathbb{R}^{2N\times 1}$, $\boldsymbol{n} = \left[\Re(\bar{\boldsymbol{n}})^T, \Im(\bar{\boldsymbol{n}})^T\right]^T \in \mathbb{R}^{2N_{\text{RF}}P\times 1}$, and

$$\mathbf{A} = \begin{pmatrix} \Re(\overline{\mathbf{A}}) & -\Im(\overline{\mathbf{A}}) \\ \Im(\overline{\mathbf{A}}) & \Re(\overline{\mathbf{A}}) \end{pmatrix} \in \mathbb{R}^{2N_{\text{RF}}P\times 2N}.$$

Thus, equation (1) can be transformed into (2).

DERIVATION FOR THE DUAL

the problem (3) is equivalently expressed by introducing slackness variable $\boldsymbol{p}, \boldsymbol{q}$:

$$\begin{aligned} \min_{\boldsymbol{p},\boldsymbol{q}} \quad & g(\boldsymbol{p}) + f(\boldsymbol{q}) \\ \text{s.t.} \quad & \boldsymbol{p} - \boldsymbol{q} = 0. \end{aligned}$$

The Lagrangian $\mathcal{L}(\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{x})$ for this problem with the Lagrange multiplier $\boldsymbol{x}$ (also known as the dual variable) is given by:

$$\mathcal{L}(\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{x}) = g(\boldsymbol{p}) + f(\boldsymbol{q}) + \boldsymbol{x}^\top(\boldsymbol{p} - \boldsymbol{q}).$$

Minimizing the Lagrangian over $\boldsymbol{p}$ and $\boldsymbol{q}$ gives:

$$\inf_{\boldsymbol{p},\boldsymbol{q}} \mathcal{L}(\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{x}) = \inf_{\boldsymbol{p}}(g(\boldsymbol{p}) + \boldsymbol{x}^\top\boldsymbol{p}) + \inf_{\boldsymbol{q}}(f(\boldsymbol{q}) - \boldsymbol{x}^\top\boldsymbol{q}).$$

The Fenchel conjugates f^* and g^* are defined as:

$$f^*(\boldsymbol{x}) = \sup_{\boldsymbol{y}}(\boldsymbol{x}^\top\boldsymbol{y} - f(\boldsymbol{y})), \quad g^*(\boldsymbol{x}) = \sup_{\boldsymbol{y}}(\boldsymbol{x}^\top\boldsymbol{y} - g(\boldsymbol{y})).$$

Thus, the dual problem is:

$$\max_{\boldsymbol{x}\in\mathbb{R}^{2N}} \left\{-g^*(-\boldsymbol{x}) - f^*(\boldsymbol{x})\right\}.$$

DERIVATION OF ALGORITHM 1

We prove this proposition by induction. For $k = 0$, we use the definition of $\boldsymbol{q}^1$ in Algorithm 1, yielding the condition $0 \in \partial f(\boldsymbol{q}^1) - \boldsymbol{\eta}^0 + \sigma\boldsymbol{q}^1$. Thus, $\boldsymbol{\eta}^0 - \sigma\boldsymbol{q}^1 \in \partial f(\boldsymbol{q}^1)$. Invoking Theorem 23.5 in [28], we deduce that

$$\boldsymbol{q}^1 \in \partial f^*(\boldsymbol{\eta}^0 - \sigma\boldsymbol{q}^1).$$

Hence, $\sigma\boldsymbol{q}^1 = \sigma\partial f^*(\boldsymbol{\eta}^0 - \sigma\boldsymbol{q}^1)$. This means that

$$\boldsymbol{\eta}^0 \in \boldsymbol{\eta}^0 - \sigma\boldsymbol{q}^1 + \sigma\partial f^*(\boldsymbol{\eta}^0 - \sigma\boldsymbol{q}^1).$$

