Phase Retrieval via Sparse Perturbed Amplitude Flow

Lan Li, Xiaoya Liu, Xiaoyan Liu, Yulong Mao

***Abstract*—Phase retrieval refers to reconstruct signal phase information only from acquired intensity or amplitude information. As the problem is underdetermined, its solution space admits multiple solutions. Additionally, the non-smooth absolute value term of the loss function may negatively impact the numerical results of Amplitude Flow. To address this issue, we propose a sparse perturbed based smooth loss function and is termed the Sparse Perturbed Amplitude Flow (SPAF) algorithm. The approach effectively constrains the signal solution space, reduces the number of required measurements. And mitigates the instability caused by near-zero of absolute value term that can lead to abrupt gradient changes. First, the initial value of the SPAF algorithm is obtained by sparse orthogonal initialization, then the exact solution is obtained after a series of hard thresholding iterations. Finally, the global convergence of the SPAF algorithm is also demonstrated. The SPAF algorithm does not require any truncation and reweighting process. Therefore, it is straightforward to achieve outstanding performance for both real and complex signals. Substantial tests confirm that the proposed algorithm significantly surpasses other state-of-the-art methods in recovery efficiency and convergence speed.**

***Index Terms*—Sparse phase retrieval, Perturbed amplitude flow, Support recovery, Gradient descent, Linear convergence.**

I. Introduction

In various scientific and engineering disciplines, reconstructing a signal from measurements without phase information is a fundamental problem, known as phase retrieval (PR). Since optical sensors record only the intensity of light rather not its phase, PR plays a critical role in related applications. It is widely used in X-ray crystallography [1], optics [2], biological imaging [3], and computational imaging [4]. Mathematically, PR is described by the following equation model:

$$y_i = |\langle \boldsymbol{a_i}, \boldsymbol{x} \rangle|, i = 1, \cdots, m, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is original signal, $\{y_i\}_{i=1}^m$ are amplitude measurements, $|\cdot|$ represent absolute value. And the sensing vectors $\boldsymbol{a_i} \in \mathbb{R}^n$ follow $\boldsymbol{a_i} \sim \mathcal{N}(0, \boldsymbol{I}_n)$ for $i = 1, \ldots, m$.

Manuscript received December 5, 2024; revised May 20, 2025. This work was supported by the Natural Science Basic Research Program in Shaanxi Province of China (Program Nos. 2025JC-YBMS-078 and 2023-JC-YB-027), the Graduate Innovation and Practical Ability Training Project of Xi'an Shiyou University (YCS23213183).

Lan Li is a professor at the School of Science, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (e-mail: lanli@xsyu.edu.cn).

Xiaoya Liu is a postgraduate student of the School of Science, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (e-mail: 2015573234@qq.com).

Xiaoyan Liu is an associate professor of School of Science, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (e-mail: liuxiaoyan@xsyu.edu.cn).

Yulong Mao is a postgraduate student of the School of Science, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China. (e-mail: 1141628827@qq.com).

The classical methods for solving the PR problem rely on alternating projection, such as the Gerchberg-Saxton (GS) algorithm [5] and Hybrid Input-Output (HIO) algorithm [6]. To address the high complexity associated with dimensionality, Candès proposed the Wirtinger flow (WF) algorithm [7]. This algorithm first obtains a relatively optimal initial estimate with spectral method, and then is followed by iterative refinement of the solution through intensity-based models:

$$\min_z f(\boldsymbol{z}) := \frac{1}{4m} \sum_{i=1}^{m} \left(\left| \boldsymbol{a}_i^T \boldsymbol{z} \right|^2 - y_i^2 \right)^2. \tag{2}$$

Based on the WF algorithm, the Truncated WF (TWF) [8] and the Reweighted WF (RWF) algorithm [9] are dicussed. Zhang [10] further demonstrated that performance can be significantly improved by minimizing the following amplitude-based models:

$$\min_z f(\boldsymbol{z}) := \frac{1}{2m} \sum_{i=1}^{m} \left(\left| \boldsymbol{a}_i^T \boldsymbol{z} \right| - y_i \right)^2, \tag{3}$$

where this method is summarized as Amplitude Flow (AF) algorithm. The AF class of algorithms had been proven to achieve more accurate recovery from $O(n)$ measurements, outperforming the WF class of algorithms [11]. To further improve the effectiveness of AF, Wang proposed Truncated AF (TAF) algorithm [12] and Reweighted AF (RAF) algorithm [13]. These algorithms incorporate truncation and reweighting operations in gradient step, respectively.

In practical applications, although problem (1) can be efficiently solved if the measurement values are overcompleted. However, a major difficulty remains to accurately recover the signal using fewer measurement values. The sparse phase retrieval algorithms extend the corresponding methods of general phase retrieval problem (1) to sparse phase retrieval problem (4).

The Stochastic Alternating Minimization (SAM) algorithm [14] and Compressive Phase Retrieval with Alternating Minimization (CoPRAM) [15] extended problem (1) by applying alternating minimization under sparsity constraints. A reweighted gradient descent approach is employed in the Sparse Reweighted Threshold Wirtinger Flow (SRThWF) algorithm [16] to progressively improve the initial estimate, and demonstrated its effectiveness when the sparsity level is unknown.

Most current sparse phase retrieval algorithms are commonly first-order methods, so an second-order Hard Threshold Tracking (HTP) algorithm [17] is proposed, which can accelerate the recovery process by a factor of several compared to conventional sparse phase retrieval algorithms. Several other methods, including the Sparse Truncated AF (SPARTA) algorithm [18], Compressed RAF (CRAF)

algorithm [19], and Sparse WF (SWF) algorithm [20], also incorporate a sparse prior to problem (1) for recovering sparse signals.

From a theoretical perspective, to distinguish gradient components that are either too large or too small during the iteration process, the TWF, TAF, and SPARTA algorithms employ truncation to eliminate components with excessive influence. In contrast, the RWF and RAF algorithms regulate these components by attenuating their weight at each update. However, these methods often lead to inconsistent search directions and increase computational complexity.

To address these theoretical deficiencies and limitations. Pinilla et al. introduced Phase Retrieval Smoothing Conjugate Gradient (PR-SCG) algorithm [21]. This approach substitutes the original loss function of phase retrieval model with a specially designed smooth function, and employs the projected gradient method to tackle the recovery problem. Additionally, Stochastic Smoothing Phase Retrieval (SSPR) algorithm is presented. This cumulative method entails an equation for each iteration, which is a fast method for large dimensions signal. In the Smoothed Amplitude Flow (SAF) algorithm [22], smooth function is introduced by Luo et al. to handle the non-smooth nature of the loss function.

In addition, Gao et al. also putted forward a Perturbed Amplitude Flow (PAF) algorithm [23], which introduces a parameter to mitigate the influence of undesirable observations. Meanwhile maintaining the smooth loss function and serving a similar role of truncation or reweighting. Xiao et al. utilized the smooth approximation of the absolute value function in smoothed amplitude flow-based phase retrieval (SAFPR) algorithm [24], which reduces the computational complexity. And J. Bacca et al. applied smooth phase retrieval algorithm to super-resolution imaging [25], enabling high-resolution images to be recovered from low-resolution in any diffraction regions.

To overcome the non-smooth characteristics of the objective function and use fewer measurement values. A Sparse Perturbated Amplitude Flow (SPAF) algorithm is presented to recover sparse signals from phaseless measurements. By controls the magnitude of the gradient component with a smoothing parameter to maintain the smoothness of the objective function. The principal contributions of this paper are categorized into the following three aspects.

(1). In contrast to previous phase retrieval methods, the SPAF algorithm incorporates sparse prior and smoothing strategy to minimize sampling and computational complexity effectively, while significantly improving recovery performance.

(2). The convergence of the SPAF algorithm is rigorously validated through mathematical analysis, providing theoretical assurance of its reliability under certain assumptions.

(3). The high recovery accuracy and robustness of the SPAF algorithm are validated by experiments on both real-valued and complex-valued signals.

