Numer. Math. Theor. Meth. Appl. Vol. **19**, No. 1, pp. 39-79
doi: 10.4208/nmtma.OA-2025-0014 February 2026

Theoretical Guarantees of Recovery Algorithms for Ternary Sparse Signals

Hanbing Liu and Chongjun Li*

School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China

Received 10 February 2025; Accepted (in revised version) 14 June 2025

Abstract. This paper focuses on the problem of recovering ternary sparse signals with s nonzero entries of 1 and -1. We propose three novel algorithms: ternary matching pursuit (TMP), ternary generalized orthogonal matching pursuit (TGOMP), and piecewise ternary generalized orthogonal matching pursuit (PTGOMP). First, inspired by the binary matching pursuit algorithm, we introduce the TMP algorithm, which assigns values of 1 or -1 based on the most correlated residual, and provide theoretical guarantees based on the mutual coherence, denoted by μ and the restricted isometry property of the measurement matrix, respectively. Second, we propose the TGOMP algorithm, by selecting multiple (M) indices at each iteration in order to improve the performance of the TMP algorithm. We establish a sufficient condition $\mu < 1/(2s-1)$ that ensures the TGOMP algorithm selects M correct indices and corresponding entries of $\boldsymbol{x}$ in each iteration. Especially, all correct indices and entries can be selected in the first iteration. Additionally, we present a sufficient condition based on the restricted isometry property that guarantees all correct indices are selected in at most s iterations. Third, we propose the PTGOMP algorithm, which employs a piecewise selection strategy at each iteration, further improves recovery performance. Theoretical guarantees for the PTGOMP algorithm are derived based on the mutual coherence, showing its advantages in ternary sparse signal recovery. Finally, we validate the effectiveness of our algorithms through simulations and numerical experiments, demonstrating that combining appropriate matrix structures with suitable sparsity patterns can significantly improve recovery performance.

AMS subject classifications: 94A12

Key words: Ternary sparse signal, mutual coherence, restricted isometry property, sparse recovery, piecewise sparsity.

1. Introduction

The problem of recovering sparse signals from an underdetermined noisy linear model has attracted significant attention [12, 13, 20, 44]. The model is typically for-

*Corresponding author. *Email address:* chongjun@dlut.edu.cn (C. Li)

mulated as

$$\boldsymbol{b} = A\boldsymbol{x} + \boldsymbol{v}, \tag{1.1}$$

where $\boldsymbol{b} \in \mathbb{R}^m$ denotes the observation vector, $A \in \mathbb{R}^{m\times n}$ denotes the measurement matrix with $m \ll n$, $\boldsymbol{x} \in \mathbb{R}^n$ is the s-sparse signal to be recovered (with s denoting the number of nonzero entries of $\boldsymbol{x}$), and $\boldsymbol{v} \in \mathbb{R}^m$ is additive noise with $\|\boldsymbol{v}\|_2 \le \epsilon$. Several algorithms have been proposed to achieve either exact or robust recovery of sparse signals, including basis pursuit (BP) [14], iterative hard thresholding (IHT) [7], compressive sampling matching pursuit (CoSaMP) [35], and orthogonal matching pursuit (OMP) [9, 25, 39], etc. Besides, a variant of the OMP algorithm, known as the generalized orthogonal matching pursuit (GOMP) [36,40] algorithm or the orthogonal multi-matching pursuit (OMMP) algorithm [17, 18, 45] has been widely studied.

The theoretical analysis of sparse recovery algorithms typically relies on two key concepts: the mutual coherence and the restricted isometry property (RIP). The mutual coherence of a matrix A is defined as [21]

$$\mu = \max_{i,j} \left|\langle \boldsymbol{a}_i, \boldsymbol{a}_j\rangle\right|,$$

where $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ are the unit-norm columns (or atoms) of the matrix A, for any $i \neq j$. Additionally, the matrix A satisfies the RIP [13] if there exists a constant $\delta \in (0,1)$ such that

$$(1-\delta)\|\boldsymbol{x}\|_2^2 \le \|A\boldsymbol{x}\|_2^2 \le (1+\delta)\|\boldsymbol{x}\|_2^2$$

for any s-sparse signal. The minimal value of δ is the restricted isometry constant (RIC), denoted δ_s.

Moreover, the inherent properties of sparse signals play an important role in the performance of recovery algorithms. Sparsity, the most widely studied property, is often referred to as global sparsity and is characterized by a single sparsity s. However, in many applications, such as magnetic resonance imaging (MRI) [24], computerized tomography (CT) [16], helium atom scattering [27], and fluorescence microscopy [37], signals exhibit additional structures that go beyond classical sparsity. Therefore, much research has focused on more complex forms of sparsity, including block sparsity [22, 23], joint sparsity [6, 8], and local sparsity in levels model [1, 2], also referred to as piecewise sparsity [30, 47].

Additionally, some sparse signals with certain magnitude and distribution of nonzero entries have been extensively studied. For instance, the binary sparse signals, where the nonzero entries are equal to 1, are applied in applications such as generalized space shift keying (GSSK) modulation detection [26] and massive connectivity detection in the Internet of things [15]. Algorithms such as the binary matching pursuit (BMP) [42] and the binary generalized orthogonal matching pursuit (BGOMP) [31] have been proposed to recover such signals. Similarly, the ternary sparse signals, where the nonzero entries are equal to 1 or -1, have been widely studied [28, 32–34, 38]. The ternary model is motivated by applications such as compressing neural networks [4, 19, 46], and the transmission of sparse signals [3, 5, 41]. The randomly aggregated unweighted least squares (RAWLS) [32] and the refined least squares (ReLS) [33] algorithms have been proposed for recovering the support of the ternary sparse signals. However, the RAWLS and ReLS algorithms perform well in recovering ternary sparse signals but are computationally expensive.

This paper focuses on the problem of recovering ternary sparse signals. Motivated by the BMP algorithm in [42], we propose the ternary matching pursuit algorithm. Unlike the BMP algorithm, which assigns the corresponding entry of the estimated sparse signal to 1 based on the highest correlation with residual, the TMP algorithm assigns the entry to 1 or -1 based on the sign of the most correlated column. Specifically, if the correlation is positive, the entry is set to 1, and if the correlation is negative, it is set to -1. We present theoretical conditions for the TMP algorithm based on the mutual coherence μ and RIP of the measurement matrix A, which ensure that the TMP algorithm recovers the support S and entries of sparse ternary signal $\boldsymbol{x}$ in s iterations. Based on the TMP algorithm, we propose the ternary generalized orthogonal matching pursuit algorithm, which improves performance by selecting multiple (M) indices in each iteration. Based on the mutual coherence of the matrix, we present a sufficient condition that ensures the TGOMP algorithm selects M indices and corresponding entries of $\boldsymbol{x}$ in each iteration or recovers the support and entries of $\boldsymbol{x}$ in s/M iterations, where s/M is an integer. In particular, when $M = s$, the TGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in the first iteration. Based on RIP, we present a sufficient condition that ensures the TGOMP algorithm recovers the support of the sparse ternary signal in at most s iterations.

We further propose the piecewise ternary generalized orthogonal matching pursuit algorithm, which employs a piecewise selection strategy in each iteration. We provide a sufficient condition that ensures the PTGOMP algorithm selects M_i correct indices and corresponding entries of $\boldsymbol{x}$ in each component in each iteration. This condition guarantees that the PTGOMP algorithm recovers the support and entries of $\boldsymbol{x}$ in $\max_{i=1,\ldots,N}\{s_i/M_i\}$ iterations, where s_i/M_i denotes an integer. In particular, when $M_i = s_i$, the PTGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in the first iteration. Finally, we validate the effectiveness of our algorithms through simulations on ternary sparse signals and experiments using sparse images from the MNIST dataset. The results indicate that combining appropriate matrix structures with suitable sparsity patterns can significantly improve recovery performance.

The rest of the paper is organized as follows. Section 2 provides some preliminaries. Section 3 introduces the TMP, TGOMP, and PTGOMP algorithms along with their theoretical results. Section 4 presents simulations and numerical experiments to demonstrate the efficiency of the proposed algorithms. Finally, Section 5 concludes the paper.

2. Notations and preliminaries

In this section, we introduce several notations, including the definitions of piecewise sparse signals and the restricted orthogonality constant (ROC). Additionally, we present some lemmas that will be used in later sections.

Let the set S denote the support of an s-sparse signal $\boldsymbol{x}$, i.e., $S = \text{supp}(\boldsymbol{x})$ and $s = |S|$. We define $\Omega = S \cup S^c$, where S^c denotes the complement of S, i.e., $S^c = \{j \in \Omega : x_j = 0\}$. The inner product is defined as $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{y}^{\mathrm{T}} \boldsymbol{x}$. For a subset $\Lambda \subseteq \Omega$, A_Λ is an $m \times |\Lambda|$ sub-matrix consisting of the columns indexed by Λ from the matrix A, and A^{T} denotes the transpose of the matrix A.

Definition 2.1 ([30, 47])**.** *A signal $\boldsymbol{x} = (\boldsymbol{x}_1^{\mathrm{T}}, \ldots, \boldsymbol{x}_N^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^n$ is partitioned into N components, where every component $\boldsymbol{x}_i \in \mathbb{R}^{n_i}$ is s_i-sparse, i.e., containing s_i nonzero entries, denoted as $\|\boldsymbol{x}_i\|_0 = s_i$ for $i = 1, \ldots, N$. Such a signal $\boldsymbol{x}$ is called a $(s_1, \ldots, s_N)$-piecewise sparse signal.*

Next, we consider the measurement matrix $A = (A_1^{\mathrm{T}}, \ldots, A_N^{\mathrm{T}})^{\mathrm{T}}$ that corresponds to the piecewise sparse signal $\boldsymbol{x} = (\boldsymbol{x}_1^{\mathrm{T}}, \ldots, \boldsymbol{x}_N^{\mathrm{T}})^{\mathrm{T}}$. In this case, we can express (1.1) as $\boldsymbol{b} = \sum_{i=1}^{N} A_i \boldsymbol{x}_i + \boldsymbol{v}$. In this paper, we consider that the matrix A is normalized, i.e., each column (or atom) $\boldsymbol{a}_j$ satisfies $\|\boldsymbol{a}_j\|_2 = 1$ for $j = 1, \ldots, n$. In the following, we introduce the concepts of mutual coherence and cumulative mutual coherence of the sub-matrices A_i, as discussed in [30, 47].

Definition 2.2 ([30, 47])**.** *The mutual coherence of the i-th sub-matrix A_i is defined as*

$$\mu^{i,i} = \max_{j \neq j'} \left| \left\langle \boldsymbol{a}_j^i, \boldsymbol{a}_{j'}^i \right\rangle \right|, \quad \boldsymbol{a}_j^i, \boldsymbol{a}_{j'}^i \in A_i, \quad i = 1, 2, \ldots, N.$$

It is clear that when A is a union of orthogonal sub-matrices, we have $\mu^{i,i} = 0$.

Lemma 2.1 ([30, 47])**.** *The mutual coherence $\mu^{i,i}$ of the i-th sub-matrix satisfies*

$$0 \leq \mu^{i,i} = \alpha_i \mu \leq \mu,$$

where $\alpha_i \in [0, 1]$ is a parameter that represents the ratio of the mutual coherence within the i-th sub-matrix to the mutual coherence of the whole matrix.

Lemma 2.2 ([30, 47])**.** *The cumulative coherence within the sub-matrix A_i is bounded by the following inequality:*

$$\mu_1^{i,i}(t) = \max_{|\Gamma_i| = t} \max_{j \notin \Gamma_i} \sum_{j' \in \Gamma_i} |\langle \boldsymbol{a}_j^i, \boldsymbol{a}_{j'}^i \rangle| \leq t \alpha_i \mu.$$

Lemma 2.3 ([30, 47])**.** *The cumulative coherence between two sub-matrices A_i and A_j is bounded by the following inequality:*

$$\mu_1^{i,j}(t) = \max_{|\Gamma_i| = t} \max_{l' \in \{1, \ldots, n_j\}} \sum_{j' \in \Gamma_i} \left| \left\langle \boldsymbol{a}_{l'}^i, \boldsymbol{a}_{j'}^j \right\rangle \right| \leq t \mu,$$

where Γ_i is the index set of t columns in sub-matrix A_i and n_j is the number of the columns in A_j.

Definition 2.3 ([11, 17]). *The restricted orthogonality constant $\theta_{s,s'}$ is defined as the smallest quantity such that for all s-sparse signals $\boldsymbol{x}$ and s'-sparse signals $\boldsymbol{x}'$ with disjoint support sets, the following inequality holds:*

$$|\langle A\boldsymbol{x}, A\boldsymbol{x}'\rangle| \leq \theta_{s,s'}\|\boldsymbol{x}\|_2\|\boldsymbol{x}'\|_2.$$

Lemma 2.4 ([11, 17]). *Let s, s' be two positive integers and $s \leq s'$. For any $c \geq 1$ such that cs' is an integer, the following inequalities hold:*

$$\begin{aligned} &\textit{monotonicity: } \delta_s \leq \delta_{s'}, \\ &\theta_{s,s'} \leq \delta_{s+s'}, && (2.1) \\ &\textit{shifting inequality: } \theta_{s,cs'} \leq \sqrt{c}\theta_{s,s'}. && (2.2) \end{aligned}$$

Lemma 2.5 ([17]). *Suppose the matrix A has the restricted isometry constant δ_s. Let Λ and Γ be two disjoint index sets, with the union of their cardinalities not exceeding s. Then the following inequalities hold:*

$$\begin{aligned} &\left(1-\delta_{|\Lambda|}\right)\|\boldsymbol{x}\|_2 \leq \left\|A_\Lambda^{\mathrm{T}} A_\Lambda \boldsymbol{x}\right\|_2 \leq \left(1+\delta_{|\Lambda|}\right)\|\boldsymbol{x}\|_2, \\ &\left\|A_\Lambda^{\mathrm{T}} A_\Gamma \boldsymbol{x}\right\|_2 \leq \theta_{|\Lambda|,|\Gamma|}\|\boldsymbol{x}\|_2. \end{aligned} \tag{2.3}$$

Lemma 2.6 ([43]). *Let $\Lambda \subseteq \Omega = \{1, 2, \ldots, n\}$ be a set such that $|\Lambda| = (k-1)M$ and $|S \cap \Lambda| = l$, where $0 \leq k-1 \leq l \leq |S|-1$. Let $W \subseteq (S \cup \Lambda)^c$ and $|W| = M$. If A satisfies the RIP of order $s+1$ and $\boldsymbol{x}$ is an s-sparse ternary signal, then the following inequality holds:*

$$\left\|A_{S\setminus\Lambda}^{\mathrm{T}} A_{S\setminus\Lambda}\boldsymbol{x}_{S\setminus\Lambda}\right\|_\infty - \frac{\|A_W^{\mathrm{T}} A_{S\setminus\Lambda}\boldsymbol{x}_{S\setminus\Lambda}\|_1}{M} \geq 1 - \sqrt{\frac{|S|-l}{M} + 1}\delta_{s+1}. \tag{2.4}$$

Note that Lemma 2.6 is a special case of [43, Lemma 1] when $P_\Lambda^\perp A_{S\setminus\Lambda}\boldsymbol{x}_{S\setminus\Lambda}$ is replaced by $A_{S\setminus\Lambda}\boldsymbol{x}_{S\setminus\Lambda}$ and $\boldsymbol{x}$ is an s-sparse ternary signal.

The BMP and BGOMP algorithms are outlined in Algorithms 2.1 and 2.2, respectively.

Algorithm 2.1 Binary Matching Pursuit Algorithm [42]

Input: $\boldsymbol{b}, A, s$.

Output: $\hat{\boldsymbol{x}}$ and Λ^k.

Initialization: $\Lambda^0 = \emptyset, k = 1, \hat{\boldsymbol{x}} = \boldsymbol{0}, \boldsymbol{r}^0 = \boldsymbol{b}$.

Repeat steps 1-5 when the stopping criterion is not met.

1: $j^k = \arg\max_{j\in\Omega\setminus\Lambda^{k-1}} |\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{r}^{k-1}|$.
2: $\Lambda^k = \Lambda^{k-1} \cup \{j^k\}$.
3: $\hat{x}_{j^k} = 1$.
4: $\boldsymbol{r}^k = \boldsymbol{r}^{k-1} - \boldsymbol{a}_{j^k}\hat{x}_{j^k}$.
5: $k = k+1$.

Stopping criterion: $k > s$ or $\|\boldsymbol{r}^k\|_2 \leq \epsilon$.

Algorithm 2.2 Binary Generalized Orthogonal Matching Pursuit Algorithm [31]

Input: $\boldsymbol{b}, A, s, M$.

Output: $\hat{\boldsymbol{x}}$ and $\hat{\Lambda} = \arg\max_{\Lambda \subseteq \Lambda^k, |\Lambda|=s} \|\hat{\boldsymbol{x}}_\Lambda\|_2$, $\hat{\boldsymbol{x}}$ satisfying $\hat{\boldsymbol{x}}_{\hat{\Lambda}} = \mathbf{1}$, and $\hat{\boldsymbol{x}}_{\Omega \setminus \hat{\Lambda}} = \mathbf{0}$.

Initialization: $\Lambda^0 = \emptyset, k = 1, \hat{\boldsymbol{x}} = \mathbf{0}, \boldsymbol{r}^0 = \boldsymbol{b}$.

Repeat steps 1-5 when the stopping criterion is not met.

1: Let T^k be the set corresponding to the M largest magnitudes of $|A^{\mathrm{T}}_{\Omega \setminus \Lambda^{k-1}} \boldsymbol{r}^{k-1}|$.

2: $\Lambda^k = \Lambda^{k-1} \cup T^k$.

3: $\hat{\boldsymbol{x}}_{T^k} = \mathbf{1}$.

4: $\boldsymbol{r}^k = \boldsymbol{r}^{k-1} - A_{T^k} \hat{\boldsymbol{x}}_{T^k}$.

5: $k = k + 1$.

$\hat{\boldsymbol{x}} = \arg\min_{\mathrm{supp}(\boldsymbol{z}) \subseteq \Lambda^k} \|\boldsymbol{b} - A\boldsymbol{z}\|_2$.

Stopping criterion: $k > s$ or $\|\boldsymbol{r}^k\|_2 \leq \epsilon$.

3. Theoretical analysis of recovery algorithms for ternary sparse signals

In this section, we present three novel algorithms for recovering ternary sparse signals: ternary matching pursuit, ternary generalized orthogonal matching pursuit, and piecewise ternary generalized orthogonal matching pursuit. These algorithms employ single, multiple (M), and piecewise selections, as demonstrated in Algorithms 3.1-3.3, respectively. We derive the theoretical conditions for the TMP and TGOMP algorithms based on the mutual coherence and the RIP, and for the PTGOMP algorithm based on the mutual coherence.