Denote that $\mathbf{M}_2 = \partial f^*$, it follows that

$$\boldsymbol{\eta}^0 \in \boldsymbol{\eta}^0 - \sigma\boldsymbol{q}^1 + \sigma\mathbf{M}_2(\boldsymbol{\eta}^0 - \sigma\boldsymbol{q}^1),$$

Let $\boldsymbol{w}^k \triangleq \mathbf{J}_{\sigma\mathbf{M}_2}(\boldsymbol{\eta}^{k-1})$, which implies

$$\boldsymbol{w}^1 = J_{\sigma\mathbf{M}_2}(\boldsymbol{\eta}^0) = \boldsymbol{\eta}^0 - \sigma\boldsymbol{q}^1.$$

In parallel, by the stipulation of $\boldsymbol{p}^1$, we observe that

$$0 \in \partial g(\boldsymbol{p}^1) + (\boldsymbol{\eta}^0 - 2\sigma\boldsymbol{q}^1 + \sigma\boldsymbol{p}^1).$$

It follows from Theorem 23.5 in [28] that

$$\boldsymbol{p}^1 \in \partial g^*(-(\boldsymbol{\eta}^0 - 2\sigma\boldsymbol{q}^1 + \sigma\boldsymbol{p}^1)).$$

Hence, $-\sigma\boldsymbol{p}^1 = -\sigma\partial g^*(-(\boldsymbol{\eta}^0 - 2\sigma\boldsymbol{q}^1 + \sigma\boldsymbol{p}^1)$, implying that $2(\boldsymbol{\eta}^0 - \sigma\boldsymbol{q}^1) - \boldsymbol{\eta}^0 \in 2(\boldsymbol{\eta}^0 - \sigma\boldsymbol{q}^1) - \boldsymbol{\eta}^0 + \sigma\boldsymbol{p}^1 - \sigma\partial g^*(-(\boldsymbol{\eta}^0 - 2\sigma\boldsymbol{q}^1 + \sigma\boldsymbol{p}^1))$.
Since $\mathbf{M}_1 = \partial(g^* \circ (-\mathbf{I}))$, we have

$$2(\boldsymbol{\eta}^0 - \sigma\boldsymbol{q}^1) - \boldsymbol{\eta}^0 \in \boldsymbol{\eta}^0 - 2\sigma\boldsymbol{q}^1 + \sigma\boldsymbol{p}^1 + \sigma M_1(\boldsymbol{\eta}^0 - 2\sigma\boldsymbol{q}^1 + \sigma\boldsymbol{p}^1),$$

which implies $\boldsymbol{x}^1 := \mathbf{J}_{\sigma\mathbf{M}_1}(2\boldsymbol{w}^1 - \boldsymbol{\eta}^0) = \boldsymbol{\eta}^0 - 2\sigma\boldsymbol{q}^1 + \sigma\boldsymbol{p}^1$.
Hence, we have $\boldsymbol{\eta}^1 := \boldsymbol{\eta}^0 + 2(\boldsymbol{x}^1 - \boldsymbol{w}^1) = \boldsymbol{\eta}^0 + 2\sigma(\boldsymbol{p}^1 - \boldsymbol{q}^1)$.
Then, it follows that

$$\begin{aligned} \boldsymbol{\eta}^1 &= \boldsymbol{\eta}^0 + 2(\boldsymbol{x}^1 - \boldsymbol{w}^1) = \boldsymbol{\eta}^0 + 2(\mathbf{J}_{\sigma\mathbf{M}_1}(2\boldsymbol{w}^1 - \boldsymbol{\eta}^0) - \mathbf{J}_{\sigma\mathbf{M}_2}(\boldsymbol{\eta}^0)) \\ &= \boldsymbol{\eta}^0 + 2(\mathbf{J}_{\sigma\mathbf{M}_1}(2\mathbf{J}_{\sigma\mathbf{M}_2}(\boldsymbol{\eta}^0)) - \mathbf{J}_{\sigma\mathbf{M}_1}(-\boldsymbol{\eta}^0) - \mathbf{J}_{\sigma\mathbf{M}_2}(\boldsymbol{\eta}^0)) \\ &= (2\mathbf{J}_{\sigma\mathbf{M}_1} - \mathbf{I})(2\mathbf{J}_{\sigma\mathbf{M}_2} - \mathbf{I})(\boldsymbol{\eta}^0) = \mathbf{R}_{\sigma\mathbf{M}_1}\mathbf{R}_{\sigma\mathbf{M}_2}\boldsymbol{\eta}^0. \end{aligned}$$