II. Related Work

A. Sparse phase retrieval problem

For large-scale problems, the enormous measurement and computational costs make it impractical to satisfy $m > n$. Therefore, it is a standard assumed that $\boldsymbol{x}$ exhibits sparsity or nearly sparsity. We consider the minimization of the composite function $f = S \circ L$, where $S : \mathbb{R}^n \to \mathbb{R}$ is a convex function and $L : \mathbb{R}^d \to \mathbb{R}^m$ is a nonlinear mapping. Then the variable sequence x_t is iteratively updated by

$$\boldsymbol{x}_{t+1} \in \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} S\left(L(\boldsymbol{x}_t) + L'(\boldsymbol{x}_t)(\boldsymbol{x} - \boldsymbol{x}_t)\right), \quad (4)$$

where $L'(\boldsymbol{x}_t)$ denotes the derivative of L with respect to $\boldsymbol{x}$ in t iterations. Due to the local approximation of L at $\boldsymbol{x}_t$ in (4), $\boldsymbol{x}_{t+1}$ is obtained a local optimum solution. Then in sparse phase retrieval problem

$$S(\boldsymbol{z}) = \|\boldsymbol{z}\|_0,$$

and since AF is better than WF in terms of recovery performance, so we have

$$L(\boldsymbol{z}) = \left(\left|\langle \boldsymbol{a}_i^T, \boldsymbol{x}\rangle\right| - y_i\right)^2, \quad i = 1, \ldots, m,$$

so, the amplitude-based sparse phase retrieval algorithm can be expressed as:

$$\min_{\|\boldsymbol{z}\|_0=k} f(\boldsymbol{z}) := \frac{1}{2m}\sum_{i=1}^{m}\left(\left|\boldsymbol{a}_i^T\boldsymbol{z}\right| - y_i\right)^2, \quad (5)$$

where k denotes the sparsity level and $\|\cdot\|_0$ denotes a zero-paradigm operator, representing the count of non-zero elements. The update rule in (4) can be described as:

$$\boldsymbol{x}_{t+1} \in \mathbf{arg}\min_{\boldsymbol{x}\in\mathbb{R}^n}\sum_{i=1}^{m}\left(\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle - sign(\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle)\cdot y_i\right)^2. \quad (6)$$

Assume that the sparsity level is known and satisfied $k \ll n$. It is also shown that the solution to problem (6) is uniquely identifiable with real general measurements if the measurement value $m = 2k$ is satisfied. Moreover, by Incorporating sparsity prior can constrain the underdetermined phase retrieval problem. It effectively narrows the solution space while mitigating noise-induced artifacts.

B. Smooth phase retrieval problem

The Eq.(5) is a nonconvex and nonsmooth function, which the nonsmooth absolute value term may degrade the numerical performance of the AF. During the optimization process, the absolute value term $|\boldsymbol{a}_i^T\boldsymbol{z}|$ in the loss function is not differentiable at $\boldsymbol{z} = 0$. Specifically, when the value of $\boldsymbol{a}_i^T\boldsymbol{z}$ approaches 0, the gradient exhibits abrupt variations. This sharp transition increases computational complexity with inconsistency in search directions. In other words, when trapped at an undesirable stationary point $\boldsymbol{z}$, a sign difference between $\boldsymbol{a}_i^T\boldsymbol{z}$ and $\boldsymbol{a}_i^T\boldsymbol{x}$ may arise, that is $\boldsymbol{a}_i^T\boldsymbol{z} \neq \boldsymbol{a}_i^T\boldsymbol{x}$.

To avoid this situation, a mathematical model for smooth phase retrieval has been offered:

$$\min_{\boldsymbol{z}} f_\epsilon(\boldsymbol{z}) := \min_{\boldsymbol{z}}\frac{1}{m}\sum_{i=1}^{m}\left(\sqrt{|\boldsymbol{a}_i^*\boldsymbol{z}|^2 + \epsilon_i^2} - \sqrt{b_i^2 + \epsilon_i^2}\right)^2, \quad (7)$$

where ϵ_i is a perturbated term and the gradient size can be controlled with a suitable choice of ϵ_i. This is essential for preventing the extremely large gradient components.

III. Proposed Method

In this paper, a model of Sparse Perturbed Amplitude-Flow(SPAF) algorithm is proposed through the following formula:

$$\min_{\|\boldsymbol{z}\|_0=k} f_{\boldsymbol{\theta}}(\boldsymbol{z}) := \frac{1}{2m}\sum_{i=1}^{m}\left(\sqrt{\left|\boldsymbol{a}_i^T\boldsymbol{z}\right|^2+\theta_i^2}-\sqrt{y_i^2+\theta_i^2}\right)^2, \tag{8}$$

where $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_m)$ are prescribed values and the smoothness of the loss function depends on the different values of θ_i. When $\theta_i = 0$, the model is abbreviated to the classical amplitude-based formula (5). If $S(\boldsymbol{z}) \neq \|\boldsymbol{z}\|_0$, formula (8) represents phase retrieval without incorporating the sparse prior, thereby simplifying to formula (7). Clearly, the loss function in (8) is transformed into a nonconvex and smooth loss function, and the next step is to derive the exact solution.

The SPAF algorithm consists of two main parts. In the initialization process, the signal's support set is inferred and assess a estimated value of signal by sparse orthogonality-promoting initialization method. Then, a gradient descent approach with Hard Thresholding is applied in the iterative phase. The following sections describe this two parts respectively.

A. Sparse Orthogonality-promoting Initialization

Correct initialization is essential in non-convex optimization to prevent convergence to unfavorable stationary points. It is pointed out in the TAF algorithm that random vectors are typically close to being orthogonal to each other in high dimensions. The normalized inner product of squares is then given by:

$$\cos^2\beta_i := \frac{|\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle|^2}{\|\boldsymbol{a}_i\|_2^2\|\boldsymbol{x}\|_2^2} = \frac{y_i^2}{\|\boldsymbol{a}_i\|_2^2\|\boldsymbol{x}\|_2^2},$$

where β_i is the angle between vectors $\boldsymbol{a}_i$ and $\boldsymbol{x}$, and $\cos^2\beta_i$ is used as a measure of the orthogonality between them, and the smaller the value, the more orthogonal they are. Thus, the key idea adopted for the initialization focuses on approximating vector $\boldsymbol{x}$ with a different vector namely maximally orthogonal to a subset $\{\boldsymbol{a}_i\}_{i\in I_0}$ of the carefully chosen sampling vectors, where $[m] := \{1, 2, \cdots, m\}$, and $I_0 \subseteq [m]$ denotes the set of indexes of the sampling vectors $\boldsymbol{a}_i$ chosen for the initialization of the computation involved.

To reduce computational complexity, the largest eigenvector can serve as an approximation to the smallest eigenvector. So, $\overline{I_0}$ represents the complement of I_0 within $[m]$. Hence, $\boldsymbol{x}$ can be estimated by a vector with index $\overline{I_0}$ as $\boldsymbol{a_i}$, then:

$$\begin{aligned}\hat{\boldsymbol{z}}_0 &:= \arg\max_{\|z\|_2=1}\frac{1}{\left|\overline{I_0}\right|}\sum_{i\in\overline{I_0}}\left|\boldsymbol{a}_i^T\boldsymbol{z}\right|^2 \\ &= \arg\max_{\|z\|_2=1}\boldsymbol{z}^T\left(\frac{1}{\left|\overline{I_0}\right|}\sum_{i\in\overline{I_0}}\frac{\boldsymbol{a}_i\boldsymbol{a}_i^T}{\|\boldsymbol{a}_i\|_2^2}\right)\boldsymbol{z}.\end{aligned} \tag{9}$$

Since the signal $\boldsymbol{x}$ to be recovered is k-sparse and $k \ll n$, the objective of this paper is to recover the sparse signals. In order to obtain informations from fewer measurements, using ℓ_0 regularization to represent the sparsity prior, the above initialization changes to:

$$\hat{\boldsymbol{z}}_0 := \arg\max_{\|z\|_2=1}\boldsymbol{z}^T\boldsymbol{Y}\boldsymbol{z} \quad s.t.\ \|\boldsymbol{z}\|_0 = k, \tag{10}$$

where

$$\boldsymbol{Y} = \left(\frac{1}{\left|\overline{I_0}\right|}\sum_{i\in\overline{I_0}}\frac{\boldsymbol{a}_i\boldsymbol{a}_i^T}{\|\boldsymbol{a}_i\|_2^2}\right).$$

The question is generally NP-hard problem due to the combinatorial constraint. So, This paper applies a orthogonality-promoting strategy to obtain a sparse initial estimate with constrained measurements. This method is divided into two primary components. And, the subsequent sections provide a detailed explanation of each part.

1) Accurate Support Recovery: For sparse signals, the available data samples are considerably fewer than the signal dimension n. Hence, it is essential to determine the support domain of the original signal, which relies on appropriate rules. Once this domain is approximated, an initial estimate of the sparsity can be meaningfully inferred from finite measurements.

Beginning with the recovery process of support domain, we assume under general premises that $\boldsymbol{x}$ exists within the support of $S \subseteq [n] := \{1, \cdots, n\}$, where the size of the support set $|S| = k$. Introduce following random variables:

$$\mathscr{W}_{i,j} = a_{i,j}^2 y_i^2, \tag{11}$$

where $j = 1, \cdots, n$. $\mathbb{E}\left[a_{i,j}^4\right] = 3$ and $\mathbb{E}\left[a_{i,j}^2\right] = 1$ are applied to normalized Gaussian variables and owing to the rotational invariance property of the Gaussian distribution, it is clear that:

$$\mathbb{E}\left[\mathscr{W}_{i,j}\right] = 2x_j^2 + \|\boldsymbol{x}\|_2^2, \tag{12}$$

where $\boldsymbol{x}_{/j}, \boldsymbol{a}_{i,/j}$ are acquired by excluding the j-the element from $\boldsymbol{x}, \boldsymbol{a}_i^T$. For $j \in S$ and $j \notin S$, there exists a separation of at least $2x_j^2$ for the expected value of $\mathscr{W}_{i,j}$. When the interval values are large enough, the set of indices associated with the k largest values of $\mathbb{E}\left[\mathscr{W}_{i,j}\right]$ is enough to recover the support domain $\boldsymbol{S}$ of $\boldsymbol{x}$. But, $\{\mathbb{E}\left[\mathscr{W}_{i,j}\right]\}$ is not available in practice. The ensemble average can be approximated by the sample mean under the strong law of large numbers, so, the average $\overline{\mathscr{W}}_{i,j}$ defined as $\frac{1}{m}\sum_{i=1}^{m}\mathscr{W}_{i,j}$ converges to the expected value $\mathbb{E}\left[\mathscr{W}_{i,j}\right]$ as m increases. So the sample mean $\overline{\mathscr{W}}_{i,j}$ is used as an estimate of the expectation value $\mathbb{E}\left[\mathscr{W}_{i,j}\right]$. Therefore, it is necessary to collect the indices corresponding to the top-k values in $\mathbb{E}\left[\mathscr{W}_{i,j}\right]$, which form the support domain of the estimate:

$$\hat{S} := \left\{j \in [n] \mid \text{indices of top-}k \text{ elements in } \left\{\overline{\mathscr{W}}_{i,j}\right\}_{j=1}^{n}\right\}. \tag{13}$$

It ensures that the elements belonging to the support set have sufficient separation from those not in the support set, and recovers $\hat{S}$ exactly with high probability.