3.1. Ternary matching pursuit algorithm

In this subsection, we introduce the TMP algorithm in Algorithm 3.1, and show that the condition $\mu < 1/(2s-1)$ ensures that the TMP algorithm selects one correct index and corresponding entry in each iteration. Furthermore, we provide another sufficient condition based on the RIP, which also guarantees the TMP algorithm selects one correct index and corresponding entry in each iteration.

Algorithm 3.1 Ternary Matching Pursuit Algorithm

Input: $\boldsymbol{b}, A, s$.

Output: $\hat{\boldsymbol{x}}$ and Λ^k.

Initialization: $\Lambda^0 = \emptyset, k = 1, \hat{\boldsymbol{x}} = \mathbf{0}, \boldsymbol{r}^0 = \boldsymbol{b}$.

Repeat steps 1-5 when the stopping criterion is not met.

1: $j^k = \arg\max_{j \in \Omega \setminus \Lambda^{k-1}} |\boldsymbol{a}_j^{\mathrm{T}} \boldsymbol{r}^{k-1}|$.

2: $\Lambda^k = \Lambda^{k-1} \cup \{j^k\}$.

3: $\hat{x}_{j^k} = \begin{cases} 1, & \text{if} \quad \boldsymbol{a}_{j^k}^{\mathrm{T}} \boldsymbol{r}^{k-1} > 0, \\ -1, & \text{if} \quad \boldsymbol{a}_{j^k}^{\mathrm{T}} \boldsymbol{r}^{k-1} < 0. \end{cases}$

4: $\boldsymbol{r}^k = \boldsymbol{b} - A_{\Lambda^k} \hat{\boldsymbol{x}}_{\Lambda^k}$.

5: $k = k + 1$.
Stopping criterion: $k > s$ or $\|\boldsymbol{r}^k\|_2 \leq \epsilon$.

Unlike the BMP algorithm, the TMP algorithm assigns values based on the sign of the highest correlation. Specifically, if $\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{r}^{k-1} > 0$, then $\hat{x}_{j^k} = 1$. Conversely, if $\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{r}^{k-1} < 0$, then $\hat{x}_{j^k} = -1$. We provide sufficient conditions to guarantee this process.

3.1.1. Sufficient condition for the TMP algorithm based on the mutual coherence

Next, we present a theoretical condition based on the mutual coherence μ of the measurement matrix A, which ensures that the TMP algorithm recovers the support S and entries of sparse ternary signal $\boldsymbol{x}$ in s iterations.

Theorem 3.1. *Assume that $\boldsymbol{x}$ is an s-sparse ternary signal and noise vector $\boldsymbol{v}$ satisfies $\|\boldsymbol{v}\|_2 \leq \epsilon$. If the mutual coherence μ of the matrix A satisfies*

$$\mu < \frac{1}{2s-1}, \tag{3.1}$$

and ϵ satisfies

$$\epsilon < \frac{1-(2s-1)\mu}{2},$$

then with the stopping criterion $\|\boldsymbol{r}^k\|_2 \leq \epsilon$, the TMP algorithm exactly recovers the support S and entries of $\boldsymbol{x}$ in s iterations.

Proof. Let S be the support of an s-sparse ternary signal $\boldsymbol{x} \in \mathbb{R}^n$ and $S = S^+ \cup S^-$, where

- $S^+ = \{i : x_i = 1, i \in S\}, \hat{s}_1 = |S^+|$,
- $S^- = \{i : x_i = -1, i \in S\}, \hat{s}_2 = |S^-|$,

then the sparsity $s = \hat{s}_1 + \hat{s}_2$. In Algorithm 3.1, let $\Lambda^{k-1} = \Lambda_+^{k-1} \cup \Lambda_-^{k-1}$, where Λ_+^{k-1} and Λ_-^{k-1} denote the sets of indices selected for positive and negative entries of $\hat{\boldsymbol{x}}$ in the first $k-1$ iterations, respectively. The residual vector $\boldsymbol{r}^{k-1}$ is given by

$$\boldsymbol{r}^{k-1} = \boldsymbol{b} - A_{\Lambda^{k-1}}\hat{\boldsymbol{x}}_{\Lambda^{k-1}}.$$

Let

$$b^k = \left\|\boldsymbol{A}_{S\setminus\Lambda^{k-1}}^{\mathrm{T}}\boldsymbol{r}^{k-1}\right\|_\infty, \quad \rho^k = \left\|A_{S^c}^{\mathrm{T}}\boldsymbol{r}^{k-1}\right\|_\infty.$$

We prove Theorem 3.1 in two parts. We first provide a sufficient condition for the step 3 of Algorithm 3.1 in each iteration, and then establish a sufficient condition that guarantees the TMP algorithm selects one correct index and corresponding entry of $\boldsymbol{x}$ in each iteration.

To ensure the validity of the step 3 of Algorithm 3.1 in each iteration, we need to guarantee that the following inequalities are satisfied in the k-th iteration for $k = 1,\ldots,s$, i.e.,

$$\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{r}^{k-1} > 0, \quad \text{if} \quad x_i = 1, \quad i \in S^+ \setminus \Lambda_+^{k-1}, \tag{3.2}$$

$$\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{r}^{k-1} < 0, \quad \text{if} \quad x_i = -1, \quad i \in S^- \setminus \Lambda_-^{k-1}. \tag{3.3}$$

Besides, to guarantee that the TMP algorithm selects a correct index and corresponding entry of $\boldsymbol{x}$, we need the following inequality to hold in the k-th iteration for $k = 1,\ldots,s$, i.e.,

$$b^k > \rho^k. \tag{3.4}$$

Assuming that the TMP algorithm selects $k-1$ correct indices and corresponding entries of $\boldsymbol{x}$ in the first $k-1$ iterations for $k = 1,\ldots,s$, then we have $\Lambda_+^{k-1} \subset S^+$, $\Lambda_-^{k-1} \subset S^-$ and $\hat{\boldsymbol{x}}_{\Lambda^{k-1}} = \boldsymbol{x}_{\Lambda^{k-1}}$. Let $|\Lambda_+^{k-1}| = k_1$ and $|\Lambda_-^{k-1}| = k_2$, then it follows that $k_1 + k_2 = k-1$. The residual vector $\boldsymbol{r}^{k-1}$ is given by

$$\boldsymbol{r}^{k-1} = \boldsymbol{b} - A_{\Lambda^{k-1}}\hat{\boldsymbol{x}}_{\Lambda^{k-1}} = A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}} + \boldsymbol{v}.$$

In the k-th iteration, for (3.2), when $x_{j^k} = 1, j^k \in S^+ \setminus \Lambda_+^{k-1}$, we have

$$\begin{aligned}
\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{r}^{k-1} &= \boldsymbol{a}_{j^k}^{\mathrm{T}}\left(A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}} + \boldsymbol{v}\right) \\
&= \boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{a}_{j^k}x_{j^k} + \sum_{j\in S^+\setminus(\Lambda_+^{k-1}\cup\{j^k\})}\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{a}_j - \sum_{j\in S^-\setminus\Lambda_-^{k-1}}\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{a}_j + \boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{v} \\
&\geq x_{j^k} - \sum_{j\in S^+\setminus(\Lambda_+^{k-1}\cup\{j^k\})}\left|\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{a}_j\right| - \sum_{j\in S^-\setminus\Lambda_-^{k-1}}\left|\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{a}_j\right| - \left|\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{v}\right| \\
&\geq 1 + \mu - \left(\left|S^+ \setminus \Lambda_+^{k-1}\right| + \left|S^- \setminus \Lambda_-^{k-1}\right|\right)\mu - \epsilon \\
&= 1 + \mu - (\hat{s}_1 - k_1 + \hat{s}_2 - k_2)\mu - \epsilon \\
&= 1 + \mu - (s-k+1)\,\mu - \epsilon.
\end{aligned} \tag{3.5}$$

Similarly, when $x_{j^k} = -1, j^k \in S^- \setminus \Lambda_-^{k-1}$, we have

$$\begin{aligned}
\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{r}^{k-1} &= \boldsymbol{a}_{j^k}^{\mathrm{T}}\left(A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}} + \boldsymbol{v}\right) \\
&= \boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{a}_{j^k}x_{j^k} - \sum_{j\in S^-\setminus(\Lambda_-^{k-1}\cup\{j^k\})}\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{a}_j + \sum_{j\in S^+\setminus\Lambda_+^{k-1}}\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{a}_j + \boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{v} \\
&\leq x_{j^k} + \sum_{j\in S^-\setminus(\Lambda_-^{k-1}\cup\{j^k\})}\left|\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{a}_j\right| + \sum_{j\in S^+\setminus\Lambda_+^{k-1}}\left|\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{a}_j\right| + \left|\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{v}\right| \\
&\leq -1 + \left|S^- \setminus \left(\Lambda_-^{k-1} \cup \{j^k\}\right)\right|\mu + \left|S^+ \setminus \Lambda_+^{k-1}\right|\mu + \epsilon \\
&= -1 + \left(\hat{s}_2 - (k_2+1) + \hat{s}_1 - k_1\right)\mu + \epsilon \\
&= -1 - \mu + (s-k+1)\mu + \epsilon.
\end{aligned} \tag{3.6}$$

Since $s-(k-1)\leq s$, by (3.5) and (3.6), when

$$\mu < \frac{1}{s-1}, \tag{3.7}$$

and ϵ satisfies

$$\epsilon < 1-(s-1)\mu, \tag{3.8}$$

we have that (3.2) and (3.3) hold.

Besides, for the left-hand side of (3.4), let $j_0 \in S \setminus \Lambda^{k-1}$, we have

$$\begin{aligned}
b^k &= \left\|A^{\mathrm{T}}_{S\setminus\Lambda^{k-1}}\boldsymbol{r}^{k-1}\right\|_\infty \\
&= \max_{j\in S\setminus\Lambda^{k-1}} \left|\boldsymbol{a}_j^{\mathrm{T}}\left(A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}+\boldsymbol{v}\right)\right| \\
&\triangleq \left|\boldsymbol{a}_{j_0}^{\mathrm{T}}\left(A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}+\boldsymbol{v}\right)\right| \\
&\geq \left|\boldsymbol{a}_{j_0}^{\mathrm{T}}A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\right| - \left|\boldsymbol{a}_{j_0}^{\mathrm{T}}\boldsymbol{v}\right| \\
&\geq |x_{j_0}| - \sum_{j\in S\setminus(\Lambda^{k-1}\cup\{j_0\})} \left|\boldsymbol{a}_{j_0}^{\mathrm{T}}\boldsymbol{a}_j\right| - \|\boldsymbol{v}\|_2 \\
&\geq 1 - \left|S\setminus\left(\Lambda^{k-1}\cup\{j_0\}\right)\right|\mu - \epsilon \\
&= 1-(s-k)\mu-\epsilon,
\end{aligned} \tag{3.9}$$

and for the right-hand side of (3.4), let $j_1 \in S^c$, we have

$$\begin{aligned}
\rho^k &= \left\|A^{\mathrm{T}}_{S^c}\boldsymbol{r}^{k-1}\right\|_\infty \\
&= \max_{j'\in S^c} \left|\boldsymbol{a}_{j'}^{\mathrm{T}}\left(A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}+\boldsymbol{v}\right)\right| \\
&\triangleq \left|\boldsymbol{a}_{j_1}^{\mathrm{T}}\left(A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}+\boldsymbol{v}\right)\right| \\
&\leq \left|\boldsymbol{a}_{j_1}^{\mathrm{T}}A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\right| + \left|\boldsymbol{a}_{j_1}^{\mathrm{T}}\boldsymbol{v}\right| \\
&\leq \sum_{j\in S\setminus\Lambda^{k-1}} \left|\boldsymbol{a}_{j_1}^{\mathrm{T}}\boldsymbol{a}_j\right| + \|\boldsymbol{v}\|_2 \\
&\leq \left|S\setminus\Lambda^{k-1}\right|\mu + \epsilon \\
&\leq \left(s-(k-1)\right)\mu+\epsilon.
\end{aligned} \tag{3.10}$$

By (3.9), (3.10), and $s-k\leq s-1$, in order to select a correct index and corresponding entry, we need the following inequalities:

$$\mu < \frac{1}{2s-1}, \tag{3.11}$$

$$\epsilon < \frac{1+\mu-2s\mu}{2}. \tag{3.12}$$

Comparing (3.7) and (3.8) with (3.11) and (3.12), we obtain that (3.11) and (3.12) are sufficient conditions for (3.7) and (3.8), respectively. That is, they guarantee that (3.2)-(3.4) hold.

Finally, we show that the TMP algorithm stops exactly under the stopping criterion $\|\boldsymbol{r}^k\|_2 \leq \epsilon$ when all correct indices are selected. When $1 \leq k \leq s$, we have

$$S \setminus \Lambda^{k-1} \neq \emptyset, \quad |S \setminus \Lambda^{k-1}| = s - |S \cap \Lambda^{k-1}| \geq 1.$$

Since $\|\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\|_2 \geq 1$, by (3.11) and (3.12), we have

$$\begin{aligned}
\|\boldsymbol{r}^{k-1}\|_2 &= \left\|A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}} + \boldsymbol{v}\right\|_2 \\
&\geq \left\|A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\right\|_2 - \|\boldsymbol{v}\|_2 \\
&\geq \sqrt{1-(|S\setminus\Lambda^{k-1}|-1)\mu}\,\|\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\|_2 - \epsilon \\
&\geq \sqrt{1-(s-k)\mu} - \epsilon \\
&\geq 1-(s-k)\mu - \epsilon \\
&> 1+\mu-2s\mu-\epsilon \\
&> 2\epsilon - \epsilon = \epsilon.
\end{aligned}$$

After the s-th iteration, we consider $S \setminus \Lambda^s = \emptyset$ and

$$\|\boldsymbol{r}^s\|_2 = \|A_{S\setminus\Lambda^s}\boldsymbol{x}_{S\setminus\Lambda^s} + \boldsymbol{v}\|_2 = \|\boldsymbol{v}\|_2 \leq \epsilon.$$

Therefore, the TMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in s iterations. This completes the proof of Theorem 3.1. □

Remark 3.1. In [42], the authors showed that the BMP algorithm can exactly recover the support S of the signal $\boldsymbol{x}$ in s iterations when the matrix A satisfies

$$\mu < \frac{1}{2s-1},$$

and if ϵ satisfies

$$\epsilon < \frac{1-(2s-1)\mu}{2}.$$

The TMP algorithm is a generalization of the BMP algorithm. If the s-sparse signal $\boldsymbol{x}$ contains only nonzero entries of 1, the TMP algorithm reduces to the BMP algorithm, and Theorem 3.1 corresponds to [42, Theorem 1].

In the noiseless case, we have the following corollary.

Corollary 3.1. *Assume that $\boldsymbol{x}$ is an s-sparse ternary signal and noise vector $\boldsymbol{v}$ satisfies $\|\boldsymbol{v}\|_2 = 0$. If the mutual coherence μ of the matrix A satisfies*

$$\mu < \frac{1}{2s-1},$$

then the TMP algorithm exactly recovers the support S and entries of $\boldsymbol{x}$ in s iterations.

Next, we show that the condition (3.1) is sharp for recovering s-sparse ternary signals.

Theorem 3.2. *Let s be a positive integer. Let $t > 0$ be any integer, and set $m = (2s-1)t$ and $n = 2st$. Then there exists an $m \times n$ matrix A with mutual coherence $\mu = 1/(2s-1)$, and two nonzero s-sparse ternary vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ with disjoint supports such that*

$$A\boldsymbol{x}_1 = A\boldsymbol{x}_2.$$

Proof. Following the approach of the proof of [10, Theorem 3.1], let G be a $2s \times 2s$ matrix such that each diagonal entry of G is 1 and each off-diagonal entry equals $-\mu$. We define a $2s \times 2s$ diagonal matrix T, where the entries at the 2nd and $(s+2)$-th positions are -1, and all other diagonal entries are 1. We obtain that $\tilde{G} = TGT$ is a positive-semidefinite matrix with rank $2s-1$. That is

$$\tilde{G} = TGT = \underbrace{\begin{pmatrix} 1 & \mu & -\mu & \cdots & -\mu \\ \mu & 1 & \mu & \cdots & \mu \\ -\mu & \mu & 1 & \cdots & -\mu \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mu & \mu & -\mu & \cdots & 1 \\ -\mu & \mu & -\mu & \cdots & -\mu \\ \mu & -\mu & \mu & \cdots & \mu \\ -\mu & \mu & -\mu & \cdots & -\mu \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mu & \mu & -\mu & \cdots & -\mu \end{pmatrix}}_{s}\underbrace{\begin{pmatrix} -\mu & \mu & -\mu & \cdots & -\mu \\ \mu & -\mu & \mu & \cdots & \mu \\ -\mu & \mu & -\mu & \cdots & -\mu \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mu & \mu & -\mu & \cdots & -\mu \\ 1 & \mu & -\mu & \cdots & -\mu \\ \mu & 1 & \mu & \cdots & \mu \\ -\mu & \mu & 1 & \cdots & -\mu \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mu & \mu & -\mu & \cdots & 1 \end{pmatrix}}_{s}.$$

Since $\tilde{G}$ has two distinct eigenvalues $2s/(2s-1)$ and 0, with the multiplicities of $2s-1$ and 1, respectively, there is an orthogonal matrix $\tilde{U}$ such that

$$\tilde{G} = \tilde{U}\mathrm{Diag}\left\{\frac{2s}{2s-1}, \frac{2s}{2s-1}, \ldots, \frac{2s}{2s-1}, 0\right\}\tilde{U}^{\mathrm{T}}.$$

Then the symmetric matrix $\tilde{G}$ can be decomposed as $\tilde{G} = \tilde{A}^{\mathrm{T}}\tilde{A}$, where $\tilde{A}$ is the following $(2s-1) \times 2s$ matrix of the rank $2s-1$:

$$\tilde{A} = \begin{pmatrix} \sqrt{\frac{2s}{2s-1}} & 0 & \cdots & 0 & 0 \\ 0 & \sqrt{\frac{2s}{2s-1}} & \cdots & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & \cdots & \sqrt{\frac{2s}{2s-1}} & 0 \end{pmatrix}\tilde{U}^{\mathrm{T}},$$

where $\tilde{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{s+1}, \boldsymbol{u}_{s+2}, \ldots, \boldsymbol{u}_{2s})$, $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{2s-1}$ are unit eigenvectors corresponding to eigenvalue $2s/(2s-1)$ and

$$\boldsymbol{u}_{2s} = \frac{1}{\sqrt{2s}}(\underbrace{1, -1, 1, \ldots, 1}_{s}, \underbrace{1, -1, 1, \ldots, 1}_{s})^{\mathrm{T}} \in \mathbb{R}^{2s}$$

is the unit eigenvector corresponding to eigenvalue 0. That is

$$\tilde{A}\boldsymbol{u}_{2s} = \mathbf{0}. \tag{3.13}$$

We define an $m \times n$ matrix A by

$$A = \mathrm{Diag}\{\underbrace{\tilde{A}, \tilde{A}, \ldots, \tilde{A}}_{t}\},$$

then the mutual coherence of A is $\mu = 1/(2s-1)$. Suppose $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^{2st}$ are given by

$$\boldsymbol{x}_1 = (\underbrace{1, -1, 1, \ldots, 1}_{s}, \underbrace{0, \ldots, 0}_{(2t-1)s})^{\mathrm{T}},$$
$$\boldsymbol{x}_2 = (\underbrace{0, \ldots, 0}_{s}, \underbrace{1, -1, 1, \ldots, 1}_{s}, \underbrace{0, \ldots, 0}_{2(t-1)s})^{\mathrm{T}},$$

then $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are s-sparse ternary vectors.