It follows that the update of $\boldsymbol{\eta}^1$ is the same as our Algorithm. Hence, we prove the statement for $k = 0$. Assume that the statement holds for some $k \geq 1$. For $k := k+1$, we can prove that the statement holds similarly to the case $k = 0$. Thus, we prove the statement holds for any $k \geq 0$ by induction.

PROOF OF THEOREM 1

Since $\text{Fix}\left(\mathbf{T}_\sigma^{\text{PR}}\right)$ is a closed convex set [30, Corollary 4.24], the fixed point is unique. Using the convergence result of [31, Theorem 22], we can directly obtain that the sequence $\{\boldsymbol{\eta}^k\}$ converges to the fixed point $\boldsymbol{\eta}^*$ of the PR iteration. Since the resolvent $\boldsymbol{J}_{\sigma M_2}$ is nonexpansive and $\boldsymbol{w}^{k+1} = \boldsymbol{J}_{\sigma M_2}\left(\eta^k\right)$ as derived in the proof of Proposition 1, we directly obtain

$$\lim_k \boldsymbol{x}^k = \lim_k \boldsymbol{J}_{\sigma M_2}\left(\boldsymbol{\eta}^k\right) = \boldsymbol{J}_{\sigma M_2}\left(\boldsymbol{\eta}^*\right).$$

As show in [18], if $\boldsymbol{\eta}^*$ is the fixed point of the PR iteration, then $\boldsymbol{x}^* = \boldsymbol{J}_{\sigma M_2}\left(\boldsymbol{\eta}^*\right)$. Hence, we prove the convergence of sequence $\{\boldsymbol{x}^k\}$ to the optimal solution $\boldsymbol{x}^*$ of (4).

Next, we prove the convergence of the sequence $\left\{\boldsymbol{q}^k\right\}$. According to Algorithm 1, we have for all $k \geq 0$,

$$\boldsymbol{q}^{k+1} = (\mathbf{A}^\top\mathbf{A} + \sigma\mathbf{I})^{-1}(\mathbf{A}^\top\boldsymbol{y} + \boldsymbol{\eta}^k).$$

Denote $\hat{f}(\boldsymbol{q}) := f(\boldsymbol{q}) + \frac{\sigma}{2}\|\boldsymbol{q}\|^2$, which is a strongly convex function. Thus, $\hat{f}^*$ is essentially smooth [28, Theorem 26.3]. The first-order optimality condition of the above iteration of $\boldsymbol{q}^{k+1}$ implies $0 \in \partial \hat{f}\left(\boldsymbol{q}^{k+1}\right) - \boldsymbol{\eta}^k$. Since $\hat{f}$ is a proper closed convex function, by [28, Theorem 23.5], the first-order optimality condition is equivalent to

$$\boldsymbol{q}^{k+1} = \nabla \hat{f}^*\left(\boldsymbol{\eta}^k\right).$$

It follows from the convergence of $\{\boldsymbol{\eta}^k\}$ and the continuity of $\nabla \hat{f}^*$ [28, Theorem 25.5] that $\{\boldsymbol{q}^k\}$ is convergent. Note that

$$\boldsymbol{\eta}^{k+1} = \boldsymbol{\eta}^k + 2\sigma(\boldsymbol{p}^{k+1} - \boldsymbol{q}^{k+1})$$

from Algorithm 1, and the sequence $\{\boldsymbol{\eta}^k\}$ is convergent, we obtain $\lim_k(\boldsymbol{p}^k - \boldsymbol{q}^k) = 0$. Then, by taking the limit, we obtain $\lim_k \boldsymbol{p}^k = \lim_k \boldsymbol{q}^k = \boldsymbol{q}^*$. Hence, $\{\boldsymbol{p}_k\}$ is convergent.