2) Orthogonality-Promoting Intialization: When the support in formula (13) is accurate, the initial value is estimated based on $\hat{S}$ and the measurement y_i can be rewritten as:

$$y_i = \left|\boldsymbol{a}_{i,\hat{S}}^T\boldsymbol{x}_{\hat{S}}\right|, \tag{14}$$

where $\boldsymbol{a}_{i,\hat{S}}$ and $\boldsymbol{x}_{\hat{S}}$ include the elements $a_{i,j}$ and $\boldsymbol{x}$ of index j belonging to $\hat{S}$, respectively. The initialization step for promoting orthogonality is utilized for the reduced-dimensional data $\left\{\left(\boldsymbol{a}_{i,\hat{S}}, y_i\right)\right\}_{i=1}^{m}$, namely:

$$\hat{\boldsymbol{z}}_{0,\hat{S}} = \underset{\|z_{\hat{S}}\|_2=1}{argmax} \frac{1}{|\overline{I}_0|} \boldsymbol{z}_{\hat{S}}^T \left(\sum_{i\in\overline{I}_0} \frac{\boldsymbol{a}_{i,\hat{S}} \boldsymbol{a}_{i,\hat{S}}^T}{\left\|\boldsymbol{a}_{i,\hat{S}}\right\|_2^2} \right) \boldsymbol{z}_{\hat{S}}. \tag{15}$$

A k-sparse n dimensional initialization $\hat{\boldsymbol{z}}_0$ is reconstructed by padding $\hat{\boldsymbol{z}}_{0,S}$ with zeros at entries with indices not included in $\hat{S}$. When $\|\boldsymbol{x}\|_2 \neq 1$, the norm estimate of $\boldsymbol{x}$ is used to rescale $\hat{\boldsymbol{z}}_0$ to obtain the final k-sparse initial estimate:

$$\boldsymbol{z}_0 = \sqrt{\frac{1}{m}\sum_{i=1}^{m} y_i} \hat{\boldsymbol{z}}_0. \tag{16}$$

B. Hard thresholding based gradient descent

To find globally optimal solution, a hard-thresholding gradient descent algorithm is utilized, with $\boldsymbol{z}_0$ serving as the initialization. From the Wirtinger derivative, the gradient $\nabla f_\theta(\boldsymbol{z}_t)$ of the loss function $f_\theta(\boldsymbol{z}_t)$ connection with $\boldsymbol{z}$ is:

$$\nabla f_\theta(\boldsymbol{z}_t) := \tfrac{1}{m}\sum_{i=1}^{m} \left(\frac{\sqrt{|\boldsymbol{a}_i^T \boldsymbol{z}_t|^2 + \theta_i^2} - \sqrt{y_i^2 + \theta_i^2}}{\sqrt{|\boldsymbol{a}_i^T \boldsymbol{z}_t|^2 + \theta_i^2}} \right) (\boldsymbol{a}_i^T \boldsymbol{z}_t) \boldsymbol{a}_i \tag{17}$$

Following the acquisition of the initial estimate z_0 in Sparse Orthogonality-promoting Initialization, this paper refines z_0 via multiple gradient updates. The k-sparse hard thresholding operation is applied in these iterations, that is:

$$\boldsymbol{z}_{t+1} := \Gamma_k(\boldsymbol{z}_t - \lambda \nabla f_\theta(\boldsymbol{z}_t)), \tag{18}$$

where t denotes iteration count, λ is the step size. $\Gamma_k(\cdot)$ is the k-sparse hard-thresholding operator, keeping the top-k entries while replacing others with zero.

It can be clearly concluded, in order to prevent abrupt changes in the gradient when the value of $|\boldsymbol{a}_i^T \boldsymbol{z}|$ approaches 0, which could lead the iteration in an undesirable direction, the smoothing mechanism adopted by the loss function mitigates the influence of most bad gradient components associated with incorrectly estimated signs. This significantly reduces the computational complexity arising from inconsistencies in the search direction. The key procedures of the proposed algorithm are summarized in Algorithm 1.

Algorithm 1 SPAF: Sparse Perturbed Amplitude Flow algorithm

Input
$\{\boldsymbol{a}_i\}_{i=1}^m$; $\{y_i\}_{i=1}^m$; the iteration upper bound T; sparsity level k; step length λ; perturbed parameter $\boldsymbol{\theta}$
Exact Support Recovery
In $\left(\frac{1}{m}\sum_{i=1}^m y_i^2 a_{i,j}^2\right)$, set support $\hat{S}$
$\hat{S} := \left\{ j \in [n] \mid \text{indices of top-}k \text{ elements in } \left\{\overline{\mathscr{W}}_{i,j}\right\}_{j=1}^n \right\}$
Initialization Evaluation
Calculate sparse initial estimates
$\hat{\boldsymbol{z}}_{0,\hat{S}} = \underset{\|\boldsymbol{z}_{\hat{S}}\|_2=1}{argmax} \frac{1}{|\overline{I}_0|} \boldsymbol{z}_{\hat{S}}^T \left(\sum_{i\in\overline{I}_0} \frac{\boldsymbol{a}_{i,\hat{S}} \boldsymbol{a}_{i,\hat{S}}^T}{\|\boldsymbol{a}_{i,\hat{S}}\|_2^2} \right) \boldsymbol{z}_{\hat{S}}$
Initialize $\boldsymbol{z}_0 = \sqrt{\frac{1}{m}\sum_{i=1}^m y_i} \hat{\boldsymbol{z}}_0$, where $\hat{\boldsymbol{z}}_0$ is determined through augmenting $\hat{\boldsymbol{z}}_{0,\hat{S}}$ with 0 at entries with their indices not in $\hat{S}$
Hard Thresholding Based Gradient Descent
For $\|\boldsymbol{z}_{t+1} - \boldsymbol{z}_t\|_2 \geq \epsilon$ to $t \leq T$
$\boldsymbol{z}_{t+1} = \Gamma_k \left(\boldsymbol{z}_t - \frac{\lambda}{m} \sum_{i=1}^m \left(\frac{\sqrt{|\boldsymbol{a}_i^T \boldsymbol{z}_t|^2 + \theta_i^2} - \sqrt{y_i^2 + \theta_i^2}}{\sqrt{|\boldsymbol{a}_i^T \boldsymbol{z}_t|^2 + \theta_i^2}} \right) (\boldsymbol{a}_i^T \boldsymbol{z}_t) \boldsymbol{a}_i \right)$
$t = t + 1$
Output
The final estimated value $\boldsymbol{z}_T$

IV. Convergence Proof

Under the Gaussian model, this section establishes the global convergence of Algorithm 1. The geometric rate of convergence for the SPAF algorithm is formally stated in the following theorem.

Lemma 1. [18] Let $\boldsymbol{x} \in \mathbb{R}^n$ is a k-sparse signal in support $\hat{S}$ and minimum non-zero entries $x_{\min} := \min_{j\in\hat{S}} |x_j|$, which is on the order of $\left(\frac{1}{\sqrt{k}}\right)\|\boldsymbol{x}\|_2$. If $\{\boldsymbol{a}_i\}_{i=1}^m$ are i.i.d standard Gaussian, support $\hat{S}$ with a probability at least $1 - \frac{6}{m}$ given that $m \geq C_0 k^2 \log(mn)$, where C_0 is a positive constant.

Lemma 2. [18] Under the conditions of Lemma 1, for $\boldsymbol{z}_0 = \sqrt{\sum_{i=1}^m y_i^2/m}\hat{\boldsymbol{z}}_0$, where $\hat{\boldsymbol{z}}_0$ obtained through the sparse orthogonality-promoting initialization, then with probability at least $1 - (m+6)e^{(-k/2)} - \frac{7}{m}$, the following holds

$$\text{dist}(\boldsymbol{z}_0, \boldsymbol{x}) \leq (1/10)\|\boldsymbol{x}\|_2$$

under the condition that $m \geq C_0' k$, where $C_0' > 0$ is a absolute constant.