Let $\boldsymbol{z} = \boldsymbol{x}_1 - \boldsymbol{x}_2 \in \mathbb{R}^{2st}$. By (3.13), we have $A\boldsymbol{x}_1 - A\boldsymbol{x}_2 = A\boldsymbol{z} = \mathbf{0}$. That is, $A\boldsymbol{x}_1 = A\boldsymbol{x}_2$. This implies that the model $\boldsymbol{b} = A\boldsymbol{x}$ is not identifiable within the class of s-sparse ternary signals. □

3.1.2. Sufficient condition for the TMP algorithm based on RIP

We provide a sufficient condition based on the RIP that ensures the TMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in s iterations.

Theorem 3.3. *Assume that $\boldsymbol{x}$ is an s-sparse ternary signal and noise vector $\boldsymbol{v}$ satisfies $\|\boldsymbol{v}\|_2 \le \epsilon$. If the RIP δ_{s+1} of the matrix A satisfies*

$$\delta_{s+1} < \frac{1}{\sqrt{s+1}},$$

and ϵ satisfies

$$\epsilon < \frac{1 - \sqrt{s+1}\,\delta_{s+1}}{2},$$

then with stopping criterion $\|\boldsymbol{r}^k\|_2 \le \epsilon$, the TMP algorithm exactly recovers the support S and entries of $\boldsymbol{x}$ in s iterations.

Proof. Similar to the proof of Theorem 3.1, we present a sufficient condition for (3.2)-(3.4) based on RIP.

Assume that the TMP algorithm selects $k-1$ correct indices and corresponding entries of $\boldsymbol{x}$ in the first $k-1$ iterations for $k=1,\ldots,s$. In the k-th iteration, when $x_{j^k}=1, j^k\in S^+\setminus\Lambda_+^{k-1}$, we have

$$\begin{aligned}
\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{r}^{k-1} &= \boldsymbol{a}_{j^k}^{\mathrm{T}}\left(A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}+\boldsymbol{v}\right)\\
&= \boldsymbol{a}_{j^k}^{\mathrm{T}}\left(\boldsymbol{a}_{j^k}x_{j^k}+A_{S\setminus(\Lambda^{k-1}\cup\{j^k\})}\boldsymbol{x}_{S\setminus(\Lambda^{k-1}\cup\{j^k\})}+\boldsymbol{v}\right)\\
&= x_{j^k}+\boldsymbol{a}_{j^k}^{\mathrm{T}}A_\Theta\boldsymbol{x}_\Theta+\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{v}\\
&\geq 1-\left|\boldsymbol{a}_{j^k}^{\mathrm{T}}A_\Theta\boldsymbol{x}_\Theta\right|-\left|\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{v}\right|\\
&\geq 1-\left\|\boldsymbol{a}_{j^k}^{\mathrm{T}}A_\Theta\boldsymbol{x}_\Theta\right\|_2-\epsilon\\
&\overset{(a)}{\geq} 1-\theta_{1,|\Theta|}\|\boldsymbol{x}_\Theta\|_2-\epsilon\\
&\geq 1-\theta_{1,s-1}\|\boldsymbol{x}_\Theta\|_2-\epsilon\\
&\geq 1-\sqrt{s-1}\theta_{1,s-1}-\epsilon\\
&\geq 1-\sqrt{s+1}\delta_{s+1}-\epsilon, \qquad (3.14)
\end{aligned}$$

where $\Theta=S\setminus(\Lambda^{k-1}\cup\{j^k\})$, $|\Theta|=s-k$, $\|\boldsymbol{x}_\Theta\|_2=\sqrt{|\Theta|}$, (a) is due to (2.3), and the last inequality is obtained by (2.1).

Similarly, when $x_{j^k}=-1, j^k\in S^-\setminus\Lambda_-^{k-1}$, we have

$$\begin{aligned}
\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{r}^{k-1} &= x_{j^k}+\boldsymbol{a}_{j^k}^{\mathrm{T}}A_\Theta\boldsymbol{x}_\Theta+\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{v}\\
&\leq -1+\left|\boldsymbol{a}_{j^k}^{\mathrm{T}}A_\Theta\boldsymbol{x}_\Theta\right|+\left|\boldsymbol{a}_{j^k}^{\mathrm{T}}\boldsymbol{v}\right|\\
&\leq -1+\left\|\boldsymbol{a}_{j^k}^{\mathrm{T}}A_\Theta\boldsymbol{x}_\Theta\right\|_2+\epsilon\\
&\leq -1+\theta_{1,|\Theta|}\|\boldsymbol{x}_\Theta\|_2+\epsilon\\
&\leq -1+\sqrt{s-1}\theta_{1,s-1}+\epsilon\\
&\leq -1+\sqrt{s+1}\delta_{s+1}+\epsilon. \qquad (3.15)
\end{aligned}$$

By (3.14) and (3.15), if

$$\delta_{s+1}<\frac{1}{\sqrt{s+1}},\quad \epsilon<1-\sqrt{s+1}\delta_{s+1}$$

are satisfied, we obtain that (3.2) and (3.3) hold.

Then, we present a sufficient condition for (3.4) based on RIP in the k-th iteration. In the left-hand side of (3.4), we have

$$\begin{aligned}
b^k &= \left\|A_{S\setminus\Lambda^{k-1}}^{\mathrm{T}}\boldsymbol{r}^{k-1}\right\|_\infty\\
&\geq \left\|A_{S\setminus\Lambda^{k-1}}^{\mathrm{T}}A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\right\|_\infty-\left\|A_{S\setminus\Lambda^{k-1}}^{\mathrm{T}}\boldsymbol{v}\right\|_\infty\\
&\geq \left\|A_{S\setminus\Lambda^{k-1}}^{\mathrm{T}}A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\right\|_\infty-\epsilon, \qquad (3.16)
\end{aligned}$$

and in the right-hand side of (3.4), we have

$$\begin{aligned}\rho^k &= \|A_{S^c}^{\mathrm{T}} \boldsymbol{r}^{k-1}\|_\infty \\ &\leq \|A_{S^c}^{\mathrm{T}} A_{S\setminus\Lambda^{k-1}} \boldsymbol{x}_{S\setminus\Lambda^{k-1}}\|_\infty + \|A_{S^c}^{\mathrm{T}} \boldsymbol{v}\|_\infty \\ &\leq \|A_{S^c}^{\mathrm{T}} A_{S\setminus\Lambda^{k-1}} \boldsymbol{x}_{S\setminus\Lambda^{k-1}}\|_\infty + \epsilon. \end{aligned} \tag{3.17}$$

Combining (3.16) and (3.17), we obtain

$$\|A_{S\setminus\Lambda^{k-1}}^{\mathrm{T}} A_{S\setminus\Lambda^{k-1}} \boldsymbol{x}_{S\setminus\Lambda^{k-1}}\|_\infty - \|A_{S^c}^{\mathrm{T}} A_{S\setminus\Lambda^{k-1}} \boldsymbol{x}_{S\setminus\Lambda^{k-1}}\|_\infty > 2\epsilon \tag{3.18}$$

is a sufficient condition for (3.4). From Lemma 2.6, we have

$$\|A_{S\setminus\Lambda^{k-1}}^{\mathrm{T}} A_{S\setminus\Lambda^{k-1}} \boldsymbol{x}_{S\setminus\Lambda^{k-1}}\|_\infty - \|A_{S^c}^{\mathrm{T}} A_{S\setminus\Lambda^{k-1}} \boldsymbol{x}_{S\setminus\Lambda^{k-1}}\|_\infty > 1 - \sqrt{s+1}\delta_{s+1},$$

then, we obtain a sufficient condition for (3.18), i.e.,

$$1 - \sqrt{s+1}\delta_{s+1} > 2\epsilon.$$

Hence, we obtain that

$$\delta_{s+1} < \frac{1}{\sqrt{s+1}}, \quad \epsilon < \frac{1 - \sqrt{s+1}\delta_{s+1}}{2}$$

are sufficient conditions for (3.4).

Similarly, when $1 \leq k \leq s$, we have

$$S \setminus \Lambda^{k-1} \neq \emptyset, \quad |S \setminus \Lambda^{k-1}| = s - |\Lambda^{k-1}| \geq 1.$$

Since $\|\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\|_2 \geq 1$, we have

$$\begin{aligned}\|\boldsymbol{r}^{k-1}\|_2 &= \|A_{S\setminus\Lambda^{k-1}} \boldsymbol{x}_{S\setminus\Lambda^{k-1}} + \boldsymbol{v}\|_2 \\ &\geq \|A_{S\setminus\Lambda^{k-1}} \boldsymbol{x}_{S\setminus\Lambda^{k-1}}\|_2 - \|\boldsymbol{v}\|_2 \\ &\geq \sqrt{1 - \delta_{s-k+1}}\|\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\|_2 - \epsilon \\ &\geq 1 - \sqrt{s+1}\delta_{s+1} - \epsilon \\ &> 2\epsilon - \epsilon = \epsilon. \end{aligned}$$

After the s-th iteration, we have $S \setminus \Lambda^s = \emptyset$ and

$$\|\boldsymbol{r}^s\|_2 = \|A_{S\setminus\Lambda^s} \boldsymbol{x}_{S\setminus\Lambda^s} + \boldsymbol{v}\|_2 = \|\boldsymbol{v}\|_2 \leq \epsilon.$$

Therefore, the TMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in s iterations. This completes the proof of Theorem 3.3. □

Accordingly, we also consider a result based on RIP in the noiseless case.

Corollary 3.2. *Assume that $\boldsymbol{x}$ is an s-sparse ternary signal and noise vector $\boldsymbol{v}$ satisfies $\|\boldsymbol{v}\|_2 = 0$. If the RIP δ_{s+1} of the matrix A satisfies*

$$\delta_{s+1} < \frac{1}{\sqrt{s+1}},$$

then the TMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in s iterations.

3.2. Ternary generalized orthogonal matching pursuit algorithm

In this subsection, similar to the BGOMP algorithm, based on the TMP algorithm, we obtain the following TGOMP algorithm by selecting multiple (M) indices in each iteration. Let s/M be an integer in the section. We establish the theoretical guarantees for the TGOMP algorithm based on the mutual coherence and the RIP. Specifically, based on the mutual coherence of the matrix, we present a sufficient condition that ensures the TGOMP algorithm selects M correct indices and corresponding entries of $\boldsymbol{x}$ in each iteration, or recovers the support and entries of $\boldsymbol{x}$ in s/M iterations. In particular, when $M = s$, the TGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in the first iteration. Besides, based on RIP, we present a sufficient condition that ensures the TGOMP algorithm recovers the support of the sparse ternary signal in at most s iterations.

Algorithm 3.2 Ternary Generalized Orthogonal Matching Pursuit Algorithm

Input: $\boldsymbol{b}, A, s, M$, where s/M is an integer.

Output: $\tilde{\Gamma} = \arg\max_{\Gamma\subseteq\Lambda^k, |\Gamma|=s} \|\hat{\boldsymbol{z}}_\Gamma\|_2$ and $\tilde{\boldsymbol{x}}$, where $\tilde{\boldsymbol{x}}_{\tilde{\Gamma}} = \text{sign}(\hat{\boldsymbol{z}}_{\tilde{\Gamma}})$ and $\tilde{\boldsymbol{x}}_{\Omega\setminus\tilde{\Gamma}} = \mathbf{0}$.

Initialization: $\Lambda^0 = \emptyset, k = 1, \hat{\boldsymbol{x}} = \mathbf{0}, \boldsymbol{r}^0 = \boldsymbol{b}$.

Repeat steps 1-5 when the stopping criterion is not met.

1: Let T^k be the set corresponding to the M largest magnitudes of $|A^{\mathrm{T}}_{\Omega\setminus\Lambda^{k-1}}\boldsymbol{r}^{k-1}|$.

2: $\Lambda^k = \Lambda^{k-1} \cup T^k$.

3: For all $j^k \in T^k$, $\hat{x}_{j^k} = \begin{cases} 1, & \text{if} \quad \boldsymbol{a}^{\mathrm{T}}_{j^k}\boldsymbol{r}^{k-1} > 0, \\ -1, & \text{if} \quad \boldsymbol{a}^{\mathrm{T}}_{j^k}\boldsymbol{r}^{k-1} < 0. \end{cases}$

4: $\boldsymbol{r}^k = \boldsymbol{b} - A_{\Lambda^k}\hat{\boldsymbol{x}}_{\Lambda^k}$.

5: $k = k + 1$.

$\hat{\boldsymbol{z}} = \arg\min_{\text{supp}(\boldsymbol{z})\subseteq\Lambda^k} \|\boldsymbol{b} - A\boldsymbol{z}\|_2$.

Stopping criterion: $k > s$ or $\|\boldsymbol{r}^k\|_2 \le \epsilon$.

The TGOMP algorithm is a generalization of the TMP algorithm, improving recovery performance by the selecting multiple (M) indices in each iteration. In Algorithm 3.2, M denotes the number of indices selected in each iteration. For all $j^k \in T^k$ in step 3, if $\boldsymbol{a}^{\mathrm{T}}_{j^k}\boldsymbol{r}^{k-1} > 0$, then $\hat{x}_{j^k} = 1$, if $\boldsymbol{a}^{\mathrm{T}}_{j^k}\boldsymbol{r}^{k-1} < 0$, then $\hat{x}_{j^k} = -1$. After the iterations stop, the current set Λ^k contains all indices selected in the first k iterations, hence, we need the following two processes to retain s nonzero entries:

(1) Perform Least squares on Λ^k, i.e., $\hat{\boldsymbol{z}} = \arg\min_{\text{supp}(\boldsymbol{z})\subseteq\Lambda^k} \|\boldsymbol{b} - A\boldsymbol{z}\|_2$.

(2) Find the s entries with the largest values of $\hat{\mathbf{z}}$, i.e., $\tilde{\Gamma} = \arg\max_{\Gamma\subseteq\Lambda^k, |\Gamma|=s} \|\hat{\boldsymbol{z}}_\Gamma\|_2$, and set $\tilde{\boldsymbol{x}}$, where $\tilde{\boldsymbol{x}}_{\tilde{\Gamma}} = \text{sign}(\hat{\boldsymbol{z}}_{\tilde{\Gamma}})$ and $\tilde{\boldsymbol{x}}_{\Omega\setminus\tilde{\Gamma}} = \mathbf{0}$.

The algorithm outputs $\tilde{\boldsymbol{x}}$ and $\tilde{\Gamma}$.

3.2.1. Sufficient condition for the TGOMP algorithm based on the mutual coherence

We provide a sufficient condition that ensures the TGOMP algorithm selects M correct indices and corresponding entries of $\boldsymbol{x}$ in each iteration, or recovers the support S and entries of $\boldsymbol{x}$ in one iteration.

Theorem 3.4. *Assume that $\boldsymbol{x}$ is an s-sparse ternary signal and noise vector $\boldsymbol{v}$ satisfies $\|\boldsymbol{v}\|_2 \leq \epsilon$, and M is the number of selected indices in each iteration by the TGOMP algorithm. If the mutual coherence μ of the matrix A satisfies*

$$\mu < \frac{1}{2s-1}, \tag{3.19}$$

and ϵ satisfies

$$\epsilon < \frac{1-(2s-1)\mu}{2}, \tag{3.20}$$

when $M \leq s$, the TGOMP algorithm selects M correct indices and corresponding entries of $\boldsymbol{x}$ in each iteration, that is, the TGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in s/M iterations. In particular, when $M = s$, the TGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in the first iteration.

The detailed proof is presented in Appendix A. The following corollary considers the noiseless case in Theorem 3.4.

Corollary 3.3. *Assume that $\boldsymbol{x}$ is an s-sparse ternary signal and noise vector $\boldsymbol{v}$ satisfies $\|\boldsymbol{v}\|_2 = 0$, and M is the number of selected indices in each iteration by the TGOMP algorithm. If the mutual coherence μ of the matrix A satisfies*

$$\mu < \frac{1}{2s-1},$$

when $M \leq s$, the TGOMP algorithm selects M correct indices and corresponding entries of $\boldsymbol{x}$ in each iteration, that is, the TGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in s/M iterations. In particular, when $M = s$, the TGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in the first iteration.

3.2.2. Sufficient condition for the TGOMP algorithm based on RIP

Next, we provide a sufficient condition based on the RIP to ensure that the TGOMP algorithm selects at least one correct index of $\boldsymbol{x}$ in each iteration.

Theorem 3.5. *Assume that $\boldsymbol{x}$ is an s-sparse ternary signal and noise vector $\boldsymbol{v}$ satisfies $\|\boldsymbol{v}\|_2 \leq \epsilon$, and M is the number of selected indices in each iteration by the TGOMP algorithm. If the RIP δ_{s+1} of the matrix A satisfies*

$$\delta_{s+1} < \frac{1}{\sqrt{s+1} + \sqrt{2(s-1)(M-1)}}, \tag{3.21}$$

and ϵ satisfies

$$\epsilon < \frac{1-\sqrt{s+1}\delta_{s+1}-\sqrt{2(s-1)}(M-1)\delta_{s+1}}{2}, \tag{3.22}$$

then the TGOMP algorithm exactly recovers the support S of $\boldsymbol{x}$ in at most s iterations.

The detailed proof is presented in Appendix B.

Remark 3.2. The TGOMP algorithm and conditions (3.21) and (3.22) are valid for the case when $\boldsymbol{x}$ is an s-sparse binary signal. The condition (3.21) is stricter than the sufficient condition

$$\delta_{s+1} < \frac{1}{\sqrt{s/M+1}+2M(s-1)}, \quad M \geq 2$$

for the BGOMP algorithm in [31], since (3.21) ensures that (B.1) and (B.2) hold, that is, the TGOMP algorithm can recover correct support for all entries 1 and -1 of an s-sparse ternary signal $\boldsymbol{x}$.