Assume that $(\boldsymbol{p}^*, \boldsymbol{q}^*, \boldsymbol{x}^*)$ is the limit point of the sequence $\{\boldsymbol{p}^k, \boldsymbol{q}^k, \boldsymbol{x}^k\}$. Since $\boldsymbol{\eta}^{k+1} = \boldsymbol{\eta}^k + 2\sigma(\boldsymbol{p}^{k+1} - \boldsymbol{q}^{k+1})$ from Algorithm 1, we can obtain $\boldsymbol{p}^* - \boldsymbol{q}^* = \mathbf{0}$ by taking the limit. It follows that

$$\lim_k \boldsymbol{w}^k = \lim_k[\boldsymbol{w}^k + \sigma(\boldsymbol{p}^k - \boldsymbol{q}^k)] = \lim_k \boldsymbol{x}^k = \boldsymbol{x}^*.$$

By Algorithm 1 and the derivation in Proposition 1, we have

$$0 \in \partial f(\boldsymbol{q}^{k+1}) - \boldsymbol{\eta}^k + \sigma \boldsymbol{q}^{k+1}.$$

Since $\boldsymbol{w}^{k+1} = \boldsymbol{\eta}^k - \sigma \boldsymbol{q}^{k+1}$, we have

$$\boldsymbol{w}^{k+1} \in \partial f\left(\boldsymbol{q}^{k+1}\right).$$

Similarly, by Algorithm 1 and the derivation details in Proposition 1, we have

$$0 \in \partial g(\boldsymbol{p}^{k+1}) + (\boldsymbol{\eta}^k - 2\sigma \boldsymbol{q}^{k+1} + \sigma \boldsymbol{p}^{k+1}).$$

Since $\boldsymbol{x}^{k+1} = \boldsymbol{\eta}^k - 2\sigma \boldsymbol{q}^{k+1} + \sigma \boldsymbol{p}^{k+1}$, we have

$$-\boldsymbol{x}^{k+1} \in \partial g(\boldsymbol{p}^{k+1}).$$

Hence, we have

$$\boldsymbol{w}^{k+1} \in \partial f\left(\boldsymbol{q}^{k+1}\right), \quad -\boldsymbol{x}^{k+1} \in \partial g\left(\boldsymbol{p}^{k+1}\right),$$

Together with $\boldsymbol{p}^* - \boldsymbol{q}^* = \mathbf{0}$ and taking limit, we have

$$\boldsymbol{x}^* \in \partial f\left(\boldsymbol{q}^*\right), \quad -\boldsymbol{x}^* \in \partial g\left(\boldsymbol{p}^*\right), \quad \boldsymbol{p}^* - \boldsymbol{q}^* = 0.$$

This completes the proof [28, Corollary 28.3.1].

Proof of Theorem 2

As indicated above, the iteration formula for $\boldsymbol{\eta}^k$ in the proposed PR-DEN can be succinctly represented as:

$$\boldsymbol{\eta}_R^{k+1} = f_{\mathrm{LT}_3} \circ f_{\mathrm{LT}_2} \circ R_\theta \circ f_{\mathrm{LT}_1}(\boldsymbol{\eta}_R^k).$$