To proceed with the discussion in this section, we revisit the principle of the Restricted Isometry Property (RIP), which serves as a fundamental condition in compressive sensing theory.

Definition 1. [26] Let $s = 1, 2, \ldots, k$ are any integer. For every k-sparse vectors $\mathbf{u}$, the isometry constant $0 < \omega_s < 1$ of a matrix $\mathbf{A}$ is defined as the smallest value ensuring the following holds:

$$(1 - \omega_k)\|\mathbf{u}\|_2^2 \leq \|\mathbf{A}\mathbf{u}\|_2^2 \leq (1 + \omega_k)\|\mathbf{u}\|_2^2$$

where $\mathbf{A}$ is a matrix with Gaussian-distributed entries that are i.i.d. standard normal variables, the matrix $\sqrt{\frac{1}{m}}\mathbf{A}$ satisfies the RIP condition with isometry constant $\omega_{3k} \leq \varepsilon$ with probability at least $1 - e^{-C_1' m}$, gived that $m \geq C_2' \varepsilon^{-2}(3k)\log(\frac{n}{3k})$, where $C_1', C_2' > 0$ are universal constants.

Additionally, if κ is a set of $\{1, 2, \ldots, n\}$ containing no more than $3k$ indices, then the following properties of $\mathbf{A}$ satisfied:

P1: For any vector $\mathbf{u}$ that is at most $3k$-sparse, the inequality holds

$$(1 - \omega_{3k})m\|\mathbf{u}\|_2 \leq \|\mathbf{A}_\kappa^T \mathbf{A}_\kappa \mathbf{u}\|_2 \leq (1 + \omega_{3k})m\|\mathbf{u}\|_2$$

P2: The inequality holds

$$\left\|\mathbf{A}_{\alpha\cup\beta}^T \mathbf{A}_{\alpha\cup\beta} - \boldsymbol{I}\right\|_2 \leq \omega_{3k}$$

where α and β are disjoint sets with combined cardinality that does not exceed $3k$.

Theorem 1. Under the conditions of Lemma 1 and Lemma 2, for any k-sparse vectors $\boldsymbol{x}$ and a suitable step size λ is chosen, the following inequality always holds in $t+1$ and t iterations:

$$\|\boldsymbol{z}_{t+1}-\boldsymbol{x}\|_2 \leq \rho\|\boldsymbol{z}_t-\boldsymbol{x}\|_2$$

where ρ is a constant and $\rho \in (0,1)$.

Proof: To begin, we introduce some notations that will be employed exclusively within this section. For any $t \geq 0$, define:

$$\boldsymbol{u}_{t+1} = \boldsymbol{z}_t - \tfrac{\lambda}{m}\sum_{i=1}^{m}\left(\frac{\sqrt{|\boldsymbol{a}_i^T\boldsymbol{z}_t|^2+\theta_i^2}-\sqrt{y_i^2+\theta_i^2}}{\sqrt{|\boldsymbol{a}_i^T\boldsymbol{z}_t|^2+\theta_i^2}}\right)(\boldsymbol{a}_i^T\boldsymbol{z}_t)\boldsymbol{a}_i$$

where represents the preliminary estimate before the hard thresholding operation is applied in equation (18). The support sets of $\boldsymbol{x}$ and $\boldsymbol{z}_t$ are represented by S^x and $\hat{S}_t$, respectively. Consequently, the reconstruction error $\boldsymbol{z}_{t+1}-\boldsymbol{x}$ lies within the support set $\Psi_{t+1} := S^x \cup \hat{S}_{t+1}$, while the error $\boldsymbol{z}_t - \boldsymbol{x}$ is supported on $\Psi_t := S^x \cup \hat{S}_t$. In addition, the set difference is specified that Ψ_t/Ψ_{t+1} as containing elements present in Ψ_t but not in Ψ_{t+1}. Due to the thresholding performed in every iteration, that is, $|S^x| = |\hat{S}_t| = k \geq |\hat{S}_{t+1}|$. So for all $t \geq 0$, $|\Psi_{t+1}| = |S^x \cup \hat{S}_{t+1}| \leq 2k$ and $|\Psi_{t+1} \cup \Psi_t| \leq 3k$.

Let $\boldsymbol{h}_t = \boldsymbol{z}_t - \boldsymbol{x}$, thus:

$$\begin{aligned}\|\boldsymbol{h}_{t+1}\|_2 = \|\boldsymbol{z}_{t+1}-\boldsymbol{x}\|_2 &\leq \left\|\boldsymbol{z}_{t+1}^{\Psi_{t+1}} - \boldsymbol{x}^{\Psi_{t+1}}\right\|_2 \\ &\leq \left\|\boldsymbol{z}_t^{\Psi_t} - \boldsymbol{u}_{t+1}^{\Psi_{t+1}}\right\|_2 + \left\|\boldsymbol{x}^{\Psi_{t+1}} - \boldsymbol{u}_{t+1}^{\Psi_{t+1}}\right\|_2, \quad (19)\end{aligned}$$

where $\boldsymbol{u}_{t+1}^{\Psi_{t+1}} := \boldsymbol{z}_t^{\Psi_{t+1}} - \frac{\lambda}{m}\nabla f_\theta^{\Psi_{t+1}}(\boldsymbol{z}_t)$. Since $\boldsymbol{z}_{t+1}^{\Psi_{t+1}}$ represents the k-best approximation of $\boldsymbol{u}_{t+1}^{\Psi_{t+1}}$ obtained through hard thresholding, and given that $|\Psi_{t+1}| \leq 2k$, it follows that:

$$\left\|\boldsymbol{x}^{\Psi_{t+1}} - \boldsymbol{u}_{t+1}^{\Psi_{t+1}}\right\|_2 \geq \left\|\boldsymbol{z}_{t+1}^{\Psi_{t+1}} - \boldsymbol{u}_{t+1}^{\Psi_{t+1}}\right\|_2.$$

Therefore, (18) can be reformulated as follows:

$$\|\boldsymbol{h}_{t+1}\|_2 \leq 2\left\|\boldsymbol{x}^{\Psi_{t+1}} - \boldsymbol{u}_{t+1}^{\Psi_{t+1}}\right\|_2. \quad (20)$$

Substituting $\boldsymbol{u}_{t+1}^{\Psi_{t+1}} := \boldsymbol{z}_t^{\Psi_{t+1}} - \frac{\lambda}{m}\nabla f_\theta^{\Psi_{t+1}}(\boldsymbol{z}_t)$ into (19), it follows that:

$$\begin{aligned}\|\boldsymbol{h}_{t+1}\|_2 &\leq 2\left\|\boldsymbol{z}_t^{\Psi_{t+1}} - \frac{\lambda}{m}\nabla f_\theta^{\Psi_{t+1}}(\boldsymbol{z}_t) - \boldsymbol{x}^{\Psi_{t+1}}\right\|_2 \\ &= 2\left\|\boldsymbol{h}_t^{\Psi_{t+1}} - \frac{\lambda}{m}\nabla f_\theta^{\Psi_{t+1}}(\boldsymbol{z}_t)\right\|_2. \quad (21)\end{aligned}$$

Due to

$$\nabla f_\theta^{\Psi_{t+1}}(\boldsymbol{z}_t) = \sum_{i=1}^{m}\left(\frac{\sqrt{|\boldsymbol{a}_i^T\boldsymbol{z}_t|^2+\theta_i^2}-\sqrt{y_i^2+\theta_i^2}}{\sqrt{|\boldsymbol{a}_i^T\boldsymbol{z}_t|^2+\theta_i^2}}\right)(\boldsymbol{a}_i^T\boldsymbol{z}_t)\boldsymbol{a}_i,$$

and let

$$L = \frac{\sqrt{|\boldsymbol{a}_i^T\boldsymbol{z}_t|^2+\theta_i^2}-\sqrt{y_i^2+\theta_i^2}}{\sqrt{|\boldsymbol{a}_i^T\boldsymbol{z}_t|^2+\theta_i^2}}.$$

Next, we will abbreviate L. When the difference between the two values $|\boldsymbol{a}_i^T\boldsymbol{z}_t|^2$ and y_i^2 becomes small after adding the bias term θ_i^2, the nonlinear effects of the square root function will diminish. Therefore, assume two proximate values $p = |\boldsymbol{a}_i^T\boldsymbol{z}_t|^2 + \theta_i^2$ and $q = y_i^2 + \theta_i^2$. Given that p and q are close to each other, we can simplify the expression by performing a Taylor expansion of $f(x) = \sqrt{x}$. In other words, within the term L, we can approximate the function $f(x) = \sqrt{x}$ expanded at $p = |\boldsymbol{a}_i^T\boldsymbol{z}_t|^2 + \theta_i^2$ to evaluate it at $q = y_i^2 + \theta_i^2$.