Corollary 3.4. *Assume that $\boldsymbol{x}$ is an s-sparse ternary signal and noise vector $\boldsymbol{v}$ satisfies $\|\boldsymbol{v}\|_2 = 0$, and M is the number of selected indices in each iteration by the TGOMP algorithm. If the RIP δ_{s+1} of the matrix A satisfies*

$$\delta_{s+1} < \frac{1}{\sqrt{s+1}+\sqrt{2(s-1)}(M-1)},$$

then the TGOMP algorithm exactly recovers the support S of $\boldsymbol{x}$ in at most s iterations.

3.3. Piecewise ternary generalized orthogonal matching pursuit algorithm

In this subsection, we introduce the piecewise ternary generalized orthogonal matching pursuit algorithm, as outlined in Algorithm 3.3 for solving the piecewise signal recovery problem $\boldsymbol{b} = A_1\boldsymbol{x}_1 + \cdots + A_N\boldsymbol{x}_N + \boldsymbol{v}$. For $i = 1, \ldots, N$, let $\boldsymbol{x}_i \in \mathbb{R}^{n_i}$ be s_i-sparse ternary signal and M_i be the number of indices selected by the PTGOMP algorithm in i-th component, where s_i/M_i are integer numbers in the section. The PTGOMP algorithm enhances the TGOMP algorithm in three ways:

(1) The algorithm provides a piecewise structure for both the measurement matrix and the signal. In each component A_i, it selects the M_i highest correlations between the residual and columns of the measurement matrix.

(2) In the final step, it selects the largest entries based on the sparsity of each component.

(3) The stopping criterion of the PTGOMP algorithm is defined as either $k > s_{\max} = \max_{i=1,\ldots,N}\{s_i\}$ or $\|\boldsymbol{r}^k\|_2 \leq \epsilon$, where ϵ is a prespecified threshold.

Algorithm 3.3 Piecewise Ternary Generalized Orthogonal Matching Pursuit Algorithm

Input: $\boldsymbol{b}, A = [A_1, \ldots, A_N], \boldsymbol{s} = [s_1, \ldots, s_N], \boldsymbol{M} = [M_1, \ldots, M_N]$, where s_i/M_i is an integer.

Output: $\tilde{\boldsymbol{x}}$ and $\tilde{\Lambda} = \cup_{i=1}^{N} \tilde{\Lambda}_i$, where $\tilde{\Lambda}_i = \arg\max_{\hat{\Lambda}_i \subseteq \Lambda_i^k, |\hat{\Lambda}_i| = s_i} \|\boldsymbol{z}_{\hat{\Lambda}_i}\|_2, i = 1, \ldots, N$, $\tilde{\boldsymbol{x}}$ satisfying $\tilde{\boldsymbol{x}}_{\tilde{\Lambda}} = \text{sign}(\boldsymbol{z}_{\tilde{\Lambda}})$ and $\tilde{\boldsymbol{x}}_{\Omega \backslash \tilde{\Lambda}} = \boldsymbol{0}$.

Initialization: $\Lambda_i^0 = \emptyset, i = 1, \ldots, N$, $k = 1$, $\hat{\boldsymbol{x}} = \boldsymbol{0}$, $\boldsymbol{r}^0 = \boldsymbol{b}$.

Repeat steps 1–5 when the stopping criterion is not met.

1: Let T_i^k be the set corresponding to the M_i largest magnitudes of $|A^{\mathrm{T}}_{\Omega_i \backslash \Lambda_i^{k-1}} \boldsymbol{r}^{k-1}|$.

2: $\Lambda^k = \cup_{i=1}^{N} \Lambda_i^k$, $\Lambda_i^k = \Lambda_i^{k-1} \cup T_i^k, i = 1, \ldots, N$.

3: For all $j^k \in T_i^k$, $\hat{x}_{j^k} = \begin{cases} 1, & \text{if} \quad \boldsymbol{a}_{j^k}^{\mathrm{T}} \boldsymbol{r}^{k-1} > 0, \\ -1, & \text{if} \quad \boldsymbol{a}_{j^k}^{\mathrm{T}} \boldsymbol{r}^{k-1} < 0. \end{cases}$

4: $\boldsymbol{r}^k = \boldsymbol{b} - A_{\Lambda^k} \hat{\boldsymbol{x}}_{\Lambda^k}$.

5: $k = k + 1$.

$\boldsymbol{z}_{\Lambda^k} = A_{\Lambda^k}^{\dagger} \boldsymbol{b}$.

Stopping criterion: $k > s_{\max} = \max_{i=1,\ldots,N}\{s_i\}$ or $\|\boldsymbol{r}^k\|_2 \leq \epsilon$.

Next, we provide a sufficient condition that ensures the PTGOMP algorithm selects M_i correct indices and corresponding entries of $\boldsymbol{x}$ in each component in each iteration. This condition guarantees that the PTGOMP algorithm recovers the support and entries of $\boldsymbol{x}$ in $\max_{i=1,\ldots,N}\{s_i/M_i\}$ iterations. In particular, when $M_i = s_i$, the PTGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in the first iteration.

Theorem 3.6. *Assume that $\boldsymbol{x}$ is an $(s_1, \ldots, s_N)$-piecewise sparse ternary signal with $s = s_1 + \cdots + s_N$, noise vector $\boldsymbol{v}$ satisfies $\|\boldsymbol{v}\|_2 \leq \epsilon$, M_i is the number of selected indices in each iteration in the i-th component by the PTGOMP algorithm for $i = 1, \ldots, N$ and $M = M_1 + \cdots + M_N$. Let $\alpha_i = \mu^{i,i}/\mu$, where $\mu^{i,i}$ denotes the coherence of the submatrix A_i. If the mutual coherence μ of the matrix A satisfies*

$$\mu < \frac{1}{2(s - (1 - \alpha_Z)s_Z) - \alpha_Z}, \tag{3.23}$$

where

$$2(1 - \alpha_Z)s_Z + \alpha_Z = \min_{i=1,\ldots,N}\{2(1 - \alpha_i)s_i + \alpha_i\},$$

and ϵ satisfies

$$\epsilon < \frac{1 - (2(s - (1 - \alpha_Z)s_Z) - \alpha_Z)\mu}{2}, \tag{3.24}$$

when $M_i \leq s_i$, the PTGOMP algorithm selects M correct indices and corresponding entries of $\boldsymbol{x}$ in each iteration, that is, the PTGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in $\max_{i=1,\ldots,N}\{s_i/M_i\}$ iterations. In particular, when $M_i = s_i$, the PTGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in the first iteration.

The detailed proof is presented in Appendix C.

Corollary 3.5. *Assume that $\boldsymbol{x}$ is an $(s_1, \ldots, s_N)$-piecewise sparse ternary signal, noise vector $\boldsymbol{v}$ satisfies $\|\boldsymbol{v}\|_2 = 0$, M_i is the number of selected indices in each iteration in the i-th component by the PTGOMP algorithm for $i = 1, \ldots, N$ and $M = M_1 + \cdots + M_N$. Let $\alpha_i = \mu^{i,i}/\mu$, where $\mu^{i,i}$ denotes the coherence of the sub-matrix A_i. If the mutual coherence μ of the matrix A satisfies*

$$\mu < \frac{1}{2(s - (1 - \alpha_Z)s_Z) - \alpha_Z},$$

where

$$2(1 - \alpha_Z)s_Z + \alpha_Z = \min_{i=1,\ldots,N}\{2(1 - \alpha_i)s_i + \alpha_i\},$$

when $M_i \leq s_i$, the PTGOMP algorithm selects M correct indices and corresponding entries of $\boldsymbol{x}$ in each iteration, that is, the PTGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in $\max_{i=1,\ldots,N}\{s_i/M_i\}$ iterations. In particular, when $M_i = s_i$, the PTGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in the first iteration.

Remark 3.3. The PTGOMP algorithm is a generalization of TGOMP algorithm, Theorem 3.6 aligns with the Theorem 3.4 when $N = 1$.

Next, we provide a sufficient condition to ensure that the PTGOMP algorithm selects all correct indices and corresponding entries of $\boldsymbol{x}$ in the first iteration when $N = 2$, where A_1 and A_2 are general matrices.

Corollary 3.6. *Assume that $\boldsymbol{x}$ is an (s_1, s_2)-piecewise sparse ternary signal with $s = s_1 + s_2$, noise vector $\boldsymbol{v}$ satisfies $\|\boldsymbol{v}\|_2 = 0$, M_i is the number of selected indices in each iteration in the i-th component by the PTGOMP algorithm for $i = 1, 2$ and $M = M_1 + M_2$. Let $\alpha_i = \mu^{i,i}/\mu$, where $\mu^{i,i}$ denotes the coherence of the sub-matrix A_i for $i = 1, 2$. If the mutual coherence μ of the matrix $A = [A_1, A_2]$ satisfies*

$$\mu < \min\left\{\frac{1}{2(\alpha_1 s_1 + s_2) - \alpha_1}, \frac{1}{2(\alpha_2 s_2 + s_1) - \alpha_2}\right\}, \tag{3.25}$$

when $M_i = s_i$, the PTGOMP algorithm exactly recovers the support S and entries of $\boldsymbol{x}$ in the first iteration.

Moreover, we consider that A_1 and A_2 are orthogonal matrices.

Corollary 3.7. *Assume that $\boldsymbol{x}$ is an (s_1, s_2)-piecewise sparse ternary signal with $s = s_1 + s_2$, noise vector $\boldsymbol{v}$ satisfies $\|\boldsymbol{v}\|_2 = 0$, M_i is the number of selected indices in each iteration in the i-th component by the PTGOMP algorithm for $i = 1, 2$ and $M = M_1 + M_2$. Let μ be the mutual coherence of the matrix $A = [A_1, A_2]$, where A_i is an orthogonal matrix for $i = 1, 2$. If the mutual coherence μ of the matrix $A = [A_1, A_2]$ satisfies*

$$\mu < \min\left\{\frac{1}{2s_2}, \frac{1}{2s_1}\right\}, \tag{3.26}$$

when $M_i = s_i$, the PTGOMP algorithm exactly recovers the support S and entries of $\boldsymbol{x}$ in the first iteration.

Example 3.1. Suppose that $A = [A_1, A_2]$ with the mutual coherence $\mu = 1/\sqrt{256}, \alpha_1 = 0.2$, and $\alpha_2 = 0.1$, respectively. We consider the sparsity upper bounds (3.19), (3.25) and (3.26), i.e., $N = 1$ and $N = 2$, respectively.

In Fig. 1, the blue solid line represents the general sparsity condition (3.19), the blue dashed line represents the condition $s < 1/\mu$ for a pair of orthogonal matrices, the red line represents condition (3.25) for a pair of general matrices, and the green line represents condition (3.26) for a pair of orthogonal matrices. Fig. 1 compares the sufficient conditions (3.19) with (3.25) and (3.26). It shows that the sparsity constraints on the sparse ternary signal are relaxed by utilizing the piecewise structure of the algorithm.

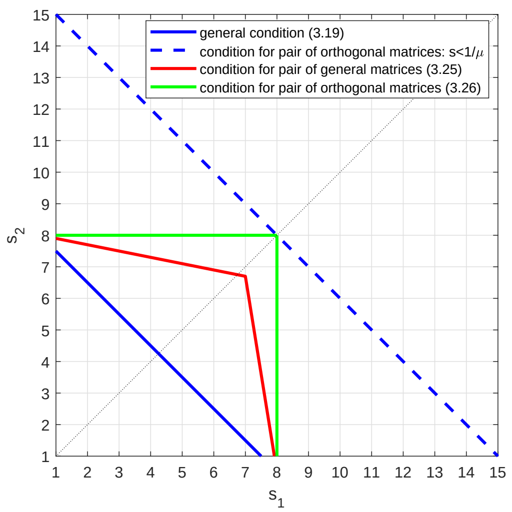


Figure 1: Comparison of the conditions of sparsity.

4. Simulations and numerical experiments

In this section, we conduct simulations for ternary sparse signals and experiments using images from the MNIST dataset to demonstrate the effectiveness of the proposed algorithms. For each sparse signal, we generate a random matrix $A \in \mathbb{R}^{m\times n}$ and add noise $\boldsymbol{v} \in \mathbb{R}^{m\times 1}$ sampled from a Gaussian distribution with standard deviation σ. Thus, the measurements can be represented as $\boldsymbol{b} = A\boldsymbol{x} + \boldsymbol{v}$. To evaluate the performance of the proposed algorithms, we use the Gaussian random matrix and the partial discrete cosine transform (pDCT) matrix as measurement matrices in Subsections 4.1 and 4.3. In Subsection 4.2, we use piecewise matrices $[I, H]$, $[I, DCT]$, $[DCT, O]$, $[H, G]$, $[G, DCT]$, and $[G, B]$ as measurement matrices, where I denotes the identity matrix, H denotes the Hadamard matrix, DCT denotes the discrete cosine transform matrix, and O denotes the orthogonalization of the random matrix, G represents the Gaussian matrix with mean 0 and variance $1/m$, and B represents a Bernoulli matrix.

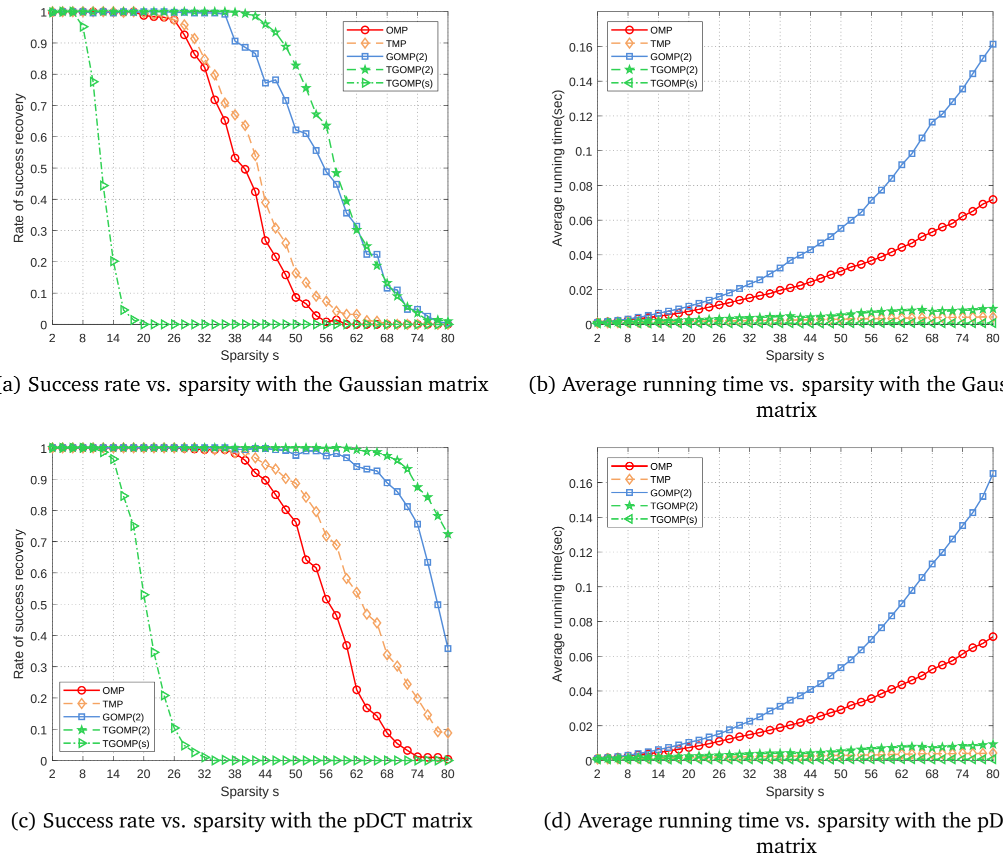
4.1. Reconstruction performance of the TMP and TGOMP algorithms

In this subsection, we compare the TMP and TGOMP algorithms with the OMP and GOMP algorithms on the reconstruction performance and phase transitions, using the Gaussian and the pDCT matrices as measurement matrices.

In the first set of simulations, we evaluate the reconstruction performance of the OMP, GOMP, TMP, and TGOMP algorithms in the presence of noise. A ternary sparse signal $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$ with the support S is generated by randomly selecting s entries from the set $\{1, 2, \ldots, n\}$. Noise vector $\boldsymbol{v}$ is also generated such that $\|\boldsymbol{v}\|_2 \leq 0.1$. A successful reconstruction is defined as

$$\frac{\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2}{\|\boldsymbol{x}\|_2} \leq 10^{-2},$$

where $\hat{\boldsymbol{x}}$ is the sparse signal estimated by the algorithm. The sparsity is set to values ranging from 2 to 80 in step size of 2, with $m = 256$, and $n = 512$. We conduct 500 tests for each simulation and record the success rates and average running times, as shown in Fig. 2.



(a) Success rate vs. sparsity with the Gaussian matrix

(b) Average running time vs. sparsity with the Gaussian matrix

(c) Success rate vs. sparsity with the pDCT matrix

(d) Average running time vs. sparsity with the pDCT matrix

Figure 2: Comparisons of reconstruction performance on success rate and running time vs. sparsity s for the Gaussian and pDCT matrices, respectively.

In Fig. 2, we compare the reconstruction performance of the TMP and TGOMP with the OMP and GOMP algorithms, and present their running times. The GOMP(2) and TGOMP(2) algorithms refer to the GOMP and TGOMP algorithms when $M = 2$, respectively, which select two indices in each iteration. We run the OMP, GOMP(2), TMP, and TGOMP(2) algorithms for s iterations. Besides, in order to verify that all correct indices are selected in the first iteration, we define TGOMP(s) as the TGOMP algorithm when $M = s$, which selects s indices in each iteration. The TGOMP(s) algorithm is run for only one iteration.

From Fig. 2, we observe that:

(1) Figs. 2(a) and 2(c) show that the TMP algorithm outperforms the OMP algorithm, and the TGOMP(2) algorithm is more effective than the GOMP(2) algorithm. It can be seen that the TGOMP(s) algorithm is indeed able to select all the correct indices in a single iteration when the sparsity is small. This result aligns with the theoretical fact presented in Section 3.

(2) Figs. 2(b) and 2(d) illustrate the average running time corresponding to Figs. 2(a) and 2(b), respectively. Specifically, the TMP and TGOMP algorithms are significantly faster than the OMP and GOMP algorithms. This speed advantage arises because the TMP algorithm avoids performing least squares during iterations, while the TGOMP algorithm only requires a single least squares computation after the iteration stopping criterion is met.

In the second set of simulations, we explore the phase transitions for the OMP, TMP, GOMP, and TGOMP algorithms with $n = 500$. We define m/n as the undersampling rate and s/m as the sparseness. This allows us to create a two-dimensional phase space $(m/n, s/m) \in [0, 1]^2$, which describes the difficulty of achieving successful recovery. We vary m/n from 0.05 to 1 in 30 steps, and s/m from 0.05 to 0.95 in 30 steps. We summarize the average results from 100 tests. The phase transitions for these four algorithms with the Gaussian matrix are shown in Fig. 3 and those with the pDCT matrix are shown in Fig. 4.