That is,

$$\begin{aligned}\boldsymbol{\eta}_R^{k+1} =& f_{\mathrm{LT}_3} \circ f_{\mathrm{LT}_2} \circ (R_\theta - \mathrm{Prox}_{\sigma^{-1}g} + \mathrm{Prox}_{\sigma^{-1}g}) \circ f_{\mathrm{LT}_1}(\boldsymbol{\eta}_R^k) \\ =& f_{\mathrm{LT}_3} \circ f_{\mathrm{LT}_2} \circ (R_\theta - \mathrm{Prox}_{\sigma^{-1}g}) \circ f_{\mathrm{LT}_1}(\boldsymbol{\eta}_R^k) \\ &+ f_{\mathrm{LT}_3} \circ f_{\mathrm{LT}_2} \circ \mathrm{Prox}_{\sigma^{-1}g} \circ f_{\mathrm{LT}_1}(\boldsymbol{\eta}_R^k).\end{aligned}$$

From the proof of proposition 1, we obtain

$$\mathbf{T}_\sigma^{\mathbf{PR}}(\boldsymbol{\eta}_R^k) = f_{\mathrm{LT}_3} \circ f_{\mathrm{LT}_2} \circ \mathrm{Prox}_{\sigma^{-1}g} \circ f_{\mathrm{LT}_1}(\boldsymbol{\eta}_R^k).$$

Thus, we derive

$$\boldsymbol{\eta}_R^{k+1} = f_{\mathrm{LT}_3} \circ f_{\mathrm{LT}_2} \circ (R_\theta - \mathrm{Prox}_{\sigma^{-1}g}) \circ f_{\mathrm{LT}_1}(\boldsymbol{\eta}_R^k) + \mathbf{T}_\sigma^{\mathbf{PR}}(\boldsymbol{\eta}_R^k). \tag{8}$$

By the Universal approximation theorem [33], there exists a network $R_\theta(\cdot)$ approximating the proximal operator $\mathrm{Prox}_{\sigma^{-1}g}$ with arbitrary precision. Suppose $\max\{\|f_{\mathrm{LT}_1}\|, \|f_{\mathrm{LT}_2}\|, \|f_{\mathrm{LT}_3}\|\} = \alpha$, $\mathrm{Lip}(\mathbf{T}_\sigma^{\mathbf{PR}}) = \beta$. There exists R_θ, such that

$$\|(R_\theta - \mathrm{Prox}_{\sigma^{-1}g})(\boldsymbol{\eta}_R^k)\| < \frac{1-\beta}{\alpha^3}\|\boldsymbol{\eta}_R^k\|. \tag{9}$$

Thus, combined (9) with (8), we derive

$$\|\boldsymbol{\eta}_R^{k+1}\| < (1-\beta)\|\boldsymbol{\eta}_R^k\| + \beta\|\boldsymbol{\eta}_R^k\| = \|\boldsymbol{\eta}_R^k\|.$$

From Banach fix point theorem, $\{\boldsymbol{\eta}_R^k\}$ is convergent.

Proof of Theorem 3

From Theorem 2, we derive the convergence of $\{\boldsymbol{\eta}_R^k\}$, which denotes the sequence produced by PR-DEN. That is

$$\lim_{k\to\infty} \boldsymbol{\eta}_R^k = \boldsymbol{\eta}_R,$$

where $\boldsymbol{\eta}_R$ is the solution of the following fix point equation:

$$\boldsymbol{\eta}_R = f_{\mathrm{LT}_3} \circ f_{\mathrm{LT}_2} \circ R_\theta \circ f_{\mathrm{LT}_1}(\boldsymbol{\eta}_R). \tag{10}$$

Additionally, since the dual problem (4) is convex, the optimal solution $\mathbf{x}_*$ is the equivalent to the solution of the following problem [18]:

$$\boldsymbol{x}_* = \mathbf{J}_{\sigma\mathbf{M_2}}(\boldsymbol{\eta}_*) \quad \text{s.t.} \quad \boldsymbol{\eta}_* = \mathbf{R}_{\sigma\mathbf{M}_1}\mathbf{R}_{\sigma\mathbf{M}_2}(\boldsymbol{\eta}_*). \tag{11}$$