Thus, by substituting into the first-order Taylor expansion, we have:

$$\sqrt{q} \approx \sqrt{p} + \frac{1}{2\sqrt{p}}(q-p).$$

Then, integrating equations $p = |\boldsymbol{a}_i^T\boldsymbol{z}_t|^2+\theta_i^2$, $q = y_i^2+\theta_i^2$ and $\boldsymbol{a_i} \in \mathbb{R}^n$, hence:

$$\sqrt{(\boldsymbol{a}_i^T\boldsymbol{z}_t)^2+\theta_i^2} - \sqrt{y_i^2+\theta_i^2} = \frac{\left((\boldsymbol{a}_i^T\boldsymbol{z}_t)^2+\theta_i^2-(y_i^2+\theta_i^2)\right)}{2\sqrt{(\boldsymbol{a}_i^T\boldsymbol{z}_t)^2+\theta_i^2}}.$$

Because $\sqrt{(\boldsymbol{a}_i^T\boldsymbol{z}_t)^2+\theta_i^2} > 0$, divide both sides by:

$$\frac{\sqrt{(\boldsymbol{a}_i^T\boldsymbol{z}_t)^2+\theta_i^2}-\sqrt{y_i^2+\theta_i^2}}{\sqrt{(\boldsymbol{a}_i^T\boldsymbol{z}_t)^2+\theta_i^2}} = \frac{1}{2}\frac{(\boldsymbol{a}_i^T\boldsymbol{z}_t)^2+\theta_i^2-(y_i^2+\theta_i^2)}{(\boldsymbol{a}_i^T\boldsymbol{z}_t)^2+\theta_i^2}.$$

In summary, it follows that:

$$\begin{aligned}L &= \frac{\sqrt{(\boldsymbol{a}_i^T\boldsymbol{z}_t)^2+\theta_i^2}-\sqrt{y_i^2+\theta_i^2}}{\sqrt{(\boldsymbol{a}_i^T\boldsymbol{z}_t)^2+\theta_i^2}} \\ &\approx \frac{(\boldsymbol{a}_i^T\boldsymbol{z}_t)^2+\theta_i^2-(y_i^2+\theta_i^2)}{(\boldsymbol{a}_i^T\boldsymbol{z}_t)^2+\theta_i^2}.\end{aligned}$$

From inequality $a^2+b^2 \geq 2ab$, then $(\boldsymbol{a}_i^T\boldsymbol{z}_t)^2+\theta_i^2$ can be simplified to $(\boldsymbol{a}_i^T\boldsymbol{z}_t)^2+\theta_i^2 \geq 2(\boldsymbol{a}_i^T\boldsymbol{z}_t)\theta_i$. Similarly, we have:

$$L \geq \frac{2(\boldsymbol{a}_i^T\boldsymbol{z}_t)\theta_i - 2y_i\theta_i}{2(\boldsymbol{a}_i^T\boldsymbol{z}_t)\theta_i} = \frac{(\boldsymbol{a}_i^T\boldsymbol{z}_t)-y_i}{(\boldsymbol{a}_i^T\boldsymbol{z}_t)}.$$

Also because $\boldsymbol{h}_t = \boldsymbol{z}_t - \boldsymbol{x}$, as a result:

$$L = \frac{\boldsymbol{a}_i^T\boldsymbol{h}_t}{\boldsymbol{a}_i^T\boldsymbol{h}_t + \boldsymbol{a}_i^T\boldsymbol{x}}. \quad (22)$$

Substitute (22) into $\nabla f_\theta^{\Psi_{t+1}}(\boldsymbol{z}_t)$:

$$\begin{aligned}\nabla f_\theta^{\Psi_{t+1}}(\boldsymbol{z}_t) &= \sum_{i=1}^{m}\left(\frac{\boldsymbol{a}_i^T\boldsymbol{h}_t}{\boldsymbol{a}_i^T\boldsymbol{h}_t+\boldsymbol{a}_i^T\boldsymbol{x}}\right)\boldsymbol{a}_i^T(\boldsymbol{h}_t+\boldsymbol{x})\boldsymbol{a}_i^{\Psi_{t+1}} \\ &= \sum_{i=1}^{m}\left(\frac{\boldsymbol{a}_i^T\boldsymbol{h}_t}{\boldsymbol{a}_i^T\boldsymbol{h}_t+\boldsymbol{a}_i^T\boldsymbol{x}}\right)\left(\boldsymbol{a}_i^T\boldsymbol{h}_t+\boldsymbol{a}_i^T\boldsymbol{x}\right)\boldsymbol{a}_i^{\Psi_{t+1}} \\ &= \sum_{i=1}^{m}\boldsymbol{a}_i^T\boldsymbol{h}_t\boldsymbol{a}_i^{\Psi_{t+1}}. \quad (23)\end{aligned}$$

The expression in (21) can be rewritten as:

$$\|\boldsymbol{h}_{t+1}\|_2 \leq 2\left\|\boldsymbol{h}_t^{\Psi_{t+1}} - \frac{\lambda}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^T\boldsymbol{h}_t\boldsymbol{a}_i^{\Psi_{t+1}}\right\|_2. \quad (24)$$

We can split $\boldsymbol{a}_i^T\boldsymbol{h}_t\boldsymbol{a}_i^{\Psi_{t+1}}$ into two parts:

$$\begin{aligned}\boldsymbol{a}_i^T\boldsymbol{h}_t\boldsymbol{a}_i^{\Psi_{t+1}} &= \boldsymbol{a}_i^{T,\Psi_{t+1}}\boldsymbol{h}_t^{\Psi_{t+1}}\boldsymbol{a}_i^{\Psi_{t+1}} \\ &\quad - \boldsymbol{a}_i^{T,\Psi_t/\Psi_{t+1}}\boldsymbol{h}_t^{\Psi_t/\Psi_{t+1}}\boldsymbol{a}_i^{\Psi_{t+1}}. \quad (19)\end{aligned}$$

As a result:

$$\begin{aligned}\|\boldsymbol{h}_{t+1}\|_2 &\le 2\left\|\boldsymbol{h}_t^{\Psi_{t+1}} - \frac{\lambda}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^T\boldsymbol{h}_t\boldsymbol{a}_i^{\Psi_{t+1}}\right\|_2 \\ &= 2\left\|\boldsymbol{h}_t^{\Psi_{t+1}} - \frac{\lambda}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^{T,\Psi_{t+1}}\boldsymbol{h}_t^{\Psi_{t+1}}\boldsymbol{a}_i^{\Psi_{t+1}}\right. \\ &\quad \left. - \frac{\lambda}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^{T,\Psi_t/\Psi_{t+1}}\boldsymbol{h}_t^{\Psi_t/\Psi_{t+1}}\boldsymbol{a}_i^{\Psi_{t+1}}\right\|_2 \\ &\le 2\left\|\boldsymbol{h}_t^{\Psi_{t+1}} - \frac{\lambda}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^{T,\Psi_{t+1}}\boldsymbol{h}_t^{\Psi_{t+1}}\boldsymbol{a}_i^{\Psi_{t+1}}\right\|_2 \\ &\quad + 2\frac{\lambda}{m}\left\|\sum_{i=1}^{m}\boldsymbol{a}_i^{T,\Psi_t/\Psi_{t+1}}\boldsymbol{h}_t^{\Psi_t/\Psi_{t+1}}\boldsymbol{a}_i^{\Psi_{t+1}}\right\|_2. \end{aligned} \quad (25)$$

As for the first term of (25), it is simple to confirm that:

$$\begin{aligned}&\left\|\boldsymbol{h}_t^{\Psi_{t+1}} - \frac{\lambda}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^{T,\Psi_{t+1}}\boldsymbol{h}_t^{\Psi_{t+1}}\boldsymbol{a}_i^{\Psi_{t+1}}\right\|_2 \\ &= \left\|\left(\boldsymbol{I} - \frac{\lambda}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^{T,\Psi_{t+1}}\boldsymbol{a}_i^{\Psi_{t+1}}\right)\boldsymbol{h}_t^{\Psi_{t+1}}\right\|_2 \\ &\le \left\|\boldsymbol{I} - \frac{\lambda}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^{T,\Psi_{t+1}}\boldsymbol{a}_i^{\Psi_{t+1}}\right\|_2 \|\boldsymbol{h}_t^{\Psi_{t+1}}\|_2 \\ &\le \max\{1-\lambda\underline{\tau}, \lambda\overline{\tau}-1\}\|\boldsymbol{h}_t^{\Psi_{t+1}}\|_2, \end{aligned} \quad (26)$$

where $\overline{\tau}$ and $\underline{\tau}$ are the largest eigenvalue and smallest eigenvalue of $\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^{\Psi_{t+1}}\boldsymbol{a}_i^{T,\Psi_{t+1}}$ respectively.