In Figs. 3 and 4, the brightness of the colors represents the success probability, with brighter colors indicating higher success probabilities and darker colors indicating lower success probabilities. From these figures, it is evident that for recovering ternary sparse signals, the TGOMP algorithm outperforms the GOMP algorithm, and the TMP algorithm outperforms the OMP algorithm.

Besides, Figs. 3(d) and 4(d) show that the TGOMP algorithm can also recover some sparse ternary signals when $m/n \in [0.9, 1]$ and $s/m \in [0.5, 0.95]$. The reason is that the TGOMP algorithm can select more indices than the GOMP algorithm, since the residuals of these two algorithms are different. In this example, the TGOMP algorithm selects two indices (or atoms) in each iteration, in some cases, all atoms of the measurement matrix may be selected after s iterations when $2s$ is close to m and n. Hence, it increases the probabilities to recover sparse ternary signals.

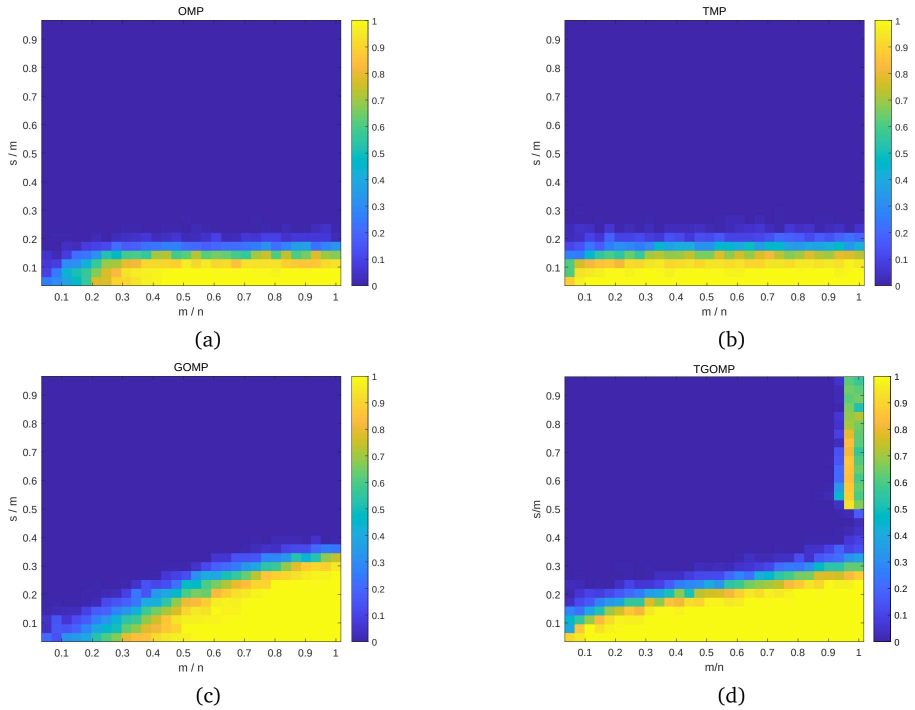


Figure 3: Phase transitions for the OMP, TMP, GOMP, and TGOMP algorithms using Gaussian matrix with $n = 500$.

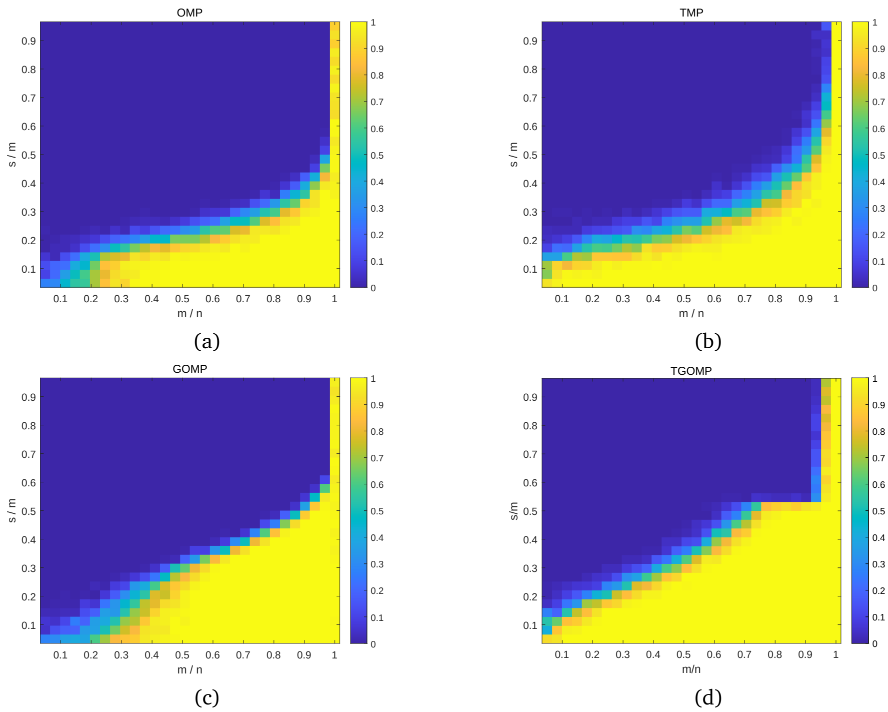
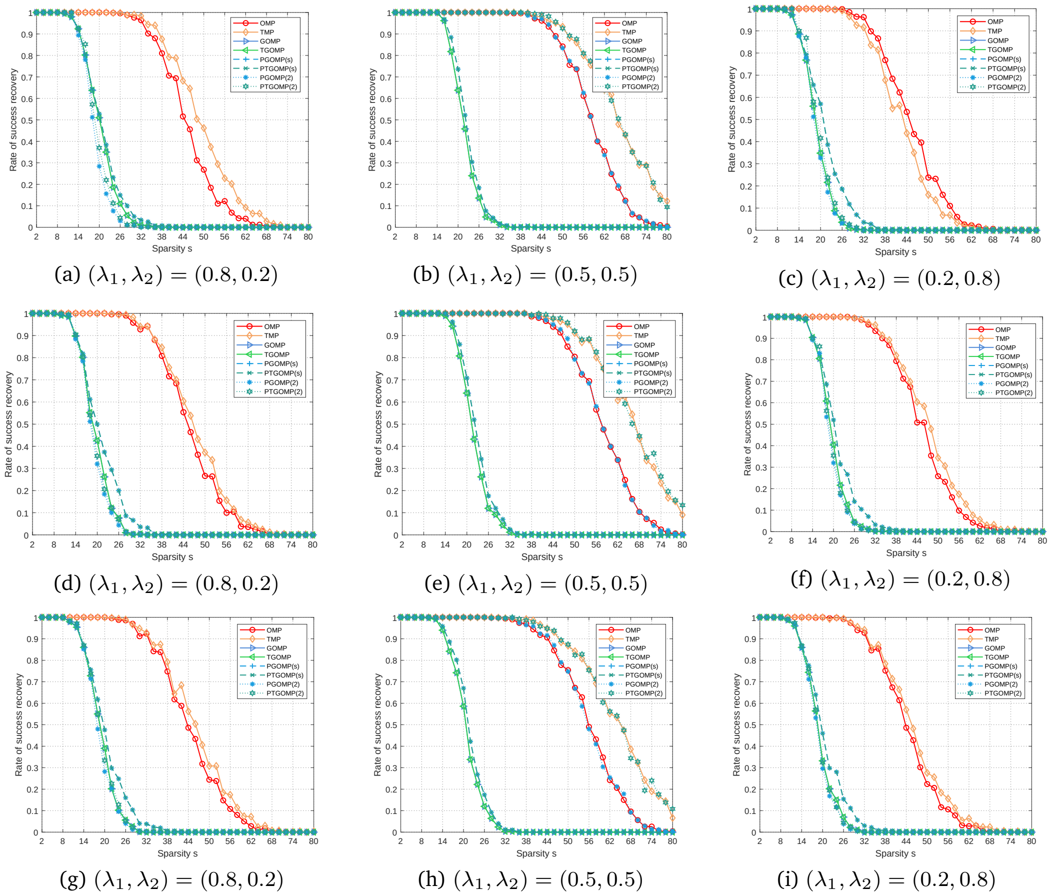


Figure 4: Phase transitions for the OMP, TMP, GOMP, and TGOMP algorithms using pDCT matrix with $n = 500$.

4.2. Reconstruction performance of the PTGOMP algorithm

For piecewise algorithms, we only test the case where $N = 2$ in this subsection. The PTGOMP(M) algorithm selects (M_1, M_2) indices in each iteration with $M = M_1 + M_2$, and the sparsity s is the multiple of M, where $s = s_1 + s_2$. Note that the piecewise strategy of the PGOMP(M) algorithm is similar to that of the PTGOMP(M) algorithm.

We compare the TMP, TGOMP(s), PTGOMP(s), and PTGOMP(2) algorithms with OMP, GOMP(s), PGOMP(s), and PGOMP(2) algorithms on the reconstruction performance with $m = 256, n = 512$. The GOMP(s), TGOMP(s), PGOMP(s), and PTGOMP(s) select s indices in each iteration and are run for one iteration. The PGOMP(2) and PTGOMP(2) algorithms select two indices in each iteration and are run for $s_{\max} = \max_{i=1,2} s_i$ iterations. We present the success recovery rates of these algorithms under different matrices and sparsity structures in Figs. 5 and 6.



(a) $(\lambda_1, \lambda_2) = (0.8, 0.2)$ (b) $(\lambda_1, \lambda_2) = (0.5, 0.5)$ (c) $(\lambda_1, \lambda_2) = (0.2, 0.8)$

(d) $(\lambda_1, \lambda_2) = (0.8, 0.2)$ (e) $(\lambda_1, \lambda_2) = (0.5, 0.5)$ (f) $(\lambda_1, \lambda_2) = (0.2, 0.8)$

(g) $(\lambda_1, \lambda_2) = (0.8, 0.2)$ (h) $(\lambda_1, \lambda_2) = (0.5, 0.5)$ (i) $(\lambda_1, \lambda_2) = (0.2, 0.8)$

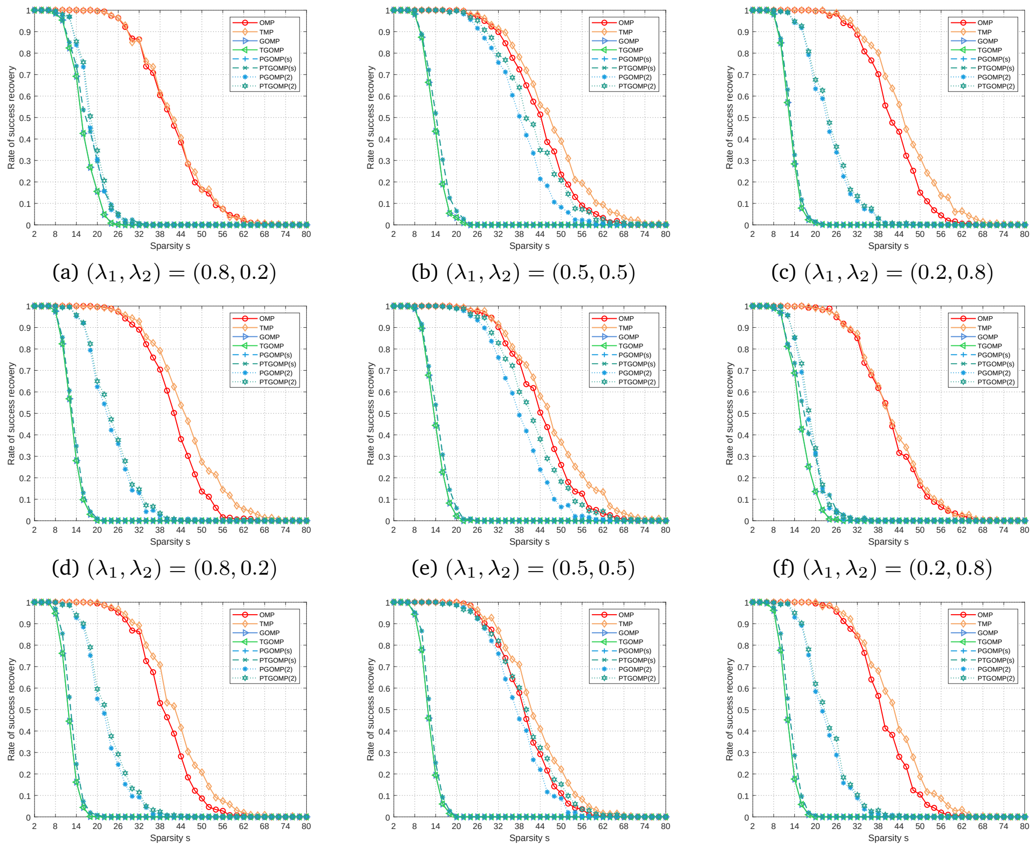
Figure 5: Reconstruction performance of the OMP, TMP, GOMP(s), TGOMP(s), PGOMP(s), PTGOMP(s), PGOMP(2), and PTGOMP(2) algorithms versus three sparsity structures with biorthogonal matrix.

In Fig. 5, we present the reconstruction performance of the OMP, TMP, GOMP(s), TGOMP(s), PGOMP(s), PTGOMP(s), PGOMP(2), and PTGOMP(2) algorithms for three sparsity structures $(\lambda_1, \lambda_2) = (s_1/s, s_2/s) = (0.8, 0.2), (0.5, 0.5), (0.2, 0.8)$. The matrices used in the experiments corresponding to Figs. 5(a)-5(c) are $[I, H]$ in the first row. The matrices corresponding to Figs. 5(d)-5(f) are $[I, DCT]$ in the second row. The matrices corresponding to Figs. 5(g)-5(i) are $[DCT, O]$ in the third row.

From Fig. 5, we observe that:

(1) By comparing the three subfigures in each row, we observe that the piecewise algorithms PGOMP(2) and PTGOMP(2) exhibit higher success rates under the sparsity structure $(0.5, 0.5)$ compared to $(0.8, 0.2)$ and $(0.2, 0.8)$.

(2) The subfigures in the second column show that the TMP algorithm has better recovery performance than the OMP algorithm when the sparsity increases under the sparsity structure $(0.5, 0.5)$.



(a) $(\lambda_1, \lambda_2) = (0.8, 0.2)$ (b) $(\lambda_1, \lambda_2) = (0.5, 0.5)$ (c) $(\lambda_1, \lambda_2) = (0.2, 0.8)$

(d) $(\lambda_1, \lambda_2) = (0.8, 0.2)$ (e) $(\lambda_1, \lambda_2) = (0.5, 0.5)$ (f) $(\lambda_1, \lambda_2) = (0.2, 0.8)$

(g) $(\lambda_1, \lambda_2) = (0.8, 0.2)$ (h) $(\lambda_1, \lambda_2) = (0.5, 0.5)$ (i) $(\lambda_1, \lambda_2) = (0.2, 0.8)$

Figure 6: Reconstruction performance of the OMP, TMP, GOMP(s), TGOMP(s), PGOMP(s), PTGOMP(s), PGOMP(2), and PTOGMP(2) algorithms versus three sparsity structures.

This suggests that if A is a biorthogonal matrix, the TMP, PGOMP(2), and PTGOMP(2) algorithms are more effective with a uniform sparsity structure.

In Fig. 6, we present the reconstruction performance of the OMP, TMP, GOMP(s), TGOMP(s), PGOMP(s), PTGOMP(s), PGOMP(2), and PTGOMP(2) algorithms for the three sparsity structures. The matrix combinations corresponding to Figs. 6(a)-6(c) are $[H,G]$ in the first row. The matrix combinations corresponding to Figs. 6(d)-6(f) are $[G,DCT]$ in the second row. The matrix combinations corresponding to Figs. 6(g)-6(i) are $[G,B]$ in the third row.

By comparing the three subfigures in each row, we observe that for the piecewise algorithms PGOMP(2) and PTGOMP(2), the success rates can be improved with different matrix combinations when matched to the appropriate sparsity structure. This suggests that selecting the proper matrix combinations with the sparsity structure can significantly improve the recovery performance of these algorithms.

Specifically, we present success recovery rates from Figs. 5 and 6 when $s=12$ in Table 1. The matrices $[I,H]$, $[I,DCT]$, $[DCT,O]$ are biorthogonal with parameters $(\alpha_1,\alpha_2)=(0,0)$. The parameters of the combinations $[H,G]$, $[G,DCT]$, $[G,B]$ are approximately $(\alpha_1,\alpha_2)=(0,1)$, $(1,0)$, and $(1,1)$, respectively.

From Table 1, we can obtain two key results:

(1) The piecewise strategy enhances the recovery performance of the algorithms.

(2) By combining different matrices with the appropriate sparsity structure, the reconstruction performance of these algorithms can be significantly enhanced under the same conditions.

Table 1: The rates of success recovery for the OMP, TMP, GOMP(s), TGOMP(s), PGOMP(s), PTGMP(s), PGOMP(2), PTGOMP(2) algorithms under different matrices and piecewise structures.