Thus, subtracting $\boldsymbol{\eta}_R$ in (10) from $\boldsymbol{\eta}_*$ in (11), we obtain

$$\boldsymbol{\eta}_R - \boldsymbol{\eta}_* = \mathbf{T}_\sigma^{\mathbf{PR}}(\boldsymbol{\eta}_R - \boldsymbol{\eta}_*) + f_{\mathrm{LT}_3} \circ f_{\mathrm{LT}_2} \circ (R_\theta - \mathrm{Prox}_{\sigma^{-1}g}) \circ f_{\mathrm{LT}_1}(\boldsymbol{\eta}_R).$$

Suppose $\max\{\|f_{\mathrm{LT}_1}\|, \|f_{\mathrm{LT}_2}\|, \|f_{\mathrm{LT}_3}\|\} = \alpha$, $\mathrm{Lip}(\mathbf{T}_\sigma^{\mathbf{PR}}) = \beta$, then

$$\|\boldsymbol{\eta}_R - \boldsymbol{\eta}_*\| \le \beta\|\boldsymbol{\eta}_R - \boldsymbol{\eta}_*\| + \alpha^3\|(R_\theta - \mathrm{Prox}_{\sigma^{-1}g}\|\|\boldsymbol{\eta}_R)\|.$$

Hence, we have

$$\|\boldsymbol{\eta}_R - \boldsymbol{\eta}_*\| \le \frac{\alpha^3}{1-\beta}\|(R_\theta - \mathrm{Prox}_{\sigma^{-1}g})\|\|\boldsymbol{\eta}_R\|.$$

We claim there exists a consistent upper bound M on $\boldsymbol{\eta}_R$.

$$\|\boldsymbol{\eta}_R\| \le \|f_{\mathrm{LT}_3} \circ f_{\mathrm{LT}_2} \circ R_\theta \circ f_{\mathrm{LT}_1}(\boldsymbol{\eta}_R)\| + 1. \tag{12}$$

By the Universal approximation theorem [33], there exists a network $R_\theta(\cdot)$ approximating the proximal operator $\mathrm{Prox}_{\sigma^{-1}g}$ with arbitrary precision. This indicates that for any $\varepsilon > 0, \boldsymbol{\eta} \in \mathbb{X}$, we have

$$\|f_{\mathrm{LT}_3} \circ f_{\mathrm{LT}_2} \circ (R_\theta - \mathrm{Prox}_{\sigma^{-1}g}) \circ f_{\mathrm{LT}_1}(\boldsymbol{\eta})\| < \epsilon$$

when R_θ sufficiently approximates $\mathrm{Prox}_{\sigma^{-1}g}$.

It demonstrates that

$$\mathrm{Lip}(f_{\mathrm{LT}_3} \circ f_{\mathrm{LT}_2} \circ R_\theta \circ f_{\mathrm{LT}_1}) < 1.$$

Combined with (12), it convinces the fact that $\|\boldsymbol{\eta}_R\|$ is bounded. Here, we denote the upper bound as M. Thus,

$$\|\boldsymbol{\eta}_R - \boldsymbol{\eta}_*\| \le \frac{\alpha^3 M}{1-\beta}\|(R_\theta - \mathrm{Prox}_{\sigma^{-1}g})\|.$$

Since $R_\theta \to \mathrm{Prox}_{\sigma^{-1}g}$, it can be indicated that

$$\boldsymbol{\eta}_R \to \boldsymbol{\eta}_*.$$

Therefore,

$$\boldsymbol{x}_R = \mathbf{J}_{\sigma\mathbf{M}_1}(\boldsymbol{\eta}_R) \to \mathbf{J}_{\sigma\mathbf{M}_1}(\boldsymbol{\eta}_*) = \boldsymbol{x}_*.$$