Next, we estimate eigenvalues $\overline{\tau}$ and $\underline{\tau}$ respectively. Because $|\Psi_{t+1}| \le 2k$, from **P1** of the RIP, it is clear that:

$$\overline{\tau} = \tau_{\max}\left(\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^{\Psi_{t+1}}\boldsymbol{a}_i^{T,\Psi_{t+1}}\right) \le 1+\omega_{2k} \quad (27)$$

$$\underline{\tau} = \tau_{\min}\left(\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^{\Psi_{t+1}}\boldsymbol{a}_i^{T,\Psi_{t+1}}\right) \ge 1-\omega_{2k} \quad (28)$$

Applying the results of (27) and (28) to (26):

$$\begin{aligned}&\left\|\boldsymbol{h}_t^{\Psi_{t+1}} - \frac{\lambda}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^{T,\Psi_{t+1}}\boldsymbol{a}_i^{\Psi_{t+1}}\boldsymbol{h}_t^{\Psi_{t+1}}\right\|_2 \\ &\le \max\{1-\lambda(1-\omega_{2k}), \lambda(1+\omega_{2k})-1\}\|\boldsymbol{h}_t^{\Psi_{t+1}}\|_2. \end{aligned} \quad (29)$$

Concerning the last term of (25), given that $|\Psi_{t+1}\cup\Psi_t| \le 3k$ and from **P2** of the RIP, then:

$$\begin{aligned}&\left\|\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^{T,\Psi_t/\Psi_{t+1}}\boldsymbol{a}_i^{\Psi_{t+1}}\boldsymbol{h}_t^{\Psi_t/\Psi_{t+1}}\right\|_2 \\ &\le \left\|\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^{\Psi_{t+1}}\boldsymbol{a}_i^{T,\Psi_t/\Psi_{t+1}}\right\|_2 \left\|\boldsymbol{h}_t^{\Psi_t/\Psi_{t+1}}\right\|_2 \\ &\le \left\|\boldsymbol{I} - \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{a}_i^{\Psi_t\cup\Psi_{t+1}}\boldsymbol{a}_i^{T,\Psi_t\cup\Psi_{t+1}}\right\|_2 \left\|\boldsymbol{h}_t^{\Psi_t/\Psi_{t+1}}\right\|_2 \\ &\le \omega_{3k}\left\|\boldsymbol{h}_t^{\Psi_t/\Psi_{t+1}}\right\|_2. \end{aligned} \quad (30)$$

By substituting (25) with (29) and (30) can be obtain:

$$\begin{aligned}\|\boldsymbol{h}_{t+1}\|_2 \le& 2\max\{1-\lambda(1-\omega_{2k}),\ \lambda(1+\omega_{2k})-1\} \\ &\|\boldsymbol{h}_t^{\Psi_{t+1}}\|_2 + 2\lambda\omega_{3k}\|\boldsymbol{h}_t^{\Psi_t/\Psi_{t+1}}\|_2, \end{aligned}$$

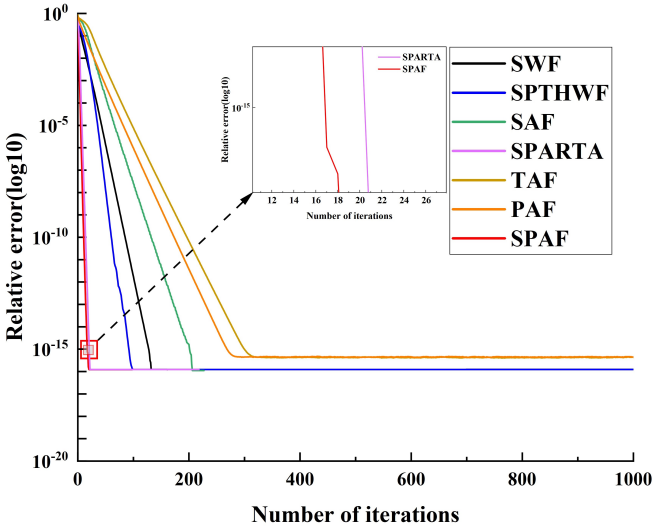

Fig. 1. Comparison of the convergence rate for different algorithms in real-valued case.

because $\Psi_{t+1}\cap(\Psi_t/\Psi_{t+1}) = \emptyset$, then:

$$\|\boldsymbol{h}_t^{\Psi_{t+1}}\|_2 + \|\boldsymbol{h}_t^{\Psi_t/\Psi_{t+1}}\|_2 \le \sqrt{2}\|\boldsymbol{h}_t\|_2.$$

So, we have

$$\begin{aligned}&\|\boldsymbol{h}_{t+1}\|_2 \quad (20)\\ \le& 2\max\{1-\lambda(1-\omega_{2k}), \lambda(1+\omega_{2k})-1\} \\ &\|\boldsymbol{h}_t^{\Psi_{t+1}}\|_2 + 2\lambda\omega_{3k}\|\boldsymbol{h}_t^{\Psi_t/\Psi_{t+1}}\|_2 \\ \le& 2\sqrt{2}\max\left\{\max\{1-\lambda(1-\omega_{2k}), \lambda(1+\omega_{2k})-1\}, \lambda\omega_{3k}\right\} \\ &\|\boldsymbol{h}_t\|_2 \\ =& \alpha\|\boldsymbol{h}_t\|_2, \end{aligned} \quad (31)$$

where $\alpha < 1$.

For sufficiently small $\omega_{3k} > 0$, we have $\rho = 1-\alpha$ and $\rho \in (0,1)$, thus establishes the result of linear convergence. □

V. Numerical Results

This section systematically tests the accuracy of theoretical derivation through multiple sets of numerical simulations. The comparison between the SPAF algorithm and other state-of-the-art gradient descent methods, containing SWF, SRThWF, SPARTA, PAF, SAF, and TAF algorithms. All experiments are completed in a 64-bit Windows 10 operating system environment equipped with an Intel Corei5-8250U processor and 8GB of memory, and MATLAB R2022a. Both real-valued and complex-valued Gaussian models are evaluated. To ensure fair comparison, the algorithmic parameters for all methods are assigned to the recommended values. Initialization for each method is achieved through 100 power iterations, followed by refinement with $T = 1000$ gradient descent iterations.

Regarding common notations used in this paper. Vectors are denoted using boldface lowercase letters, and matrices are represented by bold uppercase letters, with T standing for the transpose of a vector or matrix. $\|\cdot\|$ is the Euclidean norm. We define the distance between two vectors:

$$\text{dist}(\boldsymbol{x},\boldsymbol{z}) = \min_{\phi\in[0,2\pi)}\|\boldsymbol{z} - \boldsymbol{x}e^{i\phi}\|.$$

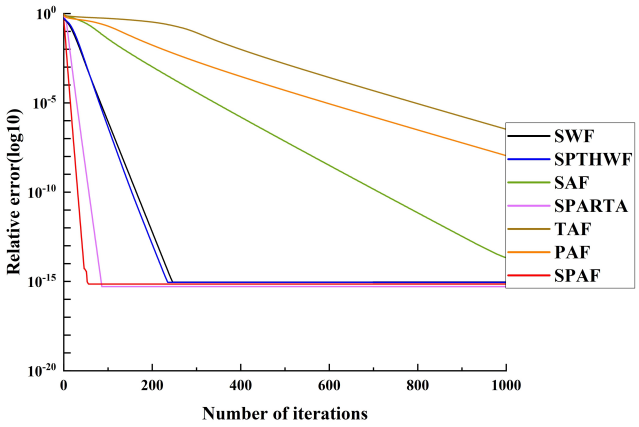

Fig. 2. Comparison of convergence rate for different algorithms in complex-valued case.

TABLE I
THE RUNNING TIMES OF DIFFERENT ALGORITHMS.

Algorithms	Real case	Complex case
PAF	3.87	-
SPARTA	0.39	1.63
TAF	4.21	-
SRThWF	2.16	16.17
SWF	2.62	16.82
SAF	2.10	-
SPAF	**0.28**	**0.95**

A. *convergence experiment*

To verify the convergence of the algorithm, the relative error of both real-valued and complex-valued signals are plotted against the iteration counts under noiseless conditions. The signal length in this experiment is set to $n = 1000$, with sparsity $k = 10$ and a sampling rate of $m/n = 3$, and a total of 1000 trials are performed. For real-valued signals (see Fig. 1), the SPAF algorithm achieves an accuracy of 10^{-15} after 17 iterations, while the SPARTA algorithm requires 21 iterations. To achieve the same accuracy, other algorithms need to undergo more than 80 iterations.

In the context of complex-valued signals, we compare SPAF algorithm with other methods (see Fig. 2). With a fixed iteration count of $T = 1000$, the SAF, PAF and TAF algorithms fail to achieve 10^{-15} accuracy, showing their limitations in dealing with complex-valued signals. As shown in Fig. 2, the proposed algorithm reaches convergence with the fewest iterations in the complex-valued case.

Table I presents the time required by each algorithm to reach a relative error of 10^{-15}, with the optimal values highlighted in bold. The results demonstrate that the SPAF algorithm achieves convergence more rapidly, achieving the target accuracy with less computational cost.