A	(λ_1,λ_2)	OMP	TMP	GOMP(s)	TGOMP(s)	PGOMP(s)	PTGOMP(s)	PGOMP(2)	PTGOMP(2)
$[I,H]$	$(0.8,0.2)$	1	1	0.98	0.98	$(0.98,0.98)$	$(1,0.98)$	$(1,1)$	$(1,1)$
	$(0.5,0.5)$	1	1	1	1	(**1, 1**)	(**1, 1**)	$(1,1)$	$(1,1)$
	$(0.2,0.8)$	1	1	0.972	0.972	$(0.972,0.972)$	$(0.972,1)$	$(1,1)$	$(1,1)$
$[I,DCT]$	$(0.8,0.2)$	1	1	0.986	0.986	$(0.986,0.986)$	$(1,0.986)$	$(0.994,0.994)$	$(1,0.994)$
	$(0.5,0.5)$	1	1	1	1	(**1, 1**)	(**1, 1**)	(**1, 1**)	(**1, 1**)
	$(0.2,0.8)$	1	1	0.986	0.986	$(0.986,0.986)$	$(0.986,1)$	$(0.992,0.992)$	$(0.992,1)$
$[DCT,O]$	$(0.8,0.2)$	1	1	0.954	0.954	$(0.956,0.956)$	$(1,0.956)$	$(0.97,0.97)$	$(1,0.972)$
	$(0.5,0.5)$	1	1	0.992	0.992	(**0.994, 0.994**)	$(0.998,0.998)$	(**1, 1**)	(**1, 1**)
	$(0.2,0.8)$	1	1	0.96	0.96	$(0.964,0.964)$	$(0.964,1)$	$(0.97,0.97)$	$(0.974,1)$
$[H,G]$	$(0.8,0.2)$	1	1	0.824	0.824	(**0.85, 0.85**)	(**1, 0.85**)	$(0.968,0.968)$	$(1,0.97)$
	$(0.5,0.5)$	1	1	0.664	0.664	$(0.722,0.722)$	$(1,0.722)$	$(1,1)$	$(1,1)$
	$(0.2,0.8)$	1	1	0.57	0.57	$(0.632,0.632)$	$(0.97,0.64)$	$(0.997,0.99)$	$(0.99,1)$
$[G,DCT]$	$(0.8,0.2)$	1	1	0.568	0.568	$(0.606,0.61)$	$(0.654,0.942)$	$(0.996,0.996)$	$(1,0.996)$
	$(0.5,0.5)$	1	1	0.678	0.678	$(0.72,0.72)$	$(0.722,0.998)$	$(1,1)$	$(1,1)$
	$(0.2,0.8)$	1	1	0.812	0.812	(**0.824, 0.824**)	(**0.824, 1**)	$(0.958,0.958)$	$(0.956,1)$
$[G,B]$	$(0.8,0.2)$	1	1	0.446	0.446	$(0.558,0.558)$	$(0.642,0.866)$	$(0.988,0.988)$	$(1,0.986)$
	$(0.5,0.5)$	1	1	0.46	0.46	$(0.55,0.55)$	$(0.734,0.748)$	$(1,1)$	$(1,1)$
	$(0.2,0.8)$	1	1	0.452	0.452	$(0.554,0.552)$	$(0.85,0.646)$	$(0.992,0.992)$	$(0.994,1)$

Specifically, firstly, by comparing the success rates in the 6-th column with those in the 8-th and 10-th columns, we find that the piecewise algorithms PTGOMP(s) and PTGOMP(2) achieve higher success rates across the three sparsity structures compared to the TGOMP(s) algorithm. Similarly, comparing the 5-th column with the 7-th and 9-th columns, we observe that the piecewise algorithms PGOMP(s) and PGOMP(2) outperform GOMP(s).

Secondly, for the piecewise matrix $A = [H, G]$ and the three sparsity structures, the success rates of the PGOMP(s) are $(0.85, 0.85)$, $(0.722, 0.722)$, and $(0.632, 0.632)$, respectively. The success rates of the PTGOMP(s) are $(1, 0.85)$, $(1, 0.722)$, and $(0.97, 0.64)$, respectively. For the matrix $A = [G, DCT]$ and the three sparsity structures, the success rates of the PGOMP(s) are $(0.606, 0.61)$, $(0.72, 0.72)$, and $(0.824, 0.824)$, respectively. The success rates of the PTGOMP(s) are $(0.654, 0.924)$, $(0.722, 0.998)$, and $(0.824, 1)$, respectively. From this, we conclude that different matrix combinations, when matched with the appropriate sparsity structure, can significantly improve the recovery performance of these algorithms under the same conditions. These numerical results align well with the theoretical results.

4.3. Support recovery of handwritten digit images from MNIST dataset

Furthermore, we consider the support recovery of handwritten digit images from the MNIST dataset [29]. The dataset consists of digit images, each with a pixel size of 28×28. Most pixels in these images are inactive (0), and the nonzero pixels are grouped together. Therefore, these images are naturally piecewise, allowing for effective recovery using the proposed algorithms. We use support matching (SM) to measure how well the support of the recovered signal matches that of the original signal. The proposed algorithms TMP, TGOMP, and PTGOMP are compared with OMP, BMP, GOMP, and BGOMP algorithms, focusing on support recovery and running time (RT) efficiency. We only select the indices with the largest positive magnitude, i.e., $\boldsymbol{z}_j = \max_{j\in\Omega\setminus\Lambda^{k-1}} \boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{r}^{k-1} > 0$. We denote these algorithms as TMPM, TGOMPM, and PTGOMPM2p, where PTGOMPM2p denotes the PTGOMPM algorithm when $N = 2$.

In the experiment, we vectorize the sparse image from Fig. 7 to be reconstructed as $\boldsymbol{x} \in \mathbb{R}^{784\times 1}$ with $N = 2$. For each sparse signal, we use both a random Gaussian matrix

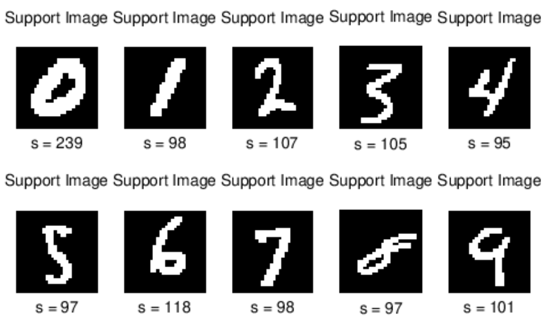


Figure 7: Original support images from the MNIST dataset. The number below each image represents the sparsity s. The labels of the images are sequentially "0"-"9".

and a pDCT matrix $A \in \mathbb{R}^{400\times784}$, and add noise $\boldsymbol{v} \in \mathbb{R}^{400\times1}$ with standard deviation $\sigma = 0.01$. We conduct 100 tests, and the average SM and RT of each algorithm, using the Gaussian and pDCT matrices, are shown in Tables 2 and 4, respectively.

Tables 2 and 4 show the average SM and RT for the OMP, BMP, TMPM, GOMP(4), BGOMP(4), TGOMPM(4), and PTGOMPM2p(4) algorithms with the Gaussian matrix, and pDCT matrix, respectively. The first row of two tables contains the labels of the original support images from Fig. 7. The second to last row represents the average SM and RT for each algorithm. Table 3 presents the average SM and RT for each algorithm for ten images, as shown in Table 2. A similar analysis is conducted for the pDCT matrix to evaluate the support recovery performance of the proposed algorithms, as shown in Table 4. The average SM and RT for each algorithm using the pDCT matrix are shown in Table 5.

From Tables 2-5, we can observe that:

(1) The OMP, BMP, and TMPM algorithms select one index each iteration. Among these, the TMPM algorithm has the best average SM with the shortest running time.

(2) The GOMP, BGOMP, TGOMPM, and PTGOMPM2p algorithms select four indices each iteration. Among these, the PTGOMPM2p algorithm has the best average SM.

Table 2: The average SMs and RTs for the OMP, BMP, TMPM, GOMP(4), BGOMP(4), TGOMPM(4), PTGOMPM2p(4) algorithms with the Gaussian matrix.

Labels	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
OMP	65.86%	94.28%	91.78%	92.36%	95.48%	94.92%	88.58%	94.68%	94.57%	93.48%
	1.5160s	0.1694s	0.2052s	0.1981s	0.1667s	0.1707s	0.2618s	0.1735s	0.1682s	0.1828s
BMP	70.78%	90.03%	88.51%	88.97%	90.48%	89.77%	86.98%	90.10%	89.84%	89.26%
	0.0228s	0.0094s	0.0100s	0.0099s	0.0095s	0.0094s	0.0112s	0.0093s	0.0092s	0.0096s
TMPM	**78.57%**	**98.31%**	**97.60%**	**97.70%**	**98.97%**	**98.40%**	**95.61%**	**98.36%**	**98.43%**	**98.12%**
	0.0217s	0.0091s	0.0097s	0.0096s	0.0094s	0.0091s	0.0109s	0.0092s	0.0089s	0.0094s
GOMP	65.65%	99.63 %	97.37%	98.93%	99.80%	99.82%	91.54%	99.67%	**100.00%**	99.54%
	1.5105s	0.4461s	0.5611s	0.5319s	0.4272s	0.4438s	0.7462s	0.4529s	1.3774s	0.4778s
BGOMP	78.98%	98.91%	97.30%	97.86%	99.27%	98.89%	94.94%	98.92%	94.10%	98.48%
	0.2950s	0.0304s	0.0350s	0.0345s	0.0302s	0.0308s	0.0419s	0.0311s	0.0724s	0.0315s
TGOMPM	81.94%	99.92%	99.38%	99.66%	99.97%	99.91%	98.33%	99.88%	99.27%	99.79%
	0.0810s	0.0234s	0.0264s	0.0257s	0.0229s	0.0237s	0.0309s	0.0234s	0.0502s	0.0240s
PTGOMPM2p	**82.95%**	**100.00%**	**99.95%**	**99.97%**	**100.00%**	**100.00%**	**99.76%**	**100.00%**	98.23%	**100.00%**
	0.2251s	0.0511s	0.0575s	0.0561s	0.0511s	0.0518s	0.0678s	0.0513s	0.0504s	0.0521s

Table 3: Evaluation of the algorithms for support recovery of real images in Fig. 7 using the Gaussian matrix, with the average SMs and RTs summarized from Table 2.

Algorithms	OMP	BMP	TMPM	GOMP	BGOMP	TGOMPM	PTGOMPM2p
SM (%)	91.30	89.16	**97.25**	97.60	97.24	**98.91**	**99.59**
RT (sec)	0.3212	0.0110	**0.0107**	0.6975	0.0633	0.0332	0.0714

Table 4: The average SMs and RTs for the OMP, BMP, TMPM, GOMP(4), BGOMP(4), TGOMPM(4), PTGOMPM2p(4) algorithms with the pDCT matrix.

Labels	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
OMP	64.62%	97.75%	96.78%	96.94%	98.50%	97.74%	94.60%	98.43%	98.59%	98.17%
	1.4692s	0.1617s	0.1951s	0.1885s	0.1504s	0.1584s	0.2436s	0.1612s	0.1594s	0.1725s
BMP	77.47%	96.55%	95.47%	95.37%	96.68%	95.87%	93.92%	96.36%	96.71%	96.29%
	0.0207s	0.0087s	0.0096s	0.0093s	0.0084s	0.0086s	0.0105s	0.0087s	0.0087s	0.0091s
TMPM	**82.98%**	**99.57%**	**99.49%**	**99.68%**	**99.65%**	**99.73%**	**99.19%**	**99.82%**	**99.72%**	**99.83%**
	0.0203s	0.0084s	0.0093s	0.0091s	0.0081s	0.0084s	0.0101s	0.0084s	0.0084s	0.0087s
GOMP	64.98%	98.49%	98.37%	98.10%	99.39%	99.19%	95.98%	99.59%	99.41%	99.46%
	1.4780s	0.4277s	0.5428s	0.5172s	0.3940s	0.4181s	0.7102s	0.4300s	1.3066s	0.4657s
BGOMP	79.55%	99.81%	99.03%	99.99%	99.99%	100.00%	99.77%	100.00%	**99.91%**	100.00%
	0.2773s	0.0288s	0.0332s	0.0324s	0.0277s	0.0284s	0.0384s	0.0290s	0.0682s	0.0304s
TGOMPM	80.89%	99.80%	99.99%	99.99%	100.00%	100.00%	99.74%	100.00%	99.82%	100.00%
	0.0704s	0.0223s	0.0251s	0.0243s	0.0213s	0.0219s	0.0287s	0.0223s	0.0399s	0.0233s
PTGOMPM2p	**84.74%**	**99.95%**	**100.00%**	**100.00%**	100.00%	100.00%	**100.00%**	100.00%	99.58%	100.00%
	0.2215s	0.0640s	0.0727s	0.0678s	0.0581s	0.0593s	0.0829s	0.0628s	0.0619s	0.0647s

Table 5: Evaluation of the algorithms for support recovery of real images in Fig. 7 using the pDCT matrix, with the average SMs and RTs summarized from Table 4.

Algorithms	OMP	BMP	TMPM	GOMP	BGOMP	TGOMPM	PTGOMPM2p
SM (%)	96.01	95.24	**99.17**	98.00	99.21	**99.42**	**99.92**
RT (sec)	0.3060	0.0102	0.0099	0.6690	0.0594	0.0299	0.0816

Note that the piecewise algorithm can achieve the best SM for almost all images in Fig. 7, as shown in Tables 2 and 4, and it performs the best in terms of the average SM for the ten images, as shown in Tables 3 and 5.

5. Conclusion

This paper addresses the problem of recovering ternary sparse signals and introduces three novel algorithms: ternary matching pursuit, ternary generalized orthogonal matching pursuit, and piecewise ternary generalized orthogonal matching pursuit. These algorithms aim to recover ternary sparse signals efficiently, and we provide theoretical results based on mutual coherence and the restricted isometry property. The TMP algorithm, inspired by binary matching pursuit, assigns values of 1 or -1 to the most correlated residual. We establish theoretical results that ensure successful recovery under suitable conditions. The TGOMP algorithm improves upon the TMP algorithm by selecting multiple (M) indices in each iteration, with theoretical results guaranteeing the recovery of all correct indices under certain conditions. Specifically, we derive a sufficient condition $\mu < 1/(2s-1)$, which ensures that all correct indices are selected in the first iteration. The PTGOMP algorithm further enhances TGOMP by using a piecewise selection strategy, improving recovery performance. We also present theoretical guarantees for PTGOMP based on mutual coherence. Finally, we validate the effectiveness of our proposed algorithms through simulations and numerical experiments.

Acknowledgments

This work was supported by the National Key R&D Program of China (Grant No. 2023YFA1009200), by the National Natural Science Foundation of China (Grant Nos. 12271079, 12494552), and by the Fundamental Research Fund for the Central Universities of China (Grant No. DUT24LAB127).

Appendix A Proof of Theorem 3.4

Proof. Let S be the support of an s-sparse ternary signal $\boldsymbol{x} \in \mathbb{R}^n$ and $S = S^+ \cup S^-$, where

- $S^+ = \{i : x_i = 1, i \in S\}, \hat{s}_1 = |S^+|$,
- $S^- = \{i : x_i = -1, i \in S\}, \hat{s}_2 = |S^-|$,

then the sparsity $s = \hat{s}_1 + \hat{s}_2$. Let $\Lambda^{k-1} = \Lambda_+^{k-1} \cup \Lambda_-^{k-1}$, where Λ_+^{k-1} and Λ_-^{k-1} denote the sets of indices selected for positive and negative entries of $\hat{\boldsymbol{x}}$ in the first $k-1$ iterations, respectively, then $|\Lambda^{k-1}| = (k-1)M$. From Algorithm 3.2, we have

$$\boldsymbol{r}^{k-1} = \boldsymbol{b} - A_{\Lambda^{k-1}} \hat{\boldsymbol{x}}_{\Lambda^{k-1}}.$$

Let $i_j \in \{i_1, \ldots, i_M\} \subseteq S \setminus \Lambda^{k-1}$, we define

$$b_{i_j}^k = \left|\boldsymbol{a}_{i_j}^{\mathrm{T}} \boldsymbol{r}^{k-1}\right|, \quad \rho^k = \left\|\boldsymbol{A}_{S^c}^{\mathrm{T}} \boldsymbol{r}^{k-1}\right\|_\infty.$$

Similar to the proof of Theorem 3.1, we provide the sufficient condition for the step 3 of Algorithm 3.2 in each iteration. Then, we establish a sufficient condition that guarantees the TGOMP algorithm selects M correct indices and corresponding entries of $\boldsymbol{x}$ in each iteration.

To ensure the validity of the step 3 of Algorithm 3.2, we need to guarantee that the following inequalities are satisfied in the k-th iteration for $k = 1, \ldots, s/M$:

$$\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{r}^{k-1} > 0, \quad \text{if} \quad x_i = 1, \quad i \in S^+ \setminus \Lambda_+^{k-1}, \tag{A.1}$$

$$\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{r}^{k-1} < 0, \quad \text{if} \quad x_i = -1, \quad i \in S^- \setminus \Lambda_-^{k-1}. \tag{A.2}$$

Besides, to guarantee that the TGOMP algorithm selects M correct indices and corresponding entries of $\boldsymbol{x}$, we need the following inequality to hold in the k-th iteration for $k = 1, \ldots, s/M$. That is, for any $i_j \in \{i_1, \ldots, i_M\} \subseteq S \setminus \Lambda^{k-1}$, we need

$$b_{i_j}^k > \rho^k \tag{A.3}$$

to hold. In particular, when $M = s$, the TGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in the first iteration.

Assuming that the TGOMP algorithm has selected $(k-1)M$ correct indices and corresponding entries in the first $k-1$ iterations for $k=1,\ldots,s/M$, then we have $\Lambda_+^{k-1} \subset S^+$, $\Lambda_-^{k-1} \subset S^-$ and $\hat{\boldsymbol{x}}_{\Lambda^{k-1}} = \boldsymbol{x}_{\Lambda^{k-1}}$. The residual vector $\boldsymbol{r}^{k-1}$ is given by

$$\boldsymbol{r}^{k-1} = \boldsymbol{b} - A_{\Lambda^{k-1}}\hat{\boldsymbol{x}}_{\Lambda^{k-1}} = A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}} + \boldsymbol{v}.$$

In the k-th iteration, when $x_i = 1$, $i \in S^+ \setminus \Lambda_+^{k-1}$, similar to (3.5), we have

$$\begin{aligned} \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{r}^{k-1} &\geq 1 + \mu - \left(\left|S^+ \setminus \Lambda_+^{k-1}\right| + \left|S^- \setminus \Lambda_-^{k-1}\right|\right)\mu - \epsilon \\ &= 1 + \mu - \left|S \setminus \Lambda^{k-1}\right|\mu - \epsilon. \end{aligned} \tag{A.4}$$

Similar to (3.6), when $x_i = -1$, $i \in S^- \setminus \Lambda_-^{k-1}$, we obtain

$$\begin{aligned} \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{r}^{k-1} &\leq -1 - \mu + \left(\left|S^+ \setminus \Lambda_+^{k-1}\right| + \left|S^- \setminus \Lambda_-^{k-1}\right|\right)\mu + \epsilon \\ &= -1 - \mu + \left|S \setminus \Lambda^{k-1}\right|\mu + \epsilon. \end{aligned} \tag{A.5}$$

Since $k \geq 1$, we have $|S \setminus \Lambda^{k-1}| = s - (k-1)M \leq s$, by (A.4) and (A.5), if

$$(s-1)\mu < 1 - \epsilon \tag{A.6}$$

is satisfied, we obtain (A.1) and (A.2) hold. That is, (A.6) is a sufficient condition for (A.1) and (A.2) in the k-th iteration.

For the left-hand side of (A.3), and any $i_j \in \{i_1,\ldots,i_M\} \subseteq S \setminus \Lambda^{k-1}$, we have

$$\begin{aligned} b_{i_j}^k &= \left|\boldsymbol{a}_{i_j}^{\mathrm{T}}\boldsymbol{r}^{k-1}\right| \\ &= \left|\boldsymbol{a}_{i_j}^{\mathrm{T}}\left(A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}} + \boldsymbol{v}\right)\right| \\ &\geq \left|\boldsymbol{a}_{i_j}^{\mathrm{T}}A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\right| - \left|\boldsymbol{a}_{i_j}^{\mathrm{T}}\boldsymbol{v}\right| \\ &\geq |x_{i_j}| - \sum_{j\in S\setminus(\Lambda^{k-1}\cup\{i_j\})}\left|\boldsymbol{a}_{i_j}^{\mathrm{T}}\boldsymbol{a}_j\right| - \|\boldsymbol{v}\|_2 \\ &\geq 1 - \left(\left|S \setminus \Lambda^{k-1}\right| - 1\right)\mu - \epsilon, \end{aligned}$$

and for the right-hand side of (A.3), we obtain

$$\begin{aligned} \rho^k &= \left\|\boldsymbol{A}_{S^c}^{\mathrm{T}}\boldsymbol{r}^{k-1}\right\|_\infty \\ &\triangleq \left|\boldsymbol{a}_{j0}^{\mathrm{T}}\left(A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}} + \boldsymbol{v}\right)\right| \\ &\leq \left|\boldsymbol{a}_{j0}^{\mathrm{T}}A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\right| + \left|\boldsymbol{a}_{j0}^{\mathrm{T}}\boldsymbol{v}\right| \\ &\leq \sum_{j\in S\setminus\Lambda^{k-1}}\left|\boldsymbol{a}_{j0}^{\mathrm{T}}\boldsymbol{a}_j\right| + \|\boldsymbol{v}\|_2 \\ &\leq \left|S \setminus \Lambda^{k-1}\right|\mu + \epsilon. \end{aligned}$$

Therefore, we can obtain that (A.3) holds if

$$2\big|S \setminus \Lambda^{k-1}\big|\mu + 2\epsilon < 1 + \mu. \tag{A.7}$$

Since $k \geq 1$, we have $|S \setminus \Lambda^{k-1}| = s - (k-1)M \leq s$, we obtain

$$s < \frac{1 + \mu - 2\epsilon}{2\mu}$$

is a sufficient condition for (A.7).

Hence, if (3.19) and (3.20) are satisfied, when $M \leq s$, the TGOMP algorithm selects M correct indices and corresponding entries of $\boldsymbol{x}$ in each iteration, that is, the TGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in s/M iterations. In particular, when $M = s$, the TGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in the first iteration. This completes the proof of Theorem 3.4. □

Appendix B Proof of Theorem 3.5

Proof. Let S be the support of an s-sparse ternary signal $\boldsymbol{x} \in \mathbb{R}^n$, $\Lambda^{k-1} = \Lambda_+^{k-1} \cup \Lambda_-^{k-1}$, where Λ_+^{k-1} and Λ_-^{k-1} denote the sets of indices selected for positive and negative entries of $\hat{\boldsymbol{x}}$ in the first $k-1$ iterations, respectively. Let $l = |S \cap \Lambda^{k-1}|$, then $|\Lambda^{k-1}| = (k-1)M$ and $|S \setminus \Lambda^{k-1}| = s - l$.

Let

$$b^k = \big\|\boldsymbol{A}_{S\setminus\Lambda^{k-1}}^{\mathrm{T}} \boldsymbol{r}^{k-1}\big\|_\infty, \quad \rho^k = \min\left\{\big|\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{r}^{k-1}\big|, i \in W_*^k\right\},$$

where W_*^k satisfies

$$\big\|\boldsymbol{A}_{W_*^k}^{\mathrm{T}} \boldsymbol{r}^{k-1}\big\|_1 = \max_{W^k \subseteq (S^c \setminus \Lambda^{k-1}), |W^k| = M} \big\|\boldsymbol{A}_{W^k}^{\mathrm{T}} \boldsymbol{r}^{k-1}\big\|_1.$$

To ensure the validity of the step 3 of Algorithm 3.2, we need to guarantee that the following inequalities are satisfied in the k-th iteration for $k = 1, \ldots, s$:

$$\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{r}^{k-1} > 0, \quad \text{if} \quad x_i = 1, \quad i \in S^+ \setminus \Lambda_+^{k-1}, \tag{B.1}$$

$$\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{r}^{k-1} < 0, \quad \text{if} \quad x_i = -1, \quad i \in S^- \setminus \Lambda_-^{k-1}. \tag{B.2}$$

Besides, to guarantee that the TGOMP algorithm selects at least one correct index and corresponding entry of $\boldsymbol{x}$, we need the following inequality to hold in the k-th iteration for $k = 1, \ldots, s$, i.e.:

$$b^k > \rho^k. \tag{B.3}$$

Assuming that the TGOMP algorithm selects at least one correct index and corresponding entry of $\boldsymbol{x}$ in the first $k-1$ iterations for $k = 1, \ldots, s$, and by the correctness of the sign identification, we have $\hat{\boldsymbol{x}}_{S\cap\Lambda^{k-1}} = \boldsymbol{x}_{S\cap\Lambda^{k-1}}$, and

$$\boldsymbol{r}^{k-1} = \boldsymbol{b} - A_{\Lambda^{k-1}} \hat{\boldsymbol{x}}_{\Lambda^{k-1}} = A_{S\setminus\Lambda^{k-1}} \boldsymbol{x}_{S\setminus\Lambda^{k-1}} - A_{\Lambda^{k-1}\setminus S} \hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S} + \boldsymbol{v}.$$

In the k-th iteration, when $x_i = 1, i \in S^+ \setminus \Lambda_+^{k-1}$, we have

$$\begin{aligned}
\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{r}^{k-1} &= \boldsymbol{a}_i^{\mathrm{T}} \big(A_{S \setminus \Lambda^{k-1}} \boldsymbol{x}_{S \setminus \Lambda^{k-1}} - A_{\Lambda^{k-1} \setminus S} \hat{\boldsymbol{x}}_{\Lambda^{k-1} \setminus S} + \boldsymbol{v} \big) \\
&= \boldsymbol{a}_i^{\mathrm{T}} \big(\boldsymbol{a}_i x_i + A_{S \setminus (\Lambda^{k-1} \cup i)} \boldsymbol{x}_{S \setminus (\Lambda^{k-1} \cup i)} - A_{\Lambda^{k-1} \setminus S} \hat{\boldsymbol{x}}_{\Lambda^{k-1} \setminus S} + \boldsymbol{v} \big) \\
&= x_i + \boldsymbol{a}_i^{\mathrm{T}} A_{\Theta_1} \tilde{\boldsymbol{x}}_{\Theta_1} + \boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{v} \\
&\geq 1 - \left| \boldsymbol{a}_i^{\mathrm{T}} A_{\Theta} \tilde{\boldsymbol{x}}_{\Theta} \right| - \left| \boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{v} \right| \\
&\geq 1 - \left\| \boldsymbol{a}_i^{\mathrm{T}} A_{\Theta_1} \tilde{\boldsymbol{x}}_{\Theta_1} \right\|_2 - \epsilon \\
&\geq 1 - \theta_{1,|\Theta_1|} \left\| \tilde{\boldsymbol{x}}_{\Theta_1} \right\|_2 - \epsilon,
\end{aligned} \tag{B.4}$$

where

$$\begin{aligned}
\Theta_1 &= \Big(S \setminus \big(\Lambda^{k-1} \cup \{i\} \big) \Big) \cup \big(\Lambda^{k-1} \setminus S \big), \\
\tilde{\boldsymbol{x}}_{\Theta_1} &= \big(\boldsymbol{x}_{S \setminus (\Lambda^{k-1} \cup \{i\})}^{\mathrm{T}}, -\hat{\boldsymbol{x}}_{\Lambda^{k-1} \setminus S}^{\mathrm{T}} \big)^{\mathrm{T}}
\end{aligned}$$

and the last inequality is due to (2.3).

When $x_i = -1, i \in S^- \setminus \Lambda_-^{k-1}$, we have

$$\begin{aligned}
\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{r}^{k-1} &= x_i + \boldsymbol{a}_i^{\mathrm{T}} A_{\Theta_1} \tilde{\boldsymbol{x}}_{\Theta_1} + \boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{v} \\
&\leq -1 + \left| \boldsymbol{a}_i^{\mathrm{T}} A_{\Theta_1} \tilde{\boldsymbol{x}}_{\Theta_1} \right| + \left| \boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{v} \right| \\
&\leq -1 + \theta_{1,|\Theta_1|} \left\| \tilde{\boldsymbol{x}}_{\Theta_1} \right\|_2 + \epsilon.
\end{aligned} \tag{B.5}$$

By (B.4) and (B.5), we obtain a sufficient condition for (B.1) and (B.2), i.e.,

$$\theta_{1,|\Theta_1|} \left\| \tilde{\boldsymbol{x}}_{\Theta_1} \right\|_2 + \epsilon < 1. \tag{B.6}$$

Since $l \geq k-1$, we have

$$|\Theta_1| = s - l - 1 + (k-1)M - l \leq s - 1 + (k-1)(M-2).$$

Next, we consider the values of M. When $M = 1, 2$, we have $|\Theta_1| < s - 1$ and $\|\tilde{\boldsymbol{x}}_{\Theta_1}\|_2 < \sqrt{s-1}$. By (2.1), we obtain

$$\delta_{s+1} < \frac{1-\epsilon}{\sqrt{s+1}} \tag{B.7}$$

is a sufficient condition for (B.6).

When $M \geq 3$, we have

$$|\Theta_1| < (s-1)(M-1), \quad \|\tilde{\boldsymbol{x}}_{\Theta_1}\|_2 < \sqrt{(s-1)(M-1)}.$$

By (2.1) and (2.2), we have

$$\theta_{1,|\Theta_1|} < \theta_{1,(s-1)(M-1)} < \sqrt{M-1}\,\theta_{1,s-1} < \sqrt{M-1}\,\delta_s,$$

then we obtain

$$\delta_{s+1} < \frac{1-\epsilon}{\sqrt{s-1}(M-1)} \tag{B.8}$$

is a sufficient condition for (B.6).

Therefore, when $M = 1, 2$, if (B.7) is satisfied, (B.1) and (B.2) hold in the k-th iteration. When $M \geq 3$, if (B.8) is satisfied, (B.1) and (B.2) hold in the k-th iteration.

Then, we prove (B.3) in the k-th iteration. For the left-hand side of (B.3), we have

$$\begin{aligned} b^k &= \big\|A^{\mathrm{T}}_{S\setminus\Lambda^{k-1}}\boldsymbol{r}^{k-1}\big\|_\infty \\ &= \big\|A^{\mathrm{T}}_{S\setminus\Lambda^{k-1}}\big(A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}} - A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S} + \boldsymbol{v}\big)\big\|_\infty \\ &\geq \big\|A^{\mathrm{T}}_{S\setminus\Lambda^{k-1}}A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\big\|_\infty - \big\|A^{\mathrm{T}}_{S\setminus\Lambda^{k-1}}A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S}\big\|_\infty - \epsilon \\ &= \big\|A^{\mathrm{T}}_{S\setminus\Lambda^{k-1}}A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\big\|_\infty - \big|\boldsymbol{a}^{\mathrm{T}}_{i_0}A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S}\big| - \epsilon, \end{aligned} \tag{B.9}$$

where $i_0 \in S \setminus \Lambda^{k-1}$ such that

$$\big|\boldsymbol{a}^{\mathrm{T}}_{i_0}A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S}\big| \triangleq \big\|A^{\mathrm{T}}_{S\setminus\Lambda^{k-1}}A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S}\big\|_\infty.$$

For the right-hand side of (B.3), we have

$$\begin{aligned} \rho^k &= \min\Big\{\big|\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{r}^{k-1}\big|, i \in W_*^k\Big\} \\ &\leq \frac{1}{M}\big\|A^{\mathrm{T}}_{W_*^k}\boldsymbol{r}^{k-1}\big\|_1 = \frac{1}{M}\big\|A^{\mathrm{T}}_{W_*^k}\big(A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}} - A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S} + \boldsymbol{v}\big)\big\|_1 \\ &\leq \frac{1}{M}\big\|A^{\mathrm{T}}_{W_*^k}A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\big\|_1 + \frac{1}{M}\big\|A^{\mathrm{T}}_{W_*^k}A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S}\big\|_1 + \frac{1}{M}\big\|A^{\mathrm{T}}_{W_*^k}\boldsymbol{v}\big\|_1 \\ &\leq \frac{1}{M}\big\|A^{\mathrm{T}}_{W_*^k}A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\big\|_1 + \big|\boldsymbol{a}^{\mathrm{T}}_{l_0}A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S}\big| + \big|\boldsymbol{a}^{\mathrm{T}}_{l_1'}\boldsymbol{v}\big| \\ &\leq \frac{1}{M}\big\|A^{\mathrm{T}}_{W_*^k}A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\big\|_1 + \big|\boldsymbol{a}^{\mathrm{T}}_{l_0}A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S}\big| + \epsilon, \end{aligned} \tag{B.10}$$

where $l_0 \in W_*^k$ such that

$$\big|\boldsymbol{a}^{\mathrm{T}}_{l_0}A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S}\big| \triangleq \big\|A^{\mathrm{T}}_{W_*^k}A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S}\big\|_\infty$$

and $l_1' \in W_*^k$ such that

$$\big|\boldsymbol{a}^{\mathrm{T}}_{l_1'}\boldsymbol{v}\big| \triangleq \big\|A^{\mathrm{T}}_{W_*^k}\boldsymbol{v}\big\|_\infty.$$

Since $l \geq k-1$, by (B.9) and (B.10), we have

$$\begin{aligned} b^k - \rho^k &= \big\|A^{\mathrm{T}}_{S\setminus\Lambda^{k-1}}\boldsymbol{r}^{k-1}\big\|_\infty - \min\Big\{\big|\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{r}^{k-1}\big|, i \in W_*^k\Big\} \\ &\geq \big\|A^{\mathrm{T}}_{S\setminus\Lambda^{k-1}}A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\big\|_\infty - \frac{1}{M}\big\|A^{\mathrm{T}}_{W_*^k}A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}\big\|_1 - 2\epsilon \\ &\quad - \Big(\big|\boldsymbol{a}^{\mathrm{T}}_{i_0}A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S}\big| + \big|\boldsymbol{a}^{\mathrm{T}}_{l_0}A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S}\big|\Big) \end{aligned}$$

$$
\begin{aligned}
&\overset{(a)}{\geq} 1-\sqrt{\frac{s-l}{M}+1}\delta_{s+1}-\left\|A_{\{i_0,l_0\}}^{\mathrm{T}}A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S}\right\|_1-2\epsilon \\
&\geq 1-\sqrt{\frac{s-l}{M}+1}\delta_{s+1}-\sqrt{2}\left\|A_{\{i_0,l_0\}}^{\mathrm{T}}A_{\Lambda^{k-1}\setminus S}\hat{\boldsymbol{x}}_{\Lambda^{k-1}\setminus S}\right\|_2-2\epsilon \\
&\overset{(b)}{\geq} 1-\sqrt{\frac{s-l}{M}+1}\delta_{s+1}-\sqrt{2}\sqrt{|\Lambda^{k-1}\setminus S|}\theta_{2,|\Lambda^{k-1}\setminus S|}-2\epsilon,
\end{aligned}
$$

where (a) is due to inequality (2.4), (b) is because of (2.3).

Since $l \geq k-1$, we have

$$
|\Lambda^{k-1}\setminus S| = (k-1)M - l \leq (k-1)(M-1),
$$

then we obtain

$$
\begin{aligned}
&1-\sqrt{\frac{s-l}{M}+1}\delta_{s+1}-\sqrt{2}\sqrt{|\Lambda^{k-1}\setminus S|}\theta_{2,|\Lambda^{k-1}\setminus S|}-2\epsilon \\
&\geq 1-\sqrt{\frac{s}{M}+1}\delta_{s+1}-\sqrt{2}\sqrt{(M-1)(k-1)}\theta_{2,(M-1)(k-1)}-2\epsilon \\
&\geq 1-\sqrt{\frac{s}{M}+1}\delta_{s+1}-\sqrt{2(s-1)}(M-1)\delta_{s+1}-2\epsilon.
\end{aligned}
$$

Hence, (B.3) holds if

$$
\delta_{s+1} < \frac{1-2\epsilon}{\sqrt{s/M+1}+\sqrt{2(s-1)}(M-1)}.
$$

Comparing

$$
\frac{1-\epsilon}{\sqrt{s+1}}, \frac{1-\epsilon}{\sqrt{s-1}(M-1)}, \frac{1-2\epsilon}{\sqrt{s/M+1}+\sqrt{2(s-1)}(M-1)},
$$

if

$$
\delta_{s+1} < \frac{1-2\epsilon}{\sqrt{s+1}+\sqrt{2(s-1)}(M-1)}
$$

is satisfied, then (B.1), (B.2) and (B.3) hold when $M \geq 1$.

Hence, if the RIP δ_{s+1} of the matrix A satisfies

$$
\delta_{s+1} < \frac{1}{\sqrt{s+1}+\sqrt{2(s-1)}(M-1)},
$$

and ϵ satisfies

$$
\epsilon < \frac{1-\sqrt{s+1}\delta_{s+1}-\sqrt{2(s-1)}(M-1)\delta_{s+1}}{2},
$$

then the TGOMP algorithm exactly recovers the support S of $\boldsymbol{x}$ in at most s iterations. The Theorem 3.5 is proved. □

Appendix C Proof of Theorem 3.6

Proof. Let S be the support of an $(s_1,\ldots,s_N)$-piecewise sparse ternary signal $\boldsymbol{x}$ and $S=\cup_{i=1}^N S_i$, where S_i is the support of $\boldsymbol{x}_i$ for $i=1,\ldots,N$. For $i=1,\ldots,N$, $S_i=S_i^+\cup S_i^-$, where

- $S_i^+=\{j|x_j=1, j\in S_i\}$,
- $S_i^-=\{j|x_j=-1, j\in S_i\}$.

Let $\Lambda_i^{k-1}=\Lambda_{i(+)}^{k-1}\cup\Lambda_{i(-)}^{k-1}$, where $\Lambda_{i(+)}^{k-1}$ and $\Lambda_{i(-)}^{k-1}$ denote the sets of indices selected for positive and negative entries of $\hat{\boldsymbol{x}}$ in the first $k-1$ iterations, respectively.

For $i=1,2,\ldots,N$, let $i_j\in\{i_1,\ldots,i_{M_i}\}\subseteq S_i\setminus\Lambda_i^{k-1}$, we define

$$b_{i_j}^k=\left|\boldsymbol{a}_{i_j}^{\mathrm{T}}\boldsymbol{r}^{k-1}\right|,\quad \rho_i^k=\left\|\boldsymbol{A}_{S_i^c}^{\mathrm{T}}\boldsymbol{r}^{k-1}\right\|_\infty.$$

Similar to the proof of Theorem 3.4, we provide the sufficient condition for the step 3 of Algorithm 3.3 in each component in each iteration. Then, we establish a sufficient condition that guarantees the PTGOMP algorithm selects M_i correct indices and corresponding entries of $\boldsymbol{x}$ in each component in each iteration.