B. *Empirical success rate of sparsity*

To investigate SPAF algorithm for recover signals with varying levels of sparsity under a fixed number of measurements. Specifically, the sparsity k is varied between from 10 to 100 in real-valued case. And from 10 to 60 in complex-valued case, as some algorithms fail at higher sparsity levels. The length of signal is fixed at $n = 1000$ and the measurement ratio is $m/n = 1.5$. The proposed

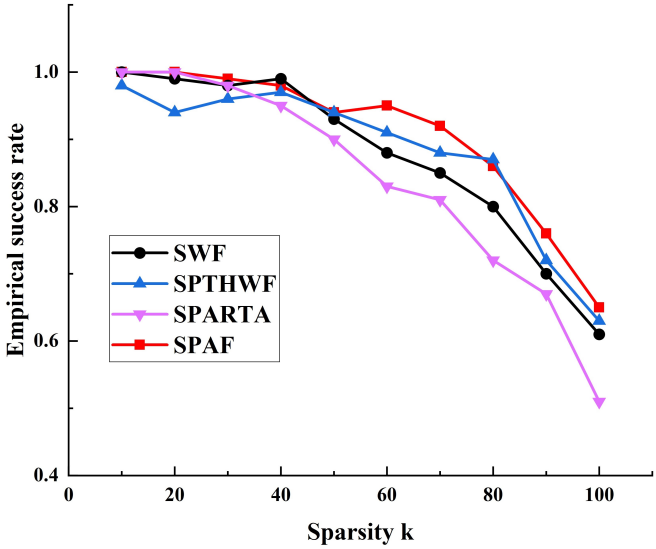

Fig. 3. Comparison of different sparsity selections k on signal recovery success rates in real-valued case.

algorithm is compared with several advanced sparse phase retrieval algorithms.

For real-valued case in Fig.3, all algorithms exhibit that the success rate of algorithms gradually decreases as sparsity k increases, reflecting that recovery becomes more difficult. However, even when k approaches 100, the SPAF algorithm maintains a success rate of approximately 65%, while the success rate of other algorithms falls below this threshold. This demonstrates that the SPAF algorithm effectively addresses signal recovery even in high sparsity scenarios.

For the complex-valued case, the experimental results are shown in Fig.4. Due to the complexity of recovering complex-valued signals, the success rate of the algorithms is lower compared to their performance on real-valued signals. At lower sparsity levels, all algorithms achieve relatively high success rates. However, when $k \geq 30$, the success rate of the SPTHWF algorithm decreases rapidly and eventually approaches zero. In contrast, the SPAF algorithm consistently outperforms others under various sparsity conditions. These results further demonstrate the significant advantages of the SPAF algorithm in complex-valued signal recovery case.

C. *Comparison of empirical success rate*

To validate the recovery success rate of the SPAF algorithm for various m/n values, the empirical success rate is compared with those algorithms from noiseless Gaussian models. Each success rate is determined from 100 independent Monte Carlo experiments, where the NMSE value is used as the criterion:

$$\text{NMSE} := \frac{\|\boldsymbol{z}_T - \boldsymbol{x}\|}{\|\boldsymbol{x}\|}.$$

When the NMSE is below 10^{-5}, a trial is considered successful. The signal dimension is specified as $n = 1000$, sparsity level is $k = 10$, the value of m/n is increased from 0 to 3 in real case and from 0 to 4 in complex case. As shown in Fig. 5, while the SPAF algorithm does not achieve the highest success rate for real-valued signals, it can also recover the signal more accurately under fewer measurement values.

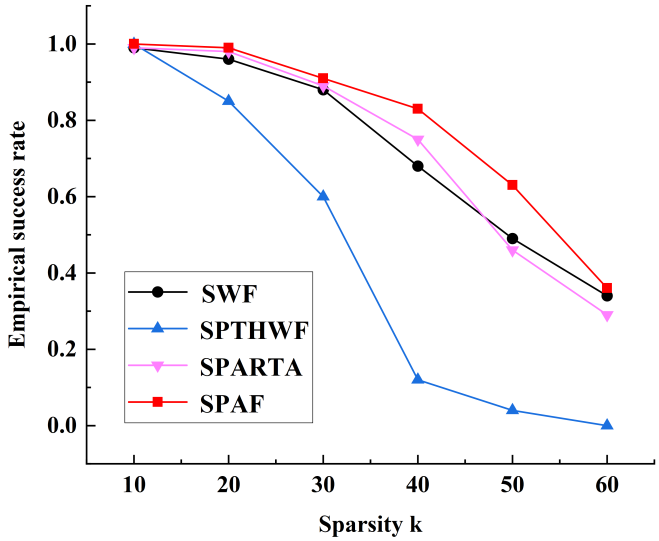

Fig. 4. Comparison of different sparsity selections k on signal recovery success rates in complex-valued case.

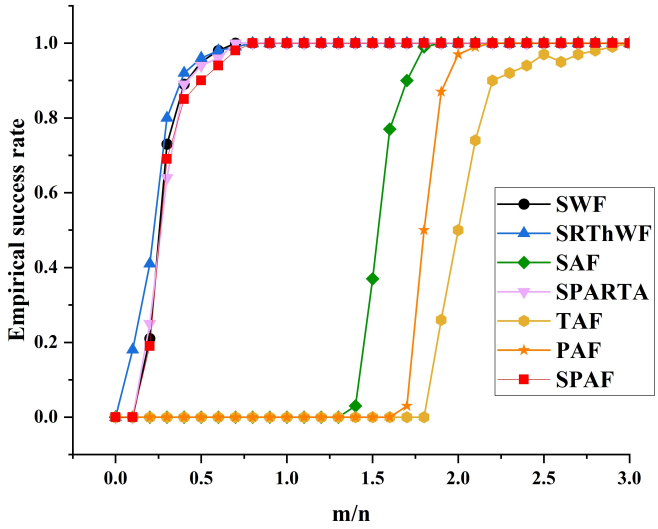

Fig. 5. The recovery success rates for various algorithms in real-valued case.

In contrast, as depicted in Fig. 6, the proposed algorithm demonstrates superior performance compared to the other algorithms in the complex case. When $m/n = 0.6$, the SPAF algorithm achieves the empirical success rate of over 95%, while the sub-optimal SPARTA algorithm reaches only 87%. This also shows the effectiveness of the proposed algorithm in complex-valued case.

Compared with SPARAF or SRThWF algorithms, these algorithms adopt truncation/reweighting to avoid too large or too small gradient component in iterative process. However, these analyses heavily depend on on each element having a sign equal to 1 or -1, making it difficult to scale to complex cases. So, the SPAF algorithm directly addresses the non-smooth nature of the loss function and effectively avoids corrupted search directions. As a result, the proposed algorithm shows excellent performance in complex-valued case.

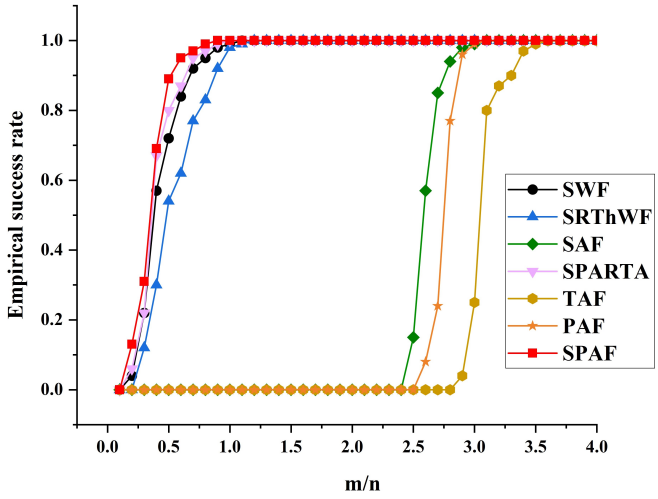

Fig. 6. The recovery success rates for various algorithms in complex-valued case.

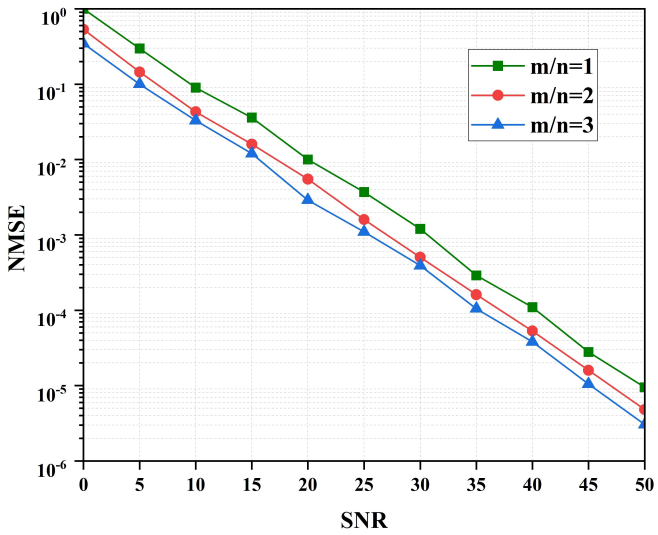

Fig. 7. NMSE vs SNR for SPAF in real-valued case.

D. Noise robustness evaluation

To evaluate the robustness of the SPAF algorithm under additive noise, the NMSE is plotted as a function of the signal-to-noise ratio (SNR) for different values of m/n. The Gaussian noise is defined as:

$$y_i = \left|\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle\right| + \eta_i,$$

where η_i is given by the following formula, that is:

$$\mathrm{SNR} = 10\log_{10}\sum_{i=1}^{m}\frac{\left|\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle\right|^2}{m\sigma^2}.$$

The SNR ranges from 0 dB to 50 dB. From Fig. 7, it can be observed that when $m = n$, $m = 2n$ and $m = 3n$ in the real-valued Gaussian model, the curves exhibit a monotonically decreasing trend. This indicates that as SNR increases, the NMSE value decreases, leading to reduced signal recovery error. Therefore, the SPAF algorithm's resilience to additive noise is demonstrated. A similar trend is observed in Fig. 8 for complex-valued Gaussian model, where the NMSE consistently decreases with increasing SNR

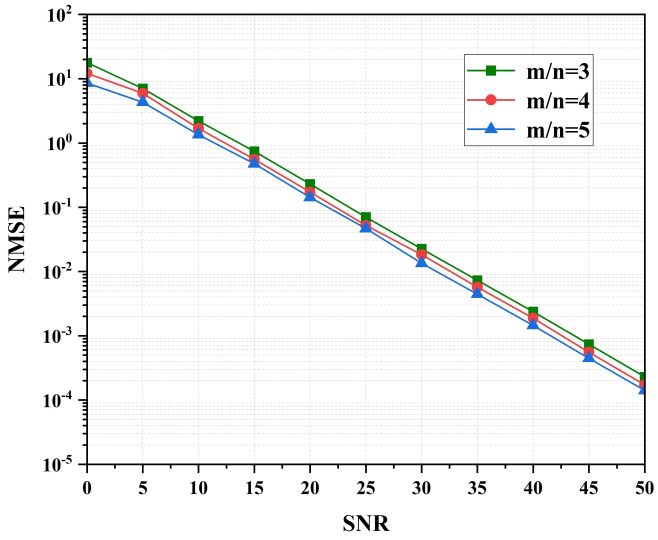

Fig. 8. NMSE vs SNR for SPAF in complex-valued case.

for $m = 3n$, $m = 4n$, and $m = 5n$. All of these results confirm the noise robustness of the SPAF algorithm.

VI. Conclusion

This paper proposes the Sparse Perturbed Amplitude Flow (SPAF) algorithm for phase retrieval of sparse signals. This algorithm incorporates smoothing parameters into the loss function to avoid excessively large or small components in gradient iterations, thus reducing complexity caused by inconsistent search directions. It first constructs a sparse support domain, then initializes value is obtained through sparse orthogonal initialization, and finally achieves the exact solution via hard thresholding iterations. Theorem 1 proves global convergence of SPAF algorithm. The SPAF algorithm requires no truncation or reweighting processes, making it is simple to implement and effective for both real and complex signals. Multiple experimentals prove that the SPAF algorithm surpasses the existing mainstream methods in both reconstruction accuracy and computational efficiency.

Future work can extend sparse phase retrieval to two-dimensional or multi-dimensional cases and incorporate deep learning, particularly convolutional neural networks, may enhance recovery performance of signal and image.

References

[1] Jianwei Miao, Charalambous Pambos, Kirz Janos and Sayre David, "Extending the Methodology of X-ray Crystallography to Allow Imaging of Micrometre Sized Non-crystalline Specimens," Nature, vol. 400, no.6742, pp342-344, 1999.

[2] Yiqing Xu, Zhibo Ren, Wong Kenneth KY and Tsia Kevin, "Overcoming the Limitation of Phase Retrieval using Gerchberg–Saxton-like Algorithm in Optical Fiber Time-stretch Systems," Optics Letters, vol. 40, no.15, pp3595-3598, 2015.

[3] Harm Walter, Roider Clemens, Jesacher Alexander, Bernet Stefan and Ritsch-Marte Monika, "Lensless Imaging through Thin Diffusive Media," Optics Express, vol. 22, no.18, pp22146-22156, 2014.

[4] Lan Li, Shasha Pu, Mingli Jing, Yulong Mao, Xiaoya Liu and Qiyv Sun, "Phase Retrieval based on the Distributed Conditional Generative Adversarial Network," Journal of the Optical Society of America A, vol. 22, no.18, pp1702-1712, 2024.

[5] Gerchberg RW and Saxton WO, "A Practical Algorithm for the Determination of Phase from Image and Diffraction Plane Pictures," Optik, vol. 35, no.2, pp237–250, 1972.

[6] Fienup James R, "Phase Retrieval Algorithms: A Comparison," Applied Optics, vol. 21, no.15, pp2758–2769, 1982.

[7] Candes Emmanuel J, Xiaodong Li and Soltanolkotabi Mahdi, "Phase Retrieval via Wirtinger Flow: Theory and Algorithms," IEEE Transactions on Information Theory, vol. 61, no.4, pp1985–2007, 2015.

[8] Yuxin Chen and Candes Emmanuel J, "Solving Random Quadratic Systems of Equations is Nearly as Easy as Solving Linear Systems," Communications on Pure and Applied Mathematics, vol. 70, no.5, pp822-883, 2015.

[9] Ziyang Yuan and Hongxia Wang, "Phase Retrieval via Reweighted Wirtinger Flow," Applied Optics, vol. 56, no.9, pp2418-2427, 2017.

[10] Huishuai Zhang, Yingbin Liang and Yuejie Chi, "A Nonconvex Approach for Phase Retrieval: Reshaped Wirtinger Flow and Incremental Algorithms," Journal of Machine Learning Research, vol. 18, no.1, pp5164– 5198, 2017.

[11] Yeh Li Hao, Dong Jonathan, Jingshan Zhong, Lei Tian, Chen Michael, Gongguo Tang, Soltanolkotabi Mahdi and Waller Laura,"Experimental Robustness of Fourier Ptychography Phase Retrieval Algorithms," Optics Express, vol. 23, no.26, pp33214–33240, 2015.

[12] Gang Wang, Giannakis Georgios B and Eldar Yonina C, "Solving Systems of Random Quadratic Equations via Truncated Amplitude Flow," IEEE Transactions on Information Theory, vol. 64, no.2, pp773-794, 2017.

[13] Gang Wang, Liang Zhang, Giannakis Georgios B and Jie Chen, "Sparse Phase Retrieval via Iteratively Reweighted Amplitude Flow," 2018 26th European Signal Processing Conference (EUSIPCO), pp712–716, 2018.

[14] JianFeng Cai, Yuling Jiao, Xiliang Lu and You Juntao, "Sample-Efficient Sparse Phase Retrieval via Stochastic Alternating Minimization," IEEE Transactions on Signal Processing, vol. 70, pp4951–4966, 2022.

[15] Jagatap Gauri and Hegde Chinmay, "Sample-Efficient Algorithms for Recovering Structured Signals from Magnitude-only Measurements," IEEE Transactions on Information Theory, vol. 65, no.7, pp4434-4456, 2019.

[16] Lan Li, Yufeng Ren and Kai Yu, "Sparse Phase Retrieval via Reweighted Thresholded Wirtinger Flow," IAENG International Journal of Applied Mathematics, vol. 54, no.4, pp683–689, 2024.

[17] JianFeng Cai, Jingzhi Li, Xiliang Lu and You Juntao, "Sparse Signal Recovery from Phaseless Measurements via Hard Thresholding Pursuit," Applied and Computational Harmonic Analysis, vol. 56, pp367–390, 2022.

[18] Gang Wang, Liang Zhang, Giannakis Georgios B, Akçakaya Mehmet and Jie Chen, "Sparse Phase Retrieval via Truncated Amplitude Flow," IEEE Transactions on Signal Processing, vol. 66, no.2, pp479-491, 2018.

[19] Liang Zhang, Gang Wang, Giannakis Georgios B and Jie Chen, "Compressive Phase Retrieval via Reweighted Amplitude Flow," IEEE Transactions on Signal Processing, vol. 66, no.19, pp5029-5040, 2018.

[20] Ziyang Yuan, Hongxia Wang and Qi Wang, "Phase Retrieval via Sparse Wirtinger Flow," IEEE Transactions on Signal Processing, vol. 355, pp162-173, 2019.

[21] Pinilla Samuel, Bacca Jorge and Arguello Henry, "Phase Retrieval Algorithm via Nonconvex Minimization using a Smoothing Function," IEEE Transactions on Signal Processing, vol. 66, no.17, pp4574-4584, 2018.

[22] Qi Luo, Hongxia Wang and Shijian Lin, "Phase Retrieval via Smoothed Amplitude Flow," Signal Processing, vol. 177, pp107719, 2020.

[23] Bing Gao, Xinwei Sun, Yang Wang and Zhiqiang Xu, "Perturbed Amplitude Flow for Phase Retrieval," IEEE Transactions on Signal Processing, vol. 68, pp5427–5440, 2020.

[24] Zhuolei Xiao, Ya Wang and Guan Gui, "Smoothed Amplitude Flow-based Phase Retrieval Algorithm," Journal of the Franklin Institute, vol. 358, no.14, pp7270–7285, 2021.

[25] Bacca Jorge, Pinilla Samuel and Arguello Henry, "Super-Resolution Phase Retrieval from Designed Coded Diffraction Patterns," IEEE Transactions on Image Processing, vol. 29, pp2598–2609, 2019.

[26] Needell Deanna and Tropp Joel A, "CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples," Applied and Computational Harmonic Analysis, vol. 26, pp301–321, 2009.