To ensure that the validity of the step 3 of Algorithm 3.3, we need the following inequalities to hold in the k-th iteration for $k=1,\ldots,\max_{i=1,\ldots,N}\{s_i/M_i\}$. That is, for $i=1,2,\ldots,N$, we need

$$\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{r}^{k-1}>0,\quad \text{if}\quad x_j=1,\qquad j\in S_i^+\setminus\Lambda_{i(+)}^{k-1}, \tag{C.1}$$

$$\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{r}^{k-1}<0,\quad \text{if}\quad x_j=-1,\quad j\in S_i^-\setminus\Lambda_{i(-)}^{k-1} \tag{C.2}$$

to hold. Besides, to guarantee that the PTGOMP algorithm selects M correct indices and corresponding entries of $\boldsymbol{x}$ in each iteration, we need the following inequality to hold in the k-th iteration for $k=1,\ldots,\max_{i=1,\ldots,N}\{s_i/M_i\}$. That is, for any $i_j\in\{i_1,\ldots,i_{M_i}\}\subseteq S_i\setminus\Lambda_i^{k-1}$, we need

$$b_{i_j}^k>\rho_i^k \tag{C.3}$$

to hold. In particular, when $M_i=s_i$, the PTGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in the first iteration.

Assuming that the PTGOMP algorithm has selected $(k-1)M_i$ correct indices for i-th component in the first $k-1$ iterations for $i=1,\ldots,N$, then we have $\Lambda_{i(+)}^{k-1}\subset S_i^+$, $\Lambda_{i(-)}^{k-1}\subset S_i^-$, $\hat{\boldsymbol{x}}_{\Lambda^{k-1}}=\boldsymbol{x}_{\Lambda^{k-1}}$, and

$$\left|\Lambda^{k-1}\right|=\left|\cup_{i=1}^N\Lambda_{k-1}^{(i)}\right|=(k-1)M,$$

where $M=\sum_{i=1}^N M_i$. By Algorithm 3.3, we have

$$\boldsymbol{r}^{k-1}=\boldsymbol{b}-A_{\Lambda^{k-1}}\hat{\boldsymbol{x}}_{\Lambda^{k-1}}=A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}}+\boldsymbol{v}.$$

For step 3 of Algorithm 3.3, when $x_{i_0} = 1$, $i_0 \in S_i^+ \setminus \Lambda_{i(+)}^{k-1}$, we have

$$\begin{aligned}\boldsymbol{a}_{i_0}^{\mathrm{T}}\boldsymbol{r}^{k-1} &= \boldsymbol{a}_{i_0}^{\mathrm{T}}\big(A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}} + \boldsymbol{v}\big)\\ &\geq x_{i_0} - \sum_{j'\in S_i\setminus(\Lambda_i^{k-1}\cup\{i_0\})} \big|\boldsymbol{a}_{i_0}^{\mathrm{T}}\boldsymbol{a}_{j'}\big| - \sum_{j\neq i}^{N}\sum_{j_1'\in S_j\setminus\Lambda_j^{k-1}} \big|\boldsymbol{a}_{i_0}^{\mathrm{T}}\boldsymbol{a}_{j_1'}\big| - \|\boldsymbol{v}\|_2\\ &\geq 1 - \big(s_i - (k-1)M_i - 1\big)\alpha_i\mu - \sum_{j\neq i}^{N}\big(s_j - (k-1)M_j\big)\mu - \epsilon, \end{aligned} \tag{C.4}$$

and when $x_{i_0} = -1$, $i_0 \in S_i^- \setminus \Lambda_{i(-)}^{k-1}$, we have

$$\begin{aligned}\boldsymbol{a}_{i_0}^{\mathrm{T}}\boldsymbol{r}^{k-1} &= \boldsymbol{a}_{i_0}^{\mathrm{T}}\big(A_{S\setminus\Lambda^{k-1}}\boldsymbol{x}_{S\setminus\Lambda^{k-1}} + \boldsymbol{v}\big)\\ &= \boldsymbol{a}_{i_0}^{\mathrm{T}}\left(A_{S_i\setminus\Lambda_i^{k-1}}\boldsymbol{x}_{S_i\setminus\Lambda_i^{k-1}} + \sum_{j\neq i}^{N} A_{S_j\setminus\Lambda_j^{k-1}}\boldsymbol{x}_{S_j\setminus\Lambda_j^{k-1}}\right) + \boldsymbol{a}_{i_0}^{\mathrm{T}}\boldsymbol{v}\\ &\leq x_{i_0} + \sum_{j'\in S_i\setminus(\Lambda_i^{k-1}\cup\{i_0\})} \big|\boldsymbol{a}_{i_0}^{\mathrm{T}}\boldsymbol{a}_{j'}\big| + \sum_{j\neq i}^{N}\sum_{j_1'\in S_j\setminus\Lambda_j^{k-1}} \big|\boldsymbol{a}_{i_0}^{\mathrm{T}}\boldsymbol{a}_{j_1'}\big| + \|\boldsymbol{v}\|_2\\ &\leq -1 + \big(s_i - (k-1)M_i - 1\big)\alpha_i\mu + \sum_{j\neq i}^{N}\big(s_j - (k-1)M_j\big)\mu + \epsilon. \end{aligned} \tag{C.5}$$

Since $(k-1)M_i \geq k-1 \geq 0$ for $i = 1,\ldots,N$, by (C.4) and (C.5), then (C.1) and (C.2) hold if

$$s_i\alpha_i - \alpha_i + \sum_{j\neq i}^{N} s_j < \frac{1-\epsilon}{\mu}. \tag{C.6}$$

Besides, for the left-hand side of (C.3) and any $i_j \in \{i_1,\ldots,i_{M_i}\} \subseteq S_i \setminus \Lambda_i^{k-1}$, we have

$$\begin{aligned} b_{i_j}^k &= \left|\boldsymbol{a}_{i_j}^{\mathrm{T}}\left(A_{S_i\setminus\Lambda_i^{k-1}}\boldsymbol{x}_{S_i\setminus\Lambda_i^{k-1}} + \sum_{j\neq i}^{N} A_{S_j\setminus\Lambda_j^{k-1}}\boldsymbol{x}_{S_j\setminus\Lambda_j^{k-1}} + \boldsymbol{v}\right)\right|\\ &\geq |x_{i_j}| - \sum_{i'\in S_i\setminus(\Lambda_i^{k-1}\cup\{i_j\})} \big|\boldsymbol{a}_{i_j}^{\mathrm{T}}\boldsymbol{a}_{i'}\big| - \sum_{j\neq i}^{N}\sum_{j'\in S_j\setminus\Lambda_j^{k-1}} \big|\boldsymbol{a}_{i_j}^{\mathrm{T}}\boldsymbol{a}_{j'}\big| - \|\boldsymbol{v}\|_2\\ &\geq 1 - \big|S_i \setminus \big(\Lambda_i^{k-1}\cup\{i_j\}\big)\big|\alpha_i\mu - \sum_{j\neq i}^{N}\big|S_j\setminus\Lambda_j^{k-1}\big|\mu - \epsilon\\ &= 1 - \big(s_i - (k-1)M_i - 1\big)\alpha_i\mu - \sum_{j\neq i}^{N}\big(s_j - (k-1)M_j\big)\mu - \epsilon, \end{aligned}$$

and for the right-hand side of (C.3), we have

$$\begin{aligned}\rho_i^k &= \big\|\boldsymbol{A}_{S_i^c}^{\mathrm{T}}\boldsymbol{r}^{k-1}\big\|_\infty \\ &\triangleq \big|\boldsymbol{a}_{i_1}^{\mathrm{T}}(A_{S\backslash\Lambda^{k-1}}\boldsymbol{x}_{S\backslash\Lambda^{k-1}}+\boldsymbol{v})\big| \\ &\le \sum_{i'\in S_i\backslash\Lambda_i^{k-1}}\big|\boldsymbol{a}_{i_1}^{\mathrm{T}}\boldsymbol{a}_{i'_0}\big| + \sum_{j\neq i}^{N}\sum_{j'\in S_j\backslash\Lambda_j^{k-1}}\big|\boldsymbol{a}_{i_1}^{\mathrm{T}}\boldsymbol{a}_{j'}\big| + \|\boldsymbol{v}\|_2 \\ &\le \big(s_i-(k-1)M_i\big)\alpha_i\mu + \sum_{j\neq i}^{N}\big(s_j-(k-1)M_j\big)\mu+\epsilon.\end{aligned}$$

Therefore, we obtain that

$$\big(s_i-(k-1)M_i\big)\alpha_i + \sum_{j\neq i}^{N}\big(s_j-(k-1)M_j\big) < \frac{1+\alpha_i\mu-2\epsilon}{2\mu} \tag{C.7}$$

is a sufficient condition for (C.3) for $i=1,\ldots,N$. Since $(k-1)M_i\ge k-1\ge 0$ for $i=1,\ldots,N$, we obtain that

$$s_i\alpha_i-\frac{\alpha_i}{2}+\sum_{j\neq i}^{N}s_j<\frac{1-2\epsilon}{2\mu} \tag{C.8}$$

is a sufficient condition for (C.7). Let

$$2(1-\alpha_Z)s_Z+\alpha_Z=\min_{i=1,\ldots,N}\{2(1-\alpha_i)s_i+\alpha_i\},$$

then we obtain that

$$2\big(s-(1-\alpha_Z)s_Z\big)\mu-\alpha_Z\mu<1-2\epsilon \tag{C.9}$$

is a sufficient condition for (C.8).

By comparing (C.6) with (C.8), we obtain (C.9) is a sufficient condition of (C.8). Hence, if (3.23) and (3.24) are satisfied, when $M_i\le s_i$, the PTGOMP algorithm selects M_i correct indices and corresponding entries of $\boldsymbol{x}$ in each component in each iteration, that is, the PTGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in $\max_{i=1,\ldots,N}\{s_i/M_i\}$ iterations. In particular, when $M_i=s_i$, the PTGOMP algorithm recovers the support S and entries of $\boldsymbol{x}$ in the first iteration. This completes the proof of Theorem 3.6. □

References

[1] B. Adcock, S. Brugiapaglia, and M. King-Roskamp, *Iterative and greedy algorithms for the sparsity in levels model in compressed sensing*, in: Conference on Wavelets and Sparsity XVIII, 11138 (2019), 1113809.

[2] B. Adcock, S. Brugiapaglia, and M. King-Roskamp, *The benefits of acting locally: Reconstruction algorithms for sparse in levels signals with stable and robust recovery guarantees*, IEEE Trans. Signal Process 69 (2021), 3160–3175.

[3] A. Akbari, M. Trocan, and B. Granado, *Sparse recovery-based error concealment*, IEEE Trans Multimedia 19 (2017), 1339–1350.

[4] H. Alemdar, V. Leroy, A. Prost-Boucle, and F. Petrot, *Ternary neural networks for resource-efficient AI applications*, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, (2017), 2547–2554.

[5] K. D. Ba, P. Indyk, E. Price, and D. P. Woodruff, *Lower bounds for sparse recovery*, in: Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, (2010), 1190–1197.

[6] J. D. Blanchard, M. Cermak, D. Hanle, and Y. R. Jing, *Greedy algorithms for joint sparse recovery*, IEEE Trans. Signal Process 62 (2014), 1694–1704.

[7] T. Blumensath and M. E. Davies, *Iterative hard thresholding for compressed sensing*, Appl. Comput. Harmon. Anal. 20 (2009), 265–274.

[8] P. T. Boufounos, P. Smaragdis, and B. Raj, *Joint sparsity models for wideband array processing*, in: Conference on Wavelets and Sparsity XIV, (2011), 81380K.

[9] T. T. Cai and L. Wang, *Orthogonal matching pursuit for sparse signal recovery with noise*, IEEE Trans. Inf. Theory 57 (2011), 4680–4688.

[10] T. T. Cai, L. Wang, and G. Xu, *Stable recovery of sparse signals and an oracle inequality*, IEEE Trans. Inf. Theory 56 (2010), 3516–3522.

[11] T. T. Cai, L. Wang, and G. Xu, *Shifting inequality and recovery of sparse signals*, IEEE Trans. Signal Process 58 (2010), 1300–1308.

[12] E. J. Candes, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inf. Theory 52 (2006), 489–509.

[13] E. J. Candes and T. Tao, *Decoding by linear programming*, IEEE Trans. Inf. Theory 51 (2005), 4203–4215.

[14] S. S. B. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM J. Sci. Comput. 20 (1998), 33–61.

[15] Z. Chen, F. Sohrabi, and W. Yu, *Sparse activity detection for massive connectivity*, IEEE Trans. Signal Process 66 (2018), 1890–1904.

[16] K. Choi, J. Wang, L. Zhu, T. S. Suh, S. Boyd, and L. Xing, *Compressed sensing based cone-beam computed tomography reconstruction with a first-order method*, Med. Phys. 37 (2010), 5113–5125.

[17] W. Dan, *Analysis of orthogonal multi-matching pursuit under restricted isometry property*, Sci. China Math. 57 (2014), 2179–2188.

[18] W. Dan, *Support recovery from noisy measurement via orthogonal multi-matching pursuit*, Numer. Math. Theor. Meth. Appl. 9 (2016), 185–192.

[19] G. Di Guglielmo et al., *Compressing deep neural networks on FPGAs to binary and ternary precision with HLS4ML*, Mach. Learn.-Sci. Technol. 2 (2021), 015001.

[20] D. L. Donoho, *Compressed sensing*, IEEE Trans. Inf. Theory 52 (2006), 1289–1306.

[21] D. L. Donoho and X. M. Huo, *Uncertainty principles and ideal atomic decomposition*, IEEE Trans. Inf. Theory 47 (2001), 2845–2862.

[22] Y. C. Eldar, P. Kuppinger, and H. Boelcskei, *Block-sparse signals: Uncertainty relations and efficient recovery*, IEEE Trans. Signal Process 58 (2010), 3042–3054.

[23] Y. C. Eldar and M. Mishali, *Block sparsity and sampling over a union of subspaces*, in: International Conference on Digital Signal Processing, IEEE, (2009), 1–8.

[24] M. GUERQUIN-KERN, M. HABERLIN, K. P. PRUESSMANN, AND M. UNSER, *A fast wavelet-based reconstruction method for magnetic resonance imaging*, IEEE Trans. Med. Imaging 30 (2011), 1649–1660.

[25] A. HUANG, R. FENG, AND S. ZHENG, *Recovery guarantee for orthogonal matching pursuit method to reconstruct sparse polynomials*, Numer. Math. Theor. Meth. Appl. 15 (2022), 793-–818.

[26] J. JEGANATHAN, A. GHRAYEB, AND L. SZCZECINSKI, *Generalized space shift keying modulation for MIMO channels*, in: Proceedings of the 2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, (2008), 1–5.

[27] A. JONES, A. TAMTOGL, I. CALVO-ALMAZAN, AND A. HANSEN, *Continuous compressed sensing for surface dynamical processes with helium atom scattering*, Sci Rep. 6 (2016), 27776.

[28] S. KEIPER, G. KUTYNIOK, D. G. LEE, AND G. E. PFANDER, *Compressed sensing for finite-valued signals*, Linear Alg. Appl. 532 (2017), 570–613.

[29] Y. LECUN, *The MNIST database of handwritten digits*, `http://yann.lecun.com/exdb/mnist/`, (1998).

[30] C. LI AND Y. ZHONG, *Piecewise sparse recovery in unions of bases*, J. Math. Res. Appl. 43 (2023), 363–378.

[31] H. LI, H. YING, AND X. LIU, *Binary generalized orthogonal matching pursuit*, Jpn. J. Ind. Appl. Math. 41 (2024), 1–12.

[32] O. LINDENBAUM AND S. STEINERBERGER, *Randomly aggregated least squares for support recovery*, Signal Process 180 (2021), 107858.

[33] O. LINDENBAUM AND S. STEINERBERGER, *Refined least squares for support recovery*, Signal Process 195 (2022), 108493.

[34] A. MALEKI AND D. L. DONOHO, *Optimally tuned iterative reconstruction algorithms for compressed sensing*, IEEE J. Sel. Top. Signal Process 4 (2010), 330–341.

[35] D. NEEDELL AND J. A. TROPP, *CoSaMP: Iterative signal recovery from incomplete and inaccurate samples*, Appl. Comput. Harmon. Anal. 26 (2009), 301–321.

[36] S. SATPATHI, R. L. DAS, AND M. CHAKRABORTY, *Improving the bound on the RIP constant in generalized orthogonal matching pursuit*, IEEE Signal Process. Lett. 20 (2013), 1074–1077.

[37] V. STUDER, J. BOBIN, M. CHAHID, H. S. MOUSSAVI, E. CANDES, AND M. DAHAN, *Compressive fluorescence microscopy for biological and hyperspectral imaging*, Proc. Natl. Acad. Sci. U.S.A 109 (2012), E1679–E1687.

[38] B. L. STURM, *A study on sparse vector distributions and recovery from compressed sensing*, arXiv:1103.6246 [cs.DS] (2011).

[39] J. A. TROPP, *Greed is good: Algorithmic results for sparse approximation*, IEEE Trans. Inf. Theory 50 (2004), 2231–2242.

[40] J. WANG, S. KWON, AND B. SHIM, *Generalized orthogonal matching pursuit*, IEEE Trans. Signal Process 60 (2012), 6202–6216.

[41] X. WANG, D. MENG, M. HUANG, AND L. WAN, *Reweighted regularized sparse recovery for DOA estimation with unknown mutual coupling*, IEEE Commun. Lett. 23 (2018), 290–293.

[42] J. WEN AND H. LI, *Binary sparse signal recovery with binary matching pursuit*, Inverse Probl. 37 (2021), 065014.

[43] J. WEN, Z. ZHOU, D. LI, AND X. TANG, *A novel sufficient condition for generalized orthogonal matching pursuit*, IEEE Commun. Lett. 21 (2017), 805–808.

[44] Z. XU, *Compressed sensing: A survey*, Sci. China Math. 42 (2012), 865–877.

[45] Z. XU, *The performance of orthogonal multi-matching pursuit under Restricted Isometry*

Property, J. Comput. Math, 33 (2015), 495–516.

[46] M. YAN, M. LI, P. ROCCARO, AND G. V. KORSHIN, *Ternary model of the speciation of I-/Br-/Cl-trihalomethanes formed in chloraminated surface waters*, Environ. Sci. Technol. 50 (2016), 4468–4475.

[47] Y. ZHONG AND C. LI, *Piecewise sparse recovery via piecewise inverse scale space algorithm with deletion rule*, J. Comput. Math. 38 (2020), 375–394.